School of Computer Science
The University of Auckland
New Zealand

Heterogeneous Network Mining and Analysis

Ranran Bian

Supervisors: *Yun Sing Koh*
Gillian Dobbie
Anna Divoli

A thesis submitted in fulfillment of the requirements of Doctor of Philosophy in Computer Science, the university of auckland, June 2019

Abstract

Nowadays, large amounts of data are being created daily through fingertips with the emergence of abundant social media. With the exponential growth of the Internet over the past decades, there has been a surge of interest in the capability to extract useful data, trends and structures on these social platforms as they act as a gateway for online commercialization and information propagation.

Heterogeneous networks model different types of objects and relationships among them. Compared to homogeneous networks, heterogeneous networks can fuse information from multiple data sources and social platforms. Therefore, it is natural to model complex objects and their relationships in big social media data with heterogeneous networks.

Despite decades of technique development for various data mining tasks, few of them target heterogeneous networks. Heterogeneity is a key element in contemporary social networks which provides diversified perception of networks. Therefore, heterogeneous network analysis has become an important topic in data mining in recent years that has been attracting increasing attention from both industry and academia, as they provide more comprehensive and interesting analysis results than their projected homogeneous networks.

Motivated by these considerations, this thesis presents a series of new techniques for knowledge discovery in heterogeneous networks. In particular, the methods proposed in this thesis have been applied to a wide range of applications including community discovery, ranking and information retrieval. For dynamic heterogeneous networks, our research presents a more effective network embedding technique when compared to the existing state-of-the-art methods. Throughout this thesis, we highlight how our methodologies were able to identify more tightly coupled communities in heterogeneous networks, more accurately rank top performing social actors and having the capability to view heterogeneous networks in a dynamic construct.

Acknowledgements

A PhD is a long journey through unknown lands, over sky-high peaks, and through deep and dark troughs. The efforts and support of many individuals have made this a pleasant and memorable journey. I owe an enormous debt of gratitude to my supervisors, *Yun Sing Koh*, *Gillian Dobbie*, and *Anna Divoli*, who have guided me to learn and grow tremendously over the progressive years. Thank you all very much for being inspiring, for your insightful suggestions and patience, for your ample research support, for your detailed comments on every single piece of my written work (the list goes on and on). It has been a great pleasure to work with you.

This PhD research has also received scholarships from different sources, for which I owe my special thanks to Pingar, Callaghan Innovation (R&D Student Fellowship Grant Contract Number: PTERN1502) and The University of Auckland.

I want to thank my fellow colleagues and friends at the KMG research group for their sensible opinions and input. I also thank my colleagues from Pingar for giving me valuable advice from industry.

I want to extend special thanks to my husband Tobey for his unconditional support and love, my parents, in-laws and close friends for their encouragement and support.

This thesis is dedicated to each and everyone who has been, and is, a part of my life, for without them I would not be where I am right now and have achieved this far.

Thank you all.

List of Publications

Parts of Chapter 3 have been published in:

- Ranran Bian, Yun Sing Koh, Gillian Dobbie, and Anna Divoli. Identifying Top-k Nodes in Social Networks: A Survey. In *ACM Computing Surveys (CSUR) Journal*, Volume 52 Issue 1 Article No. 22 pages 22:1-22:33, 2019.

Parts of Chapter 4 have been published in:

- Ranran Bian, Yun Sing Koh, Gillian Dobbie, and Anna Divoli. OHC: Uncovering Overlapping Heterogeneous Communities. In *Proceedings of the 31st Australasian Joint Conference on Artificial Intelligence*, pages 193-205, 2018. (Best Paper Award)

Parts of Chapter 6 have been published in:

- Ranran Bian, Yun Sing Koh, Gillian Dobbie, and Anna Divoli. Network Embedding and Change Modeling in Dynamic Heterogeneous Networks. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019.

Contents

Abstract i

Acknowledgements iii

List of Publications v

1 Introduction 1

1.1 Heterogeneous Network Mining and Analysis 2

1.2 Motivation 4

1.3 Problem Statement 6

1.4 Scope 7

1.5 Objectives 8

1.6 Contributions 8

1.7 Structure of Thesis 9

2 Literature Survey 13

2.1 Community Detection in Homogeneous and Heterogeneous Networks 14

2.1.1 Community Detection Methods in Homogeneous Networks 14

2.1.2 Homogeneous Community Scoring Functions 14

2.1.3 Community Detection Methods in Heterogeneous Networks 15

2.2 Ranking in Heterogeneous Networks 16

2.3 Network Embedding in Heterogeneous Networks 17

2.4 Other Data Mining Tasks in Heterogeneous Networks 18

2.4.1 Clustering 18

2.4.2 Classification 19

2.4.3 Recommendation 20

2.5 Summary of Literature 21

2.6 Performance Evaluation Metrics 22

2.7 Research Gaps 23

3 Identifying Top-k Nodes in Social Networks: A Survey 25

3.1 Introduction 26

3.1.1 Related Surveys 27

3.1.2 Contributions 28

3.1.3 Survey Organization 28

3.2 Preliminaries 28

3.2.1 Social Network 29

3.2.2 Static Network versus Dynamic Network 29

3.2.3 Social Network Graph 29

3.2.4 Node Centrality Measures 29

3.2.5 Influence Diffusion Models 30

3.2.6 Influence Maximization Definition 32

3.2.7 Influential Nodes versus Significant Nodes 32

3.2.8 Relationships among Top-k Nodes, Top-k Influential Nodes and Top-k Significant Nodes 33

3.3 Top-k Influential Nodes Identification 35

3.3.1 Greedy based Approaches and Improvements 36

3.3.2 Centrality Measure based Approaches 40

3.3.3 Topic based Approaches 42

3.3.4 Network Content or Topology based Approaches 44

3.3.5 Identifying Top-k Influential Nodes in Dynamic Social Networks 47

3.3.6 Section Summary 50

3.4 Top-k Significant Nodes Identification 51

3.4.1 Effector-based Identification and Bridge Nodes 51

3.4.2 Authoritative-based Identification 52

3.4.3 Academic Datasets 54

3.4.4 Other Work 55

3.4.5 Section Summary 56

3.5 Applications of Identifying Top-k Nodes in Various Networks . . 56

3.5.1 Twitter 57

3.5.2 Facebook 60

3.5.3 Blogosphere 60

3.5.4 Misinformation Control 62

3.5.5 Community Question Answering 62

3.5.6 Networks with Complex Topologies 63

3.5.7 Miscellaneous Applications 64
3.5.8 Section Summary 65
3.6 Research Directions 65
3.7 Conclusions 67

4 Uncovering Overlapping Heterogeneous Communities 69
4.1 Introduction 70
4.2 Problem Formalization 72
4.3 The OHC Algorithm 72
4.3.1 Three Phases of OHC 72
4.3.2 Phase One 73
4.3.3 Phase Two 73
4.3.4 Phase Three 74
4.3.5 Time Complexity Analysis of OHC 75
4.3.6 Conditions of OHC 77
4.4 Experiments and Results 77
4.4.1 Datasets 77
4.4.2 Benchmark Techniques 78
4.4.3 Evaluation Metrics 79
4.4.4 Community Quality 80
4.4.5 Case Study 82
4.4.6 Run Time 86
4.4.7 Recursion Depth 86
4.5 Conclusions and Future Work 87

5 Community Based Ranking of Objects 89
5.1 Introduction 90
5.2 Preliminaries 91
5.3 The ComRank Algorithm 92
5.3.1 Time Complexity Analysis of ComRank 97
5.4 Experimental Evaluation 98
5.4.1 Academic Datasets 99
5.4.2 Spearman Rank Correlation 101
5.4.3 Digg Dataset 102
5.4.4 Run Time 103
5.5 Conclusions and Future Work 103

6 Network Embedding and Change Modeling in Dynamic Heterogeneous Networks 105
6.1 Introduction . . . 106
6.2 Research Overview . . . 107
6.3 The change2vec Algorithm . . . 109
6.3.1 Algorithm Description . . . 112
6.3.2 Time Complexity Analysis of change2vec . . . 114
6.4 Experimental Evaluations . . . 115
6.4.1 Normalized Mutual Information (NMI) . . . 116
6.4.2 Adjusted Rand Index (ARI) . . . 117
6.4.3 Results with Introduced Open Triads . . . 118
6.4.4 Run Time . . . 119
6.4.5 Results with Deleted Edges . . . 120
6.5 Conclusions and Future Work . . . 121

7 Conclusions 123
7.1 Contributions . . . 124
7.2 Limitations . . . 125
7.3 Future Directions . . . 126

References 129

List of Figures

1.1 Example of a heterogeneous network 4

1.2 An homogeneous projection of Figure 1.1 6

1.3 Overview of mining and analysis techniques proposed in this thesis 9

3.1 Academic collaboration network 34

3.2 Core idea of greedy algorithms for identifying influential nodes . 36

4.1 Sample heterogeneous correlation network H_1 71

4.2 Output of OHC's first two phases (identified (seed) communities denoted by the dotted boundaries) 73

4.3 AAN FOMD 83

4.4 AI FOMD 84

4.5 SE FOMD 84

4.6 AAN FlakeODF 84

4.7 AI FlakeODF 85

4.8 SE FlakeODF 85

5.1 Example of an academic heterogeneous correlation network . . . 91

5.2 Example of a heterogeneous correlation network 94

6.1 Converting a network into 2D space with four different perspectives 106

6.3 Framework of change2vec 109

6.2 An illustrative example of a dynamic social network 109

List of Tables

3.1 List of some approaches for top-k influential nodes identification 35
3.2 Comparisons of some greedy based approaches for identifying influential nodes 39
3.3 Comparisons of various centrality measure based approaches . . 42
3.4 Inputs, outputs and characteristics of various topic based approaches 44
3.5 Comparisons of various network content or topology based approaches 47
3.6 List of key applications of top-k nodes identification 57
3.7 Comparisons of various techniques for identifying top-k nodes in blogosphere 61

4.1 Statistics of datasets 78
4.2 Number of detected heterogeneous communities 79
4.3 FOMD results 80
4.4 FlakeODF results 82
4.5 Average and standard deviation of FOMD across thirty runs for each threshold r of SLPAh 82
4.6 Average and standard deviation of FOMD across thirty runs for Louvainh 83
4.7 Average and standard deviation of FlakeODF across thirty runs for each threshold r of SLPAh 83
4.8 Average and standard deviation of FlakeODF across thirty runs for Louvainh 83
4.9 Run time of algorithms in seconds 86
4.10 Recursion depth (AAN dataset) 87

5.1 Rank of nodes for global importance 94
5.2 Rank of nodes for overall importance 96
5.3 Properties of academic datasets 99

5.4 Average citation count for top-k papers (IS) 99
5.5 Average citation count for top-k authors (IS) 100
5.6 Average h-index for top-k authors (IS) 100
5.7 Top 10 papers identified for AAN 100
5.8 Top 10 authors identified for CGM 101
5.9 Spearman rank correlation for the IS dataset 102
5.10 Average friend number for top-k digg users 103
5.11 Average like number for top-k digg stories 103
5.12 Run time of algorithms in seconds 103

6.1 Properties of DBLP data sets 115
6.2 Node clustering results (NMI) of each time stamp 117
6.3 Node clustering results (ARI) of each time stamp 118
6.4 Run time of each time stamp 118
6.5 Node clustering results (NMI) of each time stamp with open triads 119
6.6 Node clustering results (ARI) of each time stamp with open triads 119
6.7 Run time of each time stamp with open triads 120
6.8 Number of edges in DBLP data sets 120
6.9 Node clustering results (NMI) of each time stamp with deleted edges 121
6.10 Node clustering results (ARI) of each time stamp with deleted edges 122

To my beloved husband, Tobey Sheng-Yen Hung
To my dearest daughter, Bridgette Yen-Lin Hung
To my mother, Feng Jin
To my father, Guochen Bian

1
Introduction

"Sometimes it is the people no one imagines anything of who do the things that no one can imagine."

- Alan Turing

The quote by Alan Turning above hints of the potential of people. Nowadays, with the enormous amount of data generated every second on social networks, potential also exists in that mass of data. As a result, the fast-growing field of social network analysis emerges, focusing on using machines and algorithms to automatically search for useful patterns representing knowledge from data captured in social networks [152].

Social network analysis is the study of social structures and relationship mappings between individuals and groups of social actors within a social network. With the exponential growth of the Internet, and development and growth of social network platforms over the past decades, there has been a shift in traditional social network studies from social and behavioural sciences in fields such as marketing and economics [147] to the study of modern online social media and platforms [96]. Due to this, there has been an explosive growth and interest in this field of study due to the broad range of applications and relevancy in numerous domains including the identification of social influencers [173], viral marketing [35] and criminology [24].

Along with the research vectors comes new challenges, including data integrity, privacy and paradoxes [77], time and computational complexity with the increasing size of data sets and the constant changes and evolution of networks [80]. Hence, a large proportion of modern social network research is aimed at increasing the accuracy and efficiency of methods to allow users to extract meaningful and valuable knowledge and information from complex modern social network data.

Another shift in social network analysis is the underlying data being used. Traditionally, most research has been performed on homogeneous social networks. However in most practical real life scenarios, social networks are heterogeneous in nature with a variety of objects and relationships interacting through social actions. A rising challenge is to provide the capability of mining and extracting information from these complex networks.

1.1 Heterogeneous Network Mining and Analysis

In the past decades, social network analysis has been a hot topic due to its wide application in economics and marketing campaigns, as well as social and behavioural sciences [147, 34]. Sample application domains of social network analysis are customer value evaluation [36], customer relationship management [151], and viral marketing [113]. The majority of the current research has modeled social networks as homogeneous networks where only one type of object (node) and relationship (edge) exist. Therefore, traditional algorithms and techniques have focused on finding useful information from homogeneous social networks, using various techniques such as community detection, classification, clustering, ranking and outlier detection.

In this thesis, we argue that most social networks should be modeled as heterogeneous networks, which include more than one type of object (node) and relationship (edge). Despite a long history of structural social network and user behaviour analysis, little research has addressed heterogeneous social network mining and analysis. Traditionally, researchers focused on the analysis of homogeneous social networks. However, social networks in practice are heterogeneous due to the fact that social actors can be connected via various types of social actions. Most traditional data mining methods, which are structural and homogeneous approaches, are not capable of handling such heterogeneous social networks, which is the primary concern of this thesis. The multiple types of social actions hold valuable information about development of social actors and interests. Therefore, mining heterogeneous social networks to interpret and understand real world social networks is an important research direction.

Mining useful information is not an easy task. There is existing research that has focused on mining useful information that is accurate and precise, and also mining it efficiently and effectively. The former emphasizes minimal incorrect results of algorithms. To this end various measures have been proposed such as the NMI (Normalized Mutual Information) [129] and ARI (Adjusted Rand Index) [140] scores, which are widely used to compare the similarity between the identified results and the ground truth. The latter focuses on the execution efficiency of the techniques, where low run time and memory cost are the goal by adopting a faster processing mechanism or implementing a more efficient data structure.

In the situations where high accuracy and efficiency cannot be achieved simultaneously, a good balance between the two becomes the goal. With the abundance of multi-dimensional data embedded in heterogeneous networks, we target to optimize the balance when performing various data mining tasks in this kind of network.

1.2 Motivation

Heterogeneous networks model different types of objects and relationships between them. As compared to homogeneous networks, heterogeneous networks can fuse information from multiple data sources and social platforms. Nowadays, different types of objects are interconnected with the emergence of abundant social media. Particularly, with the rapid increase of social network content online, big data analysis becomes an important task to study. A heterogeneous information network can model complex objects and their relationships in big data effectively [127].

Heterogeneous networks exist in scenarios such as community-based question answering sites, multimedia networks and knowledge graphs [22]. The entities and relationships in these scenarios are usually of different types. Hence, the scenario can be viewed as an instance of a heterogeneous graph. Figure 1.1 gives an example heterogeneous network, which contains three types of objects (i.e., researcher, publication and department) and three types of relationships (i.e., collaboration, published and employment).

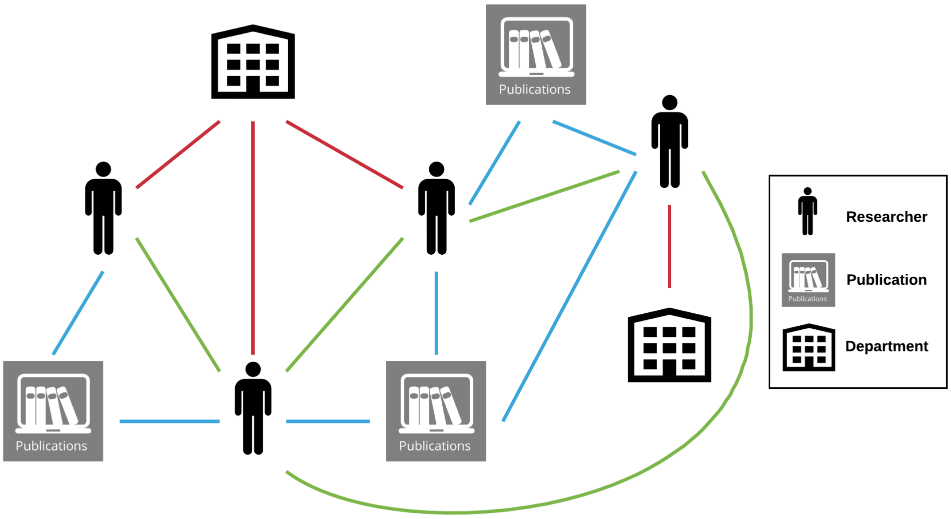


Figure 1.1: Example of a heterogeneous network

A variety of data mining tasks can be conducted in heterogeneous networks, e.g., community detection, ranking and network embedding. Because of the unique characteristics (e.g., fusion of more information and rich semantics) of heterogeneous networks, there is a potential to discover more interesting mining results in heterogeneous networks than their projected homogeneous networks. For instance, the questions that we are able to answer with the heterogeneous network in Figure 1.1 are:

1. Which researchers, publications and departments are connected more closely with each other than with others? A vice chancellor may want to segment a university into different groups to study the high-level organisational structure instead of examining each individual object.

2. What are the three most important objects considering the given relationships? Answers to this question are top-ranked researchers, departments and research topics (derived from titles of the publications), which could be useful information for making funding decisions.

3. Comparing with the network from five years ago, what objects and relationships have changed and how have they changed? This question can provide reference information to board members of a university for examining outcomes from their most recent five year strategic plan.

The above questions cannot be answered using a homogeneous network that is projected from the heterogeneous information network with significant information loss, e.g., the researcher collaboration homogeneous network in Figure 1.2.

Our main contribution in Chapter 4 is community discovery in heterogeneous networks, which corresponds to the first question above. Chapter 5 focuses on ranking of different types of objects in heterogeneous networks and corresponds to the second question. Chapter 6 studies object representation learning in dynamic heterogeneous networks and the proposed technique can be used for modeling changes in such networks as illustrated in the last question. Applying our contributions in Chapters 4, 5 and 6 to a static or dynamic heterogeneous network, we are able to answer different combinations of the questions in a more informative manner as compared to a projected homogeneous network (e.g. Figure 1.2).

Overall, the main research question is “Does heterogeneous information improve the results of network analysis?”. To answer this research question, the thesis focuses on three network analysis tasks, community detection, ranking and network embedding.

The significance of the research question is:

1. Network analysis has been studied widely in homogeneous networks (e.g., network shown in Figure 1.2) while most of real world networks are heterogeneous (e.g., network shown in Figure 1.1) . There has been little research in heterogeneous network analysis.

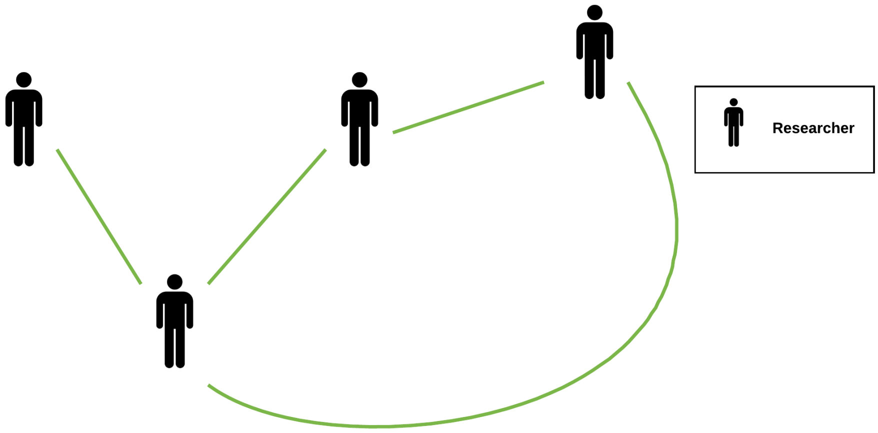


Figure 1.2: An homogeneous projection of Figure 1.1

2. Because of the unique characteristics (e.g., fusion of more information and richer semantics) of heterogeneous networks, there is potential to discover more interesting mining results in heterogeneous networks than their projected homogeneous networks. This is proven by the example questions that can be answered given the heterogeneous network in Figure 1.1, which cannot be answered comprehensively given the projected homogeneous network in Figure 1.2.

3. The three tasks that the thesis focuses on are important network analysis tasks, which can help us to understand the structure and evolution of networks. The analysed structure and evolution of networks can be useful information in a wide range of applications, such as university strategic plan review, influenza-like illness control and forecasting [3].

1.3 Problem Statement

In this thesis we focus on mining and analyzing different kinds of heterogeneous networks. In the first part we look at community discovery in both academic and social heterogeneous networks. The questions that we are interested in are:

1. Can we find heterogeneous communities with dense internal connections and loose external connections?

2. Can our proposed approach achieve good efficiency while maintaining high quality of identified heterogeneous communities?

In the second part we generate a ranking list of heterogeneous objects based on their connections and importance in a heterogeneous network. We focus on the questions:

3. How do we evaluate our heterogeneous ranking list properly?

4. Based on the evaluation metrics, how good is our ranking list?

5. Can the quality of the ranking list be improved by taking community memberships of objects into consideration?

In the last part, the problem of network embedding and change modeling in dynamic heterogeneous networks is studied and we look at the questions:

6. In a dynamic heterogeneous network, how can we produce accurate embedding vectors of objects?

7. Can structural changes in dynamic heterogeneous networks be identified in an effective and efficient manner?

1.4 Scope

In this thesis, we present a variety of methods for analyzing heterogeneous social networks, which contain multiple types of objects and relationships. Some special cases of heterogeneous networks are not in our research scope, e.g., multidimensional/mode networks [119] and composite networks [142, 141]. The multidimensional/mode network concept was proposed by Tang et al., [128] and has only one type of object and more than one kind of relationship between objects. Qiang Yang et al. introduced the composite network concept, where social actors in networks have various relationships, exhibit different behaviours in each individual network or sub-network, and share some common latent interests across networks at the same time [118].

In the first and second parts of this thesis we focus on developing novel techniques for analyzing heterogeneous networks in correlation schemas, where objects can be classified as either source type or attribute type.

Throughout the thesis, we constrain our research scope to capture only one type of relationship between objects. For example, our work considers only one relationship between an author and a paper like ‘authorship’ where the relationship between an author and a paper could be ‘authored by’, ‘co-authored by’, ‘edited by’, ‘recommended’ or ‘does not recommend’.

1.5 Objectives

In fact, social heterogeneity is a key element in contemporary online social networks that enables diversified perceptions of actors. Motivated by this consideration, this thesis seeks to improve traditional homogeneous social network analysis by taking the heterogeneity characteristic of networks into consideration, which can help unlock new potential that will allow our models to generalize across various types of social networks. To this end, our research objectives are:

1. To develop a novel algorithm that allows us to efficiently identify hetergeneous communities with strong internal connectivity and loose external connections.

2. To develop a framework that produces ranking lists of heterogeneous objects through community memberships and demonstrate that the framework outperforms existing benchmark techniques.

3. To create a heuristic that accurately embeds vectors of objects and models their changes in a dynamic heterogeneous network.

1.6 Contributions

Our contributions in social network and heterogeneous network analysis are listed below:

- Recent social network analysis seeks to leverage heterogeneous user behavior and diversified information resources. Our extensive reviews of social network analysis show that the vast majority of current techniques were proposed for homogeneous networks. Therefore, this thesis focuses on the development of a set of methods that take the heterogeneity characteristics of networks into consideration.

- We perform a comprehensive survey on modern social network analysis techniques and have highlighted the research gaps in heterogeneous network analysis.

- We show how metrics for measuring quality of homogeneous communities can be used for calculating internal and external connectivity of heterogeneous communities. We develop a novel heuristic that identifies tightly coupled communities and compare the quality of communities produced against benchmark techniques.

- We formulate a novel heterogeneous community detection and ranking technique and propose a model to rank multiple types of objects in a heterogeneous network while accounting for their importance within their respective communities.
- We propose a method to generate accurate embedding vectors of objects by modeling dynamic and heterogeneity features of networks. We then present this as a framework and evaluate dynamic heterogeneous networks and show that it can efficiently identify structural changes in the network.
- We apply our developed methods to real world heterogeneous social networks and compare them against state-of-the-art approaches.

1.7 Structure of Thesis

The primary focus of this thesis is to discover useful information and knowledge that the users can gain from different types of heterogeneous networks via a variety of techniques. An overall structure and overview of the methods developed in this thesis are shown in Figure 1.3. The mining tasks in all three chapters take a heterogeneous network as an input, process the network and output analyzed information. In addition, the heterogeneous communities identified in Chapter 4 are taken as input by Chapter 5 for community-based ranking of objects.

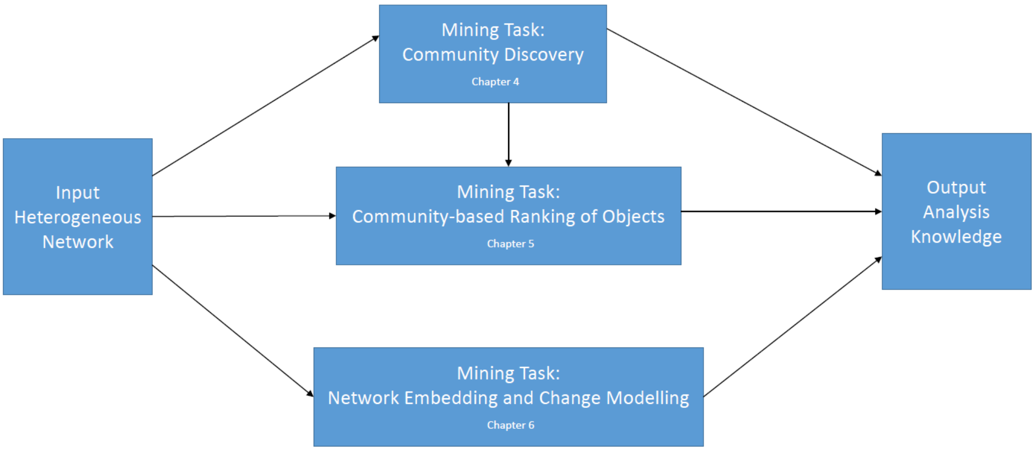


Figure 1.3: Overview of mining and analysis techniques proposed in this thesis

The remainder of this thesis is structured into the following chapters:

- *Chapter 2: Literature Survey*

  We provide an overview of relevant research in heterogeneous mining and analysis. In particular, we detail seminal research and review the overall state of the current research on community discovery, ranking and network embedding. Furthermore, we review methods for other important data mining tasks in heterogeneous networks such as clustering, classification and recommendation.

  We look at both existing homogeneous and heterogeneous techniques developed for the three network analysis tasks focused in the thesis: community detection, ranking and network embedding. To answer our research question and find out whether using heterogeneous information improves the results of network analysis, we adopt some of the homogeneous methods reviewed in this chapter as benchmark techniques in Chapters 4, 5, and 6.

- *Chapter 3: Identifying Top-k Nodes in Social Networks: A Survey*

  We provide a comprehensive survey on social network analysis with a focus on top-k nodes identification. We present a novel classification method of reviewing existing literature and discuss application domains and future research directions in the area.

  We conduct a more detailed literature review on one specific network analysis task, which identifies a research shortage in ranking in heterogeneous networks. This chapter serves as one motivation behind our research question and foundation work for Chapter 5.

- *Chapter 4: Uncovering Overlapping Heterogeneous Communities*

  We present our novel OHC algorithm, which is developed to discover overlapping heterogeneous communities in heterogeneous correlation networks. We show the advantages of OHC with extensive experimental results and analyze the results qualitatively in a case study.

  We explore whether using heterogeneous information improves the result of community detection in static heterogeneous correlation networks. To achieve the goal, two state-of-the-art homogeneous community discovery techniques are compared against the OHC algorithm.

- *Chapter 5: Community Based Ranking of Objects*

  We present our novel ComRank algorithm, which is developed to rank objects of different types in heterogeneous correlation networks by considering the object community memberships. We show the merits of

ComRank by comparing its performance in identifying top-ranked objects against three state-of-the-art baselines.

We explore whether using heterogeneous information improves the result of ranking in static heterogeneous correlation networks. To achieve the goal, a homogeneous ranking technique is compared against the ComRank algorithm.

- *Chapter 6: Network Embedding and Change Modeling in Dynamic Heterogeneous Networks*
  We present our novel change2vec algorithm, which is developed for network embedding and change modeling in dynamic heterogeneous networks. Experimental results show that change2vec outperforms two state-of-the-art methods in terms of efficiency and clustering accuracy.

  We explore whether using heterogeneous information improves the result of network embedding in dynamic heterogeneous networks. To achieve the goal, a state-of-the-art homogeneous network embedding technique is compared against the change2vec algorithm.

- *Chapter 7: Conclusions*
  We summarize the findings of this thesis, discuss limitations, and provide an outlook into the future.

2

Literature Survey

The growth of heterogeneous network mining and analysis is a research topic that has been flourishing for the past decade. Due to the complexity and broad range of sub-categories of this field of research, we narrow the scope of our review to the three core topics, community detection, ranking and network embedding. In this section, we review the most recent and relevant literature in each of these topics and draw comparisons between each proposed heuristic. Additionally, we analyse emerging techniques for staple data mining tasks in heterogeneous networks which include clustering, classification and recommendation. We then conclude this section with a discussion on gaps and weaknesses in existing research and the work described in this thesis addresses them.

To explore whether using the heterogeneous information improves the results of network analysis, we examine both homogeneous and heterogeneous techniques developed for the three core topics, community detection, ranking and network embedding. Some of the homogeneous techniques reviewed in this section are used as benchmark techniques in Chapters 4, 5 and 6.

2.1 Community Detection in Homogeneous and Heterogeneous Networks

Community Detection can be defined as the act of grouping sets of nodes that all nodes within the group are strongly and densely connected. In this section, we present existing community detection methodologies adopted for homogeneous and heterogeneous networks respectively. In addition, we draw comparisons between detection techniques of different homogeneity and highlight the additional complexities in heterogeneous network mining.

2.1.1 Community Detection Methods in Homogeneous Networks

In recent years, community detection in homogeneous networks has been researched widely from various perspectives. Some methods focus on identifying disjoint communities while others focus on overlapping communities [154]. Newman and Girvan [100] proposed that modularity can be used as a measure to divide the homogeneous network into a set of graph partitions. This idea has been influential in later community detection techniques, such as [13].

Speaker-listener Label Propagation Algorithm (SLPA) [154] has been shown empirically to be one of the best performing algorithms for both overlapping and disjoint homogeneous communities [155, 55]. The algorithm propagates all labels in each iteration to identify community membership between nodes of a given network. Louvain [13] is a popular homogeneous community detection algorithm based on modularity-optimization. This parameter-free algorithm is able to analyze a network with millions of nodes within seconds.

2.1.2 Homogeneous Community Scoring Functions

A homogeneous community scoring function assesses a group of nodes' connectivity level for representing a network community structure [157]. Triangle Participation Ratio (TPR) is a scoring function based on internal connectivity, which measures the fraction of nodes in a community that belong to a triad [148, 157, 88]. The value range of TPR is [0,1], where a higher ratio represents better internal connectivity and a value of 1 indicates a highly interconnected community. TPR has been widely recognized as a useful metric for measuring community density and cohesion. Furthermore, Yang and Leskovec's [157] experiments with 230 large real world networks highlighted TPR's high accuracy in measuring and identifying ground truth homogeneous communities.

Fraction Over Median Degree (FOMD) is another community scoring function based on internal connectivity [157]. This metric calculates the proportion of nodes in a community that have internal degree greater than the median degree of all nodes in the network. FlakeODF is a community scoring function that combines internal and external connectivity [41, 157] by calculating the fraction of nodes in a community that have fewer edges pointing inside than outside the community. Definition 1 presents the preliminary concepts for the three scoring functions.

Definition 1. Given the set of nodes S, we consider a scoring function $f(S)$ that categorizes the community quality of S. Let $G = (V, E)$ be a graph with $n = |V|$ nodes and $m = |E|$ edges. Let S be the set of nodes, where n_s is the number of nodes in S, $n_s = |S|$. In addition, we define $d(u)$ to be the degree of node u and d_m as the median degree value of $d(u)$ in V [157].

Based on the definition, the formula [157] for calculating TPR score is:

$$f(S) = \frac{|\{u : u \in S, \{(v,w) : v,w \in S, (u,v) \in E, (u,w) \in E, (v,w) \in E\} \neq \varnothing\}|}{n_s}$$

The formula for calculating FOMD score [157] is:

$$f(S) = \frac{|\{u : u \in S, |\{(u,v) : v \in S\}| > d_m\}|}{n_s}$$

The formula for calculating FlakeODF score [157, 41] is:

$$f(S) = \frac{|\{u : u \in S, |\{(u,v) \in E : v \in S\}| < d(u)/2\}|}{n_s}$$

2.1.3 Community Detection Methods in Heterogeneous Networks

The existing techniques for detecting communities in heterogeneous networks were developed for multi-relation with single-typed object schema [162], bipartite schema [93] or star schema [131]. Many of these techniques find communities that contain only a single type of object with one or more types of relationships. None of the existing techniques produce heterogeneous communities that contain multiple-typed objects and relationships.

In-depth studies has also been done on community identification in multiplex networks, which are complex heterogeneous networks with multiple dimensions or layers. Kuncheva and Monatana [81] proposed the detection technique, LART, to identify communities that are overlapping multiple layers of the network. Based on random walk, the heuristic is able to determine the inter-layer weightings between nodes while applying node dissimilarity measures to determine if they can transition between each layer. Approaching the problem from a different perspective, Hmimida and Kanawati [58] proposes a method of identifying communities in a multiplex network through a seed-centric approach. Their heuristic starts by identifying the initial set of seed (or *leader*) nodes, the remaining nodes in the network then rank the set of seeds node by a preference membership function in which it will join the top-scoring community.

Social networks are usually a popular target for community detection techniques due to the real world applications and value that they provide, hence there has recently been an increasing popularity focused sorely on analysing complex, heterogeneous, social network data. The SONAR algorithm proposed by Guy et al. [52] demonstrates ranking based clustering and formation of communities by analysing social networks from multiple sources and formed communities based on the weighted combination of each information source to provide end users with more complete and useful information.

Tang et al. [135] aimed to integrating different network dimensions and types of network interactions to produce different integration schemes. Each of the four integration scheme evaluates the network from a different perspective and utilizes different methodologies. When compared with each other the authors found that by studying the structural features of a multi-dimensional network through feature integration, they were able to produce more favourable NMI values which out-perform other evaluation schemes.

Sun et al. [130] attempts to detect the evolution of communities in a dynamic heterogeneous network. The authors adopt a Dirichlet Process Mixture generative model to identify communities in each generation. For each iteration and time stamp, the communities cluster based on historical and current networks which is evaluated by a Gibbs sampling-based inference algorithm. The authors then studied two real world data sets and proved that their heuristic can help detect the creation, convergence and destruction of communities.

