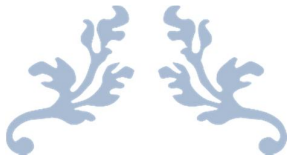
Name: **Feiyi (Aaron) Tang**

Student ID： 4505032



*Thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy*

*Title*

# Link-Prediction and its Application in Online Social Networks

College of Engineering and Science

Victoria University

Supervisory Team:

Supervisor: Professor Yanchun Zhang – Centre for Applied Informatics, Victoria University

Associate supervisor: Professor Hua Wang – Centre for Applied Informatics, Victoria University

# Abstract

Alongside the continuous development of Internet technologies, traditional social networks are running online to provide more services so as to unite the community. In the meantime, conventional web-based information systems are trying hard to utilise social networking elements to develop a virtual community so as to increase their popularity. The combination of these two domains has become what people knew as the 'online social networks'. There is much to do to reveal the knowledge behind the screen as massive amounts of user-generated data is created every second. Many people from different disciplines are using their tools and techniques to analyse and build knowledge to try understanding the evolution of it. Link Prediction, with the essence of calculating similarities of two nodes, is one of the most common techniques to analyse an online social network. It is worth mentioning that while using Link Prediction to explain online social network, we consider it as a graph with nodes and edges connecting one another where nodes represent individuals and edges represent the relations between them. Link Prediction can be utilised in many ways in this domain, where one of the most common ways is predicting links/edges that may appear in the future of an evolving network where links/edges represent connections. The meaning of these connections vary under different circumstance, such as an academia social network where they may represent co-author relationships among researchers. Therefore, one of the most common applications of Link Prediction in an online social network will be the recommendation system. Many works have been done to analyse social-oriented online networks and many turns into applications with great success such as Facebook and Twitter. However, this thesis concentrates on investigating a particular type of online social network where there is still a large gap waiting to be filled - the online academia social network. The objective of this thesis is to provide a more sensible way for people to understand the evolution of this network and develop models and algorithms that solving issues in regards to the needs of the users in this system of finding valuable research partners. Further the object is to building up an environment for future researchers to share knowledge and to carry on the work as a community. To be specific, this thesis contains four main chapters, and they are connected in some ways to develop solutions for the issues coming out during the research processes. Firstly, this thesis proposed an innovated way to analysis and understand online social networks as a whole by calculating user-

relationship strength base on the data from an academic social network website. And then the thesis will further puts forward a new link prediction algorithm that leveraging time factor as a weight on edges in co-authorship network to support the friend recommendation system of the source website. Secondly, inspiring by the flourishing Social Tagging System (STS) that is applied on famous online social websites such as Facebook, Twitter, Last.fm and various blog sites. A hybrid friend recommendation algorithm base on 'user–item - tag graph ' and 'user personal interests' has been proposed which dramatically improves the accuracy of friend recommendation. As mentioned above, there is a gap between current academic online social network and other social-oriented networks regarding application development which leads to lack of usable data for this particular group. In other words, human-labeled data in our target domain is scarce, but labelled data exist in a related field is more than sufficient. To obtain an efficient ranking model for our target domain, the fifth chapter of this thesis propose a cross-domain ranking adaption model to seek a solution to solve the data scarcity and sparsity problem in academia online social network. Finally, to reemphasize that the final goal of the thesis is to build up a community for future work. Therefore the second last chapter of the thesis introduces a live running academia online social networks websites that utilised two innovative functions to help researchers to find and establish possible connections with future research partners. More importantly, gathering people with same interest to continue developing this community in the future.

# Doctor of Philosophy Declaration

I, Feiyi Tang, declare that the PhD thesis entitled 'Link Prediction and Its Application in Online Social Network' is no more than 100,000 words in length including quotes and exclusive of tables, figures, appendices, bibliography, references and footnotes. This thesis contains no material that has been submitted previously, in whole or in part, for the award of any other academic degree or diploma. Except where otherwise indicated, this thesis is my own work.

X 

Feiyi Tang
April 2017

# Acknowledgement

First and foremost, I would like to express my sincere gratitude to my supervisor Professor Yanchun Zhang. He has always been a source of knowledge, an inspiration for new ideas, and an exemplary for career development. His attentive mentoring is the primary support that makes this doctorate pursue possible.

Also, I would like to thanks my associate supervisor Professor Hua Wang and the members of CAI and college, Professor Yuan Miao, Professor Jing He, Jiahua Du, Luyao Teng and Jingyuan He. I would like to thank Professor Hua Wang, Professor Yuan Miao and Professor Jing He for their advice on general research issues that expedites my progress. I would like to thank Luyao Teng, Jiahua Du and Jingyuan He for their insightful suggestions and constructive comments on thesis writing.