2.2 Ranking in Heterogeneous Networks

There have been numerous ranking techniques proposed for homogeneous networks, including PageRank [19] and HITS [76]. In the past decade, some co-ranking algorithms were proposed to address the heterogeneity of the network. Zhou et al. [165] were among the first that studied the co-ranking problem in heterogeneous networks. The authors proposed a PageRank-paradigm method for co-ranking authors and their publications. Their link analysis approach uses three networks: the social network of authors, the citation network of papers, and the authorship network which connects the previous two networks together. Adapting the co-ranking framework [165] to the tweet recommendation task, Yan et al. [156] developed a model to rank tweets and their authors simultaneously using three-sub networks: the network connecting the authors, the network connecting the tweets, as well as a bipartite network that ties the author and tweet networks together. While both co-ranking algorithms [165, 156] were designed for heterogeneous networks with specific sub network structures, a large proportion of research on this topic assumes the networks follow a certain schema. For example, RankClus [129] assumes that the target heterogeneous network is based on the bi-typed schema and NetClus [131] is proposed to analyse a network based on the star schema.

MultiRank [101] is a co-ranking scheme designed to evaluate and rank multiple types of objects and relationships. The framework iteratively solves sets of tensor equations to obtain stationary probability distributions that can effectively be used in ranking objects and relationships simultaneously. One of their experiments analyzed academic data sets based on author to author collaborations. MultiRank was shown to converge after ten iterations and the results indicated that the top rank authors generally have higher citations and collaborations. GPNRankClus [28] integrates clustering and ranking together on a heterogeneous network following an arbitrary schema. The method models the ranking score of each node in each cluster as a Gamma distribution, and the number of edges between two nodes as a Poisson distribution. Their experiments on the DBLP data set show that GPNRankClus is able to produce interesting and explainable co-ranking results.

2.3 Network Embedding in Heterogeneous Networks

In recent years, developing network embedding methods has attracted a considerable amount of research interest. These methods fall into three major categories based on the network type that the method was designed for. (1) Network representation learning algorithms that were designed for static homogeneous networks, such as DeepWalk [109] and node2vec [48]. The word2vec [94] framework uses the skip-gram architecture to learn the distributed representations of words in natural language. Both DeepWalk and node2vec were built on word2vec and leveraged the two-layer neural network structure in word2vec. (2) Representation learning models which were designed for static heterogeneous networks, e.g., metapath2vec++ [37] and HNE [26]. Similar to DeepWalk and node2vec, metapath2vec++ was also inspired by word2vec's skip-gram model to perform node embeddings. The algorithm captures heterogeneous structural correlations among multiple types of nodes from metapath-guided random walks. (3) Unlike the first two categories, where the network and the learned vector representations are fixed, there also exist network embedding methods for dynamic homogeneous networks [167, 172]. DynamicTriad [167] employs the triad closure process [59] to learn dynamic latent representations of nodes. Chang et al. [26] proposes a multi-layered, deep embedding function that considers both the local network content and overall topological network structures. Their approach allows them to transfer objects in a heterogeneous network to form unified vector representations.

2.4 Other Data Mining Tasks in Heterogeneous Networks

2.4.1 Clustering

Similarity evaluations are conducted between heterogeneous networks to determine how similar two networks and their underlying objects are to each other. These similarity measures are often used in conjunction with clustering heuristics to form clusters of objects that have similar attributes with each other while being dissimilar to objects of other clusters. The work by Sun et al. [128] highlights this as an algorithm, PathSim, which identifies clusters according to the metapath-based similarity. Most existing similarity measures are only applicable for homogeneous networks as each path in a heterogeneous network could have different semantic meanings and values. The authors isolate this problem by identifying peer objects, which are objects with similar attributes, properties and subtle similarity semantics.

Instead of clustering based only on topological structure of a graph from a high level perspective, Zhou et al. [169] puts heavy emphasis on vertex properties. Their proposed clustering algorithm, SA-Cluster, considers both structural and attribute similarities through a unified distance measure by splitting the network into k clusters based on node attributes. When applied to real world academic and social networks, SA-Cluster was shown to produce more dense and coefficient clusters when compared to its baseline techniques. The authors developed their heuristic further and proposed a more efficient algorithm, Inc-Cluster, that incrementally updates random walk distances based on edge weight increments [170]. This incremental process was shown to reduce run time significantly while achieving the same quality of clustering with SA-Cluster.

Developed by Sun et al. [129], the RankClus heuristic studies multi-typed heterogeneous networks and forms clusters effectively through ranking and node membership information. They focus on networks with a star network schema and produce an algorithm that recalculates cluster memberships for nodes in an iterative manner. Instead of calculating each pairwise similarity between objects, which is expensive and time-consuming, RankClus identifies an initial set of loose net-clusters then readjusts objects iteratively by building posterior probabilistic generative models, allowing them to determine if objects in an cluster should change cluster assignments. Their evaluations on DBLP academic data sets were shown to produce informative communities with more focused attributes than baseline techniques.

2.4.2 Classification

Ji et al. [65] identified the problem with transductive classifications in heterogeneous network data, where only some objects in a given network are labeled, hence, the authors aim to predict labels for the remaining objects. Their prediction heuristic, GNetMine, combines the structural information of the network, network schema and number of objects and relationships to predict the classification models of objects with non-attributed data. Their results on academic DBLP data sets show that the quality of existing labels is an important factor and significantly influences the classifications results and that their heuristic out-performed existing graph-based transductive classification heuristics such as LLGC [164], in terms of accuracy as they were mostly applicable for homogeneous networks.

Building upon their previous work, Ji et al. [64] studied the effectiveness of combining both ranking and classification of nodes when analysing heterogeneous networks. Their proposed algorithm, RankClass, is a ranking-based classification framework that iteratively builds ranking models and re-adjusts so each object is aligned with one of the pre-specified classifications and that the inter-class rankings are also adjusted. Each sub-network from a specific class was then extracted from the global network as they are defined and all the specified classifications were present, the authors applied posterior probability to each object, to determine each object's optimal class membership, similar to NetClus [129].

Kong et al. [78] studies the problem of multi-label classifications in heterogeneous networks. Real world objects represented in a heterogeneous network usually has more than one label associated due to the nature of its complexity. Additionally, label correlations are sometimes not given and could be hard to extract from the given data. Hence, the authors proposed a multi-label classification process which allows them to infer correlations with class labels effectively by studying the linkage structure of networks and class labels of sample data. They evaluated their heuristic based on metrics including Micro-F1 and Hamming Loss to show that when using heterogeneous network data, their heuristic out-performs other multi-label classification heuristics as it takes metapath-based label correlations into consideration.

2.4.3 Recommendation

The recommendation problem in the field of data mining targets the real world scenario of producing high quality suggestions and relevant recommendations based on user feedback. This problem is widespread and applicable for many domains such as streaming services, online shopping and social networks hence there has been an increasing amount of research arising in this field due to its demand.

As noted by Yu et al. [159], the recommendation problem mostly lies in "attribute-rich heterogeneous networks" where each entity has multiple labels and attributes. They propose a learning model that combines multiple vectors of relationship information along with user feedback to produce a high quality recommender system. The authors accomplish this in a two stage process, firstly by receiving user implicit feedback and applying Bayesian ranking optimization to highlight positive observations. The authors then extract latent features of the data set and capture different recommendation factors and semantics. When applied to real world IMDB and Yelp feedback data sets, their proposed global recommender system, HeteRec, achieved higher mean reciprocal rank (MRR) values than other techniques, highlighting the value of integrating learning models with similarity semantics in the network. The authors further extend this work by developing personalized recommendation systems [158], where they integrate user preference diffusion process into the metapaths of the network. With this new framework, the new recommender, HeteRec-p, accounts for all user interactions with objects in the networks and puts an associated weighting into the recommendations. Results on the same data set shows that HetecRec-p, again, outperforms baseline techniques, including their original global recommender, HeteRec.

Shi et al. [120] highlights the weakness of existing recommenders where they do not consider attribute values on links and propose the semantic path based recommender, SemRec, that applies weightings to meta paths. Through this, the authors are able to integrate personalized user weightings and preferences in the form of paths with heterogeneous information. The effectiveness of weighted meta paths is highlighted in the experiments, as shown when applied to real world data sets, SemRec was shown to achieve lower Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) values compared to other techniques, especially when there was little to no training data.

2.5 Summary of Literature

In this section, we have highlighted novel heuristics and literature in various fields of heterogeneous network mining, e.g., community detection, ranking, clustering and network embedding techniques. For each of these topics, we study the emerging techniques that solve common problems when analysing complex heterogeneous networks.

Community detection has been a widely researched topic on homogeneous networks with a large variety of scoring function to qualitative evaluations such as Triangle Participation Ratio (TPR) and Fraction Over Median Degree (FOMD). However, techniques for heterogeneous networks are less common due to the complexity of identifying communities when vertexes and edges have multiple attributes, hence existing heterogeneous community detection techniques usually focus on single types of objects with multiple types of relationships. We have outlined several techniques such as LART [81], an heuristic that detects communities with overlap multiple layers in a multidimensional network through inter-layer weightings and node similarity measures, and SONAR, an algorithm [52] that produces ranking-based clustering and community detection from propagating weighted information from multiple sources to determine community membership of nodes.

Similar to community detection, ranking techniques have been studied extensively for homogeneous networks but rarely in networks with high heterogeneity due to the large amount of vectors and variances. As one of the leading researchers in this field, Zhou et al. [165] studied the co-ranking problem in heterogeneous networks and proposed a heuristic similar to PageRank that enabled co-ranking and link analysis of complex heterogeneous academic networks that contained author, paper citation and co-authorship information. Similarly, Multirank [101] is an co-ranking scheme that ranks multiple types of objects and relationships. The heuristic iteratively solves sets of tensor equations to obtain probability distributions that is used in the evaluation of ranking of objects and relationships. Finally, GPNRankClus [28] approaches the problem by integrating both clustering and ranking to produce co-ranking results.

The study of network embedding methods has grown considerably with new techniques and evolution built upon existing frameworks such as node2vec [48] and Deepwalk [109]. Another technique noted was DynamicTriad [167] that uses triad closure process to learn dynamic latent representations of nodes.

Finally, we studied several other core data mining tasks in heterogeneous networks, including clustering, classification and recommendations. We highlighted how tightly integrated similarity evaluations are with clustering algorithms such as PathSim [128] applying this through node peerings and similarity semantics. We also noted authors that choose to solve this problem from a topological approach, with SA-Cluster [169] and Inc-Cluster [170] evaluating the structural properties of the network. Recommendations in heterogeneous networks is a widespread problem and is invaluable due to the real world applications. Yu et al. [159] proposes a solution of integrating user feedback with the recommender system, their results show that by applying user preference into the network metapaths [158], they were able to produce higher quality recommendations.

2.6 Performance Evaluation Metrics

Based on our literature review on various network mining and analysis tasks, we summarize the most common evaluation criteria used by researchers to measure the performance of algorithms designed for community detection, ranking and network embedding, which are the three main tasks focused on in this thesis.

1. Community detection: Researchers in the field of physics first proposed community structure as sets of nodes with high density of internal edges and low density external edges [83, 100]. The concept was later adopted by computer science researchers [157, 88]. The majority of existing work in the area measures internal connectivity and external connectivity of their identified community structures [55, 58, 81]. We use these two

common evaluation metrics in our heterogeneous communities detection work in Chapter 4.

2. Ranking: Zhou et al. [165] were among the first that studied the co-ranking problem in heterogeneous networks and they used domain-oriented metrics to evaluate their ranking results. For example, citation numbers for academics and publications. Adapting the co-ranking framework [165] to the tweet recommendation task, Yan et al. [156] developed a model to rank tweets and their authors simultaneously and they also used domain-oriented metrics to evaluate their ranking list. For example, number of re-tweets of tweets, popularity and influenceability of twitters. Most existing co-ranking work [101, 28, 129] selected domain-dependent metrics for evaluating the performance of their algorithms and we use this common evaluation method in Chapter 5.

3. Network embedding: Embedded vectors generated by network embedding algorithms are usually used as input for data mining tasks such as classification, clustering and recommendation [48, 109, 37]. In most cases, the outcome is then compared with ground truth information using metrics such as NMI (Normalized Mutual Information) [129]. NMI is a similarity measurement score and is in the value range of [0,1] with 1 indicating that the results are identical to the ground truth. In Chapter 6, we also use NMI to compare clustering results generated from our developed approach with Google Scholar's author and venue clusters.

2.7 Research Gaps

A common factor among existing research in the study of heterogeneous network mining and analysis is the use of static, predefined data. This is understandable due to the complexity of heterogeneous networks and authors choosing to focus on analysing singular time stamps. There is, however, no research that views and analyses networks as a stream of data. By viewing multiple time stamps of a heterogeneous network and integrating an additional dimension, time, to our analysis, we can find changes and trends in the network that highlight the evolution of communities which can help us to make predictions. This is invaluable in many real world scenarios and we attempt to analyse concurrent time stamps in Chapter 6.

Existing heterogeneous network ranking techniques focus on the attributes of individual objects and their relationships in the network before performing ranking from a topological viewpoint. However, in these heuristics, the communities where the objects reside are not considered. In Chapter 5, we propose a heuristic that integrates community detection with modern ranking methodologies to produce an object ranking heuristic that integrates their global ranking of objects as well as the local ranking within their respective communities.

3

Identifying Top-k Nodes in Social Networks: A Survey

Top-k nodes are the important actors for a subjectively determined topic in a social network. To some extent, a topic is taken as a ranking criteria for identifying top-k nodes. Within a viral marketing network, subjectively selected topics can include: Who can promote a new product to the largest number of people and who are the highest spending customers? Based on these questions, there has been a growing interest in top-k nodes research to effectively identify key players. In this chapter, we review and classify existing literature on top-k nodes identification into two major categories: top-k influential nodes and top-k significant nodes. We survey both theoretical and applied work in the field and describe promising research directions based on our review. This research area has proven to be beneficial for data analysis on online social networks (OSNs) as well as practical applications on real-life networks.

Top-k nodes identification is closely related to one of the three core topics of this thesis - ranking. We focus on this to identify research gaps in heterogeneous network analysis with a more specific topic.

3.1 Introduction

In this fast-paced digital age, people around the world are now more connected with others than ever before. Of the many communication and collaboration channels, social networks have become very popular among different communities. The general public frequently use connection social networks, such as Twitter and Facebook, to express opinions on trending topics. Marketers construct informational networks to observe consumer buying behaviors. Research collaborations are recorded in academic networks such as DBLP, which is a bibliographic reference on major computer science publications.

A social network can be modeled as a graph which consists of nodes and edges. Each node represents an actor while each edge shows a connecting relationship between two actors. Intuitively, the representation of nodes and edges varies accordingly to interested actors and relationships in the network. For example, in Twitter, each user can be represented as a node while an edge is formed when there is a "following" relationship between user X and Y. Furthermore, each Tweet (a posting message made on Twitter) can also be a node. An edge exists between Tweet A and B, if B is a retweet A's message (a Tweet message forwarded by someone else). In viral marketing networks, each consumer can be a node while an edge is formed if a consumer successfully persuades another person to purchase a new product. In DBLP, individual nodes represent different academics while edges indicate that the connected researchers have published a paper together.

An important factor for an organization's success is the ability to advertise their products to potential customers. To do this efficiently the organization may want to know who are the high-priority customers that should be targeted? Top-k nodes identification in social networks is an extension of this question. Identifying key nodes can be useful for increasing product adoption rates in advertising or searching for domain experts. Research in this field has received a lot of interest due to three benefits: (1) It brings order to search results so that the contributing nodes can be ranked by their significance, authority and/or influence [124], (2) It can be utilized to increase the efficiency of marketing and advertisement campaigns [69], (3) Its ability to improve the utility of gathered information [150].

Nowadays, with the massive amounts of data available, it is not always practical to analyze everything in a dataset due to resource constraints. Instead, sometimes, it is more effective to explore the most significant or influential actors (the top-k nodes) in a network. In this survey, we review and classify the current work on identifying top-k nodes in social networks. As seen from existing literature, this research area has applications to different domains, such as advertisements in viral marketing [69], information circulation [49] and domain experts search [171, 124].

Since the first algorithmic formulation of the top-k nodes identification problem in 2001 [36], there have been a large number of publications on various kinds of algorithms and applications in this area. With over a decade of research, it is time to perform an overview of this field and examine what more can be done in this research area. First, we provide general descriptions of concepts that are used extensively in the rest of this chapter. Please note that more detailed descriptions of these concepts are provided later in Subsections 3.2.6, 3.2.7 and 3.2.8.

Topic A topic is the area of user interest and can be used as a ranking criteria for identifying top-k nodes.

Top-k nodes (T) Top-k nodes are the k important actors for a subjectively determined topic in a social network, where k is a user-specified integer.

Top-k Influential nodes (I) Top-k influential nodes are the k actors that are capable of generating maximum influence and widest information spread to their connected nodes in a social network, where k is a user-specified integer. Top-k influential nodes will sometimes be shortened to influential nodes in this chapter.

Top-k Significant nodes (S) Given a particular topic, top-k significant nodes are the k actors whose intrinsic attributes are most relevant to the given topic in a social network, where k is a user-specified integer. Top-k significant nodes will sometimes be shortened to significant nodes in this chapter.

3.1.1 Related Surveys

In comparison with other related surveys [84, 126, 50, 111, 114], our work has three distinct differences:

1. We extend the focus from either influential or significant nodes to a broader concept - the top-k nodes. As a result, our survey includes not only research on influence maximization but also studies on identifying the significant nodes in social networks.

2. As opposed to some existing surveys, e.g., [114], which focuses on only one type of social network (Twitter in this case), our work reviews related work in a wide range of social networks, such as Amazon, DBLP and Wiki.

3. More comprehensive and recent literature are reviewed and an extensive coverage of the subject is provided, i.e., applied work in different application domains, and top-k influential nodes identification in dynamic social networks, which were not addressed in previous surveys.

3.1.2 Contributions

Our main contributions are summarized as follows. First, we define an extended concept: Top-k nodes, to provide more comprehensive coverage of the field. Second, both theoretical and applied work of identifying the top-k nodes are reviewed and classified in a novel way. Finally, some promising research directions are discussed based on our survey.

3.1.3 Survey Organization

In this chapter, we conduct a high-level overview of the top-k nodes identification algorithms, methodologies and applications. With a rich body of literature in this area, we organize our discussions into the following four topics: (1) Top-k influential nodes identification, (2) Top-k significant nodes identification, (3) Applications of identifying top-k nodes in various networks, and (4) Research directions. The remainder of the chapter is also organized in the corresponding four sections (Sections 3.3 to 3.6). In addition, we provide some preliminary concepts in Section 3.2, and conclude the survey study in Section 3.7.

3.2 Preliminaries

We introduce the social network related preliminary concepts, followed by traditional node centrality measures, and then different influence diffusion models for identifying top-k influential nodes. Finally, formal definitions of influence maximization, influential nodes and significant nodes are provided. In addition, relationships among top-k nodes, influential nodes and significant nodes are discussed.

3.2.1 Social Network

In our survey, social networks contain two concepts. Firstly, it is a network of social interactions and personal relationships. This kind of social network existed long before the likes of Facebook, Twitter and has mainly been used to describe entity relationships. Early research on this type of network was conducted by social scientists. It has become an important research area in computer science in recent years. Secondly, social networks are profile sites or virtual communities where users can share interests and ideas, or discover friends through posting comments, messages and images.

3.2.2 Static Network versus Dynamic Network

As dynamic networks evolve, new nodes and edges will be introduced. An example of this is shown in telecommunication networks, where transient links are added between two participant nodes based on texts or calls between them. These dynamic networks with transient interactions can be represented as graph streams, however due to high computational complexity and disk storage requirements, these graph streams typically require real-time methods. In contrast, static networks have fixed topology, structure and information, which can be treated as a snapshot of the dynamic network at a distinct time t. Therefore, offline computational methodologies can be applied on these static networks [4].

3.2.3 Social Network Graph

A social network can be modeled as a graph $G = \{V, E\}$, where V is a set of nodes $\{v_1, v_2, ..., v_i\}$ and E is a set of edges $\{e_1, e_2, ..., e_j\}$. In the network graph, V represents actors and E represents social interactions and relationships between the actors. For example, when an edge e_j exists between nodes v_l and v_p, the two corresponding actors are considered connected/related to each other. Neighbour nodes of v_i is represented as $N = \{n_1, n_2, ..., n_m\}$, which refers to all nodes that are directly connected to v_i. Social networks will be described occasionally as *network graphs* in this chapter.

3.2.4 Node Centrality Measures

In existing literature, a number of node-based centrality measures have been proposed and defined for evaluating the importance of nodes in selected network graphs. As this survey focuses on identifying top-k nodes, we introduce two of the measures which are relevant to this survey.

Degree Centrality

Historically first and conceptually the simplest node measure is degree centrality, where a "degree of a node is the number of edges the given node has" [147]. Degree centrality c_{v_i} of node v_i is defined to be the degree of the node (Equation 3.1):

$$c_{v_i} = deg(v_i). \tag{3.1}$$

In the academic collaboration network (Figure 3.1), the degree centrality of Diana node is 4. From the perspective of traditional social network analysis, a node with a larger degree centrality is normally considered as more important.

Closeness Centrality

"Closeness centrality is defined as the average length of shortest paths between node v_i and all reachable nodes of v_i in the network graph" [147]. The closeness centrality is represented in Equation 3.2 below:

$$Closeness(v_i) = \frac{\sum\limits_{v_j \in R \backslash v_i} dist(v_i, v_j)}{|R| - 1}, \tag{3.2}$$

Given that R is the group of nodes which can be traversed from node v_i and that R contains the node v_j, the function *dist* returns the length of shortest path between node v_i and node v_j [137]. The shortest path indicates the minimum number of edges connecting two nodes.

In the academic collaboration network (Figure 3.1), the shortest path between Diana and Lucy is 1 rather than 2 (Diana - Celina - Lucy). Following Equation 3.2, we can calculate that the closeness centrality of Diana node is 1. From the perspective of traditional social network analysis, a node with a smaller closeness centrality is normally considered more important.

3.2.5 Influence Diffusion Models

Since influential nodes are an essential component of the top-k nodes, we introduce two influence diffusion models which are widely used for exploring influence of nodes. When modeling the "spread of an innovation through a social network graph G, each node v_i can have either an *active* (an adopter of the innovation) or *inactive* (not yet an adopter) status" [69]. Two ground settings for influence diffusion models discussed in this chapter are:

1. v_i has monotonically increased tendency to become active as more of its neighbour nodes n_m (previously defined in Section 3.2.3) become active.

2. v_i can switch in only one direction: from being inactive to being active.

The influence diffusion model progresses as follows for an initially inactive node v_i: With the unfolding of this process, an increasing number of v_i's neighbour nodes n_m become active. At certain points, this might influence v_i to become active and v_i's decision might subsequently trigger a status change of the remaining inactive n_m. The process runs until no further triggering can happen [69].

Linear Threshold (LT) model [47] and Independent Cascade (IC) model [45] are two of the most basic and widely-studied influence diffusion models. LT is receiver-centric where the core idea is to find out whether a node can be activated given a fraction of active neighbour nodes. Whereas IC model is sender-centric and focuses on individual node's activation attempts on its inactive neighbour nodes. Therefore, the two models reflect different views on the influence diffusion process.

Linear Threshold Model

In the model, a node v_i is influenced by each neighbouring node n_m according to a weight w_{v_i,n_m}. For any inactive node, v_i, we select a *threshold* θ_{v_i} between the interval [0,1]. The value θ_{v_i} represents the weighted fraction of v_i's neighbour nodes n_m that must be active for v_i to become active, this represents the tendencies of nodes being influenced by neighbouring active nodes, becoming activators themselves (Equation 3.3).

$$\sum_{n_m\, active\, neighbour\, of\, v_i} w_{v_i,n_m} \geq \theta_{v_i}. \tag{3.3}$$

Given a random set of active nodes, A, and random thresholds, θ_{v_i}, for each inactive node, the diffusion process will begin to iterate in deterministic and discrete steps. For each iteration, t, all nodes that were active in steps $t-1$ or lower will remain active and if the total neighouring weight, n_m, for a target node v_i exceeds its threshold θ_{v_i}, v_i will be activated, this process will continue until no more activations are possible. [69, 47].

Independent Cascade Model

Also starting with an initial set of active nodes, A, the independent cascade model performs diffusion based on the following randomized rule. For a given iteration, t, if a target node, v_i, was activated in step $t-1$, it will have the chance to activate each inactive neighbouring node, n_m, based on a history-independent probability p_{v_i,n_m}. If v_i succeeds in activating n_m, n_m will become and remain active in steps $t+1$ onwards, repeating the same process for its neighbouring nodes and v_i will no longer be able to influence the network. [69, 45].

3.2.6 Influence Maximization Definition

Based on the influence diffusion models described in the previous subsection, the influence maximization problem is defined as follows:

Definition 2 (Influence Maximization)**.** "Influence maximization is an optimization problem, which requires selection of a good initial set of active nodes A. The influence of A is measured by the number of active nodes at the end of an influence diffusion process. The influence maximization problem aims at finding a k-node set of maximum influence" [69]. The meaning of active nodes varies with interested topics. Possible meanings can include but is not limited to: adoption of a new research area, purchase of a recommended product or receiving a new piece of information.

3.2.7 Influential Nodes versus Significant Nodes

Given the influence maximization definition in Section 3.2.6 and the definition of Topic in Section 4.1, we formally define top-k influential nodes and top-k significant nodes.

Definition 3 (Top-k Influential Nodes)**.** Let $G = \{V, E\}$ denote a social network graph, where $V = \{v_1, v_2, ..., v_i\}$ is a set of nodes and $E = \{e_1, e_2, ..., e_j\}$ is a set of edges in G. We define top-k influential nodes as a user-specified number of v_i which leads to maximum influence spreading to their connected nodes in G.

Definition 4 (Top-k Significant Nodes)**.** Let $G = \{V, E\}$ denote a social network graph, where $V = \{v_1, v_2, ..., v_i\}$ is a set of nodes and $E = \{e_1, e_2, ..., e_j\}$ is a set of edges in G. We define top-k significant nodes as a user-specified number of v_i, whose intrinsic attributes are most relevant to a specified topic in G.

3.2.8 Relationships among Top-k Nodes, Top-k Influential Nodes and Top-k Significant Nodes

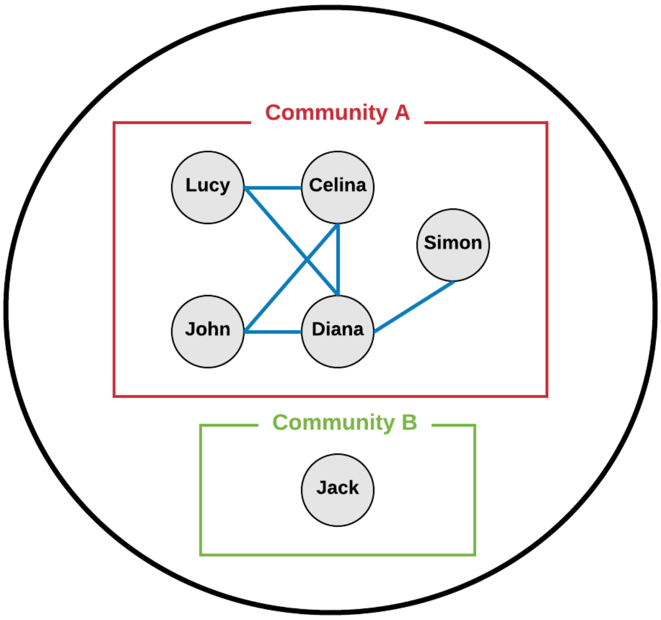
A given social network can contain a wide array of topics, such as "who are the influential users?" and "who are the subject experts?". From the perspective of viewing the network under the context of a single topic as opposed to the set of all possible topics, we illustrate the relationships between Top-k nodes (T), Top-k Influential nodes (I) and Top-k Significant nodes (S). When we consider only a single topic for a given social network, if the topic is more relevant to influence maximization of k number of nodes to their connected nodes, then the set of top-k nodes will be the same for both top-k influential and top-k significant nodes, we define this as the Top-k Influential Nodes Scenario. However, if we view the topic based on intrinsic attributes (such as authority or representativeness) of k number of nodes rather than their influence on neighbouring nodes, then the top-k nodes will be the same as the set of top-k significant nodes, which we define as the Top-k Significant Nodes Scenario. *From the perspective of all possible topics in a social network, top-k influential nodes is a subset of top-k significant nodes, denoted as* $I \subseteq S$.

In a viral marketing network, each existing or potential customer is a node (actor) while an edge is formed between two customers if they have social connections. Topics of interest such as "Who can promote a new product to the largest number of people?" are examples of influence maximization on groups of customers and can be considered as examples of Top-k Influential Nodes Scenario. Whereas topics that are focused on non-influential properties of the nodes such as "Who are the highest spending customers?" are examples of the Top-k Significant Nodes Scenario.

To further illustrate differences between the Top-k Influential Nodes Scenario and the Top-k Significant Nodes Scenario, we employ an academic collaboration network graph (Figure 3.1). In the graph, each node represents an individual academic. An edge between two nodes exists if there is a research collaboration between the two corresponding academics. Red and green rectangles indicate two separate research communities A and B in the network. The table in Figure 3.1 shows information derived from the academic collaboration network: the number of collaborators each academic has and the research community that a particular academic belongs to. Based on Figure 3.1, we provide examples of the Top-2 Influential Nodes Scenario and the Top-2 Significant Nodes Scenario below:

Top-2 Influential Nodes Scenario Topic of interest: Who are the two most influential academics in the network? By having the largest numbers of collaborators, Diana and Celina will have a higher chance of promoting new research ideas to co-workers than others. In this case, Diana and Celina are the top-2 nodes, the top-2 influential nodes and the top-2 significant nodes.

Top-2 **Significant Nodes Scenario** Topic of interest: Who are the two academics that represent all existing research communities in the network? In this case, Jack and Diana are the top-2 nodes as well as the top-2 significant nodes. Despite being an isolated researcher, Jack will still be selected as one of the top (significant) nodes due to his unique representativeness of the research community B. While the most-connected node, Diana, in research community A is the top (significant) node in her community.



Academic	Number of Collaborators	Research Community
Diana	4	A
Celina	3	A
Lucy	2	A
John	2	A
Simon	1	A
Jack	0	B

Figure 3.1: Academic collaboration network

Table 3.1: List of some approaches for top-k influential nodes identification

Category of Approach	Related Research
Greedy	Kempe et al.[69] Kimura et al.[75] Leskovec et al.[87] Chen et al.[32]
Centrality Measure	Ilyas and Radha[61] Chen et al.[27] Kim and Yoneki[72] Wei et al.[149]
Topic	Tang et al.[133] Liu et al.[90] Barbieri et al.[11] Aslay et al.[9]
Network Content(or Topology)	Wang et al.[146] Subbian et al.[125] Zhou et al.[166] Chen and He[29]
Dynamic Network	Subbian et al.[122] Zhuang et al.[174] Subbian et al.[123, 124]

In the academic collaboration network (Figure 3.1), when the topic of interest is more relevant to maximum influence (promoting new research ideas to collaborative academics), Diana and Celina are the top-2 influential and significant nodes. However, when the topic of interest is shifted to find the total number of communities in the network, the concept of influential nodes is no longer applicable and Jack and Diana become the top-2 significant nodes in the network.

3.3 Top-k Influential Nodes Identification

As we progress further into the digital age, there has been a steady increase in the importance and influence of social media and networks. The ability for users to share their thoughts, statuses and activities in these social mediums has established a new level of connectivity between different groups and niches of people, such that a single user can influence millions of their followers. Hence, it would be of interest for companies to select these influential individuals as their initial user group in order to produce the greatest level of coverage when advertising their products.

We start our survey with approaches for identifying influential nodes. Different kinds of approaches for identifying influential nodes are reviewed in Sections 3.3.1 to 3.3.4. In addition, Section 3.3.5 discusses a relatively new research area: Identifying Top-k influential nodes in dynamic social networks. Table 3.1 provides an overview of research work covered in this section.

3.3.1 Greedy based Approaches and Improvements

One important category of approaches for identifying influential nodes is based on greedy algorithms. "The core idea of the algorithm is to calculate the influence of each individual, and take turns to choose the node maximizing the marginal influence value until k number of influential nodes are selected" [146]. Figure 3.2 illustrates this core idea in a procedural way: the process of influential nodes identification begins with analysing the nodes' influence and then it iteratively maximizes the influence of an initial set of nodes. The final goal is to obtain a group of initial nodes with maximum influence, which are the influential nodes. To describe the process of information propagation, the majority of work reviewed in this section use influence diffusion models [69, 87, 31], such as the Linear Threshold [47] or the Independent Cascade model [45] (previously described in Section 3.2.5).

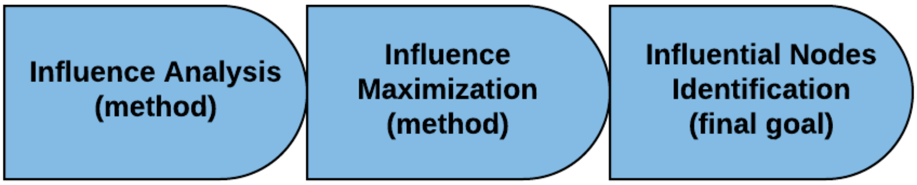

Influence Analysis (method)
Influence Maximization (method)
Influential Nodes Identification (final goal)

Figure 3.2: Core idea of greedy algorithms for identifying influential nodes

In 2001, Domingos and Richardson [36] were the first to explore information and influence propagation as a computational problem. With their proposed probabilistic solution, they designed viral marketing strategies and analyzed diffusion processes using a data mining approach. Two years later, Kempe, Kleinberg, and Tardos [69] categorized the problem of finding the most influential individuals as an optimization problem. Additionally, Kempe et al. provided the first provable approximation guarantees for the influence maximization problem, where the influence propagation process is modeled to reflect the effects of "word of mouth" for the "promotion of new products" [69]. When identifying top-k influential nodes, we are interested in finding the most influential "mouths" which can generate the largest possible influence cascade. The problem statement contains two sub-problems: "The most influential mouths" describes the problem of finding the top-k influential nodes. Whereas "generate largest possible influence cascade" corresponds to the influence analysis and maximization problem. These two sub-problems are both incorporated into Section 3.3.

Kempe et al. define influence maximization as "a problem of identifying a small set of seed nodes in a social network that maximizes the spread of influence under certain influence diffusion model" [69]. Although there are various models for influence propagation in network graphs, the authors [69] choose the Linear Threshold [47] and Independent Cascade [45] models (previously described in Section 3.2.5) in their research. On the basis of submodular functions, the proposed analysis framework demonstrated that the natural hill-climbing greedy algorithm that achieves "a solution that is provably within 63% of optimal" [69]. In conjunction to the provable guarantees, experiments were also conducted to show that their approximation algorithm significantly out-performs node-selection heuristics based on degree and closeness centrality (previously described in Section 3.2.4). One of their major findings observed that focusing only on clustered centrality-based nodes may not generate maximum influence. On the contrary, targeting nodes with most possible additional marginal gain results in superior performance over the two centrality-based heuristics.

Following their research on influence maximization through a social network [69], Kempe, Kleinberg, and Tardos [70] define a natural and general model for the influence propagation process, termed the decreasing cascade model. The model begins with a set of "active" nodes that spreads influence in a cascading way depending on a probabilistic rule. Their problem statement focuses on choosing a target set of individuals for initial activation so that the cascade process is able to result in a largest possible active set. Kempe et al. provide provable approximation guarantees for selecting a target set of size k using a simple greedy algorithm in the proposed decreasing cascade model. A research direction described is to investigate which are the most general influence diffusion models, and what provable performance guarantees can be achieved for those models. Kimura and Saito [74] proposed two natural special models (SPM, SP1M) of the Independent Cascade (IC) model [45] such that they can effectively compute the number of influenced nodes given an initial set of influential nodes. Similar to Kempe et al. [70], the authors also provided provable performance guarantees for the simple greedy algorithm in the proposed models. Their experiments with large-scale social networks demonstrated that when using the proposed two models for identifying influential nodes: (1) They can provide close approximation to the IC model if the propagation probabilities between links are small, (2) They can be scalable and much faster than the IC model.

Leskovec et al. [87] developed a "lazy-forward" optimization method for choosing a group of nodes to detect out-break, i.e., the spread of virus, as early as possible. Their proposed algorithm, CELF (Cost-Effective Lazy Forward), greatly reduces the number of calculations on the node influence propagation, which gains up to 700 times more efficiency over the simple greedy algorithm. Chen, Wang and Yang [32] attempted to further improve the efficiency of identifying influential nodes from the following two directions: (1) Develop new algorithms (NewGreedy and MixedGreedy) on top of the simple greedy algorithm. However, this resulted in unremarkable efficiency improvements, (2) Designed a new heuristic algorithm: DegreeDiscount, which is more efficient and scalable than the greedy strategy. However, the DegreeDiscount heuristic is a derivation of the *Uniform Independent Cascade model* where propagation probabilities of all edges are identical.

DegreeDiscount's limitation is later addressed in a study by Chen, Wang and Wang [31] with a new heuristic which accommodates the general Independent Cascade (IC) model [45]. The authors underline two critical weaknesses with existing influential node discovery techniques: (1) Existing algorithms have poor scalability, thus will perform poorly with large-sized graphs, i.e., Kempe et al. [69] and Leskovec et al. [87]; (2) These algorithms have either low scalability or have un-influential initial nodes and therefore have low influence spread.

To resolve these two issues, Chen et al. adopted a simple tune-able parameter for users to control "the balance between efficiency (in terms of running time) and effectiveness (in terms of influence spread) of the algorithm" [31]. Their solution results in a more efficient greedy algorithm when selecting nodes in each iteration. When comparing the performance of their algorithm with existing techniques such as simple greedy [69], their results were shown to be: (1) More scalable with linear growth in running time beyond million-sized graphs while others are exponential and (2) Faster execution time and spread of influence for both real-world and synthetic datasets.

Chen, Yuan and Zhang [30] closed an open question left by Kempe, Kleinberg, and Tardos [69] by proving that influence computation in the Linear Threshold (LT) model [47] is NP-hard. In addition, the authors showed that "computing influence in directed acyclic graphs (DAGs) can be done in linear time" [30]. Based on the fast computation in DAGs, Chen, Yuan and Zhang proposed a "scalable influence maximization algorithm tailored for the LT model" [30].

Narayanam and Narahari [98] developed a novel way of discovering influential nodes in social networks with the Shapley value concept [117], which is commonly used in cooperative game theory. The Shapley value-based Influ-

Table 3.2: Comparisons of some greedy based approaches for identifying influential nodes

Approach	Heuristic Approach	Performance Guarantees	Scalable
Simple Greedy[69]	No	Yes	No
Improved Greedy[75]	Yes	No	Yes
CELF[87]	No	Yes	No
NewGreedy[32]	No	Yes	No
MixedGreedy[32]	No	Yes	No
DegreeDiscount[32]	Yes	No	Yes

ential Nodes (SPIN) maps the information diffusion process to "the formation of coalitions in a cooperative game" [98]. For computing the network values of each node, Shapley values are used to determine the marginal contributions each node makes to the influence propagation process. The experiments with both synthetic and real-world datasets showed that SPIN works comparatively well as the simple greedy strategy [69] for maximizing influence yet by consuming much less running time.

Liu et al. [91] provided a bounded linear approach for identifying influential nodes in large-scale social networks. A quantitative metric, termed Group-PageRank, which can be computed in near constant time, is also proposed to address the scalability issue of the influence maximization problem. The authors developed two "lazy-forward" greedy algorithms based on the bounded linear approach and the Group-PageRank, respectively. The evaluation results showed that the two greedy algorithms can effectively and efficiently identify influential nodes with both of them being scalable for large-scale social networks.

We compare some greedy based approaches for identifying influential nodes in Table 3.2. The simple greedy algorithm [69] provides the first ever provable performance guarantees for the influence maximization problem. However, it is computationally expensive and not applicable for large-scale social networks and later approaches address the efficiency and scalability limitations. The improved greedy algorithm proposed by Kimura et al. [75] achieves a large reduction in computational cost by removing edges that do not contribute to information diffusion and does the propagation on a subgraph. The CELF algorithm optimizes the simple greedy algorithm using the "submodularity property of the influence maximization objective" [87] to reduce the number of calculations on the node influence propagation. Chen et al. [32] proposed three different algorithms: NewGreedy, MixedGreedy and DegreeDiscount. The key idea behind NewGreedy is to eliminate the edges that will not contribute to influence propagation from the original graph to get a smaller graph and performs the influence diffusion on the smaller graph. The first iteration of MixedGreedy employs the NewGreedy algorithm while the remaining iterations use CELF algorithm. DegreeDiscount assumes that influence spread of a node is increased with the increase in degree centrality of the node. The authors suggested that DegreeDiscount should be adopted when efficiency is essential while MixedGreedy can be used for identifying influential nodes when maximum influence spread is a priority. [146]. An interesting fact shown in Table 3.2 is that scalable greedy based approaches lacks provable performance guarantees. This interesting fact is also described as "algorithms applies various heuristics without provable approximation guarantee" [121].

3.3.2 Centrality Measure based Approaches

In terms of influence maximization, some research in Subsection 3.3.1 experimentally demonstrated that their greedy based approaches out-perform traditional centrality measures, such as the degree and closeness centrality [147] (previously described in Subsection 3.2.4). However, there are existing approaches for identifying influential nodes using centrality measures. In this subsection, we review new centrality measures that have been shown to be more effective than traditional measures for maximizing influence.

Ilyas and Radha define centrality as *"a measure to assess the criticality of a node's position"* [61]. The ability to find influential nodes is shown in many existing methodologies, such as Eigenvalue Centrality (EVC) [14]. However, these techniques often target a single set of influential nodes and cluster them within one neighbourhood of nodes. This does not reflect the properties of social network graphs, where there could be multiple influential neighbourhoods. The approach proposed by Ilyas and Radha [61] uses Principal Component Centrality (PCC) to form social hubs, groups of nodes in a network, whose centrality scores are higher than their neighbours. While EVC forms a single cluster of nodes with the highest centrality scores, PCC considers additional factors, such as the weighting of eigenvectors, when computing centrality of nodes. The authors applied both EVC and PCC to real-world datasets (i.e., Facebook and Orkut) and found a significant increase in the number of influential neighbourhoods discovered.

Chen et al. [27] underlined the issues with using traditional centrality measures for influential node identification. Degree centrality methods are simple but irrelevant while closeness centrality methodologies are inapplicable for large-scale networks due to the computational complexity and running time. The authors then proposed a semi-local centrality measure to balance between both centrality measures. Their results on four real-world networks showed much faster computational efficiency while providing more effective results than degree centrality methodologies. However, this methodology was not compared with greedy-based approaches for mining influential nodes.

Kim and Yoneki [72] studied the problem of maximizing influence diffusion through social networks. The authors underlined the weaknesses of existing techniques that use arbitrary node propagation and proposed the Influential Neighbors Selection (INS) scenario to select the most effective group of neighbouring nodes to propagate. Four selection strategies were proposed and the results showed that *highest degree selection* had favorable results for short-term diffusions but *random selection* performed better in long-term scenarios. *Highest weight and volume selections* showed similar results to *highest degree selection* but the additional communication costs meant they were less favorable.

Based on the Dempster-Shafer evidence theory [33, 116], Wei et al. [149] proposed a new evidential centrality measure for identifying influential nodes in weighted networks. Their approach considers not only degree and weight of nodes but also status of the nodes in a weighted network. By experimentally comparing with other measures such as the degree and closeness centrality, the proposed measure showed comparative performance for identifying influential nodes. Again, the evidential centrality was not compared with any greedy based approaches for top-k influential nodes identification. This common trait of centrality measure based approaches attracts our attention to the issue. Future methodologies in this category need to compare performances with greedy based approaches to be more convincing on effectiveness and efficiency of their proposed measures.

The techniques presented in this subsection showed many similarities. Both Ilyas and Radha [61] and Wei et al. [149] evaluate centrality measures while using Susceptible-Infected related models to evaluate the performances of their proposed models. The evaluation of traditional centrality measures (degree centrality, betweeness centrality and closeness centrality) was also a common theme in the work by Chen et al. [27] and Wei et al. [149] and both studies present heuristics that outperform these centrality measures in terms of influence diffusion while maintaining lower computational complexity. Kim and Yoneki [72] adopt the Independent Cascade (IC) model to select the most influential neighbours of a node. Experimental results on two evaluation metrics were presented. The first metric was Pearson correlation coefficient between closeness centrality and the proposed four selection strategies. For the second

Table 3.3: Comparisons of various centrality measure based approaches

Approach	Identify Influential Node Neighbourhoods	Identify Influential Nodes
PCC[61]	Yes	Yes
Semi-local Centrality Measure[27]	No	Yes
Four Selection Strategies[72]	Yes	Yes
Evidential Centrality Measure[149]	No	Yes

metric, those selection methods were compared against each other by computing the ratio of the number of activated nodes to the total number of nodes in the network. The similarities and differences of these approaches are summarized in Table 3.3.

3.3.3 Topic based Approaches

In a viral marketing network, top-k influential nodes are those when convinced to adopt a product, shall influence others in the network and lead to a largest possible number of new adoptions. Although real world product purchasers have different degrees of interest on various topics, e.g., some customers are only interested in latest smart-phones while others tend to pay more attention to new computer models, however, both the greedy and centrality measure based approaches are topic-blind. In this aspect, the two kinds of approaches treat all possible topics in a network as if they were the same, and focus only on general inherent attributes of nodes (such as marginal influence gain or centrality measure value) regardless of any particular topic or context. This problem is addressed in [133, 90, 11, 9] and we review these topic based approaches in this subsection.

The Topical Affinity Propagation heuristic proposed by Tang, Sun, Wang and Yang [133] describes the topic-aware influences in large-scale social networks. This tool allows users to perform meaningful and valuable influence analysis on real-world datasets by: (1) Finding the influential nodes on a given topic and (2) Identifying the social influences of the neighbouring nodes for a particular node. The authors accomplish this by integrating learning algorithms into their Topical Factor Graph (TFG) model. The features of the TFG model allow users to find both the local information of nodes (such as topic-level influences and next most probable propagation) and global information (connectivity between any two nodes). In conjunction to these components, the authors adopted distributed learning techniques to train the TFG model due to the size and complexity of large networks. The programming model, Map-Reduce, partitions large social networks into subgraphs so multiple machines can process and collect values associated with nodes in both internal and external subgraphs. When adopting the TFG model with real-world datasets, their results showed that the distributed learning approach has good scalability performance and topic-level influences can improve the performance of influential nodes finding.

Liu et al. [90] focused on investigating how to mine topic-level influence in heterogeneous networks. The authors solve the problem in two steps: (1) They proposed a model which combines the "heterogeneous link" [90] information with the text associated with nodes in the network to determine topic-aware direct influence; (2) From the direct influence that was validated, a topic-aware algorithm was proposed to determine indirect influence between nodes. Their experiments with Twitter, Digg, and citation networks showed that the proposed approach can unveil useful influence patterns in heterogeneous networks.

Barbieri et al. [11] extended the traditional Independent Cascade (IC) model [45] to a Topic-aware Independent Cascade (TIC) model. Their experiment results showed that TIC model can describe real-world influence driven propagations more accurately than the state-of-the-art topic-blind models.

As a first step towards enabling social-influence online analytics in support of viral marketing decision making, Aslay et al. [9] proposed an efficient index for a general type of topic-aware viral marketing queries. Given the computational challenges related to the enormous number of potential queries and some other aspects, the authors employed a tree-based index, INFLEX, to obtain a solution for a limited number of possible queries. Their experiment results showed that with the index, the targeted queries can be answered in milliseconds instead of several days compared to existing offline computation methodologies.

Zhou et al. [166] explored topic or preference-based top-k influential nodes identification in social networks. The proposed mining algorithm, GAUP, finds influential nodes on a given topic in two stages: Firstly, computing user preferences and projecting them into a "reduced latent space and a VSM-based model" [166]. In the second stage, GAUP utilizes the greedy based approaches, e.g., CELF [87] to find influential nodes for a particular topic. When evaluated with an academic network, GAUP was shown to maximize influence spread for a given topic.

Although identifying topical influential nodes in social networks is an interesting research direction, there hasn't been significant research performed

Table 3.4: Inputs, outputs and characteristics of various topic based approaches

Work	Inputs	Outputs	Characteristics
TFG[133]	• a social network • a prior topic distribution for each node (inferred input)	topic-wise user-to-user influence strength	• works for large-scale networks • ability to find both the local and global information of nodes
Topic-level Influence Mining in Heterogeneous Networks[90]	a heterogeneous social network with nodes which are users and documents	topic distribution and user-to-user influence	a probabilistic model for the joint inference of the topic distribution and topic-wise user-to-user influence
TIC[11]	• a log of past propagations • model parameters	more accurate descriptions of influence driven propagations	proposed Topic-aware Independent Cascade (TIC) model by extending Independent Cascade (IC) model
INFLEX[9]	a directed social graph where the edges are associated with a topic-dependent user-to-user social influence strength	a set of users that should be targeted in a viral marketing campaign for a given topic	• answer queries online within few milliseconds • based on the TIC model

in this area. From three different aspects, Table 3.4 compares four topic based approaches reviewed in this subsection.

3.3.4 Network Content or Topology based Approaches

With the increasing quantity of content in social networks, it is possible to conduct content-centric mining of influencers. In this subsection, we review network information (or network structure) based approaches for identifying top-k influential nodes, which provides a different perspective from the topic based approaches.

Community-based Greedy Algorithm (CGA) [146] takes the community property of mobile social network (MSN) into consideration. Two major components of the CGA are: An algorithm for community detection and a dynamic programming model for choosing communities to identify influential nodes. Wang et al. extended the Independent Cascade model [45] to account for the edge weight of MSNs and provide provable performance guarantees for CGA. In terms of efficiency and approximation error rate, CGA is proven to outperform some state-of-the-art greedy based approaches for identifying top-k influential nodes. Similar to Wang et al. [146], Bozorgi et al. [18] also utilized the network community structure in their research. A community-based algorithm, INCIM, was proposed for identifying influential nodes under the Linear Threshold model [47]. To summarize, the network community structure is used to find the influential communities while a node's influence is determined by a combination of its local and global influences. Their evaluation results demonstrated that INCIM outperforms other approaches, such as the simple greedy algorithm, in the quality of the identified influential nodes while maintaining a reasonably low running time and memory consumption for large graphs.

PageRank (PR) [20] and Topic-sensitive PageRank (TSPR) [56] were originally proposed for the purpose of ranking webpages. However, in recent decades, a number of research on identifying influential nodes adopt, extend and integrate PR or TSPR into their own methodologies [29, 150]. Some existing research utilize PR and (or) TSPR in their experimental comparison analysis [115, 150]. Therefore, we briefly introduce the core ideas of these two algorithms: PageRank applies academic citation literature and orders webpages by calculating the number of citations and backlinks. Detailed descriptions of the PageRank algorithm can be found in [106]. PageRank has been often adopted in research for identifying influential nodes due to numerous reasons: Firstly, in a graph composed of tens of millions of webpages, each webpage can be represented as an individual node; Secondly, Brin and Page [20] state that "*a page can have a high PageRank if there are many pages that point to it, or if there are some pages that point to it and have a high PageRank*". As correspondingly for top-k influential nodes, a node can have a high influence if there are many nodes that directly connect to it, or if there are some nodes that directly connect to it and have a high influence. In 2002, Haveliwala [56] proposed the Topic-sensitive PageRank by computing multiple PageRank vectors, biased using various topics, to capture more accurately the notion of importance regarding each particular topic.

Romero et al. [115] acknowledged the importance of user passivity when finding influential nodes in social media. The proposed general model for influence analysis uses the concept of passivity in a social network, and develops an algorithm for quantifying the influence of all the nodes in the network. Their method utilizes the network structural properties as well as the diffusion behaviors among users. The influence of a user is determined by both the size of the influenced audience and their passivity. It is claimed that the model outperforms other measures of influence, i.e., PageRank [20] and degree centrality. The authors stated that high popularity does not necessarily imply high influence. Similar opinion is also expressed in the greedy based approaches such as [69].

Subbian et al. [125] questions the lack of actual social value of network collaborations in influence diffusion models, and defined individual social captial as social value generated through collaborations with peers of high influence. Through their proposed algorithm, SoCap, they were able to find influencers based on the hypothesis that people with high social capital indicates high influence within a network. Due to this, SoCap differentiates itself from other measures of influence, e.g., degree centrality and PageRank, as it finds high social value nodes through multiple collaborations. Unlike the greedy based approaches, SoCap does not use any underlying influence diffusion models. A value-allocation model is also developed to compute the social capital and allocate the fair share of this capital to each individual involved in the collaboration.

A Context-Aware and Trust-Oriented influencer-finding method [171], named CT-Influence, incorporates social contexts, such as the preferences of participants with the social relationship and trust between participants. The experiments with two real-world datasets showed that CT-Influence greatly outperforms SoCap [125] in terms of effectiveness and efficiency for identifying influential nodes.

Chen and He [29] take hostile relations in Online Social Networks (OSNs) into consideration when integrating the PageRank algorithm [20] on signed OSNs. The authors used the integrated PageRank to discover influential nodes in OSNs with both friend and hostile relations, which correspond respectively to positive and negative edges on signed networks. Their experiment results for selecting top-k influential nodes on real-world datasets indicated that the proposed method performs better than some algorithms, such as the original PageRank [20].

In this subsection, we reviewed network topology or content based approaches for identifying top-k influential nodes. The network topology oriented methods focus on exploiting the community structure property of social networks. The content based techniques tend to extract various aspects of information from a given social network, such as friend and hostile relations, user passivity, social value of collaborations, the preferences of participants, the content or topic of tweets etc. For the presented techniques, we notice that PageRank was a popular comparison technique and a Twitter dataset was frequently analyzed. We provide comparisons of techniques discussed in

Table 3.5: Comparisons of various network content or topology based approaches

Approach	Network Topology based Approach	Network Content based Approach	Key Characteristics
CGA[146]	Yes	No	• community detection • dynamically select communities to find influential nodes
INCIM[18]	Yes	No	• influential community detection • the influence of each node is determined by its local and global influences
IPR[29]	No	Yes	• signed social networks • friend and hostile relations are represented as positive and negative edges on signed networks respectively
TwitterRank[150]	No	Yes	the topical similarity between users and the link structure are both taken into account
IP[115]	No	Yes	determines the influence and passivity of users based on their information forwarding activity
SoCap[125]	No	Yes	captures the individual social capital
CT-Influence[171]	No	Yes	considers the social trust and relationships between participants and the preferences of participants

this subsection in Table 3.5.

3.3.5 Identifying Top-k Influential Nodes in Dynamic Social Networks

In the previous four subsections, we have addressed literature on top-k influential nodes identification in static social networks. This subsection discusses the existing work on identifying influential nodes in dynamic graphs, which is a relatively new research area. The associated computational challenge is a major difference in the two network settings of static and dynamic networks. Static social network can be considered as a snapshot of dynamic network, which cannot fully represent some characteristics of real-world social networks, such as continuous network topology change, high-speed data transmission, large population of participants and uncertain information diffusion processes. More computational resources are potentially demanded due to these distinct characteristics of dynamic social networks.

Aggarwal et al. [5] are among the first who study influential nodes identification in dynamic social networks. A stochastic approach is designed by the authors to identify the information flow authorities with two types of methods: a globally optimized forward trace approach, and a locally optimized backward approach. In addition, methods for determining the approximately optimal release points for a given pattern of information spread are also proposed. The performance of proposed methods was evaluated on both DBLP and ArnetMiner citation datasets.

Zhuang et al. [174] underlined the influence maximization problem in dynamic networks as probing nodes in an unobserved network. Maximum Gap Probing (MaxG) algorithm was proposed to provide an approximate optimal solution to probe a subset of nodes which can best unravel the actual influence diffusion process in the network. The authors claimed that MaxG is a general method and can be directly used to guide online marketing decisions in social networks. Experiment datasets used for evaluating the performance of MaxG were constructed from Twitter and the coauthor network of ArnetMiner. Unlike in MaxG, which focuses on probing a subset of nodes in a dynamic network, Han et al. [54] adopt a divide-and-conquer strategy to capture the global evolution of a dynamic network by only probing the most active communities. Extensive experiments were conducted by the authors on Epinions, Slashdot, Twitter and Inventor networks.

A content and network-based flow mining approach for dynamic influence analysis is proposed by Subbian et al. [122]. In their research, sequential patterns were dynamically mined from a combination of the keywords and the dynamic network in the social stream. These were then used to discover the most influential nodes in a dynamic and evolving network. Three years later, the authors develop an algorithm, InFlowMine, [123] to determine flow patterns through content propagation on the dynamic network structure. The identified patterns were utilized for determining topic- or network-specific influential users, or their combinations. Unfortunately, not all Online Social Networks (OSNs) can provide sufficient context for discovering information flow patterns. Hence, this streaming method can not be generalized for finding social influencers across OSNs. Three datasets were used in their evaluation experiments, which are Twitter, DBLP and US patent datasets.

Vadoodparast and Taghiyareh [139] focus on maximizing product adoption in dynamic networks by proposing a multiagent framework named MAFIM. The framework contains two kinds of agents: modeling agents and solution provider agents. A dynamic network is viewed as consecutive static snapshots by these agents and according to a selected budget assignment policy, each snapshot obtains its share from the budget defined by the sales manager. Their experiment results on real-world dataset, such as Slashdot, demonstrate that it is more effective to launch many short-lived campaigns rather than few long-lived ones.

Subbian et al. [124] proposed the first general keyword-based influence query and tracking model for streaming scenarios. With the constant evolution of topics over time, which potentially results in different identified influencers, the authors developed a method to maintain real-time influence scores of users in a social stream based on topic and time-sensitive information. The core idea of the method is to track information flow patterns in a tree-like data structure across various paths of the network in a context and time-sensitive fashion. The data structure facilitates one pass computation of influencers for different contexts.

A Dynamic Independent Cascade (DIC) model and the concept of adaptive seeding strategy were proposed by Tong et al. [136]. Using a simple greedy algorithm, the authors provided a provable performance guarantee for their solution based on the DIC model. Additionally, an efficient heuristic algorithm with better scalability was also introduced. The superiority of the adaptive seeding strategies was demonstrated by empirically comparing with the state-of-art non-adaptive seeding approach [69] and random strategy. Two kinds of real-world social networks used in their experiments were Hep and Wiki. Hep contains academic collaboration data of co-authorships in physics while Wiki includes the Wikipedia voting data [86] from the inception of Wikipedia.

Song et al. [121] designed an algorithm, called UBI, to solve the influential node tracking on dynamic social network problem. The core idea of their approach is to start from the influential seed set which was identified previously and implement node replacement to maximize the influence coverage. Three datasets evaluated in their experiments are: mobile, HepPh and HepTh networks. A very recent technique proposed by Wang et al. [145] adopts the concept of sliding window to maintain “a set of k seeds with the largest influence value over the most recent social actions”. The authors collected two real-world datasets: Reddit and Twitter for their experiments.

Since many social networks, such as Twitter, are often available only in the form of social streams of user activity, there is a surge of recent research on social streams. However, our survey indicates that a very limited amount of work has been done on identifying top-k influential nodes in dynamic social networks. Among those works, a minority of them provide provable performance guarantees [136, 145, 54] or empirically compare their proposed methods with dynamic network oriented approaches [145, 54]. Three major categories of techniques presented in this subsection are: 1) overall structure and information diffusion of the dynamic network focused approaches [5, 136]; 2) network content or content flow patterns based approaches [122, 123, 124]; 3) identifying influential nodes in dynamic networks by modelling a subset of the network with the subset being a set of nodes or communities [174, 139, 121, 145, 54]. We conclude Subsection 3.3.5 by presenting two facts that exist among these works:

1. *Model Change:* Traditional influence diffusion models (such as IC [45], LT [47]) have a static network structure and edge propagation probabilities. Although they are widely used in the greedy based approaches for identifying influential nodes, they are not adopted in dynamic network settings. Instead, more flexible models are proposed to accommodate dynamic network structure and edge propagation probabilities.

2. *Content-Centric Method:* Network content, such as information flows, topic or time-sensitive data are well incorporated and utilized in approaches for identifying influential nodes in streaming social networks.

3.3.6 Section Summary

Since the first formalization of finding the influential actors problem by Kempe et al. [69], there has been an increasing amount of research and development in the area. An interesting phenomenon observed is that the focus of this research area has been shifted from efficiency and scalability to dynamic networks in recent years. With various features provided by a wide variety of algorithms, applications will be required to select suitable algorithms to accommodate for its specific problem domain when mining top-k influential nodes.

Node and edge attributes have also been an underlying theme for top-k influential node identification. The research by Chen and He [29] was based on friend and hostile attributes between nodes in the network, creating positive and negative edges on signed networks. Zu et al. [171] incorporates social contexts into the network analysis, integrating humanistic factors into the properties of participating nodes.

3.4 Top-k Significant Nodes Identification

In this section, we focus on literature relevant to significant nodes identification. By reviewing recent work, we hope to shed some light on this research area.

3.4.1 Effector-based Identification and Bridge Nodes

When we find nodes in a social network in a particular activation state, such as when a certain topic is popular, we can often observe the subset of root nodes that propagate the information and activate the neighbouring nodes in the network, known as *effectors*. Unlike influential nodes which focus on centrality measures, effector nodes are key connectors in a network due to their property as bridges between peripheral nodes and groups even when they might not have a high degree of neighbouring nodes themselves and therefore, if removed, usually cause networks to be fractured and disjoint.

Given a social network graph G and an activation vector a, Lappas et al. [85] defined k-effectors to be the set of nodes, once activated, that cause an activation pattern which is as similar as possible to the activation procedure observed at a. The authors proved that the k-effectors problem can be solved in polynomial time for social networks represented as a tree. This is accomplished through a dynamic programming approach by specifying the maximum k-effectors of sub-trees under one of the two following approaches: The root of the sub-tree is included in the set of effectors for the next recursion and then recurses on children with (k-1) budget. The second approach does not include the root and the children are recursed with k budget. With the method described, the authors were able to extract the most probable active tree that spans all the active nodes of the network. Using this, they could identify the optimal set of effectors in the social network tree and in return, they were able to extract interesting observations on the network and interactions between significant nodes.

The research by Li et al. [89] attempts to identify sets of **star nodes** (which are also the significant nodes in this case) based on the scale of connectivity loss if the nodes were removed from the network. The authors account for typical immunization methodologies, such as acquaintance immunization, and performed analysis on two generic social networks. The authors discovered that certain immunization strategies, such as selected immunization, cause entire networks to be disjoint if the targeted star nodes were removed, while random and acquaintance immunization provides less destructive results with better running times. This methodology separates itself from previous work, such as those by Lappas et al. [85] as it views effectors from a different perspective. Immunization strategies are more generally adopted to prevent outbreaks and in this case, to identify the loss of connectivity in the network if the top-k significant nodes are removed.

Borgatti [16] underlines the weaknesses of top-k node identification using centrality-based approaches, where they do not account for key players' role in a network's cohesiveness. The author proposed a model that ranks significant nodes by both their centrality measures (KPP-Pos) and loss of graph cohesion if removed (KPP-Neg).

Borgatti's [16] research incorporates concepts from Lappas et al. [85] by valuing a significant node's centrality while also weighting the consequences if the node was breached. Similar to Li et al. [89], this research underlines the importance of the top-k effectors from a loss of network connectivity perspective. This concept has also been described by researchers such as Musial and Juszczyszyn [97] as **bridge nodes**. In their research, Musial and Juszczyszyn describe bridge nodes as anchoring nodes that connect peripheral nodes with the rest of the network and if lost, can cause diffusion loss between nodes in the network. The authors conducted an experiment on the Thurman office social network in an attempt to identify and extract bridge nodes and their properties. From their results, they have established that the bridge nodes were usually nodes with the highest social position and can be categorized based on their neighbouring connections with peripheral nodes or cliques.

3.4.2 Authoritative-based Identification

With the significant growth of the Internet, the ability to find sources of information with high levels of quality and authority has become increasingly difficult. With many Online Question and Answer portals facing this problem, there has been an emergence of research aimed to distinguish the top-k set of significant nodes (or authoritative users in this context). Farahat et al. [39]

analyses documents scattered in the World Wide Web and determines the reliability and authoritativeness of these documents based on textual, non-topical cues. In their research, the authors discovered that when querying certain subjects, documents found by PageRank [20] were sometimes uninformative and even controversial. To estimate the authority of documents more accurately, the authors combined the analysis of textual content of documents with its linguistic features and found that this approach was often able to rank authoritative documents produced by professionals and subject-matter experts higher than PageRank.

Zhang et al. [161] attempts to identify a set of users with high expertise on a Java Forum. In their research, they evaluated several network-based ranking algorithms such as HITS Authority [76] and separated users into five expertise ratings. From their results, they found several behavioral patterns among users of different expertise levels, such as newbies making few posts and experts answering other users' questions while asking very few themselves. These asker-helper interactions overlap with several properties of effector-based identification where information from a single source (which in this case, is knowledge of the answer to a specific question) is able to activate downstream lower level users.

Jurczyk and Agichtein [67] present authoritative node identification in a more controlled environment. The authors crawled through almost half a million questions on Yahoo! Answers portal with three million corresponding answers in the attempt to estimate the authority of users while using the HITS algorithm [76] as a baseline. For each question, the authors extracted the number of answers, the sum of the answers' voted values and the average number of stars of the best answer's author and drew comparisons between each of these categories based on the Pearson correlation coefficient. What they found was that authority was easier to identify in particular subject domains and using votes and stars produced results based on the popularity and quality of the feedback respectively. Unlike the research by Farahat et al. [39], Jurczyk and Agichtein's dataset contains several properties which the authors leveraged.

Unlike scattered documents in the World Wide Web, the datasets from Zhang et al. [161] and Jurczyk and Agichtein [67] separated questions into distinct subjects and domains which reduces the possibility of irrelevant nodes or answers being found for a particular query. Secondly, with each question, there exists a set of user-defined properties (such as the Voting and Star system in Yahoo! Answers) which exposes a new layer to evaluate the quality and authority of a particular node, underlining the value of useful node attributes to users and researchers. All of the research presented attempts to identify top-k significant nodes from various angles but the effectiveness and accuracy of the algorithms are heavily influenced by the dataset and its attributes.

3.4.3 Academic Datasets

In terms of identifying top-k significant nodes, there are some other studies investigating the significance of researchers in academic co-author networks. Nascimento et al. [99] investigated the co-authorship graph obtained from all papers published at SIGMOD between 1975 and 2002. They utilized the evolution of minimum closeness centrality scores as the ranking criteria to evaluate the significance of authors. Moreover, Liu et al. [92] built a weighted directional model to represent the co-authorship network, for which they defined AuthorRank as an indicator for the prestige of an individual author in the network. Under the assumption that program committee members can be regarded as prestigious actors in the field, their results are validated against conference program committee members in the same time period.

The evaluation results showed that the use of AuthorRank has clear advantages over traditional centrality measures, e.g., degree and closeness metrics. Zhou et al. [165] acknowledge the remarkable success demonstrated by graph-theoretic approaches for ranking networked entities. However, they also pointed out that the majority of the methodologies can only be utilized on homogeneous networks, such as the citation network. To rank significant authors and documents, Zhou et al. proposed a framework for co-ranking entities of two different types in heterogeneous networks. Their method couples two random walks into a combined one to co-rank authors and their publications using information retrieved from several networks: the social network connecting the authors, the citation network connecting the publications, as well as the authorship network that ties the previous two together. Their results suggest that the rankings of authors and documents depend on each other. Zaïane et al. [160] used an iterative random walk algorithm to evaluate the relevance between significant authors for the purpose of discovering research communities. Their core idea is to use a random walk on the bipartite or tripartite model of DBLP data and generate a relevance score to measure the closeness between two entities. The relevance score is then used to rank entities based on importance of a given relationship.

3.4.4 Other Work

Among the numerous existing top-k significant node heuristics, PageRank [20] and the HITS [76] algorithms are the most popular and considered as baselines for many later techniques. As described previously in Subsection 3.3.4, PageRank is a top-k heuristic that determines the quality and relevance of a webpage based on the search topic, which behaves like our definition of significant node identification. The HITS algorithm queries the World Wide Web in two steps: (1) A sampling stage that constructs a collection of webpages which are likely to be relevant authorities. (2) A weight propagation step which iteratively estimates a hub and authority score for each webpage and returns the highest scoring authorities of each topic. HITS has led to a number of future works such as work by Joel et al. [95] which enhances the original HITS algorithm with exponential inputs and the work by Wu et al. [153] which integrates collaborative tagging to support community detection, user and document recommendations.

Tseng and Chen [137] proposed a novel node ranking methodology for general real-world applications. This technique contains an unsupervised learning algorithm which requires no training data to produce a ranking list of top-k significant nodes. The authors implemented their methodology and experimented on a co-author network of from the DBLP computer science bibliography. The network was structured as an undirected graph where each node represents an author, and each edge between two nodes indicates publication collaboration between two authors. This process consists of two parameters and major phases. The user-defined parameters are the set of desired features and the number of significant nodes wanted. The first phase was an offline procedure where several features are extracted from each author in the network and sorted lists of author features are prepared, these lists are used as the input of the ranking algorithm in the second phase. The primary principle of the ranking strategy is to find top-k significant nodes with overall ranks higher than others. The methodology was able to generate different ranking lists when diverse sets of desired features are considered.

In addition to the ranking methodology, Tseng and Chen [137] also claimed that the definition of significant nodes is application-dependent as it varies with circumstances in different networks formed by diverse kinds of social connections. In order to accommodate different application characteristics, the proposed methodology requires a set of desired features as one user-specified parameter. While this technique provides a wide range of features to meet a variety of service demands, it has the trade-off of requiring more feature-related data and lowering efficiency for processing the two separate phases. The preparation of sorted author lists demands large volumes of input data on various features to be processed offline, which might not always be realistic for real-world applications. Additionally, adequate data might not be readily available and the application could require streaming data to be processed online rather than offline. The degree and closeness centrality measures are the two of four desired features in the experiment evaluation, where Tseng et al. avoid giving a specific weight on each desired feature for the purpose of providing a more objective assessment. It is highly likely that the resulting weighting-free ranking algorithm is not suitable for applications requiring subjective assessment.

3.4.5 Section Summary

From the categories described in Section 3.3, we found that processes for identifying top-k influential nodes were mostly greedy based approaches. Whereas for top-k significant nodes, we find a wider array of different methodologies adopted. Section 3.4.1 describes the characteristics of effectors, star nodes and bridge nodes in social networks. While in Section 3.4.2, methodologies for identifying authoritative nodes are presented.

Additionally, we emphasized the importance of node attributes and network properties, on how they can affect the methodology and process for identifying significant nodes and studying the network. The study by Zhang et al. [161] and Jurczyk and Agichtein [67] both contained datasets with very specific node and edge properties which influenced the direction of their research and heuristics.

In addition, there is a paper by Lappas et al. [84], where more details on topics such as algorithms and systems for expert location in social networks are discussed.

3.5 Applications of Identifying Top-k Nodes in Various Networks

With the tremendous increase of usage in Online Social Networks (OSNs), research in the identification of top users of OSNs such as Facebook, Twitter and LinkedIn has increased over the last decade. In this section, we review numerous applications of identifying top-k nodes and some of these applications showed how the use of content can enhance the identification process. It is evi-

Table 3.6: List of key applications of top-k nodes identification

Domain	Related Research
Twitter	Cha et al.[25] Weng et al.[150] Bakshy et al.[10] Drakopoulos et al.[38]
Facebook	Heidemann et al.[57] Kim and Han[71]
Blogosphere	Gruhl et al.[49] Java et al.[63] Java et al.[7] Huang et al.[60]
Misinformation Control	Budak et al.[21] Nguyen et al.[102]
Community Question Answering	Opsahl et al.[105] Zhang et al.[161] Guo et al.[51] Pal and Konstan[108]
Networks with Complex Topologies	Opsahl et al.[105] Wei et al.[149] Zhou et al.[165]
Miscellaneous Applications	Ghosh and Lerman[43] Aral and Walker[8]

dent from the discussion of this section that top-k nodes identification is useful for a wide variety of inter-disciplinary domains such as Twitter, or Blogosphere. Due to the limitation of length, we focus on only a small number of successful applications: (1) Twitter (Sect. 3.5.1), (2) Facebook (Sect. 3.5.2), (3) Blogosphere (Sect. 3.5.3), (4) Misinformation Control (Sect. 3.5.4), (5) Community Question Answering (Sect. 3.5.5), (6) Networks with complex topologies (Sect. 3.5.6) and (7) Miscellaneous Applications (Sect. 3.5.7). An overview of the key applications are summarized in Table 3.6.