I acknowledge the enormous support from the research team in China, Professor Jianguo Li, Associate Professor Jieming Chen, Doctor Chengzhou Fu and Doctor Shiping Huang. I would like to thank A/P Jieming Chen for her initial enlightenment and constant encouragement on chasing the research dream to me. I would like to thank Professor Jianguo Li for his sharp critique and endorsement in development my technique skills. I am also thankful to Doctor Chengzhou Fu and Doctor Shiping Huang for their effort and help in developing our studies as co-authors. I thank all the other people around me in the college and our research group, for our fruitful collaborations, thought-provoking discussions and enduring friendships.

Also I am grateful to my parents, for their unconditional love, support and encouragement also for their financial assistance that funds the research in this thesis. Last but not least, my sincere gratetude goes to my partner Luyao Teng, who is always be there by my side and encourage me to fight towards my goals.

# Preface

Papers are corresponding to the work reported in this thesis has been published[1].

These articles are listed below:

2016 – Chapter 6 & Chapter 1

[1] Feiyi Tang, Jia Zhu, Chaobo He, Chengzhou Fu, Jing He, and Yong Tang. SCHOLAT: An Innovative Academic Information Service Platform. The 27th Australasian Database Conference (ADC2016), Sydney, Australia, 28-29 Sept 2016 pp.453-456

[2] F. Tang, J. Zhu, Y. Cao, S. Ma, Y. Chen, J. He, C. Huang, G. Zhao and Y. Tang. PARecommender: A Pattern-Based System for Route Recommendation. Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16), New York，USA，July 9–15, 2016, 4272-4273

2016 – Chapter 4

[3] JieMin Chen, Feiyi Tang, Jing Xiao, JianGuo Li, Jing He, Yong Tang(*). CogTime RMF: regularised matrix factorization with drifting cognition degree for collaborative filtering. Cluster Computing, June 2016, Volume 19, Issue 2, pp 821-835

2015

[4] Luyao Teng, Xi Yang, Feiyi Tang, Shaohua Teng, Wei Zhang, Xiufen Fu. 3-Dimension Evaluation Method for Stock Investment Based on 2-Tuple Linguistic. Lecture Notes in Computer Science, Vol. 8944

2014 –Chapter 5 & Chapter 1

[5] Feiyi, Tang, Jing He, Yong Tang, Zewu Peng, Luyao Teng. Sparse Ranking Model Adaptation for Cross-Domain Learning to Rank, Journal of Internet Technology, Vol. 15 No. 6, PP. 949-962, 11 2014

2014 –Chapter 3 & Chapter 1

[6] Shiping Huang, Yong Tang, Feiyi Tang, Jianguo Li. Link Prediction Based on Time-varied Weight in Co-authorship Network. Proceedings of the 2014 IEEE 18th International Conference on Computer Supported Cooperative Work in Design (CSCWD), May 21-23, 2014 National Tsing Hua University, Hsinchu, Taiwan,706-709

[1] Paper listed are partially contributed to the Chapters indicated

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1

## Introduction

*Chapter 1 introduces this thesis as a whole, starting from presenting the background knowledge about what is a social network and how it is evolved. Then I will demonstrate how link prediction problem is related to online social network problems meanwhile explains my understanding towards what is the essence of online social network systems/sites and how to utilise link prediction methods to improve their service such as recommendation service. After a statement describing the significance of the study, I will briefly summarise the research outcome of the thesis as contributions. And in the last section of this chapter, the thesis outline is provided. (It is worth mentioning that in this thesis, online social network system and online social networking sites can be used interchangeably. Also when we consider the online social network as a graph or system, the word 'graph' and 'network' and 'system' implies the same thing and they can be used interchangeably as well.)*

## 1.1 Social Network Evolution

The word 'Social Network' was first used by J.A. Barnes in the Class and Committees in a Norwegian Island Parish in 1954 to explain human relations [136, 141]. Social network appears as a social structure consisting of many different network nodes, and each 'node' stands for a particular individual or an organisation. Generally speaking, a social network is a map of all the nodes and connections marked up as shown in Figure 1.1. Every node represents a unique existent; they can be either a person or a group. In this case, they are representing authors of publications. Many connections/edges are linking the nodes together; these links can be our relations with our family, friends as well as colleagues and so on. In this case, links are representing co-authorship relationship. In short, a social network is a social structure made of nodes that are tied by one or more specific types of interdependency, just like a map of all of the relevant ties between the nodes are being studied [136]. From studying this map/graph and its dynamic evolution processes, we can usually find out some valuable information that can help us solve practical problems in our real world [37, 83].