Networks with Complex Topologies Miscellaneous Applications

3.5.1 Twitter

With the rising popularity and usage of online social networking mediums in the past decade, knowledge on how information is shared and distributed to users on these mediums has become increasingly valuable to many corporations that seek to advertise their products and services. Among these networking services, one of the most common is Twitter.

The social infrastructure of one of the most notable micro-blogging services, Twitter, is composed of twitterers, users that publish/provide tweets (with a limit of 140 characters) or content, to all of their connected followers. The information distributed by a few core and influential twitterers (or tweets) could have a much greater impact and distribution of information than a large group of random ones. Hence, there has been an increasing interest in finding these core and influential twitterers (or tweets) from various perspectives, i.e., [150], [10].

Weng et al. [150] presented an approach of identifying influential twitters by employing an extension of the PageRank algorithm [20] (previously described in Subsection 3.3.4). The proposed algorithm, TwitterRank, considers "both the topical similarity between users and the link structure" [150]. In contrast to many other existing research on Twitter datasets, the authors pointed out that the number of followers alone may not accurately represent the influence due to "reciprocity", where users follow their followers as an act of social courtesy as opposed to "homophily", where users follow due to mutual topic interests. To establish if "homophily" exists in the Twitter community, the authors proposed two underlying questions:

1. Are twitterers with "following" relationships more similar than those without according to the topics they are interested in?

2. Are twitterers with "reciprocal following" relationships more similar than those without according to the topics they are interested in?

To answer these questions, Weng et al. analyzed large volumes of unlabeled tweets (content) to automatically distill topics and determine if the relationship between influential twitterers and followers is due to topic sensitive influence. Their experiment results showed that: Firstly, "homophily" does exist in the context of Twitter and the authors claim that they are the first to report this. Their observation justifies that there are some twitterers who in-fact "follow" someone due to common topical interests instead of just playing a "number game". Secondly, their proposed approach outperforms the benchmark technique that is currently used by Twitter and other related algorithms, e.g., in-degree (i.e., the number of followers) and the original PageRank [20].

In addition to TwitterRank [150], three other Twitter-specific ranking algorithms are TunkRank [138], inDegreeRanking [82] and IARank [23]. TunkRank adapts PageRank [20] and defines influence as the attention a user is able to give to the tweets he receives, combined with the attention that his followers can give to him. Kwak et al. [82] proposed to rank users based on the number of followers (in-degree), and found the produced ranking was similar to PageRank. Unfortunately, a high in-degree could be made up by simply creating fake usernames that follow a user whose ranking is to be increased, making it an easy loophole for spammers. The key characteristic of IARank is its ability to rank users on Twitter in near real-time. The basis of IARank is information amplification potential of a user, which is evaluated by the capacity of the user to increase the audience of a tweet or another twitter that they would find interesting, by retweeting or mentioning.

TunkRank [138] converges to the final ranking in an iterative way. The convergence time is determined by the number of users considered in the ranking process. As opposed to IARank [23], TunkRank is not capable of producing a ranking list in real time. TwitterRank [150] focuses on topical-oriented influential twitters mining by distilling available topics and it is also not amenable to be calculated in real time.

Cha et al. [25] compares three different measures of influence in Twitter: Number of followers, number of retweets, number of mentions and finds that the most followed users do not necessarily score highest on the other two measures. Number of followers (in-degree) represents a user's popularity, however, it is not directly correlated to other important perspectives of influence such as engaging audience. *Retweets are driven by the content value of a tweet, while mentions are driven by the name value of the user. Such subtle differences lead to dissimilar groups of the top influential Twitter users.* Focusing on retweets and mentions, the authors investigate the dynamics of user influence across topics and time using a large amount of data collected from Twitter.

The research by Bakshy et al. [10] studies the information propagation of influential sources when compared to "ordinary influential users". One of the key functionalities of Twitter is the ability for a user to "retweet" or repost the contents of another twitterer. However, due to the 140 character limit of tweets, a common strategy adopted by twitterers is to attach content to shortened URLs (e.g., bit.ly) to effectively condense content in an easily distinguishable form. Additionally, because of these unique tokens, it is easy for the authors to identify the depth or level of cascade a tweet is propagated from a single influential source. Therefore, the authors study the URLs that twitterers add to their tweets and the overall cost-effectiveness of marketing through a small group of key influential twitterers as opposed to a large group of average or under-performing ones. The focus of the Bakshy et al.'s [10] study is explicitly on the overall influence of a post rather than the traditional qualitative measure of user influence, hence, their methodologies are based on the "influence score" of URL posts and the diffusion level of URL reposts from the original "seed" until termination of the cascade. Their results showed that while the majority of posted URLs do not spread (with an average cascade size of 1.14), some distinct ones are able to spread as far as nine generations of repost. Using these results, the authors were able to compare the effectiveness of "seed" influencers with word-of-mouth campaigns and find that using a quantitative approach of influence of a post has similar results to the qualitative study of influential users. Both methodologies showed that since the majority of posts from a single source do not spread at all, targeting certain "seeds" with high rates of diffusion as opposed to multiple low performing ones generates the most coverage.

Pal and Counts [107] discovered topical authorities in Twitter. There are three steps in their approach: Firstly, characterize Twitter users with social media-specific features including both nodal and topical metrics. Secondly, cluster the users over feature space. Lastly, produce a list of the important authors for a specific topic utilizing a within-cluster ranking procedure.

In 2016, Drakopoulos et al. [38] studied five Twitter-specific metrics for ranking Twitter influence. Based on concepts from system theory, a methodology for evaluating influence metrics is also proposed. In addition, the authors implemented the five metrics in Java over Neo4j, which is a graph database that provides production grade front or back-end social graph storage.

3.5.2 Facebook

Being the largest social network, Facebook has more than 400 million active users with the average of 130 friends [2]. Facebook data set has been utilized in various top-k nodes identification research.

Heidemann et al. [57] proposed an adapted PageRank algorithm to identify influential users in a social network according to activity links. The approach was evaluated on a Facebook data set and found that more active users that are retained can be identified when drawing on users' prior communication activities or centrality measures e.g., degree centrality. Kim and Han [71] identify influential users in a network graph by first calculating degree centrality based on social links and then estimating an activity index. The proposed method was evaluated by observing the influence diffusion of a Facebook game. Their experiment results show that by targeting the identified influential users, game growth rates and number of new game adopters can be increased.

Although Facebook has been frequently used as experiment data for identifying top-k nodes, our literature research shows that so far only few Facebook-oriented algorithms exist. This phenomenon differs significantly from the Twitter domain as a number of top-k nodes identification approaches have been designed specifically for Twitter, such as TunkRank [138], TwitterRank [150] and IARank [23].

3.5.3 Blogosphere

With an increasing amount of blog posters and readers, Blogospheres (a network of blogs) are an effective and inexpensive medium for companies to evaluate their advertising campaigns. Hence, it is interesting to observe the emerg-

Table 3.7: Comparisons of various techniques for identifying top-k nodes in blogosphere

Technique	Identify Influential Bloggers	Identify Influential Blogs	Utilize Blog Contents	Analyze Influence Diffusion Models
Gruhl et al.[49]	Yes	No	Yes	No
Java et al.[63]	Yes	No	No	Yes
Java et al.[7]	No	Yes	Yes	No
Huang et al.[60]	Yes	No	Yes	No

ing research that has begun identifying top-k nodes in the weblog domain.

Gruhl et al. [49] model the diffusion of topics between blogs, determined by the textual content of the weblog. The authors managed to characterize information propagation from two perspectives: Topics and individuals. The proposed model makes it possible to "identify particular individuals who are highly effective at contributing to the spread of infectious topics" [49].

Java et al. [63] presented an analysis of influence models, i.e., Linear Threshold model [47], Independent Cascade model [45], on a large-scale and real-world blog graph. Their evaluation results suggest PageRank [20] is an inexpensive approximation to the simple greedy algorithm [69] for selecting an influential set of bloggers, which maximizes the spread of information on the blogosphere. In order to recommend feeds and identify influential blogs automatically, the same authors [7] found that "blog feeds that matter" for a specific topic using folder names and subscriber counts.

Huang et al. [60] presented a framework which contains a heuristic quantification model for ranking key microbloggers. The two major parts in the framework were: (1) Based on content of posts, a classifying approach with sliding-window to specify interested domains of microbloggers, (2) A method for quantifying key microbloggers by taking both the influence and user activity into consideration.

Overall, in the blogosphere domain, it is observed that a number of research, e.g., [49], [7], [60], utilize the contents of blogs in order to identify top-k nodes in the weblog networks. We further compare the techniques discussed in this subsection in Table 3.7.

3.5.4 Misinformation Control

Despite the benefits of interconnectivity provided by online social networks, there are existing threats such as spread of misinformation that can cause undesirable effects such as widespread panic to the general public. Budak et al. [21] described the misinformation control problem as "identifying a subset of individuals that need to be convinced to adopt the good campaign so as to minimize the number of people that adopt the bad campaign" [21]. In addition, they formulated their description as an optimization problem, proved that it was NP-hard, and then provided performance guarantees for a greedy strategy.

From a different aspect of controlling misinformation, Nguyen et al. [102] focused on how to limit rumor spread in Online Social Networks (OSNs) by finding the smallest set of influential nodes, whose decontamination with good information helps to control the viral propagation of misinformation. Their solution includes a greedy-based algorithm, *Greedy Viral Stopper* (GVS) and a community-based heuristic method. GVS greedily adds nodes with the best influence gain to the current solution, and shows that the algorithm selects a small fraction of the total nodes from the optimal solution. The community-based method returns a selection of nodes to decontaminate in a timely manner. The authors verified their approaches on real-world OSNs such as Facebook.

There are two major differences in the two aforementioned research by Budak et al. [21] and Nguyen et al. [102]: (1) The approach by Budak et al. was limited by a k-nodes budget where k is a constraint imposed by the user. This was not presented in Nguyen et al.'s approach. (2) The heuristic proposed by Budak et al. assumes a high level of propagation, where the probability for good information spread is either one or zero. In contrast, Nguyen et al. accounts for arbitrary probabilities.

3.5.5 Community Question Answering

Top-k nodes identification has also been researched widely in the Community Question Answering (CQA) domain with a number of models proposed. Users can often find detailed information from subject-matter experts on Q&A portals such as "Stack Overflow" and "Yahoo! Answers". However, the quality of information can range from detailed solutions to unconstructive criticism. Additionally, if the feedback for a topic is sparse, it can be difficult to differentiate the different quality of answers. Hence, the ability for users to identify the members that provide reliable and accurate answers is critical for the CQA domain.

Fisher et al. [40] detects key authors ("answer people") in Usenet newsgroups. Their heuristic network analysis methodology uses nodal features to find "answer people" with high out-degree and low in-degree. Zhang et al. [161] models CQA as an expertise graph, and evaluates a number of ranking measures and algorithms for identifying users with high expertise in differently structured networks. Extended from Zhang's approach [161], Guo et al. [51] proposed topic-based models to identify appropriate users to answer a specific question. Jurczyk and Agichtein [66] presented an analysis of the link structure of a general-purpose Q&A community to identify authoritative users. Agichtein et al. [6] uses data from a web-scale community question answering portal and extracts graph-based features, e.g., the degree distribution of users and their PageRank, hubs scores to determine individual user's relative importance. A model to identify authoritative actors based on the number of best answers provided by users is presented in [17], whereas Pal et al. [108] discriminates experts on top of their preference in answering position.

3.5.6 Networks with Complex Topologies

The majority of research described up until now performs analysis on unweighted networks with very basic properties and node-to-node relationships. However, these simple network structures will often obfuscate many important properties of a relationship such as intensity, duration, human emotional factors and therefore, are unable to characterize the complexity of real-world networks and relationships. Barrat et al. [12] describe weighted networks as graphs that go beyond the topological point and underline the capacity and intensity of connections between complex entities.

Opsahl et al. [105] presents an approach on evaluating node centrality in weighted networks. In this research, the authors proposed a generalized degree centrality measure that incorporates both the number of edges and their weights on Freeman's Electronic Information Exchange System (EIES) network [42]. As it is difficult to non-subjectively define a weight for each of these properties, the authors' analysis contains a proposed threshold value, a, that alters the weighting of each attribute. The non-deterministic and complex nature of weighted networks can also be seen in a similar research by Wei et al. [149]. Also using Freeman's EIES dataset, the authors performed centrality measures based on Dempster–Shafer's theory of evidence [33, 116], using the degree and strength of the node. To evaluate the effects of different weightings of an edge, the authors also applied small thresholds to their centrality measures.

Another complex network topology that we are underlining is the homogeneity of social networks for top-k node identification. A homogeneous network has a singular type of object and relationship, which are represented as nodes and edges, respectively. Whereas, heterogeneous networks are networks with multiple types of nodes and node-to-node relationships. The vast majority of the research in this survey has predominately been on homogeneous networks, such as [69, 146, 150], this is likely due to homogeneous network data being more readily available and easier to analyze. On the other hand, there is very little existing research in identifying top-k nodes in heterogeneous networks. One example presented in this survey is the research by Zhou et al [165]. The requirement to co-rank two different types of entities significantly increases the complexity of their heuristic, which involves coupling two random walks in order to rank authors with their respective publications in the heterogeneous network.

3.5.7 Miscellaneous Applications

Ghosh and Lerman [43] are among the few researchers who study the problem of predicting influential users in online social networks. They classified influence diffusion process as non-conservative if it depends not only on the network structure, but also on details of the dynamic processes occurring on it. The authors experimented with the social news aggregator, Digg, which allows users to post and vote for news stories. In their scenario, influence was defined as the dynamic number of in-network votes a user's post generates, which represents non-conservative information flow. A number of influence models were compared and their experiment results showed that non-conservative models which capture the actual details of the dynamic process are better at predicting influential users on Digg. In addition, Ghoah and Lerman's experiments also found that the normalized α-centrality metric is one of the best predictors of influential users.

Aral and Walker [8] conducted a randomized experiment, which identifies influence and susceptibility to influence in the product adoption decisions of a representative sample of 1.3 million Facebook users. The experiment includes "a random manipulation of influence-mediating messages sent from a commercial Facebook application" [8], which allows users to share information about various products. Their methodology avoids the biases inherent in traditional estimates of social contagion by randomly manipulating who receives the influence-mediating messages. Some interesting findings from their experiments are: (1) Younger users tend to be more susceptible to influence than older users, (2) Influential users are less susceptible to influence than non-influential ones and they cluster in the network while susceptible users do not, (3) Unlike some previous research, which claim that an individual's influence is determined only by his or her personal attributes, Aral and Walker's experiment results showed that the combination of influence, susceptibility, and the likelihood of spontaneous adoption contributes to an individual's importance to the diffusion of behaviors.

3.5.8 Section Summary

This section presents examples of real-world applications integrating top-k node detection techniques. Several techniques that analyze OSNs such as Facebook and Twitter have been reviewed including those targeting social dynamics such as misinformation control. We have also reflected on work that targets weighted and heterogeneous networks and underlined the increased complexity of identifying top-k nodes in these kinds of networks when compared to traditional networks.

Emphasis on node and edge attributes can be seen throughout the research in this section. (1) Research on Twitter including [138, 150], all contain node attributes, such as the number of followers and retweets. (2) Community Question Answering datasets [40, 6] contain integrated user voting and scoring systems. (3) Networks with complex heterogeneous structures [165] contain multiple types of nodes and edges resulting in multi-layered analysis of different node and edge types.

3.6 Research Directions

With the abundant amount of literature published in research into identifying top-k nodes, one may wonder whether we have solved most of the critical problems related to top-k nodes identification such that the solutions provided are refined enough for most of the social network analysis tasks. However, in our view, there are still several critical research problems that need to be solved before top-k nodes identification can become a sufficient tool for analyzing social networks.

Exploration of factors that create the top k-nodes So far, very little literature has focused on the exploration of factors that establish the top k-nodes being the most significant and/or influential and this can be an interesting research area. In a situation where a social network comes with some known top-k nodes, the question is what are the factors that distinguish those top nodes from others. Understanding these factors is essential to improve interpretation and usability of top-k nodes mining.

Devotion of more systematic research effort on top-k significant nodes In general, more progress has been made on the top-k influential nodes as compared to the top-k significant nodes. In-depth research can be conducted in the future to investigate the top-k significant nodes in various domains. With the proliferation of social networks, i.e., email communication network, user interactive question answering network, organization hierarchy, OSNs etc., more intelligent and practical solutions can be applied to the top-k significant nodes, e.g., maximizing the quality of identified top-k significant nodes by capturing and utilizing individuals' skill sets and their social interactions as well as user influence.

Identification of top-k nodes in dynamic social networks In Section 3.3.5, we pointed out that identifying top-k nodes in dynamic social network is a relatively new area. Since many social networks, such as Twitter, are only available in the form of social streams of user activities, there is an increasing value of research in identifying top-k nodes in dynamic social networks.

Identification of top-k nodes for multiple topics Based on our review, there has been very little research into the problem of identifying top-k nodes for multiple topics. The vast majority of existing literature in the field identifies top-k nodes only for a single topic. However, it is more practical and useful to perform top-k nodes mining for multiple topics, especially for companies and organizations with heterogeneous social networks.

Enhancement of searching variety for a single topic When dealing with top-k nodes, we are interested in the subjectively determined topic. A wide range of searching capabilities can be utilized to identify top-k nodes for a given topic. For example, when searching for top researchers of a particular subject in an area, it is useful if the functionality of searching on subject name is integrated within the algorithm.

Development of more efficient, scalable and performance guaranteed algorithms for top-k influential nodes As we have categorized and summarized in Section 3.3, abundant research has been dedicated to generic algorithm development for top-k influential nodes. A number of these research evaluated their approaches in large-scale social networks using metrics such as time consumption, memory usage, and true positive rate. Therefore, one possible future direction is to focus on efficiency, scalability and performance guarantee improvement of such generic algorithms. This direction is also described in a literature survey [111] on finding influential users as "*efficiency and validity are crucial*".

3.7 Conclusions

In this chapter, we present an overview of the current status and future directions of top-k nodes identification in social networks. We reviewed and classified existing literature in this area into two major categories: top-k influential nodes and top-k significant nodes. In general, we feel that considerable progress has been made on the top-k influential nodes as compared to top-k significant nodes. The applications of top-k nodes identification in social networks are quite diverse and have been discussed in detail.

This survey work aims to show that numerous research have attempted to solve the top-k nodes identification problem by proposing various algorithms, methodologies, frameworks. Interesting scope exists in future research targeting top-k significant nodes since the work on this area is quite limited. Furthermore, the vast majority of the work on identifying influential nodes is designed for static networks, therefore, the research area of mining top-k nodes in dynamic networks is still relatively new.

Based on our review of earlier literature, we constrain the scope of the specific problems to be covered and addressed in Chapters 4, 5 and 6 as follows:

1. The heterogeneous network schema studied in Chapter 4 and Chapter 5 is a correlation schema.

2. Heterogeneous networks containing more than one type of object and relationship are studied in the three chapters.

3. There can be only one relationship type between objects of the same type in the three chapters.

The specific heterogeneous network mining and analysis tasks studied in the three chapters are: overlapping heterogeneous communities discovery in static heterogeneous correlation networks (Chapter 4); community based ranking of objects in static heterogeneous correlation networks (Chapter 5); network embedding and change modeling in dynamic heterogeneous networks (Chapter 6).

4

Uncovering Overlapping Heterogeneous Communities

A heterogeneous correlation network represents relationships (edges) among source-typed and attribute-typed objects (nodes). It can be used to model an academic collaboration network, describing connections among authors (source-typed objects) and published papers (attribute-typed objects). To date, there has been little research into mining communities in heterogeneous networks. The objective of our research is to discover overlapping communities that include node and edge of each type in a heterogeneous correlation network. In this chapter, we describe an algorithm, OHC (**O**verlapping **H**eterogeneous **C**ommunity). Inspired by a homogeneous community scoring function, Triangle Participation Ratio (TPR), OHC finds target heterogeneous communities then expands them recursively with triangle-forming nodes. Experiments on different real world networks demonstrate that OHC identifies heterogeneous communities that are tightly connected internally according to two traditional scoring functions. Additionally, analyzing the top ranking heterogeneous communities in a case study, we evaluate the results qualitatively.

Overall, our main research question is "Does heterogeneous information improve the results of network analysis?". To answer this research question, the thesis focuses on three network analysis tasks: community detection, ranking and network embedding. In this chapter, we focus on our first network analysis task, which is community detection in static heterogeneous correlation networks. In doing so, we compare the performance of the OHC algorithm against two state-of-the-art homogeneous community discovery techniques. Specifically, we compare internal and external connectivity of communities identified by the different algorithms and quantitatively evaluate the top ranking heterogeneous communities identified.

4.1 Introduction

A homogeneous network represents relationships between one object type. A wide variety of methods for detecting communities in homogeneous networks have been proposed [100, 13, 154]. Researchers in the fields of Computer Science [157, 88] and Physics [83, 100] describe such communities as sets of nodes with high density of internal edges and low density external edges. In contrast, a heterogeneous network represents relationships (edges) between multiple types of interacting objects (nodes). To date, there has been limited research that detects communities in heterogeneous networks. Existing homogeneous community detection techniques cannot be used to detect communities that retain the complex characteristics of heterogeneous networks, such as information gathered from multiple sources [22]. The aim of our research is to find a group of tightly-connected objects with their closely-interacting other types of objects in a heterogeneous network. For example, in an academic heterogeneous network, we want to identify academics that collaborate frequently together with the research papers they collaborated on. The group has dense internal connections and loose external connections with academics that collaborated infrequently. Another example is social network users that co-comment on the same topics together with their co-commented topics. Finding these groups can help us to understand the structure of a network, so collaborations can be encouraged or topics can be promoted among loose connections. In the meantime, dense connections can be loosened if it is not the ideal type of connections.

To represent the metastructure formed by multiple types of objects or relationships in heterogeneous networks, researchers have proposed different types of network schemas, including: Multi-relation with single-typed object schema [162] , Bipartite schema [93], Star schema [131] and Correlation schema [79]. We constrain our work to correlation schema, where objects can be categorized as either Source Type (ST) or Attribute Type (AT). Figure 4.1 presents a bibli-

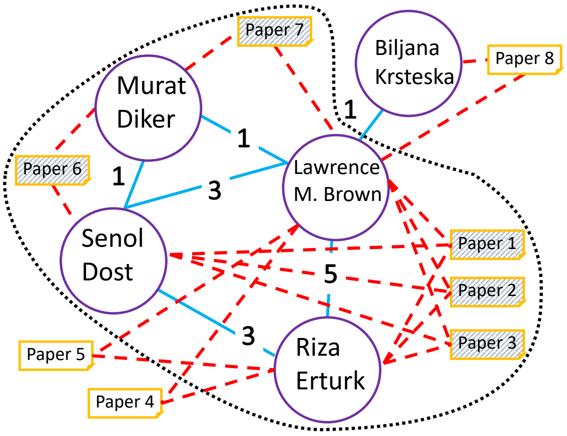


Figure 4.1: Sample heterogeneous correlation network H_1

ographic heterogeneous correlation network H_1, where authors are the source-typed objects (nodes) and papers are the attribute-typed objects (nodes) (represented by circles and rectangles respectively). The rectangles with a striped pattern represent papers that share a common theme, namely they are about texture spaces. The co-authorship relationship between authors is denoted by solid lines. The relationship weighting represents the number of papers the pair of authors have co-published together. Relationships between authors and papers denoted by dashed lines, indicate the authors of a paper.

Applying heterogeneous community detection techniques on H_1, we can identify communities of authors that publish together, and the papers that are most relevant in this community. Our **O**verlapping **H**eterogeneous **C**ommunity detection algorithm, OHC, detects one heterogeneous community, denoted by the dotted boundary in Figure 4.1: {Lawrence M. Brown, Riza Erturk, Senol Dost, Murat Diker, Paper 1, Paper 2, Paper 3, Paper 6, Paper 7}. The authors were shown to have a common interest in the field of texture spaces (as highlighted by the papers with stripped patterns in Figure 4.1). This example illustrates that OHC can detect communities of authors that are interested in a particular research sub-field.

The main contributions of this chapter are: 1) We propose a novel algorithm, OHC, that detects communities which contain multi-typed objects (nodes) and relationships (edges) in heterogeneous correlation networks. Evaluation experiments and case studies on real world datasets validate the effectiveness of the algorithm. 2) We demonstrate that traditional metrics can be used to evaluate the quality of detected heterogeneous communities.

4.2 Problem Formalization

In graph theory, a heterogeneous network is a graph that contains more than one type of node with multiple kinds of node-to-node relationships.

Definition 5. Given the undirected graph $G = (V, E)$, we define heterogeneous networks to be graphs that contain two or more types of object $\{v_1, v_2, .., v_n\} \in V$ and multiple types of relationships $\{e_1, e_2, .., e_n\} \in E$. Each object type v_x has relationships to the same or other object types defined by an edge type e_x.

We define overlapping heterogeneous communities as subgraphs of a given heterogeneous network, where each community contains nodes of all object types and edges of all relationship types that exist in the network. The aim of our research is to identify heterogeneous communities with dense internal connections and loose external connections in a heterogeneous correlation network.

4.3 The OHC Algorithm

In this section, we present our greedy community detection technique, OHC (**O**verlapping **H**eterogeneous **C**ommunity). Details of OHC and how it transforms our example heterogeneous network shown in Figure 4.1 into a heterogeneous community are provided. Java source code for the OHC algorithm and experiment datasets are available for download[1].

4.3.1 Three Phases of OHC

Our proposed approach is composed of three main phases. Phase One processes the input data and generates a set of seed communities. Inspired by Triangle Participation Ratio (TPR) scoring function [157, 148] described in Section 2.1, Phase Two produces a set of heterogeneous communities by adding triangle-forming source-typed nodes with associated attribute-typed nodes to each seed community. The output of Phase Two is fed into Phase Three, where we remove the duplicate and subset heterogeneous communities. The pseudocode for OHC is detailed in Algorithm 1. Figures 4.1 and 4.2 demonstrate outputs of OHC's third phase and the first two phases respectively. The data presented in these figures is a subset of real-world datasets used in our experiments.

[1] http://bit.ly/2vEfOQU

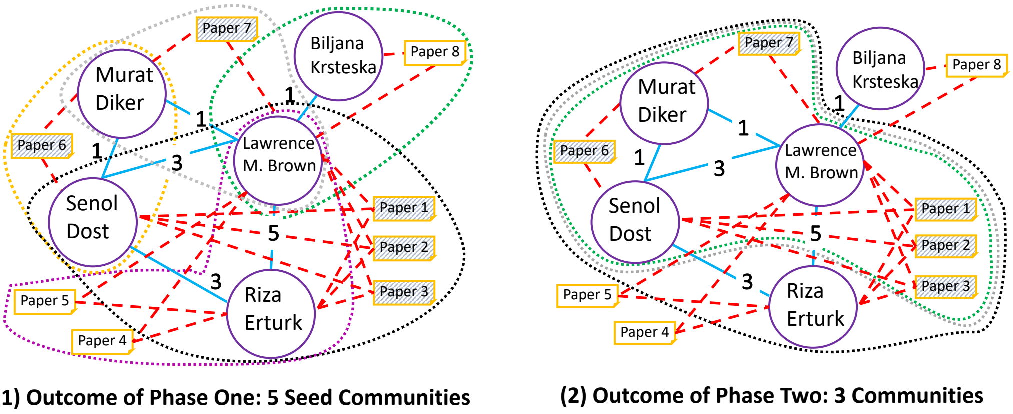


Figure 4.2: Output of OHC's first two phases (identified (seed) communities denoted by the dotted boundaries)

4.3.2 Phase One

The heterogeneous network processed by OHC in the initial phase is composed of a homogeneous network that contains relationships among source type (ST) nodes and a heterogeneous network containing relationships between both ST nodes and attribute type (AT) nodes. For example, the heterogeneous network processed by OHC in Figure 4.1, is constructed from an author-collaboration homogeneous network, containing co-authorship relationships between authors and an authorship heterogeneous network, which contains the author-to-paper relationships. This phase generates a list of heterogeneous seed communities based on a set of distinct and interconnected source-typed nodes. Each of the seed communities must contain more than one ST node, at least one commonly linked node of AT, and the corresponding relationship edges. Seed communities with solo ST nodes are eliminated due to their inability to form triangles in the next phase. When applied to the heterogeneous network shown in Figure 4.1, Phase One produces the set of seed communities, C_{seeds}, denoted by the dotted boundaries in Figure 4.2(1).

4.3.3 Phase Two

When expanding an individual seed community, C_{a_seed} in C_{seeds}, OHC takes a depth-first search approach to iterate through all pairs of ST (Source Type) nodes (ST_x, ST_y) in the original and expanded C_{a_seed}, and recursively finds all triangle-forming triads (ST_x, ST_y, ST_z) in an exhaustive manner by examining all other seed heterogeneous communities generated from Phase One. The expansion process introduces the eligible triangle-forming ST nodes, the associated AT (Attribute Type) nodes, and the corresponding relationship edges that do not already exist in C_{a_seed}. The recursive process for each pair of ST nodes continues until no further triangle-forming ST nodes can be found. The expansion processes for seed communities are deterministic. Additionally, resulting communities that have fewer than three AT nodes or do not form triangles with any ST nodes will be removed from the output of Phase Two.

To become a triangle-forming ST node, a node must have at least one distinct edge with each member of the pair (ST_x, ST_y). The triangle-forming ST node, ST_z, with the associated AT nodes will be added to C_{a_seed} if they are not already present. In the situation where ST_z is not included in the initial set of nodes of C_{a_seed}, ST_z forms two new pairs (ST_z, ST_x) and (ST_z, ST_y) with each member of the original pair. The recursive process starts on each of the new pairs to find further triangle-forming ST nodes. While in the situation where ST_z is included in the initial set of nodes of C_{a_seed}, OHC will explore further triangle-forming ST nodes using the pairs (ST_z, ST_x) and (ST_z, ST_y) only when both ST_x and ST_y are not included in the initial set of nodes of C_{a_seed}.

To better illustrate the mechanisms of this phase, we refer to the seed community, s, {Lawrence M. Brown, Murat Diker, Paper 7} in Figure 4.2(1) as an example, with the author pair (Lawrence M. Brown, Murat Diker) being a pair of ST nodes. OHC detects that Senol Dost has one distinct edge with Lawrence M. Brown (due to the associated AT nodes: Paper 1, Paper 2, Paper 3) and another distinct edge with Murat Diker (resulting from Paper 6), therefore, Senol Dost is a triangle-forming ST node for s, which expands s into the community {Lawrence M. Brown, Murat Diker, Paper 7, Senol Dost, Paper 1, Paper 2, Paper 3, Paper 6}. The output of Phase Two is denoted by the dotted boundaries in Figure 4.2(2). Algorithm 2 describes the main logic procedure of phase two.

4.3.4 Phase Three

For the final phase, we iterate through the expanded communities from Phase Two and remove the ones that are duplicates or subsets of other communities. These communities are removed as information contained in a subset or duplicate community has already been replicated in their counterpart superset communities. Attributes observed in the OHC-detected heterogeneous community from our example network are: 1) Authors in a community have strong connections with other community members. 2) Papers in the same community tend to share common topics with one another.

In our running example, the seed community, {Lawrence M. Brown, Riza

Erturk, Senol Dost, Paper 1, Paper 2, Paper 3} expanded to {Lawrence M. Brown, Riza Erturk, Senol Dost, Paper 1, Paper 2, Paper 3, Murat Diker, Paper 6, Paper 7}. This results in the two duplicated sub-communities removed from OHC's final output. The final result is denoted by the dotted boundaries in Figure 4.1.

Algorithm 1 OHC

```
Input: A given heterogeneous network: $H$
Output: Heterogeneous communities detected: $C_{output}$

Phase One
The list of heterogeneous seed communities obtained after processing $H$: $C_{seed}$

for all $c$ in $C_{seed}$ do
    $m$ = number of $ST$ nodes in $c$
    $n$ = number of $AT$ nodes in $c$
    if ($m$ <2 or $n$ <1) then
        $C_{seed}$.remove($c$)
    end if
end for
$C_{output}$ = Phase Two($C_{seed}$)
Phase Three
for all $c$ in $C_{output}$ do
    for all $k$ in $C_{output}$ do
        if (indexOf($k$) $\neq$ indexOf($c$) and $c \subseteq k$) then
            $C_{output}$.remove($c$)
        end if
    end for
end for
```

4.3.5 Time Complexity Analysis of OHC

To analyze the complexity of the proposed algorithm we use $|x|$ ST nodes, $|y|$ AT nodes and $|s|$ seed communities. Here we first analyze the best case performance of OHC. Based on a set of distinct and interconnected ST nodes, Phase One iterates through $|y|$ AT nodes to generate qualified seed communities with O($|y|\times|x|$) run time. The best case for Phase Two is when only one pair of ST nodes needs to be processed in all the seed communities and none of them have triangle-forming ST nodes, which leads to a run time of O($|s|^2$). During Phase Three, a linear execution time of O($|s|$) is required when the communities produced from the previous phase can all be merged into one su-

Algorithm 2 Phase Two

Input: Processed C_{seed} from phase one
Output: C_{seed} updated with triangle-forming nodes

```
for all $c$ in $C_{seed}$ do
   cliqueFormed = $false$
   $N$ = set of all $\boldsymbol{ST}$ nodes in $c$ and given $x, y \in N$
   for all unique combination $(x, y)$ in $N$ do
      $S$ = stack with $(x, y)$
      while S is not empty do
         S.pop() → $(x,\ y)$
         $Z$ = $\boldsymbol{ST}$ nodes connected to both $x$ and $y$
         for all $z$ in $Z$ do
            $a_{xz}$ = $\boldsymbol{AT}$ nodes linked to both $x$ and $z$
            $b_{yz}$ = $\boldsymbol{AT}$ nodes linked to both $y$ and $z$
            if $a_{xz} \neq b_{yz}$ then
               if  $z \notin N$ then
                  cliqueFormed = $true$
                  if $z \notin c$ then
                     add $z$ to $c$
                  end if
                  if $a_{xz} \notin c$ then
                     add $a_{xz}$ to $c$
                  end if
                  if $b_{yz} \notin c$ then
                     add $b_{yz}$ to $c$
                  end if
                  add $(x, z)$ and $(y, z)$ to the top of $S$
               else
                  if $a_{xz} \notin c$ then
                     cliqueFormed = true
                     add $a_{xz}$ to $c$
                  end if
                  if $b_{yz} \notin c$ then
                     cliqueFormed = true
                     add $b_{yz}$ to $c$
                  end if
                  if $(x \notin N$ and $y \notin N)$ then bibliographic add $(x, z)$ and $(y, z)$ to the top of $S$
                  end if
               end if
            end if
         end for
      end while
   end for
   $i$ = number of $\boldsymbol{AT}$ nodes in $c$
   if (cliqueFormed == $false$ and $i$ <3) then
      $C_{seed}$.remove($c$)
   end if
end for
```

perset community. Combining run time of the three phases together we have a complexity of O($|y|\times|x|+|s|^2+|s|$). Usually, $|x|$ and $|y|$ are orders of magnitude larger than $|s|$, which means the best case run time of OHC reduces to O($|y|\times|x|+|s|^2$).

In a rare case where all ST nodes are processed in every seed community, Phase Two's run time becomes O($|x|^2\times|s|^2$). Execution time of the worst case in the third phase is ($|s|^2$) where no duplicate or subset communities can be eliminated. As a result, OHC's worst case performance is O($|y|\times|x|+|x|^2\times|s|^2$ $+|s|^2$), which can be reduced to O($|x|^2\times|s|^2$).

4.3.6 Conditions of OHC

Our proposed approach works for a given heterogeneous network H, which satisfies the following two conditions:

1. The metastructure of H is correlation schema based.

2. H is a static network.

4.4 Experiments and Results

To study the effectiveness of OHC, we conducted experiments on various publicly accessible real-world heterogeneous networks. All the experiments were performed on a computer with 3.40 GHz i7 CPU, 8 GB RAM and Windows 10 operating system.

4.4.1 Datasets

Bibliographic data is widely used in heterogeneous network experiments in the existing literature. For our experiments, we analyzed the ACL Anthology Network **(AAN)** dataset [112] and the following five bibliographic datasets from ArnetMiner [134]: Data Mining database information retrieval **(DM)** dataset, Software Engineering **(SE)** dataset, Computer Graphics Multimedia **(CGM)** dataset, Artificial Intelligence **(AI)** dataset and Interdisciplinary Studies **(IS)** dataset. Statistics of the six datasets are shown in Table 4.1. For each dataset, we constructed an author-collaboration network and authorship network. In the author-collaboration network, authors are represented as nodes while the number of edges between two nodes indicates the number of papers that the corresponding authors have co-published. For the authorship network, there exist two types of nodes: author and paper nodes, an edge between these two

Table 4.1: Statistics of datasets

Dataset	No. of Authors	No. of Papers	Average Node Degree*
DM	5856	2640	3.3521
SE	8127	3923	3.8214
CGM	25961	16599	4.4163
AI	41478	27596	4.4372
IS	46097	18583	7.6488
AAN	14464	18041	8.2469

* Measured based on author-collaboration network

types of nodes represents that the author had authored the paper. We define author nodes to be our source type (ST) nodes while the paper nodes are described as attribute type (AT) nodes.

4.4.2 Benchmark Techniques

We are unable to compare OHC with existing heterogeneous community detection techniques because they were designed for a different purpose (as described in Section 2.1.3). Hence, we build benchmark techniques based on two state-of-the-art homogeneous community detection algorithms: SLPA and Louvain. For simplicity, we notate the SLPA-based method as **SLPAh** and the Louvain-based method as **Louvainh**. SLPAh is built on top of SLPA [154], where SLPA is initially applied to the author-collaboration network to detect homogeneous communities of the ST (author) nodes. Appropriate AT (paper) nodes that are connected to all ST nodes in a community are then appended to each of these detected homogeneous communities. In our experiments, the number of iterations T was set to 100 to guarantee SLPAh's stable performance and the threshold r, which affects the number of detected overlapping communities, was varied: 0.01, 0.25 and 0.45.

The construction process of Louvainh is similar as SLPAh. We built Louvainh on top of Louvain [13], where Louvain is applied to the author-collaboration network first to detect homogeneous communities of the ST nodes. Afterwards, appropriate AT nodes that are connected to all ST nodes in a community are appended to each of the detected homogeneous communities.

Both SLPAh and Louvainh are non-deterministic, therefore, we repeated our experiments 30 times for each technique and report their average performance and standard deviation values. In addition, we repeated our experiments 30 times for each value of SLPAh's threshold, r.

Table 4.2: Number of detected heterogeneous communities

Dataset	OHC	SLPAh	Louvainh
DM	71	1411	1720
SE	237	1735	2222
CGM	1685	4863	6746
AI	2808	7562	10596
IS	3211	6359	8839
AAN	2385	1604	2434

Clustering and classification algorithms such as GenClus, RankClus and RankClass reviewed in Sections 2.4.1 and 2.4.2 were not selected as benchmark techniques because they were designed for different purposes and are not comparable with OHC. For example, Section 2.4.1 reviews algorithms developed for doing clustering in heterogeneous networks. These algorithms focus on determining how similar underlying objects in a network are to each other (structural and attribute similarities). Section 2.4.2 reviews algorithms developed for doing classification in heterogeneous networks. These algorithms focus on assigning categorized labels to objects in a network. As for OHC, we are interested in identifying communities in heterogeneous networks, where the internal and external connections of objects are the major concern.

4.4.3 Evaluation Metrics

Without ground-truth heterogeneous communities and particular metrics for evaluating heterogeneous communities available, we adopt the homogeneous community scoring functions, FOMD and FlakeODF, to measure the inter and intra connectivity of the identified communities. Details of FOMD and FlakeODF scoring functions are provided in Section 2.1.2. Edge (relationship) weightings are used for calculating FOMD and FlakeODF scores. The value range of FOMD is [0,1], where a higher FOMD score represents better internal connectivity and a value of 1 indicates a highly interconnected community. The value range of FlakeODF is [0,1] as well. As the FlakeODF score approaches 0, this indicates that a community is highly connected internally while being more disconnected from the rest of the network. For our evaluations, we aim to find heterogeneous communities with high FOMD score and low FlakeODF score.

Table 4.3: FOMD results

Method-Dataset	Min	Max	Median	Average	SD
OHC-DM	**0.1933**	1.0000	**0.5000**	**0.5711**	**0.1723**
SLPAh-DM	0.0000	1.0000	0.0000	0.1965	0.3922
Louvainh-DM	0.0000	1.0000	0.0000	0.0733	0.2544
OHC-SE	**0.1811**	1.0000	**0.5000**	**0.5526**	**0.1583**
SLPAh-SE	0.0000	1.0000	0.0000	0.1788	0.3794
Louvainh-SE	0.0000	1.0000	0.0000	0.0882	0.2673
OHC-CGM	**0.1100**	1.0000	**0.4000**	**0.4871**	**0.1612**
SLPAh-CGM	0.0000	1.0000	0.0000	0.1385	0.3366
Louvainh-CGM	0.0000	1.0000	0.0000	0.0677	0.2293
OHC-AI	**0.1222**	1.0000	**0.4000**	**0.4813**	**0.1611**
SLPAh-AI	0.0000	1.0000	0.0000	0.1466	0.3248
Louvainh-AI	0.0000	1.0000	0.0000	0.0768	0.2457
OHC-IS	0.0000	1.0000	**0.4000**	**0.4822**	**0.1913**
SLPAh-IS	0.0000	1.0000	0.0000	0.1233	0.2965
Louvainh-IS	0.0000	1.0000	0.0000	0.0539	0.2244
OHC-AAN	**0.0900**	1.0000	**0.5000**	**0.4412**	**0.1522**
SLPAh-AAN	0.0000	1.0000	0.0000	0.1477	0.3388
Louvainh-AAN	0.0000	1.0000	0.0000	0.0866	0.2678

4.4.4 Community Quality

In our experiments, we evaluated the performance of benchmark algorithms, SLPAh and Louvainh, and compared them with OHC. We calculate the minimum, maximum, median and average values for FOMD and FlakeODF for the six datasets and collate them in Tables 4.3 and 4.4 respectively, with the best results highlighted for each dataset. As SLPAh and Louvainh are non-deterministic, we present their best results from multiple runs. Additionally, the number of heterogeneous communities detected by each technique is shown in Table 4.2 (for SLPAh and Louvainh, number of communities are recorded from their best runs).

Notice in Table 4.3 that SLPAh and Louvainh despite having the same maximum value as OHC, their median FOMD value is 0. This indicates that at least half of their communities scored very poorly resulting in an overall lower average. On the other hand, OHC's positive results were reinforced by its higher median and lower standard deviation values, indicating the distribution between their communities was less volatile. The average FOMD values for OHC communities were approximately five times higher than the other two techniques. Table 4.4 shows that the median FlakeODF score for SLPAh and Louvainh is either 0 or 1, indicating the communities produced by these algorithms have FlakeODF scores at either extreme. The average FlakeODF score for OHC communities are considerably lower than other techniques, particularly in the AAN dataset.

Figures 4.3 to 4.8 presents the distributions of FOMD and FlakeODF value ranges across different heterogeneous community detection heuristics. From the results of the three datasets presented, we identify common trends that underline the performances of OHC, SLPAh and Louvainh. From Figures 4.3 to 4.5, we see that both SLPAh and Louvainh have a larger percentage of communities in the lowest value range (0-0.2). This indicates that most of SLPAh and Louvainh's resulting communities have low internal connectivity. On the other hand, OHC's result presents a more normal distribution curve, with the majority of communities scoring between 0.4 and 0.6, indicating that OHC produces communities with denser internal connectivity on average. The FlakeODF value ranges in Figures 4.6 to 4.8 again indicate favorable results for OHC. The results from SLPAh and Louvainh communities form clusters on the value ranges of both extremes, indicating around half of the communities have poor quality. Whereas the trend from OHC presents a sharp negative slope, with the majority of communities in the 0 to 0.4 value ranges. Notice that the described common trends were observed consistently in all of the six datasets.

Due to SLPAh and Louvainh's non-deterministic nature, we ran each of them thirty times and recorded their average FOMD and FlakeODF and standard deviation values (denoted as σ in our tables) of average FOMD and FlakeODF in Tables 4.5 to 4.8. There were minor variation in the average values obtained from each run, indicating both SLPAh and Louvainh had fairly stable performance.

In SLPAh and Louvainh we appended papers which were published by all authors within a community. However, this methodology does not include papers that have been authored by a subset of authors within a community. We produced two additional sets of results, firstly adding all papers that were authored by one or more authors within a community and secondly adding all papers that were authored by two or more authors within a community. In both cases, OHC retains higher minimum, median and average FOMD values across all datasets. However, the modified SLPAh and even Louvainh slightly outperform OHC in some cases in terms of FlakeODF scores. The results indicate that when adding papers that require at least two internal authors, the modified SLPAh and Louvainh produce a more tightly internally bound

Table 4.4: FlakeODF results

Method-Dataset	Min	Max	Median	Average	SD
OHC-DM	**0.6700**	0.0000	0.1000	**0.1812**	**0.1523**
SLPAh-DM	1.0000	0.0000	**0.0000**	0.2667	0.4566
Louvainh-DM	1.0000	0.0000	1.0000	0.7485	0.4069
OHC-SE	**0.8633**	0.0000	0.2000	**0.2411**	**0.1615**
SLPAh-SE	1.0000	0.0000	**0.0000**	0.3379	0.4777
Louvainh-SE	1.0000	0.0000	1.0000	0.7437	0.4384
OHC-CGM	**0.7800**	0.0000	0.1900	**0.2111**	**0.1516**
SLPAh-CGM	1.0000	0.0000	**0.0000**	0.4074	0.4947
Louvainh-CGM	1.0000	0.0000	1.0000	0.6997	0.4566
OHC-AI	**0.8923**	0.0000	0.2000	**0.2233**	**0.1544**
SLPAh-AI	1.0000	0.0000	**0.0000**	0.3925	0.4911
Louvainh-AI	1.0000	0.0000	1.0000	0.6811	0.4677
OHC-IS	**0.9333**	0.0000	0.3000	**0.3466**	**0.1910**
SLPAh-IS	1.0000	0.0000	**0.0000**	0.4211	0.4922
Louvainh-IS	1.0000	0.0000	1.0000	0.6194	0.4888
OHC-AAN	**0.8900**	0.0000	**0.1000**	**0.2214**	**0.1422**
SLPAh-AAN	1.0000	0.0000	1.0000	0.5188	0.4986
Louvainh-AAN	1.0000	0.0000	1.0000	0.5934	0.4744

Table 4.5: Average and standard deviation of FOMD across thirty runs for each threshold r of SLPAh

Dataset	r=0.01	σ(r=0.01)	r=0.25	σ(r=0.25)	r=0.45	σ(r=0.45)
DM	0.1912	0.0026	0.1900	0.0021	0.1873	0.0031
SE	0.1701	0.0025	0.1733	0.0023	0.1700	0.0041
CGM	0.1233	0.0046	0.1322	0.0032	0.1321	0.0052
AI	0.1242	0.0043	0.1311	0.0041	0.1311	0.0061
IS	0.0910	0.0033	0.1013	0.0023	0.1133	0.0035
AAN	0.1311	0.0051	0.1322	0.0062	0.1313	0.0056

community and more loosely externally connected community as compared to adding papers of each author, likely due to the decrease in loosely connected papers.

4.4.5 Case Study

The purpose of the case study is to evaluate our results qualitatively. For each dataset, we examined the top five heterogeneous communities with the highest FOMD and lowest FlakeODF scores generated by OHC, SLPAh and Louvainh and found: 1) By analyzing common keywords across the paper titles of a community, OHC was often able to identify additional information such as the research sub-field that the community focuses on, 2) OHC clusters authors

Table 4.6: Average and standard deviation of FOMD across thirty runs for Louvainh

Dataset	Louvainh	σ(Louvainh)
DM	0.0683	0.0007
SE	0.0803	0.0021
CGM	0.0611	0.0006
AI	0.0683	0.0004
IS	0.0512	0.0004
AAN	0.0815	0.0011

Table 4.7: Average and standard deviation of FlakeODF across thirty runs for each threshold r of SLPAh

Dataset	r=0.01	σ(r=0.01)	r=0.25	σ(r=0.25)	r=0.45	σ(r=0.45)
DM	0.2883	0.0021	0.2932	0.0011	0.2911	0.0030
SE	0.3410	0.0053	0.3466	0.0048	0.3526	0.0072
CGM	0.4211	0.0071	0.4166	0.0052	0.4133	0.0065
AI	0.4580	0.0161	0.4011	0.0143	0.4422	0.0175
IS	0.4393	0.0202	0.4533	0.0200	0.4462	0.0201
AAN	0.5311	0.0116	0.5233	0.0112	0.5277	0.0122

Table 4.8: Average and standard deviation of FlakeODF across thirty runs for Louvainh

Dataset	Louvainh	σ(Louvainh)
DM	0.7911	0.0021
SE	0.7521	0.0023
CGM	0.7032	0.0024
AI	0.6942	0.0007
IS	0.6222	0.0011
AAN	0.6013	0.0032

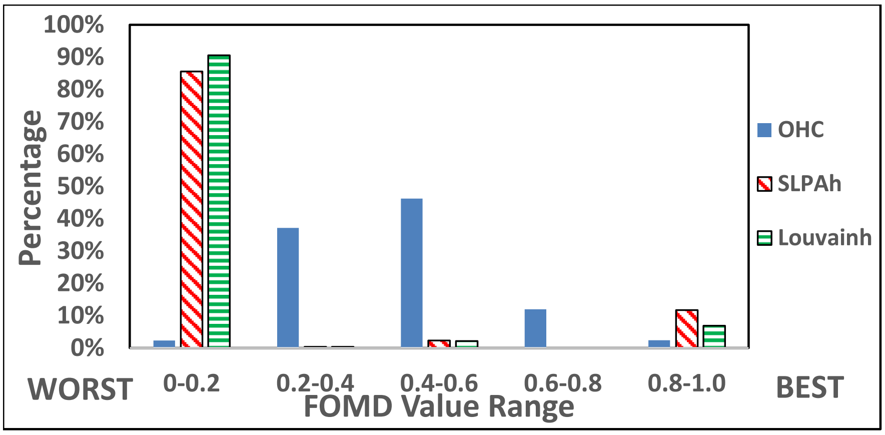


Figure 4.3: AAN FOMD

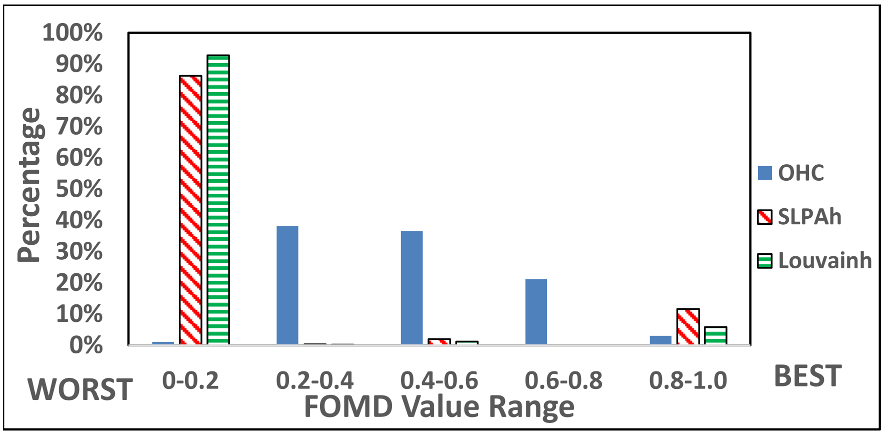


Figure 4.4: AI FOMD

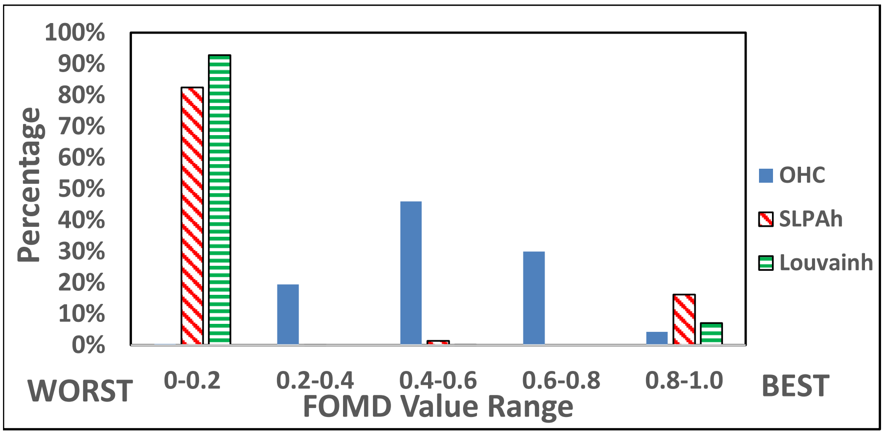


Figure 4.5: SE FOMD

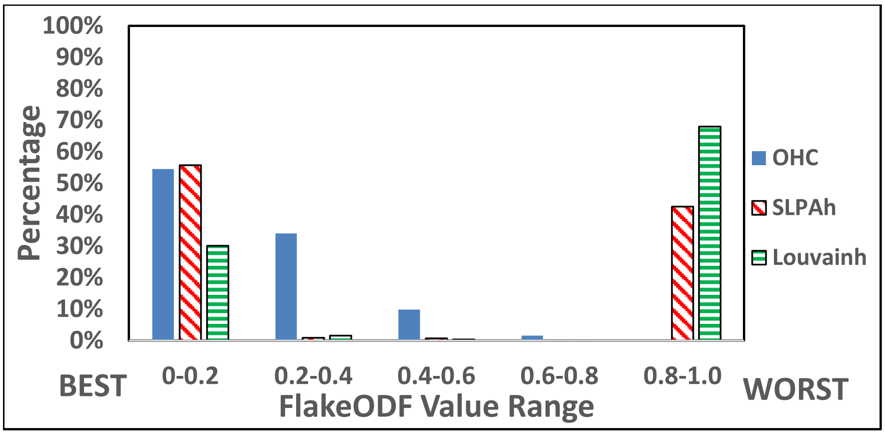


Figure 4.6: AAN FlakeODF

that frequently publish in the same field of research despite there being no co-publication between all of these authors, 3) Regardless of the paper appending methodology used, the vast majority of the top five scoring communities identified by SLPAh and Louvainh do not have these two properties.

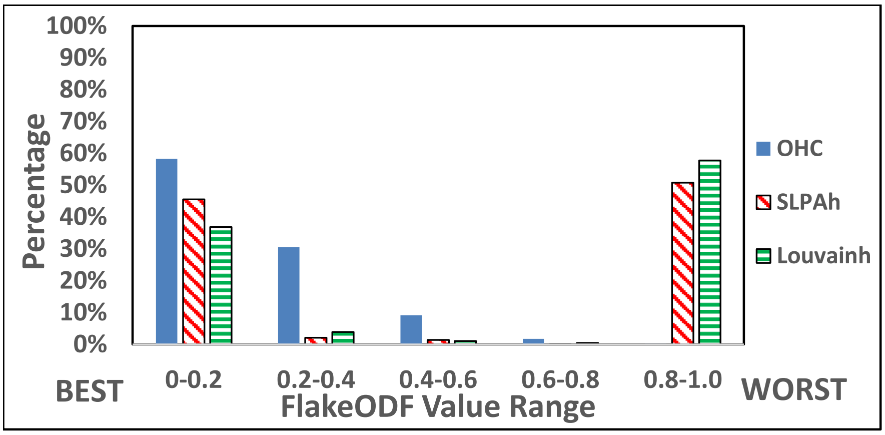


Figure 4.7: AI FlakeODF

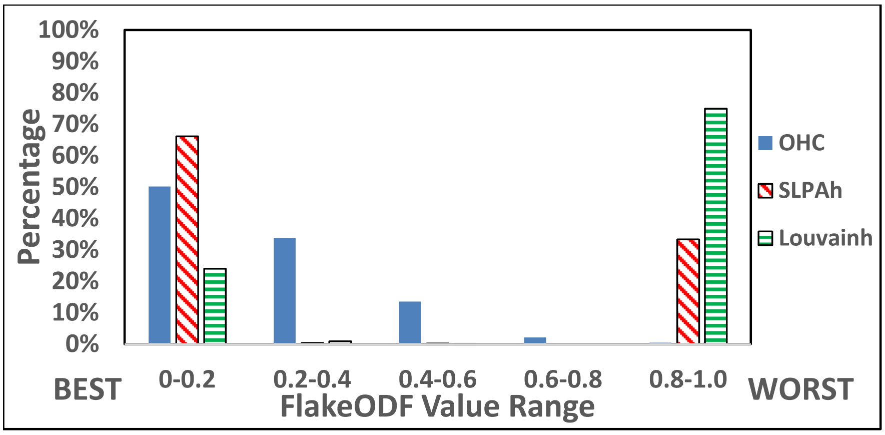


Figure 4.8: SE FlakeODF

We illustrate our findings further by analyzing the Top-1 AAN heterogeneous community detected by OHC which achieved a FOMD score of 1 and a FlakeODF score of 0. The common keyword across the five paper titles in this Top-1 community is "metaphor". In addition, by examining the AAN dataset, we found that the authors in this community had not published any papers together and only partial co-authorship exists among them. To reduce bias, we evaluated SLPAh and Louvainh communities that contain one or more nodes in the Top-1 OHC community and found that the same property was not demonstrated in these communities.

To present our findings in a systematic way, we analyze the top five AAN communities detected by each of OHC, SLPAh and Louvainh. Each of the top five OHC communities have common keywords across their paper titles. In contrast, for both SLPAh and Louvainh, only one of the top communities had common keywords. Additionally, for all of the SLPAh and Louvainh top five communities, there are only authors that have published the same papers together, which differs substantially from OHC. Overall, based on our analysis of the top scoring communities, OHC clusters authors with papers for a specific research sub-field despite the authors not being fully interconnected.

4.4.6 Run Time

Table 4.9 shows the execution time of each technique across various datasets. As expected, OHC consumed more time than both SLPAh and Louvainh. The difference increased with the growth of dataset size. As an example, OHC executed for 12860 seconds (3.5 hours) to detect heterogeneous communities in the AAN dataset, which we suspect was caused by high overlapping density and high overlapping diversity of the dataset. Notice that we show the average execution time and standard deviation of average execution time across thirty runs for both SLPAh and Louvainh in Table 4.9.

Table 4.9: Run time of algorithms in seconds

Dataset	OHC	SLPAh	σ(SLPAh)	Louvainh	σ(Louvainh)
DM	11	7	1	**2**	1
SE	26	9	2	**3**	1
CGM	492	62	1	**23**	1
AI	535	200	1	**58**	1
IS	658	254	2	**69**	1
AAN	12860	20	2	**9**	1

4.4.7 Recursion Depth

To identify possible improvements to the run time of OHC, we attempted to reduce the complexity of our algorithm by applying thresholds to the depth of recursions. We have accomplished this by setting a maximum depth threshold for our recursion process, which limits the number of pairs of ST nodes that can form cliques. This threshold therefore reduces the run time of OHC significantly, as we always stop at a constant depth, although it also sacrifices some community quality as we are no longer using a greedy approach.

Table 4.10 presents the results when limiting the recursion depth for our longest run time dataset, AAN. Our original run time without setting a recursion threshold results in a run time of 12860 seconds (3.5 hours), however, by setting a threshold of 3, we see an significant decrease in run time to only 198 seconds, which is approximately 1.5% of the original run time. As noted

Table 4.10: Recursion depth (AAN dataset)

Depth	Time(seconds)	No. of Communities	FOMD	σ(FOMD)	FlakeODF	σ(FlakeODF)
Not Set	12860	2385	0.4412	0.1522	0.2214	0.1422
3	198	2696	0.4322	0.1610	0.2866	0.1311
5	206	2667	0.4342	0.1678	0.2773	0.1328
10	268	2658	0.4388	0.1546	0.2762	0.1389

before, we see a minor decrease in community quality as the average FOMD value decreased by 0.0090 and average FlakeODF value increased by 0.0652, which accounts for the few outlier communities that recurse deeply. To view balancing effects of speed against quality, we tested higher thresholds and have found very minor differences in run time and community quality, indicating that when a threshold is not set, there could be a few expansive communities that contain deep recursions that encapsulate a large proportion of the heterogeneous network, causing the run time of OHC to spike.

Setting an appropriate threshold for the recursion depth can be a challenging task and it is also dependent on specific dataset characteristics. In addition, the threshold should be specified at a level where the balance between run time and community quality will be maintained.

4.5 Conclusions and Future Work

In this chapter, we identify heterogeneous communities by integrating and utilizing multiple node-to-node relationships that exist in heterogeneous correlation networks. The proposed OHC algorithm uncovers overlapping heterogeneous communities and has been shown to outperform benchmark techniques through higher FOMD values. In addition, by analyzing the top scoring communities in our case study, OHC clustered authors for specific research topics with indirect authorships more effectively. Future work will include improving OHC's efficiency by limiting depth of the recursive process in Phase Two, adapting OHC to find an evolution of communities in dynamic heterogeneous networks.

In this chapter, we find a positive answer to our main research question, which is heterogeneous information can improve the results of network analysis. We reach our conclusion by looking at one specific network analysis task, community detection. We compare the performance of our developed algorithm against two state-of-the-art homogeneous community detection methods. Unlike the homogeneous community detection methods, which can only generate communities of a single type of object, our algorithm processes relationships among multiple types of objects for identifying communities that contain more than one type of objects. Our experiment results show that our algorithm identifies communities with denser internal connectivity as compared to the two homogeneous methods.

5

Community Based Ranking of Objects

Ranking of objects in homogeneous networks has been studied extensively. However, real world networks are usually heterogeneous, containing multiple types of objects and relationships. To date, there has very limited amount of studies at analysing the ranking of objects in heterogeneous networks [165, 156] and to the best of our knowledge, none of these works consider inherent community structures. In this chapter, we propose a community-based approach, ComRank, for ranking different types of objects in a heterogeneous correlation network. ComRank determines the overall importance of objects by analyzing their connections and community memberships in the network. Therefore, the importance of an object is not only measured by the direct links it has with its one-hop neighbours, but also measured by the non-direct links the object has with its peers which reside in the same communities as the object. To evaluate the merits of our method, we compare it against PageRank, MultiRank and GPNRankClus in experiments on four real world datasets. According to external measurements, ComRank outperforms the three baseline techniques for identifying top rank objects.

Overall, our main research question is "Does heterogeneous information improve the results of network analysis?". To answer this research question, the thesis focuses on three network analysis tasks: community detection, ranking and network embedding. In this chapter, we focus on our second network analysis task, which is to co-rank different types of objects in static heterogeneous correlation networks. In doing so, we compare the performance of the ComRank algorithm against a homogeneous ranking technique. Specifically, based on domain-dependent evaluation metrics, we compare quality of top objects ranked by the different algorithms.

5.1 Introduction

There have been a variety of ranking algorithms developed for homogeneous networks [19, 76, 104, 15] but few for heterogeneous networks, which contain more than one type of object (node) and relationship (edge). Ranking of objects of different types in heterogeneous networks has many useful real world applications, for example, recommending top rank (top-k) publications and authors [165] to researchers or journal editors.

A heterogeneous correlation network [79] categorizes objects as either Source Type (ST) or Attribute Type (AT) and describes relationships among the two types of objects. A heterogeneous correlation network can be used to specify a unique dependency constraint on the relationships among multiple types of objects. ST objects act as the hub objects and are connected to different AT objects, whereas AT objects determine the relationships among their linked ST objects. Figure 5.1 provides an example of an academic correlation network. In this example, we choose to represent authors and papers as ST objects and AT objects respectively. The number of authorships between two authors is determined by the number of their co-published papers. Two heterogeneous communities are highlighted by dotted lines, where one community contains Authors A, B and Papers 1, 2 while the other community contains Authors B, C and Paper 3. The two communities share one common object (Author B), therefore they overlap with each other. Author B is the only object in the network that belongs to more than one heterogeneous communities.

We propose the ComRank (**Com**munity-based **Rank**ing) algorithm to co-rank objects of different types in heterogeneous correlation networks. Our approach calculates the importance of an object by analyzing connections and community memberships of the object in a network. To the best of our knowledge, ComRank is the first co-ranking scheme that takes the community structures into account.

The novelty of our approach lies in three aspects: (1) ComRank determines

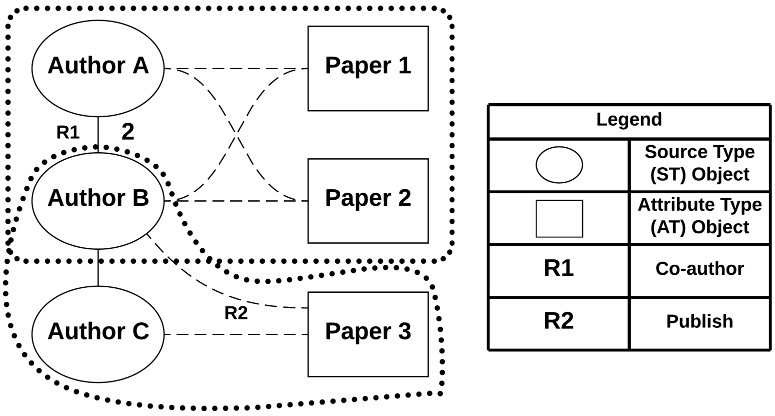


Figure 5.1: Example of an academic heterogeneous correlation network

the importance of each object according to direct and indirect relationships between the object and its peers within the communities in which the object resides, which is unlike the existing ranking techniques that measure the importance of an object solely based on connections it has with its one-hop neighbour nodes; (2) In contrast to homogeneous networks, ComRank uses information of multiple sources in a heterogeneous correlation network to calculate the object importance; (3) ComRank normalizes the importance of each community in the network and calculates the importance of an object in all the communities in which the object resides.

To evaluate the effectiveness of our method, we compare top objects ranked by ComRank against three other benchmark techniques, PageRank [19], MultiRank [101] and GPNRankClus [28] with both academic and social network data. External validation metrics for measuring influence or popularity of the top objects are used in our evaluations.

5.2 Preliminaries

We introduce two algorithms that are adopted in our community-based co-ranking scheme in this section. PageRank [19] is a link analysis algorithm used by Google's search engine. The algorithm was originally designed for the webgraph, treating all World Wide Web pages as nodes and hyperlinks as edges. The PageRank value of a particular webpage, p, represents the page's importance. The value of p is calculated recursively and determined by the number and PageRank values of all pages that are connected to p. The underlying assumption of the PageRank algorithm is that more important websites tend to have more links from other important sites, therefore, a page that is connected to many pages with high PageRank values will also receive a high PageRank. A simplified PageRank algorithm can be expressed as:

$$PR(p) = \sum_{q \in C_p} \frac{PR(q)}{L(q)} \tag{5.1}$$

where $PR(p)$ is the PageRank value for p, that is dependant on the PageRank value of each page q in the set C_p (the set containing all pages connected to the page p), divided by $L(q)$ (the number of outbound links from page q).

The OHC (**O**verlapping **H**eterogeneous **C**ommunities) algorithm was proposed in Chapter 4 to identify communities in heterogeneous correlation networks. The algorithm is comprised of three main phases. During the initial phase, heterogeneous seed communities are identified based on a distinct set of interconnected Source Type (ST) nodes. Phase Two recursively expands each seed community with triangle-forming ST nodes and associated Attribute Type (AT) nodes. In general, ST nodes work as anchoring objects for structuring triangles and appending tightly-connected AT nodes. The last phase removes duplicate and subset heterogeneous communities generated from Phase Two.

The OHC algorithm has been experimentally shown to uncover overlapping heterogeneous communities with denser internal and looser external connectivity as compared to the baseline techniques. For example, the interrelationships of objects within the OHC-detected communities are more cohesive than objects within communities identified by the benchmark techniques. In addition, the degree of independence to objects in other OHC-detected communities is higher than their counterparts identified by the baselines.

5.3 The ComRank Algorithm

Our proposed algorithm, ComRank, produces a ranked list for objects of different types in a heterogeneous correlation network with five main steps. In the first step, the PageRank value of each node in the network is calculated and used to represent the global importance of the node. The second step calculates the importance of all OHC-detected heterogeneous communities. In step three, the importance of a node for a particular community that the node resides in is calculated first, and then by aggregating importance of the node in all communities to which the node belongs, we generate the local importance of the node. Step four outputs the overall importance of a node by combining its global importance and local importance. In the final step, ComRank ranks all objects in the given heterogeneous correlation network according to their overall importance.

Definition 6. A Heterogeneous Correlation Network is an undirected and weighted graph $G = (V, E, T)$ in which each node v and each edge e is associated with their mapping functions $\phi(v) : V \rightarrow T_V$ and $\varphi(e) : E \rightarrow T_E$ respectively. T_V and T_E represent the sets of object and relationship types, where $|T_V| + |T_E| > 2$. In addition, T_V can be categorized as either source type or attribute type.

In a heterogeneous network, an object tends to connect to a certain group of objects with particular characteristics, therefore, objects form communities within the network [28]. The importance of an object is dependent on the number and importance of the object's connections, the community memberships of the object and the role that the object plays in the communities that it belongs to. In general, ComRank awards a high overall importance score to an object, that is linked with many other high-scoring objects (high global importance) and is a key member of important communities (high local importance). We now describe details of the algorithm step by step with a running example.

Step 1: Calculate global importance of node

ComRank first applies PageRank on a heterogeneous correlation network to obtain each object's global importance, as represented in Equation 5.2 where $GI(i)$ and $PR(i)$ are global importance and PageRank value of an object, i, respectively. Global importance of the object is calculated recursively and dependent on the number and global importance values of all objects that connected to the object in the network.

$$GI(i) = PR(i) \tag{5.2}$$

While it was initially designed for directed networks, PageRank has also been applied to undirected networks [143, 110]. In our first step, PageRank is applied on an undirected heterogeneous network, where the out-degree of a node is equal to the in-degree of the node. In this step, PageRank takes all objects existing in the network as nodes and relationships between objects as edges. All objects existing in the given network are treated as the same type when applying PageRank on the network. Figure 5.2 shows an example undirected heterogeneous correlation network, where circles and rectangles represent two different types of objects. While solid lines and dashed lines represent two different types of relationships. The example network is a weighted graph, the

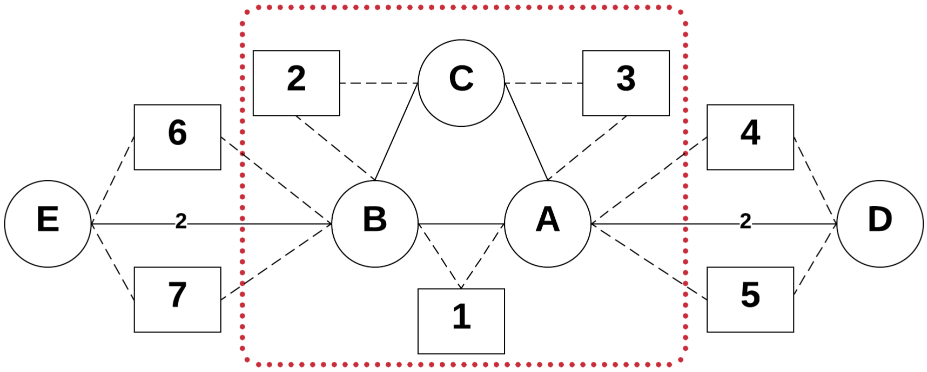


Figure 5.2: Example of a heterogeneous correlation network

Table 5.1: Rank of nodes for global importance

Rank	Nodes	Global Importance
#1	A, B	0.1787
#2	C, D, E	0.0943
#3	2, 3, 4, 5, 6, 7	0.0515
#4	1	0.0505

probability of following an edge out of the object, i, is equal to the weight of the edge over the sum of weights of all edges of i. Calculated global importance of nodes in the example network is ranked and presented in Table 5.1.

Step 2: Calculate importance of heterogeneous community

Being a community-based co-ranking technique, ComRank measures the importance of an OHC-detected heterogeneous community, c', by aggregating the global importance of each object, i, in c' (Equation 5.3).

$$I_c(c') = \sum_{i \in c'} GI(i) \tag{5.3}$$

As denoted by the dotted lines in Figure 5.2, there exists only one heterogeneous community in the example network. The importance of the community is calculated by combining the global importance of its members, which are A, B, C, 1, 2 and 3.