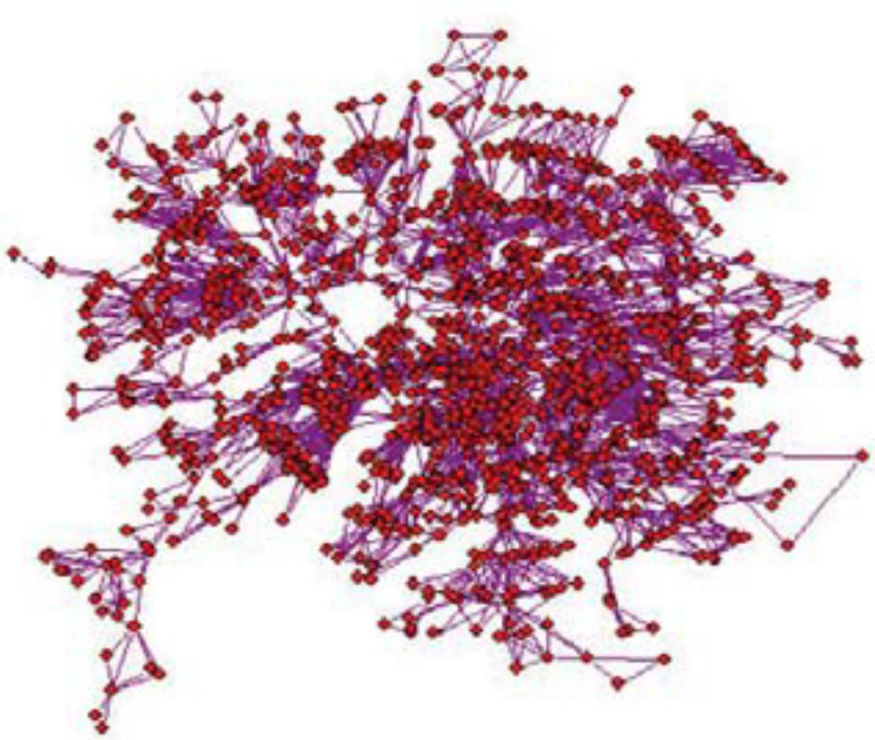


Figure 1.1 Visualisation of a part of Co-authorship Network of international conference on CSCWD

In comparison to the old-time Social Network when connections were usually built via face-to-face communications, nowadays people tend to use virtual networking application to create

relationships known as the ‘Online Social Networking System’(OSNS). The development of web-based technology removes the geographical limitation constraints of a conventional social network. Many popular online social network systems are thriving over the Internet such as Facebook, Twitter, QQ, LinkedIn, WeChat, SCHOLAT[1]. There is no denying that OSNS is in an era of high prosperity.

Social networks are very dynamic objects [2] since new edges/links and vertices/nodes are adding to the graph over the time. Moreover, the infrastructure of the modern online social network nowadays can support a rich diversity of data analytic applications such as text analysis, image analysis and sensor application, etc. Link prediction takes a really important role in social network analysis. And the link prediction techniques has been wildly adopted in many application domains [5, 13, 14, 68, 114, 117]. A significant interest emerged in methods that consider social network as a form of a graph in recent years. [144]. Therefore to make predictions is to compare the similarity between two nodes in the graph utilising topological and semantic measures [64, 72, 89, 94, 134]. These methods are commonly known as ‘Link-Prediction Methods’.

Link-prediction methods can be utilised in many application domains[89] such as 1) analyzing historical navigation data to generate tools for more efficient navigation recommendation[137]; 2) overcoming the data sparsity problem in recommendation systems[135]; 3) accelerating establishment of some mutually beneficial connections between academic professionals in academia social networks[138] and so on.

In the remainder of Chapter 1, I will briefly describe the context of how solving link-prediction problems is related to various online social network problems. From the application perspective, the link prediction methods have a great impact on the current recommendation system in terms of improving its accuracy. Together, I will elaborate the challenges as well as opportunities emerged for link-prediction application in the context of online social networking systems that drives the motivation of this study. Last, the contributions of this thesis are summarised, and the outline of the thesis is presented.

## 1.2 Link-Prediction in Online Social Networking Systems

Many systems in the real world can be described as networks [22], such as social, biological, geographical and information system. As mentioned, nodes in the network map

denote individuals, or entities and links denote relations or interactions between nodes. These systems are also known as 'complex network' and studying their evolving mechanism can benefit a lot in different disciplines, such as community detection in Sociology or protein interaction in Bioinformatics. The study of the complex network has therefore become a thorny problem of many branches of science. This thesis focuses on the *Online Social Network* as the target complex network type to study.

First, I will introduce what so-called Online Social Networking System/Sites are, then we can further explore what are the link-prediction problems that associated with it are. Thanks to the development of network technology, on the one hand, most information systems are now web-based and run through the Internet. On the other hand, traditional social networks nowadays utilise various information systems to provide its original services online and therefore thrive over the Internet. From my view, *Online Social Networks* can be regarded as the combination of '*Social Network*' and '*Information System (Network)*'. So it has the attributes and features of both complex networks. This means there are problems requiring study in both domains to meet the need of providing related services. Great efforts have been made to understand the evolution of this new network from different branches of science domain [13, 168]. In this section, I will introduce how we can transform various online social network problems into link-prediction problem and solve them by utilising knowledge and methods from various domains. To fully understand the connection between online social networking systems and link-predication problem, I will demonstrate the link-prediction problems in *social networks* and *information systems network* separately.

### *1.2.1 Link Prediction in Social Network*

Human interactions are the primary element that defines the traditional social network. When speaking of 'social networks' in the past, people usually associate it with the classical study of sociology that often requires tireless endeavour to measure and collect interactions between people manually. What makes it even harder is that social networks are very dynamic objects since new people (vertices) and new relationships (edges) are added to the community (graph) over time that makes it sometimes impossible for a researcher to get the real big picture of the communities [19].

Along with the development of Internet and web-based technologies we are now in the era of information. As mentioned, this technology advancement eliminates constraints of the geographical limitation. By utilising various information systems, many different types of social networks, from popular social-oriented sites such as Facebook to a specific career-oriented site like Linkedin are florising over the Internet. When the internet enabled devices such as PC, smartphones and tablets (i.e. iPad) became more affordable and convenient to use, social networks are becoming even more popular online. So speaking of social networks today, we no longer just consider it to be related to sociology but many other science domains, especially computer science. Nonetheless, 'human interaction' is still the unique factor that marks the boundary from a study on social networks to a study purely on conventional information systems.

In today's cases, the human interactions are usually centred on some specific services. One of the major services provided by these social network systems is to recommend friends or information that their users may interested in. If we can accurately predict the edges (possible connections) that will be added to the network during the interval from time t to a given future time t' (t'>t)[65]. We can have a better understanding of what is the dynamics that driving the evolution of the social network so as to provide more accurate and meaningful recommendations. This problem is commonly known as the link prediction problem. Link prediction can be utilised in a range of applications in social networks where the most classic one would be recommender system[87] and many other applications which helps the user find potential friends or collaborations among scientists. Thus, link prediction has become an important task in social network analysis [117, 147].

In this thesis, I will focus more on the academia social network, since little work has been done for this particular group while great demands exist. Some of my early studies concentrate on the problem of link prediction particularly in the context of evolving co-authorship network, where nodes represent researchers and links/edges denote the co-authorship relations over a particular time. In this case, the research output is recommending potential experts that have future collaboration opportunities. Co-authorship networks have been extensively studied in the field of social network analysis [73, 115, 125]. A classical way of predicting academic relationships is through the bibliographic systems via co-authorship or citation data such as DBLP. However, the result could be too plain that is usually not accurate or sufficient enough to make a prediction for providing a recommendation. There are many other features that we can

use as parameters to make a prediction more comprehensive. For example, the latest publications today are richer in content such as images and even audios or videos as an appendix in demonstrations. Moreover, since the academic-oriented social networking platform[2] we built is getting more and more popular online, numerous user-generated data about academic information are being created every minute. Therefore, we have even richer content associated with the underlying papers that are explicitly archived in our networks. I think if we can use these sources in conjunction with the citation data, in the meantime combining methods that have been employed in different domains, we can get more interesting ideas from the big picture. So in Chapter 3 of this thesis, I will introduce a new method that can analyse the user relationship strength and reveal it clearly in a visible way. Moreover, a 'Time Varied Weight' link prediction algorithm has been proposed. This algorithm considers time as a new feature where more weights can be put onto different links since it is obvious that old events are less likely to be relevant to determining the future linkages than recent ones[122, 143].

### *1.2.2 Link Prediction in Information System*

As mentioned, more and more information systems today are utilising the Internet to take its function to the air. In the last sub-section, we also mentioned that the data created nowadays are richer in content in terms of images, audios and even videos. A web-based information system is more likely to become a platform of a data pool that everyone can share information online, and others who interested in this information can browse or even retrieve them by downloading to local drives. Inevitably, during these uploading, searching and retrieving processes people would like to communicate with each other for recommendations and suggestions. Moreover, some people may intend to find individuals who sharing the same interests that may be able to build up connections and become friends. On the other hand, the service provider would like to see people talking about their content that will also increase their system's popularity. Hence, more and more information systems start to allow users to leave 'Comments', to give 'Rate' or 'Stars' and even directly sent a message to communicate with the owner of the source information in real time. From these facts, an interesting idea emerged in my mind that any online information system or application (either PC based or portable device based) that offers some kinds of

[2]SCHOLAT website http://www.scholat.com will be introduce in Chapter 6

‘human interaction’ elements as social experiences can be regarded as a form of the online social network [2, 10, 96].

Therefore, from my view, a much broader range of sites (conventional information systems) or phone-based applications can be included in this category such as YouTube, Flickr, Instagram and Zomato. For example, YouTube and Flickr are primarily content-sharing orientated sites that are not typically perceived as social networks. Instead, they are more like a massive ‘online file distribution system’ that are recognised as ‘information systems’. Today they allow the user to rate interests of the content and leave comments to the content owner, even in a real-time manner. These functions they provided create a lot of human interaction data online. And this fact makes them pass over the boundary as a conventional ‘information systems’ and joining the bigger category as ‘online social networks’ from my view. Nonetheless, link prediction methods are closely related to the original information system problems which can be utilised in dealing the conjunct problems in online social networks as well.