Step 3: Calculate local importance of node

ComRank determines the local importance $LI(i)$ of an object, i, using Equations 5.4 and 5.5. Equation 5.4 calculates the importance of the object, i, for a particular community, c', which i resides in. The contribution that i makes in c' is measured by the fraction of i's global importance over the importance of the community, c', ($I_c(c')$). To normalize the importance of c' in the heterogeneous network, we first utilize the maximum function, Max, to return the highest community importance score in the network (note that $I_c(C)$ gives us the importance of all heterogeneous communities, C, in the network). The importance of c' is then divided by the value returned from the Max function to obtain the normalized importance of c'. The final step of Equation 5.4 is to calculate i's importance in c' by multiplying the contribution that i makes in c' with the normalized importance of c' in the network.

$$I_n(i \in c') = \frac{GI(i)}{I_c(c')} \times \frac{I_c(c')}{Max(I_c(C))} \tag{5.4}$$

$$LI(i) = \sum_{c' \in C_i} I_n(i \in c') \tag{5.5}$$

Since it is possible for an object, i, in a heterogeneous correlation network to reside in more than one community, to obtain the local importance of i, we use Equation 5.5 to aggregate the importance of i across all communities, C_i, that i resides in. Since our example network in Figure 5.2 has only one heterogeneous community, the local importance for each object in the community is calculated using Equation 5.4.

One of PageRank's main weaknesses is that it disregards new objects as they do not have many connections. ComRank mitigates this issue by considering the community memberships of newcomers, where the overall importance of new objects that reside in important communities can be increased by their high local importance scores.

Step 4: Calculate overall importance of node

The algorithm derives the overall importance of an object, i, by aggregating its global and local importance. The roles played by the global and local importance in determining the object's overall importance is considered as equivalent by ComRank, hence balanced weights are given to them during the aggregation process. Equation 5.6 reflects this object importance, where $OI_n(i)$ represents the overall importance of i.

$$OI_n(i) = GI(i) + LI(i) \tag{5.6}$$

Calculated overall importance of nodes in our example network is ranked and presented in Table 5.2. Compared with their scores in Table 5.1, only objects

Table 5.2: Rank of nodes for overall importance

Rank	Nodes	Overall Importance
#1	A, B	0.4740
#2	C	0.2500
#3	2, 3	0.1367
#4	1	0.1339
#5	D, E	0.0943
#6	4, 5, 6, 7	0.0515

included in the OHC-detected community have changed scores.

Step 5: Rank nodes by their overall importance

Lastly, ComRank produces a heterogeneous ranking list according to the overall importance of nodes presented in a descending order. It is obvious that the rank of nodes in Table 5.2 have changed from Table 5.1. For example, node C no longer shares the same rank with nodes D and E, despite having the same number of connections as D and E (two connections with each type of object), by residing in the same community as the other two important nodes (A and B), node C is ranked higher than D and E by ComRank after taking C's community membership into consideration.

Algorithm 3 describes the main ideas of our proposed approach. The importance of all OHC-detected communities is calculated from line 1 to 6. Afterwards, from line 7 to 13, ComRank loops through all objects in the network and their community memberships to calculate the local importance of them. The last piece (lines 14 to 17) of the algorithm outputs the overall importance of an object by aggregating its global and local importance. Algorithm 3 shows two characteristics of ComRank: 1) Source Type (ST) and Attribute Type (AT) objects are not differentiated in their global, local and overall importance computation, the two types of nodes are only differentiated in the community detection process; 2) The overall importance (or rank) of an object is positively correlated with the number of communities in which the object resides.

The novelty of ComRank lies in three main aspects: 1) Unlike probability distribution based co-ranking methods [101, 28], which focus only on direct relationships between the objects, ComRank considers community structures to leverage implicit relationships between the community members; 2) Unlike PageRank [19] which treats multiple types of objects as the same type, different types of objects are differentiated in ComRank's adopted heterogeneous community construction process; 3) The importance of OHC-detected communities is normalized by the maximum community importance in the network.

The performance of ComRank partially depends on the OHC algorithm because ComRank determines the overall importance of nodes by analysing their community memberships as well as their connections (represented by their PageRank values). ComRank can adopt other community detection algorithms to calculate heterogeneous community importance and node local importance in Step 2 and 3 respectively, which may result in a different final rank list.

Algorithm 3 ComRank

Input: OHC-detected communities: C
PageRank value of each object i**:** $PR(i)$
Output: Object ranking list: R

```
for all $c$ in $C$ do
    for all $i$ in $c$ do
        $GI(i) = PR(i)$
        $I_c(c)$ += $GI(i)$
    end for
end for
for all $i$ in $I$ do
    for all $c$ in $C$ do
        if $i \subseteq c$ then
            $LI(i)$ += ($GI(i)$ / $I_c(c)$) * ($I_c(c)$ / max($I_c(c)$))
        end if
    end for
end for
for all $i$ in $I$ do
    $OI_n(i) = GI(i) + LI(i)$
    $R$.add($OI_n(i)$)
end for
SORT($R$. DESCENDING)
```

5.3.1 Time Complexity Analysis of ComRank

The time complexity of ComRank is determined by the number of objects and OHC-detected communities, which are denoted as n and m respectively. The execution time for calculating the importance of all heterogeneous communities is m. ComRank iterates through all objects and communities to calculate the local importance of each object, resulting in an execution time of $(n * m)$. In addition, calculating the overall importance of all objects and sorting them into descending order requires a maximum execution time function of $(n+n^2)$. Finally, with all parts summarized, the time complexity of ComRank is $O(n*m+n^2)$.

5.4 Experimental Evaluation

To evaluate the performance of ComRank, we apply the algorithm on various real world datasets and compare the results against PageRank [19], Multi-Rank [101] and GPNRankClus [28]. The damping factor of PageRank in our experiments is consistently set as 0.85. A mixed ranking list of different types of objects is generated by each technique. Objects of different types cannot be compared directly, therefore we extract a ranking list for each individual type and evaluate them separately. All the experiments were performed on a standard desktop PC with 3.40 GHz i7 CPU, 8 GB RAM and Windows 10 operating system. A Java implementation of ComRank and datasets used in our experiments are available for download[1].

Unlike the majority of the existing work, which evaluate their algorithms only on academic heterogeneous networks, we include social network datasets to understand ComRank's co-ranking capability in different domains. The three academic datasets used in our evaluations are the ACL Anthology Network **(AAN)** [112] dataset[2], Interdisciplinary Studies **(IS)** and Computer Graphics Multimedia **(CGM)** bibliographic datasets[3]. For the social network dataset, we analyze the Digg dataset[4] retrieved from ArnetMiner [134].

To compare the performance of each ranking method, we adopt an external validation technique which evaluates the importance of top ranked objects with domain-dependent metrics. For academic datasets, we use the metrics: citation count and h-index value for authors, and citation count for papers. For the Digg dataset, we use the number of friends as the metric for Digg users and the number of likes for stories. These domain-dependent metrics were selected as our evaluation measures as they represent desired characteristics of the top ranked objects.

[1] https://github.com/comrank/ComRank.git

[2] http://clair.eecs.umich.edu/aan/index.php

[3] https://www.aminer.cn/citation

[4] https://aminer.org/heterinf

Table 5.3: Properties of academic datasets

Dataset	No. of Authors	No. of Papers	No. of Collaborations	No. of Authorships
AAN	18968	17653	62925	42236
IS	50996	18583	176134	77165
CGM	29359	16599	57251	48422

Table 5.4: Average citation count for top-k papers (IS)

	@ Top 10	@ Top 20	@ Top 30	@ Top 40	@ Top 50
ComRank	**331**	**195**	**142**	**166**	**159**
PageRank	246	166	129	109	92
MultiRank	245	176	137	118	105
GPNRankClus	97	76	63	159	136

5.4.1 Academic Datasets

For each academic dataset, we construct a heterogeneous correlation network, containing two types of objects: authors and papers. Figure 5.1 shows an example structure of our input network. The edge between two authors indicates the number of papers that the corresponding authors have co-published (R1: denoted by the solid line in Figure 5.1). An unweighted and undirected edge existing between an author and a paper represents the author-to-paper relationship (R2: denoted by the dashed line in Figure 5.1). The properties of each academic dataset is presented in Table 5.3.

As shown in Table 5.4, for the IS dataset, the top papers identified by ComRank have a higher average citation count compared to the other ranking algorithms. We evaluate the average paper citation count for every 10th interval of top papers and find that ComRank consistently outperforms other benchmark techniques, with more than double the average citations when measured against GPNRankClus for the top 30 papers. From these results, we show that ComRank is able to rank papers more successfully based on their sphere of influence and their importance in the network.

This trend is also exhibited for the IS author ranking results, as shown in Tables 5.5 and 5.6, where the top authors identified by ComRank have higher average citation counts and h-indexes. We find two exceptions where MultiRank and PageRank produce better results for h-index at the top 10 and top 50 mark respectively. However, these differences are minor and could be influenced by the benchmark techniques identifying outlier authors with high h-index values but relatively lower citation counts.

For the other two academic datasets, we observed similar trends. We list the top 10 ranked papers for AAN and the top 10 ranked authors for CGM.

Table 5.5: Average citation count for top-k authors (IS)

	@ Top 10	@ Top 20	@ Top 30	@ Top 40	@ Top 50
ComRank	**32939**	**34931**	**29052**	**25992**	**22729**
PageRank	26973	25657	23291	21064	20399
MultiRank	27233	28095	23266	20427	20101
GPNRankClus	13459	11608	11228	11339	11478

Table 5.6: Average h-index for top-k authors (IS)

	@ Top 10	@ Top 20	@ Top 30	@ Top 40	@ Top 50
ComRank	63	**66**	**60**	**57**	53
PageRank	62	59	55	55	**55**
MultiRank	**64**	62	57	55	54
GPNRankClus	49	49	46	46	46

Table 5.7: Top 10 papers identified for AAN

ComRank		PageRank		MultiRank		GPNRankClus	
Paper ID	c *	Paper ID	c	Paper ID	c	Paper ID	c
W09-1201	202	L08-1355	5	W09-1201	202	W99-0304	0
P02-1056	76	E09-1054	3	L08-1467	20	L08-1609	0
W11-2101	48	L08-1084	2	H01-1038	7	L08-1373	5
H93-1042	39	W11-0903	15	W99-0304	10	W08-1506	6
P07-2045	4206	W09-3006	4	W10-1108	12	H93-1042	39
H91-1060	581	L08-1609	0	W10-1829	15	E03-1012	1
C96-2120	65	W10-1844	10	H05-2013	2	E03-2001	1
L08-1581	4	P08-2046	2	N07-4012	7	L08-1084	2
C94-1072	62	W11-2157	9	L08-1355	5	W05-1106	78
W09-2411	63	L08-1058	3	P10-4009	4	W04-2507	12

* c = Citation Number

Table 5.7 presents the top 10 papers identified by each ranking algorithm when evaluating the AAN dataset. From these results, we find that in most cases ComRank detects top papers with higher citation counts, in particular, it identifies the top 8 papers with the highest citation counts in our list. These characteristics reflect ComRank's ability to accurately identify highly influential publications and rank them accordingly.

When ranking authors, ComRank also produces more favorable results when compared to the benchmark techniques. Table 5.8 presents the top 10 authors identified by each ranking technique for the CGM dataset and from these results, we find that in general, ComRank is able to identify authors with higher citation counts and h-index values. Although each technique would often identify a common set of influential authors (e.g. M. Sharir), ComRank

Table 5.8: Top 10 authors identified for CGM

ComRank			PageRank			MultiRank			GPNRankClus		
Name	h *	c *	Name	h	c	Name	h	c	Name	h	c
M.Sharir	53	11547	M.Sharir	53	11547	M.Sharir	53	11547	R.T.Farouki	29	2951
D.Tao	82	21801	A.Katsaggelos	53	12180	A.Katsaggelos	47	9446	M.Moonen	48	8927
S.Yan	66	22179	R.Chellappa	72	27211	G.Elber	25	2360	I.Song	15	828
H. E.	53	12462	G.Elber	28	2909	M.Unser	59	17983	P.Stoica	67	19657
L.J.Guibas	71	20936	M.Unser	71	24344	H.P.Seidel	69	18974	S.C.Pei	53	10904
J.Pach	31	4810	N.Thalmann	45	7120	N.Thalmann	43	6394	G.Bi	21	1771
S.M.Hu	38	6464	H.P.Seidel	69	18858	X.Wu	37	7454	R.Scopigno	37	4878
W.Gao	57	16731	I.Pitas	56	12227	R.Chellappa	63	20506	A.Farina	44	7053
B.Aronov	26	1894	B.Jüttler	26	2951	B.Jüttler	25	2742	I.Djurovic	25	2720
T.S.Huang	97	50022	X.Wu	37	7573	I.Pitas	53	10810	Z.Bao	45	8724

* h = H-Index, c = Citation Number

is shown to detect strong influencers that are different from other algorithms (such as T.S. Huang and S. Yan), which contributes towards its positive results.

5.4.2 Spearman Rank Correlation

Since each algorithm produces a list of authors and papers with rankings, we can apply Spearman rank correlation [62] to compare the results of ComRank and the other algorithms to determine the degree of rank correlations between them. Table 5.9 presents the Spearman rank correlation for ranked authors and papers between ComRank and the benchmark techniques in the IS dataset.

PageRank exhibits the strongest author rank correlation with a high score of 0.7357 (with 1 indicating two ranking lists are identical), this is to be expected as ComRank utilizes the PageRank algorithm to obtain global importance of nodes. However, with a paper rank correlation score of 0.1512, the level of similarity between how ComRank and PageRank rank papers is shown to be low.

MultiRank is shown to have a moderate author rank correlation score of 0.5215, indicating some similarities in how both algorithms rank authors. However, despite only scoring 0.3774 for the paper rank correlation, MultiRank is shown to rank papers in the closest manner to ComRank when compared to the other two techniques. This indicates in terms of paper rankings, ComRank is shown to have little similarity with the three other techniques and there is weak to no association between how ComRank and the other algorithms rank papers.

With an author rank correlation score of 0.0846 and paper rank correlation score of -0.0437, GPNRankClus is shown to have almost no correlation with ComRank in the way they rank authors and even slightly dissimilar in terms

Table 5.9: Spearman rank correlation for the IS dataset

ComRank	PageRank	MultiRank	GPNRankClus
Author Ranking Correlation	0.7357	0.5215	0.0846
Paper Ranking Correlation	0.1512	0.3774	-0.0437

of paper rankings. With both scores being near 0, we can state that there is little ranking correlation between ComRank and GPNRankClus.

5.4.3 Digg Dataset

Digg [1] is a news and social platform, which specifically organizes stories on different topics for the Internet audience such as technology, science, trending political issues. We construct a heterogeneous correlation network, based on Digg users and their actions (reply and comment) with respect to the stories. Two types of objects in the constructed network are Digg users and stories. The number of edges between two users indicates the number of stories that the corresponding Digg users have both replied to or commented on. An unweighted and undirected edge existing between a user and a story represents the user-reply-to (or comment-on story) relationship. The number of Digg users in our experiment dataset is 10293, the number of Digg stories is 12924, the number of co-comment edges is 536248 and the number of user-comment-on edges is 78687.

Table 5.10 shows that ComRank is able to detect top 50 users with more friends when compared to benchmark techniques, indicating ComRank's ability to identify more active and popular users. A similar trend is observed for the top stories, where Comrank identifies the top 50 stories with the highest average number of likes as shown in Table 5.11. In addition, by analysing the topics and keywords of each top story, ComRank is shown to detect stories that were trending and more relevant for the Digg dataset's timestamp (January 2009). Topics such as Obama's election and iPhone's release in 2008 are common themes among the top 50 stories ranked by ComRank. Despite this, we acknowledge that it is difficult to determine a qualitative ranking of trending stories since its value and interest to individuals is subjective. Our results for ComRank on the Digg dataset are overall positive, however, since the dataset is a subset of data scraped from Digg in early 2009, we have to acknowledge that the underlying data could be incomplete and imbalanced to some degree. In addition, we only represent 'recall' in our experiment results as it was difficult to evaluate the accuracy of the results of each of the exper-

Table 5.10: Average friend number for top-k digg users

	@ Top 10	@ Top 20	@ Top 30	@ Top 40	@ Top 50
ComRank	**50**	**32**	**26**	**25**	**21**
PageRank	31	18	16	17	14
MultiRank	30	18	21	16	14
GPNRankClus	43	22	23	20	16

Table 5.11: Average like number for top-k digg stories

	@ Top 10	@ Top 20	@ Top 30	@ Top 40	@ Top 50
ComRank	**49**	**39**	**40**	**38**	**41**
PageRank	23	32	37	**38**	21
MultiRank	33	31	38	31	31
GPNRankClus	14	21	29	26	33

iments without the ground-truth information of the data sets. For example, the h-index values of all authors and the like numbers of all digg stories in the data sets were not available.

5.4.4 Run Time

Table 5.12 shows the execution time (in seconds) of each ranking technique across various datasets. Overall, ComRank took more time in ranking than the benchmark techniques in all four datasets. PageRank and GPNRankClus were the fastest algorithms. In fact, the vast majority of time taken by ComRank was spent in running the OHC algorithm to identity heterogeneous communities in the datasets.

Table 5.12: Run time of algorithms in seconds

Dataset	ComRank	PageRank	MultiRank	GPNRankClus
AAN	12895	**33**	42	39
IS	688	**26**	55	29
CGM	517	24	41	**23**
Digg	232	**20**	61	21

5.5 Conclusions and Future Work

In this chapter, we propose the algorithm, ComRank, that determines the importance of objects of different types based on their connections and community memberships in a heterogeneous correlation network. Experiments on real world data show that ComRank outperforms PageRank, MultiRank and GPNRankClus for the top ranked objects according to external validation metrics. In addition, using Spearman rank correlation on the co-ranked lists generated by ComRank and the benchmark techniques, we observe an obvious discrepancy in the ways that objects are ranked by different methods. To the best of our knowledge, ComRank is the first approach that utilizes community structures for co-ranking. We hope our work will inspire future researchers to consider using community memberships of objects when developing new heterogeneous ranking methods.

In this chapter, we again find a positive answer to our main research question, which is heterogeneous information can improve the results of network analysis. We reach our conclusion by looking at one specific network analysis task, ranking. We compare the performance of our developed algorithm against PageRank, a popular homogeneous ranking method. Unlike PageRank, which considers different types of objects in a network as one type, our algorithm considers the network heterogeneity. Our experiment results show that as compared to PageRank, our algorithm ranks top objects with better quality according to domain-dependent external metrics.

Future work includes adapting ComRank to dynamic heterogeneous correlation networks so the algorithm can efficiently identify top-k objects in heterogeneous data streams, monitoring paradigm shifts of identified top objects in streaming heterogeneous data and explaining the causes behind the shifts. In addition, we are also interested in implementing and experimenting with variations of the ComRank algorithm. For example, to differentiate Source Type and Attribute Type objects in ranking computation.

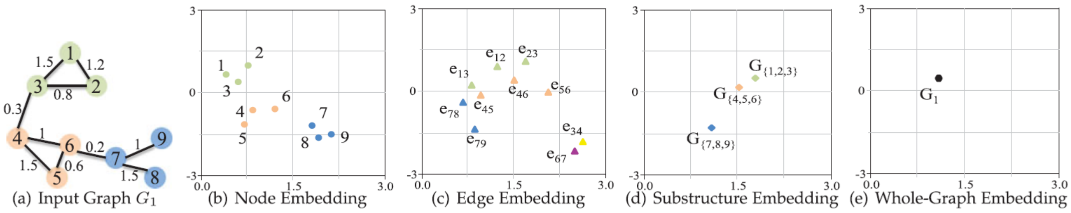
6
Network Embedding and Change Modeling in Dynamic Heterogeneous Networks

The target of network embedding is to learn the vector representations of a given network. Most real world networks are heterogeneous and evolve over time. Although studying network embedding would be useful in mining and analyzing real life networks, there has been very little research in developed so far on this topic. In this chapter, we address the problem of learning the vector representations that capture the structural relationships among the objects at any time step for a given dynamic heterogeneous network. We develop a novel network embedding method, change2vec, which considers a dynamic heterogeneous network as snapshots of static networks with different time stamps. Instead of processing the whole network at each time stamp, change2vec models changes between two consecutive static networks. It does this by capturing newly-added or deleted nodes with their neighbouring nodes in addition to newly-formed or deleted edges where there are core structural changes known as triad closure or open processes. Change2vec leverages metapath based node embedding and change modeling to preserve both heterogeneous and dynamic features of a network. Experimental results show that change2vec outperforms two state-of-the-art methods in terms of clustering performance and efficiency.

Overall, our main research question is "Does heterogeneous information improve the results of network analysis?". To answer this research question, the thesis focuses on three network analysis tasks: community detection, ranking and network embedding. In this chapter, we focus on our third network analysis task, which is network embedding in dynamic heterogeneous networks. In doing so, we compare the performance of the change2vec algorithm against a state-of-the-art homogeneous network embedding technique. Specifically, we compare accuracy of the embedding vectors generated by the different algorithms.

6.1 Introduction

Network (graph) embedding is also known as representation learning, which focuses on mapping a network into a low dimensional space (vectors) in which the graph structural information and graph properties are maximally preserved [22]. Figure 6.1 shows four different ways of converting a toy network into a 2D space. We base this chapter on node embedding, which maps a given network into a low dimensional space via learning object representation vectors. Node embedding has a vital role in important data mining tasks such as object clustering and classification [103, 144], entity retrieval and recommendation [163].



(a) Input Graph G_1 (b) Node Embedding (c) Edge Embedding (d) Substructure Embedding (e) Whole-Graph Embedding

Figure 6.1: Converting a network into 2D space with four different perspectives [22]

Most real world networks are heterogeneous and evolve constantly. Typical examples of these networks are social networks and biological networks, where multiple types of objects (nodes) and relationships (edges) exist. Recently, a number of network embedding methods [109, 172, 168, 37, 128, 132] have been proposed to learn low-dimensional vector representation of a node based on its neighbourhood relationships. To the best of our knowledge, none of these approaches were tailored to dynamic heterogeneous networks.

Heterogeneous networks model different types of objects and relationships among them. As compared to homogeneous networks, heterogeneous networks can fuse information from multiple data sources and social platforms. Heterogeneous networks are used in scenarios such as community-based question answering sites, multimedia networks and knowledge graphs [22]. Heterogeneous networks are not always static, especially in real life scenarios, e.g., social networks in Twitter, citation networks in DBLP. Existing network embedding mainly focuses on embedding the static homogeneous network and embedding dynamic heterogeneous networks has been overlooked. Unlike static homogeneous network embedding, the techniques for dynamic heterogeneous networks needs to be scalable and incremental to deal with the dynamic changes efficiently in addition to model different types of interacting objects and their relationships. This makes most of the existing network embedding methods, which suffer from the low efficiency and single type oriented problems unsuitable. How to design effective network embedding methods for dynamic heterogeneous networks remains an open question. With the current development of network embedding techniques, there is still significant work to be done in scaling node and network embedding approaches to truly massive data sets (e.g., billions of nodes and edges) [53].

6.2 Research Overview

The intuition behind our proposed method, change2vec, is to utilize triads (i.e., a set of three nodes) to capture the structural changes of dynamic heterogeneous networks. A triad is a fundamental block of a network [148] and can be open or closed. In a closed triad, there exists a relationship between any two objects. Whereas in an open triad, there are only two edges, indicating that there is no relationship between two of the nodes. To capture formation and evolution of dynamic heterogeneous networks, we model how an open triad transitions into a closed triad through the *triad closure process* and how a closed triad becomes an open triad through the *triad open process* [59]. Figure 6.2 presents a dynamic heterogeneous social network where the circles represent users while squares denote topics of interest that users subscribe to. The network contains two types of relationships: friendship between two users represented by a solid line and subscription between a user and a topic denoted by a dashed line. The transition from time step t to $t + 1$ shows that two new edges were introduced to the network (edge e_{BC} connecting B and C and edge e_{BT} connecting B and T), which resulted in two triad closure processes. As the network evolves from time step $t + 1$ to $t + 2$, a triad open process occurs due to the disappearance of the edge e_{AC} between A and C. To generate up-to-date latent vectors (vector coordinates that reflect objects' positions in a low dimensional space) for all nodes in the network, traditional static network embedding methods such as Node2vec [48], DeepWalk [109] and metapath2vec++ [37] need to process the whole network at each time step, which is unmanageable for large dynamic heterogeneous networks. Instead, change2vec captures network structural changes by only updating vector representations for nodes that are involved in triad closure or open processes during two consecutive time steps. In general, it is natural for networks to evolve and transit smoothly over time, instead of volatile restructures between each time step [168]. This characteristic equips change2vec with advantages over static network embedding methods which process the whole network at each time stamp. For example, consider the network in Figure 6.2, at time step $t + 1$, change2vec only updates vector representations for B, C and T, at time step $t + 2$, only vector representations for A and C are updated, which is more efficient than approximating latent vectors for all nodes at each time step.

Metapath is defined in a heterogeneous network as a sequence of relationships between different object types, which describes a new composite relationship between its starting type and ending type [127, 37]. The metapath scheme provides a useful change capture mechanism by incorporating heterogeneous features of the network. The dynamic social network in Figure 6.2 contains two types of objects: users (U) and topics (P), which can form useful metapath schemes such as "UUU" and "UPU" for modeling triad closure and open processes. Change2vec is designed to capture, analyze and describe changes in evolving heterogeneous networks. Dynamic vector representations of nodes learned by change2vec can benefit various heterogeneous network mining tasks. For example, the embedding vector of each node can be used as the feature input of node clustering and similarity search, edge reconstruction and prediction tasks. Therefore, it is a critical requirement for network embedding methods to generate vector representations that will reflect the dynamic and heterogeneous characteristics of nodes [37].

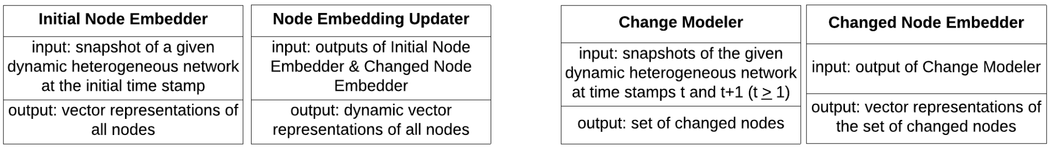

Initial Node Embedder
input: snapshot of a given dynamic heterogeneous network at the initial time stamp
output: vector representations of all nodes
Node Embedding Updater
input: outputs of Initial Node Embedder & Changed Node Embedder
output: dynamic vector representations of all nodes
Change Modeler
input: snapshots of the given dynamic heterogeneous network at time stamps t and t+1 (t ≥ 1)
output: set of changed nodes
Changed Node Embedder
input: output of Change Modeler
output: vector representations of the set of changed nodes

Figure 6.3: Framework of change2vec

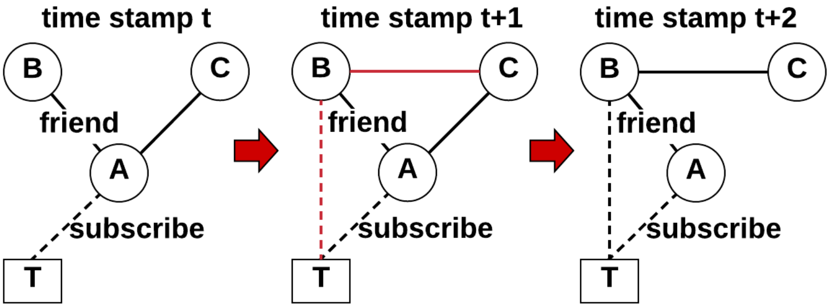


Figure 6.2: An illustrative example of a dynamic social network

Our research contributions are as follows: (1) We propose the method, change2vec, which incorporates both dynamic and heterogeneous features of a given network for the representation learning task; (2) When applying learned embedding vectors of nodes, change2vec outperforms two state-of-the-art network embedding methods in terms of Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI) scores; (3) Compared to a static heterogeneous network embedding method, change2vec is shown to be more run time efficient for networks that are constantly evolving.

6.3 The change2vec Algorithm

Change2vec processes a dynamic heterogeneous network as snapshots of static networks with different time stamps. Instead of reconstructing the network in each time step, we identify certain sets of changed nodes between two consecutive time steps, which will be considerably more run time efficient than processing the whole network at each step. Our framework aims to learn the embedding vectors of nodes by modeling the dynamic and heterogeneous properties of a network.

Definition 7. A Dynamic Heterogeneous Network is a constantly evolving undirected graph $G = (V, E, T)$ in which each node v and each edge e is associated with their mapping functions $\phi(v) : V \rightarrow T_V$ and $\varphi(e) : E \rightarrow T_E$ respectively. T_V and T_E represent the sets of object and relationship types, where $|T_V| + |T_E| > 2$. In addition, between two consecutive time stamps t and

$t+1$, $|V^t| \neq |V^{t+1}|$ or $|E^t| \neq |E^{t+1}|$, where $|V^t|$ and $|V^{t+1}|$ denote the number of objects at time stamps t and $t+1$ respectively, $|E^t|$ and $|E^{t+1}|$ represent the number of relationships at time stamps t and $t+1$ respectively. T_V and T_E remain unchanged across all time stamps t.

For example, the network in Figure 6.2 can be represented with users (U), and topics (P) as nodes, wherein edges denote the friendship ($U-U$), and subscription ($U-P$) relationships. The number of edges changes in the network between different time steps. We now formalize the vector representation learning problem in dynamic heterogeneous networks.

Problem 1. Dynamic Heterogeneous Network Representation Learning Given a dynamic heterogeneous network G, which is represented as a sequence of heterogeneous network snapshots G^t at different time steps t ($t =$ 1,...,z), the task is to learn the dynamic d-dimensional vector representations $X^t \in \mathbb{R}^{|V^t| \times d}$, $d \ll |V^t|$ that capture the structural relationships among the objects in G^t at the time step t.

The result is the low-dimensional matrix X^t, with the v^{th} row being a d-dimensional vector of node v at the time step t. Representations of different types of nodes in G are mapped into the same latent space. The learned embedding vectors of nodes can be used as feature inputs of various heterogeneous network mining tasks.

Figure 6.3 shows the framework of change2vec. The input of the Initial Node Embedder component is G^1, which is the snapshot of a given dynamic heterogeneous network G at the initial time stamp $t = 1$. This component leverages the metapath2vec++ algorithm [37] to use metapath based random walks as input training data for learning vector representations of all nodes in G^1.

When G evolves from time step t to $t+1$, change2vec models the differences between the network snapshots of the two consecutive time steps, G^t and G^{t+1}, by producing a set of changed nodes V_{change}. Nodes that are involved in the following four scenarios are included in V_{change}:

(1) added nodes and their one-hop neighbour nodes.

$$V_{change} = \{v, u : v, u \in V^{t+1}, v \notin V^t, (v, u) \in E^{t+1}\} \tag{6.1}$$

(2) deleted nodes and their one-hop neighbour nodes.

$$V_{change} = \{v, u : v, u \in V^t, v \notin V^{t+1}, (v, u) \in E^t\} \tag{6.2}$$

(3) formed edges which caused triad closure processes.

$$\begin{aligned} V_{change} = \{v, u : (v, u) \notin E^t, \{(v, u) \in E^{t+1}, \\ (v, w) \in E^{t+1}, (u, w) \in E^{t+1}\} \neq \emptyset\} \end{aligned} \tag{6.3}$$

(4) deleted edges which caused triad open processes.

$$\begin{aligned} V_{change} = \{v, u : (v, u) \in E^t, (v, u) \notin E^{t+1}, \\ \{(v, u) \in E^t, (v, w) \in E^t, (u, w) \in E^t\} \neq \emptyset, \\ \{(v, u) \in E^{t+1}, (v, w) \in E^{t+1}, (u, w) \in E^{t+1}\} = \emptyset\} \end{aligned} \tag{6.4}$$

Notice that if a node is involved in more than one of the scenarios described above, the node is included in V_{change} only once. An example of the triad closure process in Figure 6.2 is when the network transits from time step t to $t + 1$ and the edge (e_{BC}) between B and C is introduced, resulting in the closure process of the triad formed by A, B and C. An example of the triad open process in Figure 6.2 is when the network transits from time step $t+1$ to $t+2$ and the edge (e_{AC}) between A and C disappears, resulting in the open process of the triad formed by A, B and C. After the Change Modeler component obtains the complete set of changed nodes, the set is used as input data by the Changed Node Embedder component. Similar to the Initial Node Embedder component, the Changed Node Embedder component leverages the metapath2vec++ algorithm to generate embedding vectors of nodes. The differences between the two components are the input and output data. Instead of processing all the nodes in the network, the Changed Node Embedder component only works on the changed nodes and produces vector representations of them. The main functionality of the Node Embedding Updater component is to generate up-to-date vector representations of all nodes at each time stamp $t + 1$ from t ($t \geq 1$). Basically, X^t is updated to X^{t+1} by adding embedding vectors of the newly-added nodes ($\{v : v \in V^{t+1}, v \notin V^t\}$), removing vector representations of the deleted nodes ($\{v : v \in V^t, v \notin V^{t+1}\}$), replacing latent vectors of the existing but changed nodes ($\{v : v \in V^t, v \in V^{t+1}, X_v^t \neq X_v^{t+1}\}$).

Usually, the objective of network embedding methods is to maximize a given network probability in terms of neighbourhoods of nodes [37, 94, 48, 109]. The change2vec technique inherits the common objective and maximizes the probability of the network $G = (V, E, T)$ at each different time stamp t as follows:

$$\underset{\theta}{\operatorname{argmax}} \sum_{v \in V^t} \sum_{tp \in T_V} \sum_{c_{tp} \in N_{tp}^t(v)} \log p(c_{tp}|v;\theta) \tag{6.5}$$

where $N_{tp}^t(v)$ represents node v's one-hop neighbourhood with the tp^{th} type of nodes at the time step t, $p(c_{tp}|v;\theta)$ denotes the conditional probability of having a context node c_{tp} given a node v and is commonly defined as a softmax function [37], that is:

$$p(c_{tp}|v;\theta) = \frac{e^{X_{c_{tp}}^t \cdot X_v^t}}{\sum_{u_{tp} \in V_{tp}^t} e^{X_{u_{tp}}^t \cdot X_v^t}} \tag{6.6}$$

where X_v^t is the v^{th} row of X^t, denoting the vector representation for node v at the time stamp t, V_{tp}^t is the node set of type tp in the network at t. For example, consider the social network in Figure 6.2, the neighbourhood of one user node A can be structurally close to other users (e.g., B & C), topics (e.g., T). The Initial Node Embedder and Changed Node Embedder components leverage the same heterogeneous negative sampling [94, 37] and stochastic gradient descent algorithms used in metapath2vec++ [37] for optimizing the derived objective function.

Overall, change2vec preserves the heterogeneous feature of a network through metapath based schemes for representation learning and open or closure triad modeling. The Initial Node Embedder and Changed Node Embedder components use metapath guided random walks as input training data to generate vector representations of multiple types of nodes. The Change Modeler component follows specific metapaths for modeling triad open and closure processes, for example, in the network in Figure 6.2, triads formed by "UUU" and "UPU" can be specified for modeling.

6.3.1 Algorithm Description

Algorithms 4 and 5 describe the main ideas of the Change Modeler component of change2vec. In Algorithm 4, we produce the set of changed nodes, V_{change}, between two consecutive time stamps, t and $t+1$. V_{change} is used to generate random walks for metapath2vec++ in the Changed Node Embedder component of change2vec. At each time stamp, all node connections are recorded in an adjacency list for the specific time step. From the adjacency lists of t and $t+1$, we are able to identify the common nodes between the two time stamps. Old nodes being deleted and new nodes being introduced result in changes in sections of the network, therefore, we are also interested in the neighbouring

Algorithm 4 Changed Nodes Generation

Input: Node adjacency lists at time step t and $t+1$: A^t, A^{t+1}
Output: Set of changed nodes from t to $t+1$: V_{change}

```
for all $Key$ in $A^t$ do
    if $Key$ not in $A^{t+1}$ then
        $V^t$ += $Key$
    else
        $V_{common}$ += $Key$
    end if
end for
for all $Key$ in $A^{t+1}$ do
    if $Key$ not in $A^t$ then
        $V^{t+1}$ += $Key$
    end if
end for
One-hop neighbours of $V^t$ at $t$: $V^t_{neighbourOld}$
for all $v$ in $V^t$ do
    $V^t_{neighbourOld}$ += $A^t[v]$
end for
One-hop neighbours of $V^{t+1}$ at $t+1$: $V^{t+1}_{neighbourNew}$
for all $v$ in $K^{t+1}$ do
    $V^{t+1}_{neighbourNew}$ += $A^{t+1}[v]$
end for
$V_{difference} = V_{common}$ - $V^t_{neighbourOld}$ - $V^{t+1}_{neighbourNew}$
$V_{triad}$ = TriadModeling($V_{difference}$, $A^t$, $A^{t+1}$)
$V_{change}$ += $V^{t+1}$
$V_{change}$ += $V^t_{neighbourOld}$
$V_{change}$ += $V^{t+1}_{neighbourNew}$
$V_{change}$ += $V_{triad}$
```

nodes of old and new nodes, forming the list $V^t_{neighbourOld}$ and $V^{t+1}_{neighbourNew}$. From the lists, we can then identify common nodes between t and $t+1$ that are not neighbours of old nodes or new nodes, $V_{difference}$, and check if they form open or closed triads between the two consecutive time stamps in Algorithm 5.