One of the major issues relevant to analysing an information system is called information retrieval (IR)[26, 66] From the literal meaning of information retrieval, we can perceive the IR problem as simple as finding a book in a library where you may give the name of a book to a librarian. Then the librarian will get it for you by looking up the capital letter of the name that matches the numbered rows or columns of the shelves. Later, after we have computers and database system installed in the library, all the books and collections are being tagged either using a conventional bar code or latest RFID tags. Either way in this situation, a relational database will be used to store the information of the book associated with its location in the library. In this case, IR problem transfer to the problem about “*database querying*”, computation language are used not limited to SQL or CQL[129], etc. Currently, as mentioned, most of the information systems are providing their service online and creating unprecedented large amounts of data every second. Most of these data are generated by users and commonly deemed as “unstructured” in nature; hence, information retrieval problem under current situation has changed. In my understanding, given the essence of our goal here remain unchanged as finding information from a data pool, the current situation is that one querying can return hundreds and thousands of results back. If all these information retrieved is pulled into the server at once without any selection process, it will not only cause a burden to the server but also overwhelm the inquirer. This situation is commonly known as ‘information overload’, and then ranking has

become a crucial task for information retrieval systems today[26]. In chapter 4 of this thesis, a 'user-item-tag graph and user personal interest' model has been proposed to improve friend recommendation efficiency.

## 1.3 Research Issues in this Thesis

As discussed in above two sub-sections, a few research issues are revealed and studied in this thesis. First of all, given a data pool of an online social network/sites, how can we describe the current users' relationship more efficiently and accurately in a more tangible and visible way? Then, after the relationships of existing users in the network have been analysed, all the connections/edges of existing nodes are understood. How can we predict the new edges that are going to appear in the future?

On the other hand, as Internet technologies thrives, many issues emerge such as information overload issue and data sparsity issue. However, on the bright side, there are many interesting ideas that people work out to cope with these matters such as Description Tags' and 'Cross-Domain Learning'. How to utilise these newly invented ideas from the Internet to implement and improve the current services of the online social sites today has become a very useful and practical domain.

Last but not least, as mentioned in the previous sections, current popular online social networking sites are mainly targeting general social communities which are more life-oriented social networks. While we do have some specific career-oriented sites such as LinkedIn, there are many distinct communities that are indeed vast in numbers and desperately require some customised services such as the academic community[8]. Therefore, what unique services they need and how we can satisfy the needs have become a very practical issue in the industry.

## 1.4 Significance of the Study

Social network analysis has become a popular research topic in computer science. It is well-known that predicting on social networks is tough because: the online social networks have large quantities of users with millions nodes or more, and even billions of edges. Additionally, the data of online social networks have high dynamic quality, and the social activities of users are unpredictable where the joining or exiting of users, as well as the emergence or eliminating of edges, could happen at any time. Thirdly, the relationships in the social networks exhibit a lot of

diversity, the different kinds of systems have various types of relations, the degree of strength, and whether they have directions, etc.

If we can accurately predict the edges that will be created between two nodes in the network during a time interval from t to a given future time t'(t'>t)[65], we can understand how a social network evolves and what the dynamics that drive it from behind are. More importantly, since the links from the network stands for their maintenance and quality that reflect social behaviours of individuals and communities, hence link prediction research on them can be very helpful in the quantitative and qualitative assessment of human relationships in this information era where more people are participating in virtual communities online. Thus, link prediction is an important task in social network analysis. Last but not least, since link prediction algorithms and methods can apply to a wide variety of areas like bibliographic domain, molecular biology, terrorism and criminal investigations and any commercial recommender systems as mentioned earlier, therefore the result of this study may be extended or mutated to adapt to multi-discipline of studies.

## 1.5 Contributions of the Thesis

The main contributions of this thesis are to exploring and utilising link prediction related approaches to improve the services in Online Social Networking Sites in terms of their recommendation efficiency and accuracy. Contributions are listed as follows:

*1)* ***A User Relationship Strength Fusing Multiple Factors Model (URSMF)*** has been proposed which integrate various factors that will affect user relationship strength. URSMF took three types of factors into account and integrated them into the calculation namely the User Profile Similarity Degree, the User Friendship Network Structure Similarity Degree and the User Interaction Intensity. URSMF has been applied to real OSN data to calculate the relationship strength to support the friend recommendation function, and the result turns out positive.

2) ***A Link Prediction Algorithm Based on Time Varied Algorithm*** has been proposed which take into account time-varied weights for similarity indices. Take the co-authorship network

as an example. The weight of the links usually calculated from the number of co-authorized papers and the coauthored time of those papers. It is obvious to see that co-authorized papers that are published with a longer period of time from now are less likely to be relevant than recent ones in terms of predicting the future collaboration. Traditional link prediction algorithm similarity indices only simply consider the binary relations among nodes while time dimension and frequency of link occurrences are being neglected. However, a link established at different time interval will have a very different effect in the real world.

3) ***User-Item-Tag Graph based on Collaborative Filtering Model (UITGCF)*** has been proposed. UITGCF is a hybrid recommendation algorithm combining the diffusion on the user-item-tag graph and user personal interest model for friend recommendations. Firstly, it calculates similarities between users by mass diffusion method in the tripartite graph. Secondly, it introduces the relationship between users and tag graph of users and detects communities in the tag graphs of users for representing topics of user interests. By integrating similarities between users from mass diffusion method in tripartite graph and similarities of users from interest-based user recommendation in social tagging systems finally two kinds of similarities are integrated for user recommendation by the harmonic mean method.

4) ***Sparse Ranking Model Adaptation Framework for Cross-Domain Learning to Rank.*** This framework, which utilises $\ell_1$ Regularisation to transfer the most confident prior knowledge from the source domain to the target domain. Due to the sparsity-inducing property of the $\boldsymbol{\ell_1}$ regularization, the framework is able to reduce the negative effects of the feature gap between source domain data and target domain data. However, the optimization problem formulated by the framework is non-differentiable. It is difficult to obtain the solution by the most popular methods. To address this problem, we design an efficient algorithm from the primal-dual perspective.

5) ***SCHOLAT: an academic online social network solution.*** As mentioned in the section of research issues that academic community requires some specialised services that can

enhance the potential user's working efficiency in terms of communication, searching papers, researching ideas, teaching, organising conference and workshops, etc. By utilising and implementing the models and methods studied and created in this thesis, a live running online social networking sites dedicated to supporting teachers and researchers are created. In this thesis two of the novel systems that implemented into the website have been introduced as XPSearch and XSRecom.

## 1.6 Thesis Outline

The remainder of this thesis is organised as follows.

- **Chapter 2** *presents a literature review on the related areas of the problems studied in this thesis where many classic algorithms are introduced and categorised based on my understanding of my research topic area.*
- **Chapter 3** *focuses on the issue of link prediction particularly in the context of evolving co-authorship. Firstly, this chapter will analyse the user relationship by calculating the relationship strength. To understand this better, tools such as Pajek have been used to visualise the evolution of this network. Finally, a hybrid link prediction approach utilising time-varied weight information of links has been proposed, and the experiment result has proved that the new method proposed can achieve better results than the traditional link prediction algorithm.*
- **Chapter 4** *presents a hybrid collaborative filtering recommendation algorithm by utilising the diffusion on the user-item-tag graph and users' personal interests. A distinct feature of this model is that it integrates similarities between users from mass diffusion method in tripartite graph and similarities of users from interest-based user recommendation in social tagging systems.*
- **Chapter 5** *proposes a sparse ranking method to resolve the cross-domain learning to rank problem, which utilises ℓ1 regularisation to transfer the most confident prior knowledge of the source domain to target domain. Learning to rank has attracted a lot of attentions in recent years, and many algorithms have been proposed as supervised learning methods, which require sufficient labelled data to train precise ranking models. However, in some applications, it is impractical to collect sufficient labelled data, while*

*there are plenty of labelled data in a related domain named as source domain. Hence it results in a new problem named cross-domain learning to rank problem, which is to utilise the labelled data in the source domain to improve the ranking accuracy in the target domain. The solution of this chapter will help to solve the recommendation problem while less resource is available on one particular social network.*

- **Chapter 6** *presents a system called SCHOLAT, which is implemented as a scholar-oriented social network that aims to form an academic community to let users establish a connection with other researchers. SCHOLAT provides two innovative professional services that are useful to researchers, namely, XPSearch and XSRecom. XPSearch is a service that provides vertical searching of research papers with author name disambiguation. XSRecom uses a topic community-based method to provide users with a list of "Recommend Scholars", which can help them find potential collaborators who share the same research interests and may be interested in building a collaborative relationship. These two services boost the efficiency of searching papers and discover research opportunities for scholars.*

# Chapter 2

## Literature Review

*In the previous Chapter, I have introduced the basic idea and my understanding of what is a social network and how link prediction is related to Social Network and Information System as they can be regarded as 'online social network' when combined. Moreover, how recommendation system comes across this domain as a universal application. In this Chapter, I will first present the overview of link-prediction problem as a whole in terms of its problem definition, link prediction method categorization and some of my understandings towards the problem. To be specific, I have created two graphs that overview the link prediction technique and problem. In the current research field of link prediction, many generic, classic link prediction metrics have been created and used for many years which some of them using information of nodes, topology and social theory to calculate the similarities of node pairs. They are all good starting points and also the footstone of my research. Then I will go through a literature review on the link prediction metrics and methods systematically from section 2.2 to section 2.4. Recently, learning-based link prediction methods are becoming more popular in solving link prediction problem, so I will also present the literature review on the learning-based methods that are closely related to the application as recommendation system in section 2.5. It is worth mentioning that I will not go too deep for some of the metrics and methods as they are beyond my research focus for this thesis.*

## 2.1 Link Prediction Problem Overview

### *2.1.1 Mathematical Description of the Problem*

Mathematically, the link prediction task can be formulated as followed:

> Given a set of data instances at a particular moment G = (A, V) and the node $A_i$ and the node $A_j$, we have $\boldsymbol{A} = \{\boldsymbol{A_i}\}_{\boldsymbol{i=1}}^{\boldsymbol{n}}$ which is organized in this form of a social graph $G$ where **A** is the set of observed links. Let $\mathbf{v_{ij}} \notin$ A as an unobserved link that does not exists between a pair of nodes $A_i$, $A_j$ in the given data network. Hence the link prediction task here is to calculate the likelihood of the existence of these links **V**.