For each node v in $V_{difference}$, we initially check if its adjacency list entries between time stamps t and $t+1$ are the same. If the two entries differ, this means either existing neighbour nodes were removed or new neighbour nodes were introduced at $t+1$, hence we check for open and closed triad formations for v. To identify closed triads, we examine all the new neighbour nodes of v

Algorithm 5 Open and Closed Triad Modeling

Input: Common nodes except neighbour nodes: $V_{difference}$

Output: Set of nodes that triggered triad open or close process: V_{triad}

```
for all $v$ in $V_{difference}$ do
    if $A^t[v] \neq A^{t+1}[v]$ then
        $A_{diff} = A^{t+1}[v] \setminus A^t[v]$
        for all $Entry$ in $A_{diff}$ do
            $d_{common} = A^{t+1}[Entry] \cap A^{t+1}[v]$
            if $d_{common} \geq 1$ then
                $V_{close}$ += $v$ {Find nodes that triggered triad close process}
            end if
        end for
    Closed triads of $v$ at $t$: $t_{close}$
        for all $Entry$ in $A^t[v]$ do
            $d_{common} = A^t[Entry] \cap A^t[v]$
            if $d_{common} \geq 1$ then
                $t_{close}$ += $d$
            end if
        end for
        if $t_{close} \not\subseteq A^{t+1}[v]$ then
            $V_{open}$ += $v$ {Find nodes that triggered triad open process}
        end if
    end if
end for
$V_{triad} = V_{close} + V_{open}$
```

and for each of these nodes, we check if there are any common neighbouring nodes between itself and v, if there are common neighbouring nodes, we add v to the list of nodes that triggered the triad close process, V_{close}. For open triads, we identify all closed triads of v at time stamp t, if any of these nodes do not exist in v's entry of the $t+1$ adjacency list, then we add v to the list of nodes that caused triad open process, V_{open}. Formation of closed and open triads indicate the formation and loss of strong inter-connectivity respectively, hence we add both V_{close} and V_{open} to V_{change}.

6.3.2 Time Complexity Analysis of change2vec

The time complexity of change2vec is determined by the number of nodes in two sequential time stamps, t and $t+1$, which are denoted as n^t and n^{t+1} respectively. The first stage of finding the common keys and the relative complements iterates through both adjacency lists, resulting in an execution time function of $(n^t + n^{t+1})$. Finding all the neighbour nodes of old nodes in the worst case iterates through all nodes in time stamp t, giving us an execution time of (n^t). Similarly, finding all the neighbour nodes of new nodes in the worst case iterates through all nodes in time stamp $t+1$, resulting in an execution time of (n^{t+1}). The closed triad examination process iterates through all new nodes and their neighbours, giving us a maximum execution time function of $(n^{t+1}log(n^{t+1}))$. While the open triad examination process will in the worst case iterate through neighbours of all nodes in t, producing an execution time of $(n^t log(n^t))$. Finally, the final time complexity of change2vec is $O(n^t log(n^t) + n^{t+1}log(n^{t+1}))$.

6.4 Experimental Evaluations

We apply change2vec to Digital Bibliographic Library Project (DBLP) data sets to evaluate its performance and run time when compared to existing state-of-the-art baselines. Metapath2vec++[37] and DynamicTriad [168] were selected as they have been empirically shown to outperform other network embedding techniques in their respective fields. All experiments were performed on virtual machines with 4 vCPUs and 16GB RAM. Our source code is available for download[1].

Table 6.1: Properties of DBLP data sets

Data Set Year	Time Sequence	Authors	Venues	Papers
2010	1st	140351	108	155117
2011	2nd	147024	110	163207
2013	3rd	185987	115	212953

For our experiments, we used three sequential time stamps of DBLP data sets extracted from AMiner[2]. Similar to Dong. et al. [37], we processed each data set and have only kept entries that matched the top 20 conferences in 8 research categories[3] found on Google Scholar[4]. These research categories are used as ground truth information when we perform clustering tasks on venues

[1] https://github.com/Change2vec/change2vec.git

[2] https://aminer.org/citation

[3] 1. Computational Linguistics, 2. Computer Graphics, 3. Computer Networks & Wireless Communication, 4. Computer Vision & Pattern Recognition, 5. Computing Systems, 6. Databases & Information Systems, 7. Human Computer Interaction, and 8. Theoretical Computer Science.

[4] https://scholar.google.com/citations?view_op=top_venues&hl=en&vq=eng. Accessed on January, 2019.

(V) and authors (A). At each time step, we map an author to a particular cluster or research category if over half of the papers they publish are from conferences of that research category. The properties of each processed data set are presented in Table 6.1.

Like metapath2vec++, change2vec adopts the metapath "AVA" to guide random walks. In addition, "AAA" and "AVA" are the metapaths specified in our experiments for modeling triad open or closure processes. For metapath2vec++ and change2vec, we used the optimal parameters as proposed by the original authors [37] with the number of clusters set to 8, number of walks to 1000, length of walks to 100, and dimensions to 128. For DynamicTriad, we also set the number of dimensions to 128. Due to the nondeterministic nature of random walk based schemes in metapath2vec++ and change2vec, we repeated each evaluation 30 times for both methods and present the average and standard deviation results. DynamicTriad is a deterministic algorithm, we run it only once to obtain its results and there is no need to give a standard deviation.

We performed clustering on venues and authors separately as proxy tasks to evaluate the accuracy of learned node embedding vectors, which makes more sense than evaluating those vectors directly without applying a data mining task. More accurate vectors should generate better clustering results. The embedding vectors of author and venue nodes produced by each method were used as feature inputs by the k-means clustering algorithm. From the resulting clusters, we then evaluated performance of each method with the NMI [129] and ARI [140] scores which show how close the resulting clusters are to the author and venue ground truths. Both scores are in the value range of [0,1] with 1 indicating that the results are identical to the ground truth clusters.

6.4.1 Normalized Mutual Information (NMI)

Table 6.2 presents the venue and author clustering NMI values for each method after processing the data set with the specific time stamp. For example, change2vec obtained a high venue clustering NMI value of 0.8073 and an author clustering value of 0.7296 for the 2013 time stamp. The best result in each set of data is highlighted in bold. Since change2vec and metapath2vec++ both follow the same steps for the first time stamp, we can see that the values for them are identical (as underlined in the table). However, from 2011 onwards, we notice a higher NMI values for venue clusters of change2vec. This demonstrates that change2vec was able to achieve clustering results that is closer to the ground truth while being time efficient (as shown in Table 6.4). The time efficiency could easily be explained as we only target nodes with change, however the improvement in NMI could be due to the fact that change2vec considers the state of nodes of both time stamps. Having information and visibility of state of nodes in multiple time stamps will provide us with details on how a node's connections change over time. This provides the heuristic we use in change2vec to more accurate identify which nodes to target and which cluster they belong in. This is reflected by our results as change2vec was shown to produce more accurate clustering scores in most scenarios.

As expected, since DynamicTriad was designed for dynamic homogeneous networks and cannot differentiate between author and venue nodes as separate node types, the method consistently produced low NMI values. In addition, the relatively low standard deviation values (denoted as σ in our tables) of change2vec showed more consistent and stable outcomes over benchmark techniques.

Table 6.2: Node clustering results (NMI) of each time stamp

Method	Year	Sequence	Venue	σ (Venue)	Author	σ (Author)
DynamicTriad	2010	1st	0.1806	N/A	0.0444	N/A
Metapath2vec++	2010	1st	**0.8023**	0.0511	**0.7944**	0.0402
Change2vec	2010	1st	**0.8023**	0.0511	**0.7944**	0.0402
DynamicTriad	2011	2nd	0.3112	N/A	0.0727	N/A
Metapath2vec++	2011	2nd	0.6751	0.0521	**0.7138**	0.0333
Change2vec	2011	2nd	**0.7137**	0.0334	0.7127	0.0541
DynamicTriad	2013	3rd	0.3255	N/A	0.1366	N/A
Metapath2vec++	2013	3rd	0.7313	0.0422	**0.7372**	0.0501
Change2vec	2013	3rd	**0.8073**	0.0211	0.7296	0.0406

6.4.2 Adjusted Rand Index (ARI)

The Adjusted Rand Index [140] is another commonly adopted clustering evaluation metric. Table 6.3 highlights change2vec's effectiveness as seen in 2011 and 2013 data sets where it produced considerably higher ARI values for venue clusters and noticeable increases for author clusters. The large difference in the ARI results between change2vec and metapath2vec++ could be due to there being only a small number of venues in total, hence small changes and improvements in agreement to ground truth could result in large increases in clustering accuracy. The increase in author ARI again, could be due to the change2vec having visibility of both time stamps and identifying which nodes have changed, reducing noise. Similar to NMI, DynamicTriad was shown to produce lowest cluster similarity scores due to the method being agnostic to node types. In addition, the relative low standard deviation values in Table 6.3 demonstrate metapath2vec++ and change2vec's stable clustering performances.

Table 6.3: Node clustering results (ARI) of each time stamp

Method	Year	Sequence	Venue	σ (Venue)	Author	σ (Author)
DynamicTriad	2010	1st	0.1037	N/A	0.0314	N/A
Metapath2vec++	2010	1st	**0.6911**	0.0391	**0.7977**	0.0242
Change2vec	2010	1st	**0.6911**	0.0391	**0.7977**	0.0242
DynamicTriad	2011	2nd	0.0251	N/A	0.0521	N/A
Metapath2vec++	2011	2nd	0.4623	0.0215	0.6416	0.0404
Change2vec	2011	2nd	**0.5587**	0.0144	**0.6677**	0.0232
DynamicTriad	2013	3rd	0.2380	N/A	0.0227	N/A
Metapath2vec++	2013	3rd	0.5686	0.0423	0.7421	0.0304
Change2vec	2013	3rd	**0.8073**	0.0346	**0.7596**	0.0311

Table 6.4: Run time of each time stamp

Method	Year	Sequence	Time (mins)
DynamicTriad	2010	1st	**7**
Metapath2vec++	2010	1st	20
Change2vec	2010	1st	20
DynamicTriad	2011	2nd	**7**
Metapath2vec++	2011	2nd	20
Change2vec	2011	2nd	10
DynamicTriad	2013	3rd	**8**
Metapath2vec++	2013	3rd	25
Change2vec	2013	3rd	18

6.4.3 Results with Introduced Open Triads

To broaden the scope of structural changes in the DBLP data sets, we have attempted to find nodes that form open triads, where one or more edges of a closed triad that the nodes previously formed is lost from time stamp t to $t+1$. With the unique characteristic of academic data sets, we cannot expect disappearance of any edges between two consecutive time stamps since a publication does not disappear over time, therefore we consider an edge is lost between two authors at time stamp $t+1$ when the two authors have not co-published any paper from time stamp t to $t+1$. We then examine whether the lost edge caused the transformation of a closed triad to an open triad between the two consecutive time stamps. Around 2% (11482 author-to-author edges) and 3% (24188 author-to-author edges) of the edges were removed from the

2011 and 2013 data sets respectively to introduce the described open triads. Tables 6.5 and 6.6 represent the NMI and ARI values for the DBLP data sets with deliberately introduced open triads. From these tables, we can see that there is an average increase in clustering results across all time stamps and node types with change2vec. Our approach was also shown to slightly outperform metapath2vec++ in the 2013 time stamp where it was not able to in the previous experiments. This indicates that with the extra consideration of nodes that lose the closed triad relationships, change2vec is able to get more accurate clustering evaluations.

Table 6.5: Node clustering results (NMI) of each time stamp with open triads

Method	Year	Sequence	Venue	σ (Venue)	Author	σ (Author)
DynamicTriad	2010	1st	0.1806	N/A	0.0444	N/A
Metapath2vec++	2010	1st	**0.8023**	0.0216	**0.7944**	0.0342
Change2vec	2010	1st	**0.8023**	0.0216	**0.7944**	0.0342
DynamicTriad	2011	2nd	0.3112	N/A	0.0727	N/A
Metapath2vec++	2011	2nd	0.6751	0.0145	**0.7181**	0.0236
Change2vec	2011	2nd	**0.7294**	0.0271	0.6993	0.0175
DynamicTriad	2013	3rd	0.3255	N/A	0.1366	N/A
Metapath2vec++	2013	3rd	0.7313	0.0311	0.7372	0.0213
Change2vec	2013	3rd	**0.8154**	0.0166	**0.7382**	0.0321

Table 6.6: Node clustering results (ARI) of each time stamp with open triads

Method	Year	Sequence	Venue	σ (Venue)	Author	σ (Author)
DynamicTriad	2010	1st	0.1037	N/A	0.0314	N/A
Metapath2vec++	2010	1st	**0.6911**	0.0211	**0.7977**	0.0139
Change2vec	2010	1st	**0.6911**	0.0211	**0.7977**	0.0139
DynamicTriad	2011	2nd	0.1251	N/A	0.0521	N/A
Metapath2vec++	2011	2nd	0.4623	0.0217	0.6416	0.0222
Change2vec	2011	2nd	**0.5624**	0.0197	**0.6695**	0.0244
DynamicTriad	2013	3rd	0.2380	N/A	0.0227	N/A
Metapath2vec++	2013	3rd	0.5686	0.0155	0.7421	0.0213
Change2vec	2013	3rd	**0.8181**	0.0119	**0.7612**	0.0344

6.4.4 Run Time

Table 6.4 and 6.7 present the time taken for each method to process the data set with the specific time stamp. Overall, DynamicTriad was considerably faster than change2vec and metapath2vec++. From the 2nd time sequence onwards, we observe a consistent decrease in run time for change2vec as compared to metapath2vec++. Comparing Table 6.7 with Table 6.4, we notice an increase

Table 6.7: Run time of each time stamp with open triads

Method	Year	Sequence	Time (mins)
DynamicTriad	2010	1st	**7**
Metapath2vec++	2010	1st	20
Change2vec	2010	1st	20
DynamicTriad	2011	2nd	**7**
Metapath2vec++	2011	2nd	20
Change2vec	2011	2nd	12
DynamicTriad	2013	3rd	**8**
Metapath2vec++	2013	3rd	25
Change2vec	2013	3rd	22

in run time for change2vec due to extra validation and introduction of more changed nodes caused by the deliberately introduced open triads.

6.4.5 Results with Deleted Edges

Table 6.8 presents the number of edges in the original DBLP data sets. To stress test the three methods' embedding ability with noisy data, we perform the same clustering experiments in the data sets with randomly deleted edges. As shown in Table 6.8, there are two types of edges in the data sets: author-to-author edge and author-to-venue edge. To perform the stress test, we randomly deleted 10%, 20% or 30% of the total edges in each data set (type of the edge being deleted is also determined randomly) and apply the three methods to the remaining data set.

Tables 6.9 and 6.10 present NMI and ARI values on different generated data sets respectively. Notice that we repeated the experiments 30 times of random edge deletion for each individual value in the two tables. Overall, clustering accuracy of all three methods dropped as compared to experimenting with the original data sets. However, change2vec outperformed the other two baselines in most cases, showing its ability of learning more accurate embedding vectors in networks with noise. The relatively low standard deviation values of metapath2vec++ and change2vec demonstrate their stable performance when dealing with noisy data.

Table 6.8: Number of edges in DBLP data sets

Data Set Year	Time Sequence	Author-to-Author	Author-to-Venue
2010	1st	537357	424617
2011	2nd	574086	450521
2013	3rd	806263	608624

Table 6.9: Node clustering results (NMI) of each time stamp with deleted edges

Method	Year	Edges Deleted	Author	σ (Author)	Venue	σ (Venue)
DynamicTriad	2010	10%	0.0880	0.0261	0.2263	0.0688
Metapath2vec++	2010	10%	**0.4606**	0.1528	**0.5279**	0.1674
Change2vec	2010	10%	**0.4606**	0.1528	**0.5279**	0.1674
DynamicTriad	2010	20%	0.0514	0.0215	0.1495	0.0122
Metapath2vec++	2010	20%	**0.5472**	0.1341	**0.5026**	0.0969
Change2vec	2010	20%	**0.5472**	0.1341	**0.5026**	0.0969
DynamicTriad	2010	30%	0.0384	0.0120	0.0835	0.0383
Metapath2vec++	2010	30%	**0.4294**	0.1090	**0.5594**	0.0941
Change2vec	2010	30%	**0.4294**	0.1090	**0.5594**	0.0941
DynamicTriad	2011	10%	0.0649	0.0304	0.2041	0.0843
Metapath2vec++	2011	10%	0.6133	0.1536	0.5335	0.0454
Change2vec	2011	10%	**0.6613**	0.1028	**0.7303**	0.0595
DynamicTriad	2011	20%	0.0745	0.0221	0.1420	0.0795
Metapath2vec++	2011	20%	0.5290	0.1485	0.5793	0.2349
Change2vec	2011	20%	**0.6738**	0.0446	**0.6972**	0.1069
DynamicTriad	2011	30%	0.0408	0.0123	0.0756	0.0151
Metapath2vec++	2011	30%	0.5815	0.0916	0.5107	0.1680
Change2vec	2011	30%	**0.6944**	0.0303	**0.6995**	0.0176
DynamicTriad	2013	10%	0.0379	0.0210	0.1878	0.0606
Metapath2vec++	2013	10%	0.4716	0.1341	0.4397	0.0314
Change2vec	2013	10%	**0.6892**	0.1251	**0.5919**	0.2053
DynamicTriad	2013	20%	0.0462	0.0219	0.1120	0.0363
Metapath2vec++	2013	20%	0.4888	0.1396	0.5466	0.1608
Change2vec	2013	20%	**0.6637**	0.1179	**0.6189**	0.2532
DynamicTriad	2013	30%	0.0272	0.0055	0.0581	0.0065
Metapath2vec++	2013	30%	0.5449	0.1588	0.4149	0.1014
Change2vec	2013	30%	**0.6114**	0.1411	**0.5110**	0.1426

6.5 Conclusions and Future Work

In this chapter, we propose a framework, change2vec, to capture changes and heterogeneous features in a dynamic heterogeneous network. The framework captures structural changes between two consecutive network snapshots by modeling the triad open or closure processes and describes such changes with up-to-date embedding vectors of nodes. change2vec was experimentally shown to outperform metapath2vec++ in terms of embedding accuracy and run time efficiency on dynamic heterogeneous networks. Although being less efficient than DynamicTriad, change2vec consumed up to three times more time in order to gain up to 78% more accurate embedding vectors as compared to DynamicTriad.

In this chapter, we again find a positive answer to our main research question, which is heterogeneous information can improve the results of network analysis. We reach our conclusion by looking at one specific network analy-

Table 6.10: Node clustering results (ARI) of each time stamp with deleted edges

Method	Year	Edges Deleted	Author	σ (Author)	Venue	σ (Venue)
DynamicTriad	2010	10%	0.0376	0.0143	0.0913	0.0633
Metapath2vec++	2010	10%	**0.5073**	0.0654	**0.6283**	0.0279
Change2vec	2010	10%	**0.5073**	0.0654	**0.6283**	0.0279
DynamicTriad	2010	20%	0.0234	0.0117	0.0748	0.0420
Metapath2vec++	2010	20%	**0.5630**	0.2275	**0.5740**	0.1164
Change2vec	2010	20%	**0.5630**	0.2275	**0.5740**	0.1164
DynamicTriad	2010	30%	0.0199	0.0059	0.0235	0.0172
Metapath2vec++	2010	30%	**0.4573**	0.0653	**0.5468**	0.2162
Change2vec	2010	30%	**0.4573**	0.0653	**0.5468**	0.2162
DynamicTriad	2011	10%	0.0293	0.0156	0.1469	0.0304
Metapath2vec++	2011	10%	**0.5516**	0.1875	0.4733	0.2089
Change2vec	2011	10%	0.5381	0.2249	**0.5254**	0.1642
DynamicTriad	2011	20%	0.0308	0.0040	0.1148	0.0265
Metapath2vec++	2011	20%	0.4723	0.2567	**0.5626**	0.1056
Change2vec	2011	20%	**0.5918**	0.0727	0.5274	0.1738
DynamicTriad	2011	30%	0.0155	0.0130	0.0579	0.0122
Metapath2vec++	2011	30%	**0.6528**	0.1668	0.4637	0.1896
Change2vec	2011	30%	0.6323	0.0538	**0.5471**	0.0013
DynamicTriad	2013	10%	0.0336	0.0063	0.0722	0.0396
Metapath2vec++	2013	10%	0.5162	0.0236	0.5817	0.1572
Change2vec	2013	10%	**0.6803**	0.1439	**0.6126**	0.1277
DynamicTriad	2013	20%	0.0182	0.0088	0.0733	0.0420
Metapath2vec++	2013	20%	0.6002	0.0568	0.6044	0.1218
Change2vec	2013	20%	**0.6289**	0.1223	**0.6237**	0.1166
DynamicTriad	2013	30%	0.0092	0.0071	0.0455	0.0392
Metapath2vec++	2013	30%	0.4512	0.0548	0.4331	0.0989
Change2vec	2013	30%	**0.6433**	0.1330	**0.6606**	0.0560

sis task, network embedding. We compare the performance of our developed algorithm against DynamicTriad, a state-of-the-art homogeneous network embedding method. Unlike DynamicTriad, which considers different types of objects in a network as one type, our algorithm considers the network heterogeneity. Our experiment results show that as compared to DynamicTriad, our algorithm generates node embedding vectors that are closer to the ground truth.

Future work includes generalizing change2vec to diverse types of dynamic heterogeneous networks, automatically learning useful metapaths and using learned embedding vectors in other heterogeneous network mining tasks. This includes node similarity search and classification.

7 Conclusions

In most related literature, the definition of a social network is a homogeneous graph, in which each node represents an object, and each edge denotes a relationship between two objects. However, the single type of relationship in traditional social network definitions is not enough to capture real world associations. Very often, a variety of types of relationships exists among objects, with different semantics associated with each relationship. Examples of heterogeneous types of relationships among objects are social collaborations [68], citations [44], and annotations [46].

In addition, multiple types of objects involved in social networks are also missing in the traditional definition of a social network. With a comprehensive literature survey on one specific aspect of social network analysis, the notion of network heterogeneity comes into focus, and as a result this thesis focused on developing new knowledge discovery and data mining techniques for analyzing heterogeneous social networks.

Chapter 4 and Chapter 5 were developed for heterogeneous networks with correlation schemas, while Chapter 6 was developed for any type of heterogeneous network. Heterogeneous networks with correlation schemas were targeted in Chapters 4 and 5 because the schema represents an important network structure and defines a constraint relationship between source type and attribute type objects.

A series of novel methods are presented for knowledge discovery in heterogeneous social networks. The methods proposed in this thesis have been applied to a wide range of applications including community discovery, ranking, information retrieval, network embedding, and change modeling. In this final chapter we summarize our findings and results presented in the thesis, and discuss future directions for our research.

7.1 Contributions

Overall, we developed novel algorithms and applied them to a wide range of applications including community discovery, ranking and network embedding. To answer the main research question raised in Section 1.2: ”Does heterogeneous information improve the results of network analysis?”, we have embedded heterogeneous information in our developed approaches and shown an improvement in network analysis tasks over their counterpart homogeneous heuristics. In Chapter 4, OHC was experimentally shown to detect communities with denser internal connectivity than two homogeneous community detection techniques. In Chapter 5, ComRank was experimentally shown to identify top rank objects with better quality than PageRank. In Chapter 6, change2vec was experimentally shown to generate more accurate embedding vectors of objects than a state-of-the-art homogeneous network embedding method. The consistent performance shows that the results of network analysis can definitely be improved by embedding heterogeneous information, answering our research question.

We next summarize our contributions to the areas of social network analysis and heterogeneous network analysis by chapter:

Chapter 3

- We conducted a comprehensive literature review on social network analysis with a focus on top-k nodes identification. We then outlined a series of applicable domains for this research topic and highlighted some promising future directions.

- One of the conclusions drawn from the literature survey is that the vast majority of existing techniques for social network analysis were proposed for homogeneous social networks. Typical techniques includes community discovery, ranking and network embedding. The survey lays a foundation of our research on heterogeneous network analysis.

Chapter 4

- We presented a novel algorithm for discovering overlapping heterogeneous communities and demonstrated how by integrating the TPR scoring function into community evaluation, we were able to detect communities with stronger internal connectivity and looser external connectivity than benchmark techniques.
- We empirically showed the merits of our methodology and analyzed the performance of our proposed method qualitatively in a case study.

Chapter 5

- We presented a novel approach for ranking objects of different types in heterogeneous networks by integrating community memberships as a co-ranking mechanism. ComRank was shown to produce higher quality top-k rankings of objects than benchmark techniques.
- We showed the merits of our proposed approach by comparing its performance in identifying top-rank objects against three state-of-the-art baselines.

Chapter 6

- We presented a novel model for network embedding and change modeling in dynamic heterogeneous networks.
- We showed that by modelling concurrent time stamps and analysing the structural changes of triads, change2vec outperforms two state-of-the-art methods in terms of time efficiency and clustering accuracy through NMI and ARI scores.

7.2 Limitations

This thesis has presented algorithms and approaches that contribute to performing mining and analysis tasks in heterogeneous networks. Despite the contributions, there are some limitations to the proposed algorithms and approaches.

The proposed methods in Chapters 4 and 5 were developed to analyse heterogeneous correlation networks. The methods would need to be revisited for other kinds of networks, such as heterogeneous networks with other schema types such as star and bipartite schemas. Additionally, as stated in our research scope (Section 1.4), our heuristics are not designed to evaluate composite networks [142, 141] and complex networks with non-trivial topological features and patterns of connection between its elements that are neither purely regular nor purely random [73, 118].

The time complexity for our algorithms was shown to be at least $0(n^2)$. While they performed well against the benchmark and proposed data sets, we would not expect them to analyse larger heterogeneous networks (over 1 TiB of data) effectively due to time and computational complexities. We aim to address this in the future.

7.3 Future Directions

In this thesis, we have proposed heterogeneous network analysis and mining techniques. However, this field of research is still relatively new and there are more challenges and ideas that we have yet to examine, providing us with some potential research directions that we can explore in the future. We will first discuss the future work for each of the individual topics in the chapters followed by plans for enhancing our heuristics to tackle existing heterogeneous community mining challenges of dynamic data streams, larger and more complex data structures and by applying our heuristics to a larger variety of heterogeneous networks.

Uncovering Overlapping Heterogeneous Communities

We proposed the OHC algorithm to facilitate the discovery of overlapping heterogeneous communities in Chapter 4. Our future work in this part includes improving OHC's scalability and efficiency by limiting its recursion depth, automatically distinguishing between Source Type and Attribute Type objects, adapting OHC to find an evolution of communities in dynamic heterogeneous networks. Currently a limitation for OHC is that it can only evaluate heterogeneous networks with correlation schemas, investigation into expanding OHC to perform evaluations on other heterogeneous schemas such as star or bipartite schema would allow us to evaluate a wider range of heterogeneous networks.

Community Based Ranking of Objects

In Chapter 5 we presented the idea of ranking objects of different types in heterogeneous networks by taking the object community memberships into consideration. Our proposed approach, ComRank, ranks communities based on the authors, papers and the relationships between them. However, this does not account for attributes such as the ranking of the conferences in which the papers were published or if the topic of papers in the community is similar which could impact the integrity of the community. For future work, we want to evolve ComRank so it can perform evaluations in a multi-dimensional way. Along with the core data sets, we can introduce new ground truth data sets such as conference ranking information, and paper index terms to perform richer and more robust analysis that includes real world factors. Similar to OHC, a limitation for ComRank is that it only evaluates networks with a correlation schema, and we would like to expand this so ComRank can perform rankings on heterogeneous networks with other schema types.

Network Embedding and Change Modeling in Dynamic Heterogeneous Networks

Change2vec, a framework for network embedding and change modeling in dynamic heterogeneous networks was presented in Chapter 6. In the future we intend to implement change2vec into real data streams because currently we only perform analysis on snapshots (time stamps) of dynamic heterogeneous networks. To this end, we need to solve challenges such as handling of incomplete or missing data and performance optimization. Furthermore, we also want to adapt change2vec to automatically learn useful metapaths and enable it to evaluate composite and complex network structures.

General Directions

As most of our existing experiments have been performed on tested and proven academic data sets or data sets proposed by authors with related work, a direction that can be explored is application of our heuristics on more untested heterogeneous networks in different domains, such as large corporate collaboration data or biological data. By developing and integrating new clustering, recommendation and classification techniques upon these data sets, there is a large room of potential to find undiscovered results and findings.

The study of data streams is also a topic that this thesis has barely scratched the surface of, and it is our intention to enhance our heuristics to more effectively process and analyse dynamic streams of data and monitor paradigm shifts of top objects identified in streams while providing an explanation for what causes the shifts.

The scope of our research considers only one relationship type between objects, for example, in Chapter 4, the relationship considered between authors is always ‘co-authorship’ and the relationship considered between an author and a paper is always ‘authorship’; in Chapter 5, the relationship considered between Digg users is always ‘co-comment’ and the relationship considered between a Digg user and a story is always ‘user-comment-to’; in Chapter 6, the relationship considered between an author and a venue is always ‘published at’. Since, relationships between objects can have different meanings, an interesting challenge would be to find the best way to extend the developed algorithms to take different types of relationships between objects into account.

Finally, we intend to enhance the proposed methods or develop new approaches to handle heterogeneous networks with more complex network construction, richer semantic information, or bigger networked data. Last but not least, we are interested in shaping and demonstrating more potential applications of heterogeneous network mining and analysis.

References

[1] *Digg Homepage*, accessed on April 8, 2018. http://digg.com.

[2] *Facebook Statistics*, accessed on September 10, 2017. http://www.facebook.com/press/info.php?statistics.

[3] Bijaya Adhikari, Xinfeng Xu, Naren Ramakrishnan, and B. Aditya Prakash. Epideep: Exploiting embeddings for epidemic forecasting. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 577–586, 2019.

[4] Charu Aggarwal and Karthik Subbian. Evolutionary network analysis: A survey. *ACM Computing Surveys*, 47(1):10:1–10:36, 2014.

[5] Charu C. Aggarwal, Shuyang Lin, and Philip S. Yu. On influential node discovery in dynamic social networks. In *Proceedings of the 2012 SIAM International Conference on Data Mining*, pages 636–647, 2012.

[6] Eugene Agichtein, Carlos Castillo, Debora Donato, Aristides Gionis, and Gilad Mishne. Finding high-quality content in social media. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pages 183–194, 2008.

[7] Tim Finin Anupam Joshi Akshay Java, Pranam Kolari and Tim Oates. Feeds That Matter: A Study of Bloglines Subscriptions. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM 2007)*. Computer Science and Electrical Engineering, University of Maryland, Baltimore County, March 2007.

[8] Sinan Aral and Dylan Walker. Identifying influential and susceptible members of social networks. *Science*, 337(6092):337–341, 2012.

[9] Çigdem Aslay, Nicola Barbieri, Francesco Bonchi, and Ricardo A Baeza-Yates. Online topic-aware influence maximization queries. In *Proceedings*

of the 17th International Conference on Extending Database Technology, pages 295–306, 2014.

[10] Eytan Bakshy, Jake M. Hofman, Winter A. Mason, and Duncan J. Watts. Everyone's an influencer: Quantifying influence on twitter. In *Proceedings of the 4th International Conference on Web Search and Data Mining*, pages 65–74, 2011.

[11] Nicola Barbieri, Francesco Bonchi, and Giuseppe Manco. Topic-aware social influence propagation models. *Knowledge and Information Systems*, 37(3):555–584, 2013.

[12] Alain Barrat, Marc Barthélemy, and Alessandro Vespignani. Modeling the evolution of weighted networks. *Physical Review*, 70(6):066149, 2004.

[13] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[14] Phillip Bonacich. Power and centrality: A family of measures. *American Journal of Sociology*, 92(5):1170–1182, 1987.

[15] Stephen P. Borgatti. Centrality and network flow. *Social Networks*, 27(1):55–71, 2005.

[16] Stephen P Borgatti. Identifying sets of key players in a social network. *Computational & Mathematical Organization Theory*, 12(1):21–34, 2006.

[17] Mohamed Bouguessa, Benoît Dumoulin, and Shengrui Wang. Identifying authoritative actors in question-answering forums: The case of yahoo! answers. In *Proceedings of the 14th International Conference on Knowledge Discovery and Data Mining*, pages 866–874, 2008.

[18] Arastoo Bozorgi, Hassan Haghighi, Mohammad Sadegh Zahedi, and Mojtaba Rezvani. Incim: A community-based algorithm for influence maximization problem under the linear threshold model. *Information Processing Management*, 52(6):1188 – 1199, 2016.

[19] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the 7th International Conference on World Wide Web*, pages 107–117, 1998.

[20] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117, 1998.

[21] Ceren Budak, Divyakant Agrawal, and Amr El Abbadi. Limiting the spread of misinformation in social networks. In *Proceedings of the 20th International Conference on World Wide Web*, pages 665–674, 2011.

[22] H. Cai, V. W. Zheng, and K. C. Chang. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Transactions on Knowledge and Data Engineering*, 30(9):1616–1637, 2018.

[23] R. Cappelletti and N. Sastry. Iarank: Ranking users on twitter in near real-time, based on their information amplification potential. In *Proceedings of the 2012 International Conference on Social Informatics*, pages 70–77, 2012.

[24] Peter J Carrington. Crime and social network analysis. *The SAGE handbook of social network analysis*, pages 236–255, 2011.

[25] Meeyoung Cha, Hamed Haddadi, Fabricio Benevenuto, and Krishna Gummadi. Measuring user influence in twitter: The million follower fallacy. In *Proceedings of the 4th International AAAI Conference on Web and Social Media*, 2010.

[26] Shiyu Chang, Wei Han, Jiliang Tang, Guo-Jun Qi, Charu C. Aggarwal, and Thomas S. Huang. Heterogeneous network embedding via deep architectures. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 119–128, 2015.

[27] Duanbing Chen, Linyuan Lü, Ming-Sheng Shang, Yi-Cheng Zhang, and Tao Zhou. Identifying influential nodes in complex networks. *Physica A: Statistical Mechanics and its Applications*, 391(4):1777 – 1787, 2012.

[28] Junxiang Chen, Wei Dai, Yizhou Sun, and Jennifer Dy. Clustering and ranking in heterogeneous information networks via Gamma-Poisson model. In *Proceedings of the 2015 SIAM International Conference on Data Mining*, pages 424–432, 2015.

[29] Shubo Chen and Kejing He. Influence maximization on signed social networks with integrated PageRank. In *Proceedings of the 2015 IEEE*

International Conference on Smart City/SocialCom/SustainCom, pages 289–292, 2015.

[30] W. Chen, Y. Yuan, and L. Zhang. Scalable influence maximization in social networks under the linear threshold model. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, pages 88–97, 2010.

[31] Wei Chen, Chi Wang, and Yajun Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th International Conference on Knowledge Discovery and Data Mining*, pages 1029–1038, 2010.

[32] Wei Chen, Yajun Wang, and Siyu Yang. Efficient influence maximization in social networks. In *Proceedings of the 15th International Conference on Knowledge Discovery and Data Mining*, pages 199–208, 2009.

[33] Arthur P. Dempster. Upper and lower probabilities induced by a multivalued mapping. *The Annals of Mathematical Statistics*, 38(2):325–339, 1967.