To visualise a link prediction task:

> Given snapshots of a social network from time $t_i$ to time $t_k$ as shown below. Hence our task is to determine where the edges will be added to the network during the time interval.

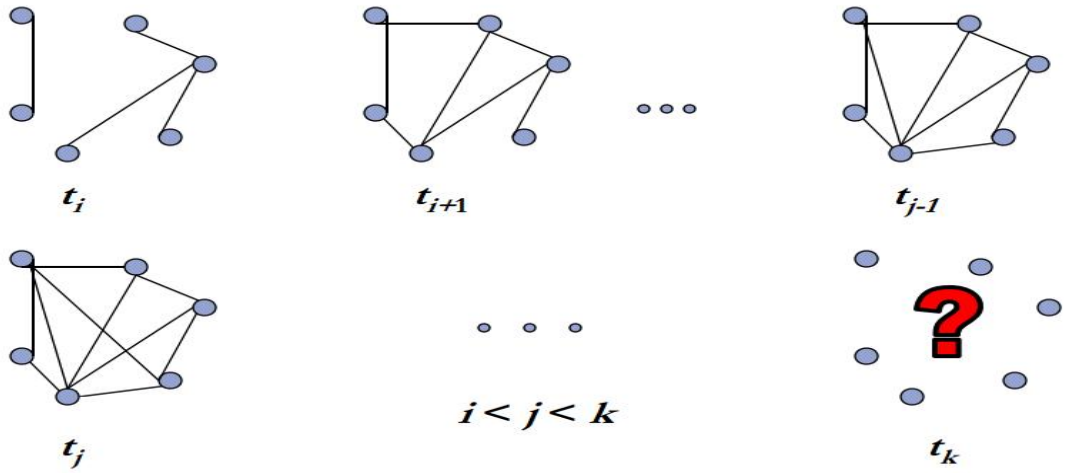



Figure 2.1 Visualised Processes of Link Prediction Task

Usually, this problem can be solved by similarities calculation between node pairs and some ranking techniques[35].

### *2.1.2 Link Prediction Problem Categories*

Link prediction problems can be viewed from many different perspectives. There are mainly two categories to start with: **Link Based Problem** and **Network Problem**. The first one

focuses more on the link characteristics and the second one concerns the network graph as a whole, as known as network evolution analysis.

To be specific, we can see that the network problem category has divided into Temporal Network, Location Network and Heterogeneous Network [4]. To briefly describe the meaning of these sub-categories: the **Temporal Network** which considers the time in particular as the network mode. For example, in my research, I can answer questions like "who is likely to publish at DBLP with whom as author or coauthor net year". By doing that, a simple way is to give more weight to more recent links in the network. **Location Network** can be regarded as link prediction problem in route recommendation application as analysing trajectory data. Last but not least, the **Heterogeneous Network** which means a network that can best represent the real world where a complete attribute value of the nodes is always hard to identify and obtain. Moreover, link prediction on such network is a non-trivial task, so most of the existing link prediction work considers the network as homogeneous one which means only one type of nodes and links exist.