[34] Zhou Ding. *Mining Social Documents And Networks.* PhD thesis, The Pennsylvania State University, 2008.

[35] Pedro Domingos. Mining social networks for viral marketing. *IEEE Intelligent Systems*, 20(1):80–82, 2005.

[36] Pedro Domingos and Matt Richardson. Mining the network value of customers. In *Proceedings of the 7th International Conference on Knowledge Discovery and Data Mining*, pages 57–66, 2001.

[37] Yuxiao Dong, Nitesh V. Chawla, and Ananthram Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd International Conference on Knowledge Discovery and Data Mining*, pages 135–144, 2017.

[38] Georgios Drakopoulos, Andreas Kanavos, and Athanasios Tsakalidis. Evaluating twitter influence ranking with system theory. In *Proceedings of the 12th International Conference on Web Information Systems and Technologies*, pages 113–120, 2016.

[39] Ayman Farahat, Geoff Nunberg, and Francine Chen. Augeas: Authoritativeness grading, estimation, and sorting. In *Proceedings of the 11th*

International Conference on Information and Knowledge Management, pages 194–202, 2002.

[40] D. Fisher, M. Smith, and H. T. Welser. You are who you talk to: Detecting roles in usenet newsgroups. In *Proceedings of the 39th Annual Hawaii International Conference on System Sciences*, volume 3, pages 59b–59b, 2006.

[41] Gary William Flake, Steve Lawrence, and C. Lee Giles. Efficient identification of web communities. In *Proceedings of the 2000 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 150–160, 2000.

[42] Linton C Freeman. Centrality in social networks conceptual clarification. *Social networks*, 1(3):215–239, 1978.

[43] Rumi Ghosh and Kristina Lerman. Predicting influential users in online social networks. *CoRR*, abs/1005.4882, 2010.

[44] C. Lee Giles, Kurt D. Bollacker, and Steve Lawrence. Citeseer: an automatic citation indexing system. In *Proceedings of the 3rd ACM Conference on Digital Libraries*, pages 89–98, 1998.

[45] Jacob Goldenberg, Barak Libai, and Eitan Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing Letters*, 12(3):211–223, 2001.

[46] Scott A. Golder and Bernardo A. Huberman. Usage patterns of collaborative tagging systems. *Journal of Information Science*, 32(2):198–208, 2006.

[47] Mark Granovetter. Threshold models of collective behavior. *American Journal of Sociology*, 83(6):1420–1443, 1987.

[48] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd International Conference on Knowledge Discovery and Data Mining*, pages 855–864, 2016.

[49] Daniel Gruhl, R. Guha, David Liben-Nowell, and Andrew Tomkins. Information diffusion through blogspace. In *Proceedings of the 13th International Conference on World Wide Web*, pages 491–501, 2004.

[50] Adrien Guille, Hakim Hacid, Cecile Favre, and Djamel A. Zighed. Information diffusion in online social networks: A survey. *SIGMOD Rec.*, 42(2):17–28, 2013.

[51] Jinwen Guo, Shengliang Xu, Shenghua Bao, and Yong Yu. Tapping on the potential of q&a community by recommending answer providers. In *Proceedings of the 17th Conference on Information and Knowledge Management*, pages 921–930, 2008.

[52] Ido Guy, Michal Jacovi, Elad Shahar, Noga Meshulam, Vladimir Soroka, and Stephen Farrell. Harvesting with sonar: the value of aggregating social network information. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1017–1026, 2008.

[53] William L. Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. *CoRR*, abs/1709.05584, 2017.

[54] Meng Han, Mingyuan Yan, Zhipeng Cai, Yingshu Li, Xingquan Cai, and Jiguo Yu. Influence maximization by probing partial communities in dynamic online social networks. *Transactions on Emerging Telecommunications Technologies*, 28(4):e3054, 2017.

[55] Steve Harenberg, Gonzalo Bello, L. Gjeltema, Stephen Ranshous, Jitendra Harlalka, Ramona Seay, Kanchana Padmanabhan, and Nagiza Samatova. Community detection in large-scale networks: a survey and empirical evaluation. *Wiley Interdisciplinary Reviews: Computational Statistics*, 6(6):426–439, 2014.

[56] Taher H. Haveliwala. Topic-sensitive pagerank. In *Proceedings of the 11th International Conference on World Wide Web*, pages 517–526, 2002.

[57] Julia Heidemann, Mathias Klier, and Florian Probst. Identifying key users in online social networks: a pagerank based approach. In *International Conference of Information Systems*, 2010.

[58] Manel Hmimida and Rushed Kanawati. Community detection in multiplex networks: a seed-centric approach. *Networks and Heterogeneous Media*, 10(1):71–85, 2015.

[59] H. Huang, J. Tang, L. Liu, J. Luo, and X. Fu. Triadic closure pattern analysis and prediction in social networks. *IEEE Transactions on Knowledge and Data Engineering*, 27(12):3374–3389, 2015.

[60] Lidong Huang, Wenxian Wang, Xiaozhou Chen, and Junhua Chen. A heuristic method of identifying key microbloggers. In *Proceedings of the 14th International Conference on Smart Homes and Health Telematics*, pages 417–426, 2016.

[61] Muhammad U. Ilyas and Hayder Radha. Identifying influential nodes in online social networks using principal component centrality. In *Proceedings of the 2011 International Conference on Communications*, pages 1–5, 2011.

[62] Ronald L. Iman and W. J. Conover. A distribution-free approach to inducing rank correlation among input variables. *Communications in Statistics - Simulation and Computation*, 11(3):311–334, 1982.

[63] Akshay Java, Pranam Kolari, Tim Finin, and Tim Oates. Modeling the spread of influence on the blogosphere. In *Proceedings of the 15th International World Wide Web Conference*, pages 22–26, 2006.

[64] Ming Ji, Jiawei Han, and Marina Danilevsky. Ranking-based classification of heterogeneous information networks. In *Proceedings of the 17th International Conference on Knowledge Discovery and Data Mining*, pages 1298–1306, 2011.

[65] Ming Ji, Yizhou Sun, Marina Danilevsky, Jiawei Han, and Jing Gao. Graph regularized transductive classification on heterogeneous information networks. In *Proceedings of the 2010 Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 570–586, 2010.

[66] Pawel Jurczyk and Eugene Agichtein. Discovering authorities in question answer communities by using link analysis. In *Proceedings of the 16th Conference on Information and Knowledge Management*, pages 919–922, 2007.

[67] Pawel Jurczyk and Eugene Agichtein. Hits on question answer portals: exploration of link analysis for author ranking. In *Proceedings of the 30th Annual International Conference on Research and Development in Information Retrieval*, pages 845–846, 2007.

[68] Henry Kautz, Bart Selman, and Mehul Shah. Referral web: Combining social networks and collaborative filtering. *Communications of the ACM*, 40(3):63–65, 1997.

[69] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the 9th International Conference on Knowledge Discovery and Data Mining*, pages 137–146, 2003.

[70] David Kempe, Jon Kleinberg, and Éva Tardos. Influential nodes in a diffusion model for social networks. In *Proceedings of the 32nd International Colloquium on Automata, Languages and Programming*, pages 1127–1138, 2005.

[71] E. S. Kim and S. S. Han. An analytical way to find influencers on social networks and validate their effects in disseminating social games. In *Proceedings of the 2009 International Conference on Advances in Social Network Analysis and Mining*, pages 41–46, 2009.

[72] Hyoungshick Kim and Eiko Yoneki. Influential neighbours selection for information diffusion in online social networks. In *Proceedings of the 21st International Conference on Computer Communications and Networks*, pages 1–7, 2012.

[73] Jongkwang Kim and Thomas Wilhelm. What is a complex graph? *Physica A: Statistical Mechanics and its Applications*, 387(11):2637 – 2652, 2008.

[74] Masahiro Kimura and Kazumi Saito. Tractable models for information diffusion in social networks. In *Proceedings of the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 259–271, 2006.

[75] Masahiro Kimura, Kazumi Saito, and Ryohei Nakano. Extracting influential nodes for information diffusion on a social network. In *Proceedings of the 2007 Association for the Advancement of Artificial Intelligence*, volume 7, pages 1371–1376, 2007.

[76] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 668–677, 1998.

[77] Jon M Kleinberg. Challenges in mining social network data: processes, privacy, and paradoxes. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 4–5, 2007.

[78] Xiangnan Kong, Bokai Cao, and Philip S Yu. Multi-label classification by mining label and instance correlations from heterogeneous information networks. In *Proceedings of the 19th International Conference on Knowledge Discovery and Data Mining*, pages 614–622, 2013.

[79] Xiangnan Kong, Bokai Cao, and Philip S. Yu. Multi-label classification by mining label and instance correlations from heterogeneous information networks. In *Proceedings of the 19th International Conference on Knowledge Discovery and Data Mining*, pages 614–622, 2013.

[80] Gueorgi Kossinets and Duncan J Watts. Empirical analysis of an evolving social network. *Science*, 311(5757):88–90, 2006.

[81] Zhana Kuncheva and Giovanni Montana. Community detection in multiplex networks using locally adaptive random walks. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 1308–1315, 2015.

[82] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th International Conference on World Wide Web*, pages 591–600, 2010.

[83] Andrea Lancichinetti and Santo Fortunato. Community detection algorithms: A comparative analysis. *Phys. Rev. E*, 80:056117, 2009.

[84] Theodoros Lappas, Kun Liu, and Evimaria Terzi. A survey of algorithms and systems for expert location in social networks. *Social Network Data Analytics*, pages 215–241, 2011.

[85] Theodoros Lappas, Evimaria Terzi, Dimitrios Gunopulos, and Heikki Mannila. Finding effectors in social networks. In *Proceedings of the 16th International Conference on Knowledge Discovery and Data Mining*, pages 1059–1068, 2010.

[86] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Wikipedia vote network, 2010. data retrieved from http://snap.stanford.edu/data/wiki-Vote.html.

[87] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th International Conference on Knowledge Discovery and Data Mining*, pages 420–429, 2007.

[88] Jure Leskovec, Kevin J. Lang, and Michael Mahoney. Empirical comparison of algorithms for network community detection. In *Proceedings of the 19th International Conference on World Wide Web*, pages 631–640, 2010.

[89] Hongbo Li, Jianpei Zhang, Jing Yang, Jinbo Bai, and Yuming Zhao. Efficient star nodes discovery algorithm in social networks based on acquaintance immunization. In *Proceedings of the 2nd International Conference on Image, Vision and Computing*, pages 937–940, 2017.

[90] Lu Liu, Jie Tang, Jiawei Han, Meng Jiang, and Shiqiang Yang. Mining topic-level influence in heterogeneous networks. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, pages 199–208, 2010.

[91] Qi Liu, Biao Xiang, Enhong Chen, Hui Xiong, Fangshuang Tang, and Jeffrey Xu Yu. Influence maximization over large-scale social networks: A bounded linear approach. In *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management*, pages 171–180, 2014.

[92] Xiaoming Liu, Johan Bollen, Michael L. Nelson, and Herbert Van de Sompel. Co-authorship networks in the digital library research community. *Information Processing & Management*, 41(6):1462 – 1480, 2005.

[93] Bo Long, Zhongfei (Mark) Zhang, and Philip S. Yu. Co-clustering by block value decomposition. In *Proceedings of the 11th ACM International Conference on Knowledge Discovery and Data Mining*, pages 635–640, 2005.

[94] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26:3111–3119, 2013.

[95] Joel C Miller, Gregory Rae, Fred Schaefer, Lesley A Ward, Thomas LoFaro, and Ayman Farahat. Modifications of Kleinberg's hits algorithm using matrix exponentiation and web log records. In *Proceedings of the 24th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 444–445, 2001.

[96] Alan Mislove, Massimiliano Marcon, Krishna P Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and analysis of online social networks. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 29–42. ACM, 2007.

[97] Katarzyna Musiał and Krzysztof Juszczyszyn. Properties of bridge nodes in social networks. *Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems*, pages 357–364, 2009.

[98] Ramasuri Narayanam and Yadati Narahari. A Shapley value-based approach to discover influential nodes in social networks. *IEEE Transactions on Automation Science and Engineering*, 8(1):130–147, 2011.

[99] Mario A. Nascimento, Jorg Sander, and Jeffrey Pound. Analysis of sigmod's co-authorship graph. *SIGMOD Rec.*, 32(3):8–10, 2003.

[100] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69, 2004.

[101] Michaek Kwok-Po Ng, Xutao Li, and Yunming Ye. MultiRank: Coranking for objects and relations in multi-relational data. In *Proceedings of the 17th International Conference on Knowledge Discovery and Data Mining*, pages 1217–1225, 2011.

[102] Nam P. Nguyen, Guanhua Yan, My T. Thai, and Stephan Eidenbenz. Containment of misinformation spread in online social networks. In *Proceedings of the 4th Annual ACM Web Science Conference*, pages 213–222, 2012.

[103] Feiping Nie, Wei Zhu, and Xuelong Li. Unsupervised large graph embedding. In *Proceedings of the 2017 Association for the Artificial Intelligence*, 2017.

[104] Zaiqing Nie, Yuanzhi Zhang, Ji-Rong Wen, and Wei-Ying Ma. Object-level ranking: Bringing order to web objects. In *Proceedings of the 14th International Conference on World Wide Web*, pages 567–574, 2005.

[105] Tore Opsahl, Filip Agneessens, and John Skvoretz. Node centrality in weighted networks: Generalizing degree and shortest paths. *Social networks*, 32(3):245–251, 2010.

[106] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, 1999.

[107] Aditya Pal and Scott Counts. Identifying topical authorities in microblogs. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*, pages 45–54, 2011.

[108] Aditya Pal and Joseph A. Konstan. Expert identification in community question answering: Exploring question selection bias. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, pages 1505–1508, 2010.

[109] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th International Conference on Knowledge Discovery and Data Mining*, pages 701–710, 2014.

[110] Nicola Perra and Santo Fortunato. Spectral centrality measures in complex networks. *Phys. Rev. E*, 78:036107, 2008.

[111] Florian Probst, Laura Grosswiele, and Regina Pfleger. Who will lead and who will follow: Identifying influential users in online social networks. *Business & Information Systems Engineering*, 5(3):179–193, 2013.

[112] Dragomir R. Radev, Pradeep Muthukrishnan, Vahed Qazvinian, and Amjad Abu-Jbara. The ACL anthology network corpus. *Language Resources and Evaluation*, 47(4):919–944, 2009.

[113] Matthew Richardson and Pedro Domingos. Mining knowledge-sharing sites for viral marketing. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 61–70, 2002.

[114] Fabián Riquelme and Pablo González-Cantergiani. Measuring user influence on twitter: A survey. *Information Processing & Management*, 52(5):949 – 975, 2016.

[115] Daniel M. Romero, Wojciech Galuba, Sitaram Asur, and Bernardo A. Huberman. Influence and passivity in social media. In *Proceedings of the 2011 European Conference in Machine Learning and Knowledge Discovery*, pages 18–33, 2011.

[116] Glenn Shafer. *A mathematical theory of evidence*, volume 1. Princeton University Press, 1976.

[117] L. S. Shapley. *A value for N-person games in contributions to the theory of games*, volume 1. Princeton University Press, 1950.

[118] C. Shi, Y. Li, J. Zhang, Y. Sun, and P. S. Yu. A survey of heterogeneous information network analysis. *IEEE Transactions on Knowledge and Data Engineering*, 29(1):17–37, 2017.

[119] Chuan Shi, Ran Wang, Yitong Li, Philip S. Yu, and Bin Wu. Ranking-based clustering on general heterogeneous information networks by network projection. In *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management*, pages 699–708, 2014.

[120] Chuan Shi, Zhiqiang Zhang, Ping Luo, Philip S Yu, Yading Yue, and Bin Wu. Semantic path based personalized recommendation on weighted heterogeneous information networks. In *Proceedings of the 24th International Conference on Information and Knowledge Management*, pages 453–462, 2015.

[121] G. Song, Y. Li, X. Chen, X. He, and J. Tang. Influential node tracking on dynamic social network: An interchange greedy approach. *IEEE Transactions on Knowledge and Data Engineering*, 29(2):359–372, 2017.

[122] Karthik Subbian, Charu Aggarwal, and Jaideep Srivastava. Content-centric flow mining for influence analysis in social streams. In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, pages 841–846, 2013.

[123] Karthik Subbian, Charu Aggarwal, and Jaideep Srivastava. Mining influencers using information flows in social streams. *ACM Transaction Knowledge Discovery Data*, 10(3):26:1–26:28, 2016.

[124] Karthik Subbian, Charu C. Aggarwal, and Jaideep Srivastava. Querying and tracking influencers in social streams. In *Proceedings of the 9th ACM International Conference on Web Search and Data Mining*, pages 493–502, 2016.

[125] Karthik Subbian, Dhruv Sharma, Zhen Wen, and Jaideep Srivastava. Finding influencers in networks using social capital. *Social Network Analysis and Mining*, 4(1):1–13, 2014.

[126] Jimeng Sun and Jie Tang. *A survey of models and algorithms for social influence analysis*, pages 177–214. 2011.

[127] Yizhou Sun and Jiawei Han. Mining heterogeneous information networks: Principles and methodologies. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 3(2):1–159, 2012.

[128] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S Yu, and Tianyi Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. In *Proceedings of the 2011 VLDB Endowment*, pages 992–1003, 2011.

[129] Yizhou Sun, Jiawei Han, Peixiang Zhao, Zhijun Yin, Hong Cheng, and Tianyi Wu. Rankclus: Integrating clustering with ranking for heterogeneous information network analysis. In *Proceedings of the 12th International Conference on Extending Database Technology*, pages 565–576, 2009.

[130] Yizhou Sun, Jie Tang, Jiawei Han, Manish Gupta, and Bo Zhao. Community evolution detection in dynamic heterogeneous information networks. In *Proceedings of the Eighth Workshop on Mining and Learning with Graphs*, pages 137–146. ACM, 2010.

[131] Yizhou Sun, Yintao Yu, and Jiawei Han. Ranking-based clustering of heterogeneous information networks with star network schema. In *Proceedings of the 15th International Conference on Knowledge Discovery and Data Mining*, pages 797–806, 2009.

[132] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1067–1077, 2015.

[133] Jie Tang, Jimeng Sun, Chi Wang, and Zi Yang. Social influence analysis in large-scale networks. In *Proceedings of the 15th International Conference on Knowledge Discovery and Data Mining*, pages 807–816, 2009.

[134] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: Extraction and mining of academic social networks. In *The 14th International Conference on Knowledge Discovery and Data Mining*, pages 990–998, 2008.

[135] Lei Tang, Xufei Wang, and Huan Liu. Community detection via heterogeneous interaction analysis. *Data mining and knowledge discovery*, 25(1):1–33, 2012.

[136] Guangmo Tong, Weili Wu, Shaojie Tang, and Ding-Zhu Du. Adaptive influence maximization in dynamic social networks. *IEEE/ACM Transaction Network*, 25(1):112–125, 2017.

[137] Chi-Yao Tseng and Ming-Syan Chen. Significant node identification in social networks. In *Proceedings of the 2011 Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 459–470, 2012.

[138] Daniel Tunkelang. A twitter analog to pagerank. In *The Noisy Channel*, 2009.

[139] M. Vadoodparast and F. Taghiyareh. A multi-agent solution to maximizing product adoption in dynamic social networks. In *Proceedings of the 2015 International Symposium on Artificial Intelligence and Signal Processing*, pages 71–78, 2015.

[140] Silke Wagner and Dorothea Wagner. *Comparing clusterings: an overview.* Universität Karlsruhe, Fakultät für Informatik Karlsruhe, 2007.

[141] Chenguang Wang, Yangqiu Song, Ahmed El-Kishky, Dan Roth, Ming Zhang, and Jiawei Han. Incorporating world knowledge to document clustering via heterogeneous information networks. In *Proceedings of the 21st ACM International Conference on Knowledge Discovery and Data Mining*, pages 1215–1224, 2015.

[142] Chi Wang, Jiawei Han, Qi Li, Xiang Li, Wen-Pin Lin, and Heng Ji. Learning hierarchical relationships among partially ordered objects with heterogeneous attributes and links. In *Proceedings of the 2012 SIAM International Conference on Data Mining*, pages 516–527, 2012.

[143] Jinghua Wang, Jianyi Liu, and Cong Wang. Keyword extraction based on pagerank. In *Proceedings of the 11th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 857–864, 2007.

[144] Xiao Wang, Peng Cui, Jing Wang, Jian Pei, Wenwu Zhu, and Shiqiang Yang. Community preserving network embedding. In *Proceedings of the 2017 AAAI Conference on Artificial Intelligence*, 2017.

[145] Yanhao Wang, Qi Fan, Yuchen Li, and Kian-Lee Tan. Real-time influence maximization on dynamic social streams. In *Proceedings of the 2017 VLDB Endowment*, volume 10, pages 805–816, 2017.

[146] Yu Wang, Gao Cong, Guojie Song, and Kunqing Xie. Community-based greedy algorithm for mining top-k influential nodes in mobile social networks. In *Proceedings of the 16th International Conference on Knowledge Discovery and Data Mining*, pages 1039–1048, 2010.

[147] Stanley Wasserman and Katherine Faust. *Social network analysis: Methods and applications*, volume 8. Cambridge University Press, 1994.

[148] Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440, 1998.

[149] Daijun Wei, Xinyang Deng, Xiaoge Zhang, Yong Deng, and Sankaran Mahadevan. Identifying influential nodes in weighted networks based on evidence theory. *Physica A: Statistical Mechanics and its Applications*, 392(10):2564 – 2575, 2013.

[150] Jianshu Weng, Ee-Peng Lim, Jing Jiang, and Qi He. Twitterrank: Finding topic-sensitive influential twitterers. In *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining*, pages 261–270, 2010.

[151] Scott White and Padhraic Smyth. Algorithms for estimating relative importance in networks. In *Proceedings of the 9th ACM International Conference on Knowledge Discovery and Data Mining*, pages 266–275, 2003.

[152] Ian H. Witten, Eibe Frank, Mark A. Hall, and Christopher J. Pal. *Data Mining: Practical Machine Learning Tools and Techniques.* Morgan Kaufmann, 2005.

[153] Harris Wu, Mohammad Zubair, and Kurt Maly. Harvesting social knowledge from folksonomies. In *Proceedings of the 7th Conference on Hypertext and Hypermedia*, pages 111–114, 2006.

[154] J. Xie, B. K. Szymanski, and X. Liu. Slpa: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process. In *Proceedings of the 2011 IEEE International Conference on Data Mining*, pages 344–349, 2011.

[155] Jierui Xie, Stephen Kelley, and Boleslaw K. Szymanski. Overlapping community detection in networks: The state-of-the-art and comparative study. *ACM Comput. Surv.*, 45(4):43:1–43:35, 2013.

[156] Rui Yan, Mirella Lapata, and Xiaoming Li. Tweet recommendation with graph co-ranking. In *The 50th Annual Meeting of the Association for Computational Linguistics.*, pages 516–525, 2012.

[157] Jaewon Yang and Jure Leskovec. Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems*, 42(1):181–213, 2013.

[158] Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norick, and Jiawei Han. Personalized entity recommendation: A heterogeneous information network approach. In *Proceedings of the 7th International Conference on Web Search and Data Mining*, pages 283–292, 2014.

[159] Xiao Yu, Xiang Ren, Yizhou Sun, Bradley Sturt, Urvashi Khandelwal, Quanquan Gu, Brandon Norick, and Jiawei Han. Recommendation in heterogeneous information networks with implicit user feedback. In *Proceedings of the 7th Conference on Recommender Systems*, pages 347–350, 2013.

[160] Osmar R. Zaïane, Jiyang Chen, and Randy Goebel. Mining research communities in bibliographical data. In *Advances in Web Mining and Web Usage Analysis*, pages 59–76, 2009.

[161] Jun Zhang, Mark S. Ackerman, and Lada Adamic. Expertise networks in online communities: Structure and algorithms. In *Proceedings of the 16th International Conference on World Wide Web*, pages 221–230, 2007.

[162] Erheng Zhong, Wei Fan, Yin Zhu, and Qiang Yang. Modeling the dynamics of composite social networks. In *Proceedings of the 2013 International Conference on Knowledge Discovery and Data Mining*, pages 937–945, 2013.

[163] Chang Zhou, Yuqiong Liu, Xiaofei Liu, Zhongyi Liu, and Jun Gao. Scalable graph embedding for asymmetric proximity. In *Proceedings of the 2017 AAAI Conference on Artificial Intelligence*, 2017.

[164] Dengyong Zhou, Olivier Bousquet, Thomas N Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems*, pages 321–328, 2004.

[165] Ding Zhou, Sergey A. Orshanskiy, Hongyuan Zha, and C. Lee Giles. Co-ranking authors and documents in a heterogeneous network. In *Proceedings of the 7th IEEE International Conference on Data Mining*, pages 739–744, 2007.

[166] Jingyu Zhou, Yunlong Zhang, and Jia Cheng. Preference-based mining of top-k influential nodes in social networks. *Future Generation Computer Systems*, 31:40 – 47, 2014.

[167] Lekui Zhou, Yang Yang, Xiang Ren, Fei Wu, and Yueting Zhuang. Dynamic network embedding by modeling triadic closure process. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, 2018.

[168] Lekui Zhou, Yang Yang, Xiang Ren, Fei Wu, and Yueting Zhuang. Dynamic network embedding by modeling triadic closure process. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, 2018.

[169] Yang Zhou, Hong Cheng, and Jeffrey Xu Yu. Graph clustering based on structural/attribute similarities. In *Proceedings of the VLDB Endowment*, number 1, pages 718–729, 2009.

[170] Yang Zhou, Hong Cheng, and Jeffrey Xu Yu. Clustering large attributed graphs: An efficient incremental approach. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, pages 689–698, 2010.

[171] Feng Zhu, Guanfeng Liu, Yan Wang, An Liu, Zhixu Li, Pengpeng Zhao, and Lei Li. A context-aware trust-oriented influencers finding in online social networks. In *Proceedings of the 2015 IEEE International Conference on Web Services*, pages 456–463, 2015.

[172] Linhong Zhu, Dong Guo, Junming Yin, Greg Ver Steeg, and Aram Galstyan. Scalable temporal latent space inference for link prediction in dynamic social networks. *IEEE Transactions on Knowledge and Data Engineering*, 28(10):2765–2777, 2016.

[173] Dong Zhuang, Benyu Zhang, Heng Zhang, Jeremy Tantrum, Teresa Mah, Hua-Jun Zeng, Zheng Chen, and Jian Wang. Identifying influential persons in a social network, January 22 2013. US Patent 8,359,276.

[174] Honglei Zhuang, Yihan Sun, Jie Tang, Jialin Zhang, and Xiaoming Sun. Influence maximization in dynamic social networks. In *Proceedings of the 13th International Conference on Data Mining*, pages 1313–1318, 2013.

THE UNIVERSITY OF AUCKLAND
Te Whare Wānanga o Tāmaki Makaurau
NEW ZEALAND

School of Graduate Studies
AskAuckland Central
Alfred Nathan House
The University of Auckland
Tel: +64 9 373 7599 ext 81321
Email: postgradinfo@auckland.ac.nz

Declaration to Accompany a Doctoral Thesis on Submission

Section 1. Candidate's details

Candidate name:	RANRAN BIAN	ID number:	1706953
Thesis title:	Heterogeneous Networks Mining and Analysis		
Address in full:	32B Centorian Drive, Windsor Park Auckland 0632		

Section 2. Candidate's declaration

I do solemnly and sincerely declare that:

1. I am making a submission to the University of Auckland to be examined for the degree of

   Doctor of Philosophy

2. I am currently registered as a candidate for the above degree at the University of Auckland.

3. I have attached to this declaration one copy of a thesis/work and the necessary documentation for examination. The electronic copy of the thesis included with the submission and the temporary-bound copy are identical.

4. The thesis/work in substantially its present form has not been submitted or accepted previously for the award of a degree or diploma in this or any other tertiary institution, and is not being submitted for a degree or diploma in any other tertiary institution or for another degree or diploma at this institution.

   Candidates must state below if the thesis incorporates any material that has been submitted or accepted for any other degree or diploma, and note the extent and nature of that re-use.

THE UNIVERSITY OF
AUCKLAND
Te Whare Wānanga o Tāmaki Makaurau
NEW ZEALAND

School of Graduate Studies
AskAuckland Central
Alfred Nathan House
The University of Auckland
Tel: +64 9 373 7599 ext 81321
Email: postgradinfo@auckland.ac.nz

Declaration to Accompany a Doctoral Thesis on Submission

Section 2. Candidate's declaration (cont'd)

5. Regarding the thesis/work submitted for examination:

(a) The research in the chapters of the thesis/work (the development of methodologies, experimentation, interpretation of results and preparation of the manuscript) were my own:

☑ in full ☐ in part *[please tick one]*

If in part, outline the parts to which others have contributed:

....

....

....

Where the thesis contains any jointly authored research work (published or unpublished) or includes research reported in any co-authored works (published or unpublished) I have included the required Co-Authorship Form/s within the thesis.

(b) The work includes a creative practice component ☐ yes ☑ no

If a creative practice component is included, this component is in the form of a:

☐ performance ☐ exhibition ☐ other *[please specify]*

....

....

Where any creative practice component was co-produced I have included the required Co-Production Form/s within the thesis.

6. Written permission has been obtained for any third-party copyright material reproduced in the thesis that represents a "substantial part" of the other work.

7. The information provided in Sections 1 and 2 of this form is complete and accurate, no relevant information has been withheld that I am aware of, and I have complied with all of the University's requirements in the statutes and regulations associated with my degree.

I make this solemn declaration conscientiously believing the same to be true, and by virtue of the Oaths and Declarations Act 1957.

Signed ....

DECLARED at Auckland

On 13th day of June month 2019 year

Before me Stanley Joseph Saw

Signed S J Saw
(Justice of the Peace or ~~Solicitor of High Court~~)

S.J. Saw, JP
#8101
AUCKLAND
Justice of the Peace for New Zealand

THE UNIVERSITY OF AUCKLAND
Te Whare Wānanga o Tāmaki Makaurau
NEW ZEALAND

School of Graduate Studies
AskAuckland Central
Alfred Nathan House
The University of Auckland
Tel: +64 9 373 7599 ext 81321
Email: postgradinfo@auckland.ac.nz

Declaration to Accompany a Doctoral Thesis on Submission

Section 3. Additional material

I have attached additional material in the form of:

□ CD/DVD ROM   □ other *[please specify]* ....

Provide a description of the content of this additional material and explain how it is related to the thesis:

....

....

....

This additional material does not contain any direct appeal to the examiners.

....
Signed by Candidate   Date

Where additional material is included in this submission and listed above, the Main Supervisor and the Head of Department must, by signing below, confirm that the description and explanation given above are complete and correct and that the additional material does not contain any direct appeal to the examiners.

....
Signed by Main Supervisor   Date   Signed by HoD/HoS   Date

Name: ....   Name: ....

THE UNIVERSITY OF AUCKLAND
Te Whare Wānanga o Tāmaki Makaurau
NEW ZEALAND

School of Graduate Studies
AskAuckland Central
Alfred Nathan House
The University of Auckland
Tel: +64 9 373 7599 ext 81321
Email: postgradinfo@auckland.ac.nz

Co-Authorship Form

This form is to accompany the submission of any PhD that contains published or unpublished co-authored work. **Please include one copy of this form for each co-authored work.** Completed forms should be included in all copies of your thesis submitted for examination and library deposit (including digital deposit), following your thesis Acknowledgements. Co-authored works may be included in a thesis if the candidate has written all or the majority of the text and had their contribution confirmed by all co-authors as not less than 65%.

Please indicate the chapter/section/pages of this thesis that are extracted from a co-authored work and give the title and publication details or details of submission of the co-authored work.

Included in: Chapter 6 | Paper Title: Network Embedding and Change Modeling in Dynamic Heterogeneous Networks | Published in: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval

Nature of contribution by PhD candidate	Idea formulation, code development, experimental evaluation, and writing of paper
Extent of contribution by PhD candidate (%)	80

CO-AUTHORS

Name	Nature of Contribution
Yun Sing Koh	Supervision, guidance, and proof-reading of paper
Gillian Dobbie	Supervision, guidance, and proof-reading of paper
Anna Divoli	Supervision, guidance, and proof-reading of paper

Certification by Co-Authors

The undersigned hereby certify that:

- the above statement correctly reflects the nature and extent of the PhD candidate's contribution to this work, and the nature of the contribution of each of the co-authors; and
- that the candidate wrote all or the majority of the text.

Name	Signature	Date
Gillian Dobbie	G. C Dobbie	11/6/19
Yun Sing Koh	YS Koh	12/6/2019
Anna Divoli		23/6/19

Last updated: 28 November 2017

THE UNIVERSITY OF
AUCKLAND
Te Whare Wānanga o Tāmaki Makaurau
NEW ZEALAND

School of Graduate Studies
AskAuckland Central
Alfred Nathan House
The University of Auckland
Tel: +64 9 373 7599 ext 81321
Email: postgradinfo@auckland.ac.nz

Co-Authorship Form

This form is to accompany the submission of any PhD that contains published or unpublished co-authored work. **Please include one copy of this form for each co-authored work.** Completed forms should be included in all copies of your thesis submitted for examination and library deposit (including digital deposit), following your thesis Acknowledgements. Co-authored works may be included in a thesis if the candidate has written all or the majority of the text and had their contribution confirmed by all co-authors as not less than 65%.

Please indicate the chapter/section/pages of this thesis that are extracted from a co-authored work and give the title and publication details or details of submission of the co-authored work.

Included in: Chapter 3 | Paper Title: Identifying Top-k Nodes in Social Networks: A Survey | Published in: ACM Computing Surveys (CSUR) Journal

Nature of contribution by PhD candidate	Literature survey, idea formulation, and writing of paper
Extent of contribution by PhD candidate (%)	80

CO-AUTHORS

Name	Nature of Contribution
Yun Sing Koh	Supervision, guidance, and proof-reading of paper
Gillian Dobbie	Supervision, guidance, and proof-reading of paper
Anna Divoli	Supervision, guidance, and proof-reading of paper

Certification by Co-Authors

The undersigned hereby certify that:

- the above statement correctly reflects the nature and extent of the PhD candidate's contribution to this work, and the nature of the contribution of each of the co-authors; and
- that the candidate wrote all or the majority of the text.

Name	Signature	Date
Yun Sing Koh	YS Koh	12/6/2019
Gillian Dobbie	G. C Dobbie	11/6/19
Anna Divoli		23/6/19

THE UNIVERSITY OF
AUCKLAND
Te Whare Wānanga o Tāmaki Makaurau
NEW ZEALAND

School of Graduate Studies
AskAuckland Central
Alfred Nathan House
The University of Auckland
Tel: +64 9 373 7599 ext 81321
Email: postgradinfo@auckland.ac.nz

Co-Authorship Form

This form is to accompany the submission of any PhD that contains published or unpublished co-authored work. **Please include one copy of this form for each co-authored work.** Completed forms should be included in all copies of your thesis submitted for examination and library deposit (including digital deposit), following your thesis Acknowledgements. Co-authored works may be included in a thesis if the candidate has written all or the majority of the text and had their contribution confirmed by all co-authors as not less than 65%.

Please indicate the chapter/section/pages of this thesis that are extracted from a co-authored work and give the title and publication details or details of submission of the co-authored work.

Included in: Chapter 4 | Paper Title: OHC: Uncovering Overlapping Heterogeneous Communities | Published in: Proceedings of the 31st Australasian Joint Conference on Artificial Intelligence

Nature of contribution by PhD candidate	Idea formulation, code development, experimental evaluation, and writing of paper
Extent of contribution by PhD candidate (%)	80

CO-AUTHORS

Name	Nature of Contribution
Yun Sing Koh	Supervision, guidance, and proof-reading of paper
Gillian Dobbie	Supervision, guidance, and proof-reading of paper
Anna Divoli	Supervision, guidance, and proof-reading of paper

Certification by Co-Authors

The undersigned hereby certify that:

- the above statement correctly reflects the nature and extent of the PhD candidate's contribution to this work, and the nature of the contribution of each of the co-authors; and
- that the candidate wrote all or the majority of the text.

Name	Signature	Date
Yun Sing Koh	YS Koh	12/6/2019
Gillian Dobbie	G. C Dobbie	11/6/19
Anna Divoli		23/6/19