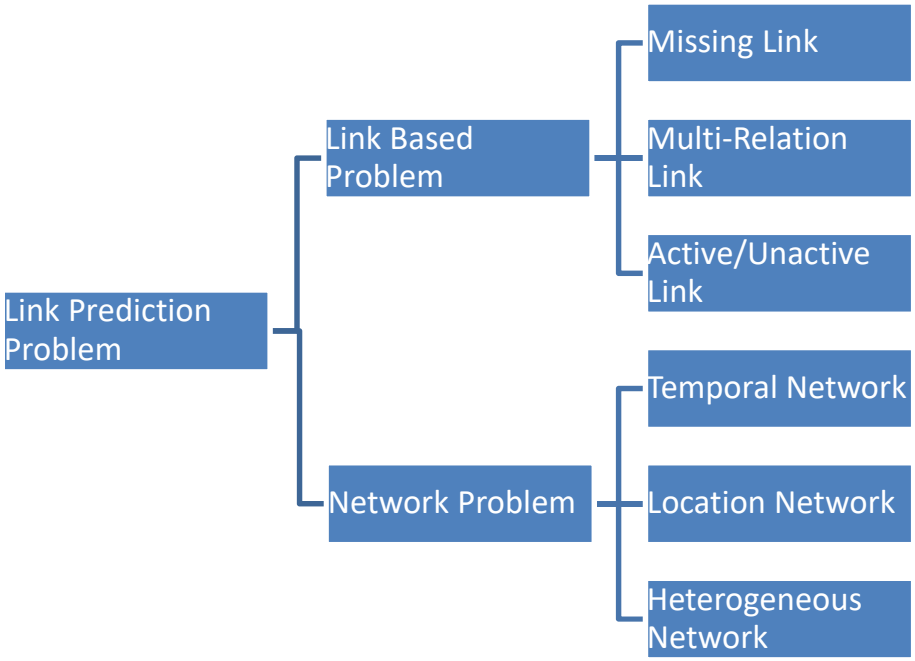



Figure 2.2 Link Prediction Problem Categories

On the other hand, the link category can be split into three sub-categories as Missing Link [111], Multi-Relation Link [151] and Active/Un-active Link. I will skip the first two as they are very intuitionally understandable. Active/Un-active Link is highly related to the Temporal Network as mentioned above. Giving an assumption that if a node pair interacts with each other

recently, then we can consider these two links has become 'Active' so that the time stamp of the last interaction has become the most important feature of the node pair [33]. Many specific and detailed link prediction techniques derived from these problems and I will present and discuss them later in different chapters of the thesis. In my research, I will focus more on the Link Based Problem but combine the idea of the second category since they are all closely related after all.

In the first group, Link prediction mainly addresses four different problems as shown in Figure 2.3 below. Currently, most of the research papers that intend to solve a link prediction problem concentrate on the issue of **Link Existence** which standards for whether there is a new link between two nodes will exist in the future or not. In my opinion, it is also very obvious to see that the first problem is the most fundamental factor that determines whether a study can be extended to the other three problems.

**Link Weight** is a condition that links have different weights associated with them

**Link Cardinality** is a condition that there is more than one link between an existing pair of nodes in a social network.

**Link Type** prediction is a bit different from the above two which gives different roles to the relationship between two objects.

According to the description of the survey as well as interpretations from other research papers [69, 93], I have modified the framework as shown in the right-hand side of Figure 2.3 to emphasise the importance of the fundamental step for link prediction as the Link Existence problem. Moreover, link existence can be regarded as new, missing or unobserved links.

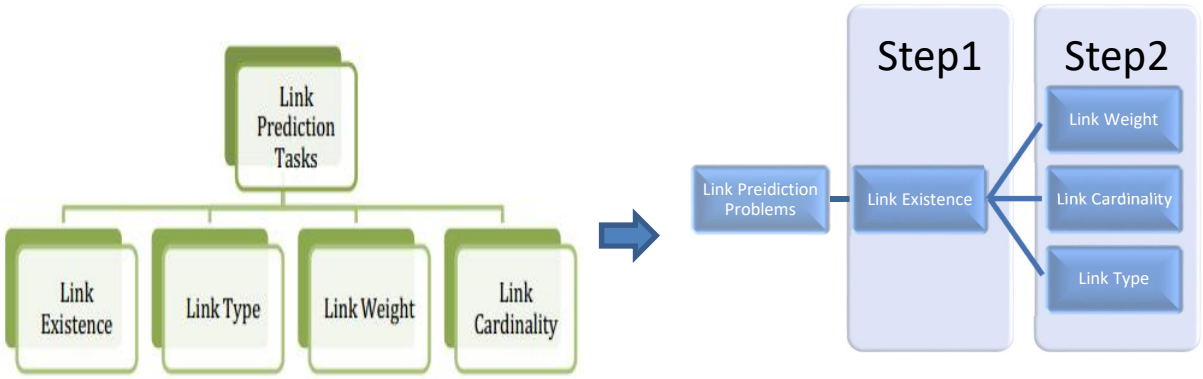



Figure 2.3 Traditional Problem of Link Prediction (A) converts to Modified Basic Tasks of Link Prediction (B)

### 2.1.3 Link Prediction Problem Approaches

There are many ways to approach a Link Prediction problem; some treat it as a probabilistic problem where prediction is deemed as measuring the joint probability between two nodes in an undirected graph[99]. Some treat it as a classification problem where predicting tasking can be regarded as ranking the similarity scores between two nodes[167]. These approaches can be categorised into two groups, one is so-called 'network evolution modelling', and the other is called 'feature based link prediction'. The former one predicts the future edges of a network taking some well-known attributes of social networks from **Social Theory** such as the power law distribution and small world phenomenon. The latter one is mainly using methods related to solving supervised classification task where a classification model is built to predict the unknown nodes via a training set. It is worth mentioning that machine learning techniques are highly related to this approach and can be extended into broader domains [6, 71, 107]. Figure 2.4 shows an overview of the listed the categorization of the common link prediction techniques. In my opinion, the **Node Based technique** is the most intuitive solution to solve a link prediction problem. The basic idea is that the more similar the node pairs are, the more likely that they will join with a 'link' if that link is not currently existing in the graph or network[3].

[3] *In this thesis I will interchangeably use 'graph' or 'network' to describe a figure denoted with nodes and links*

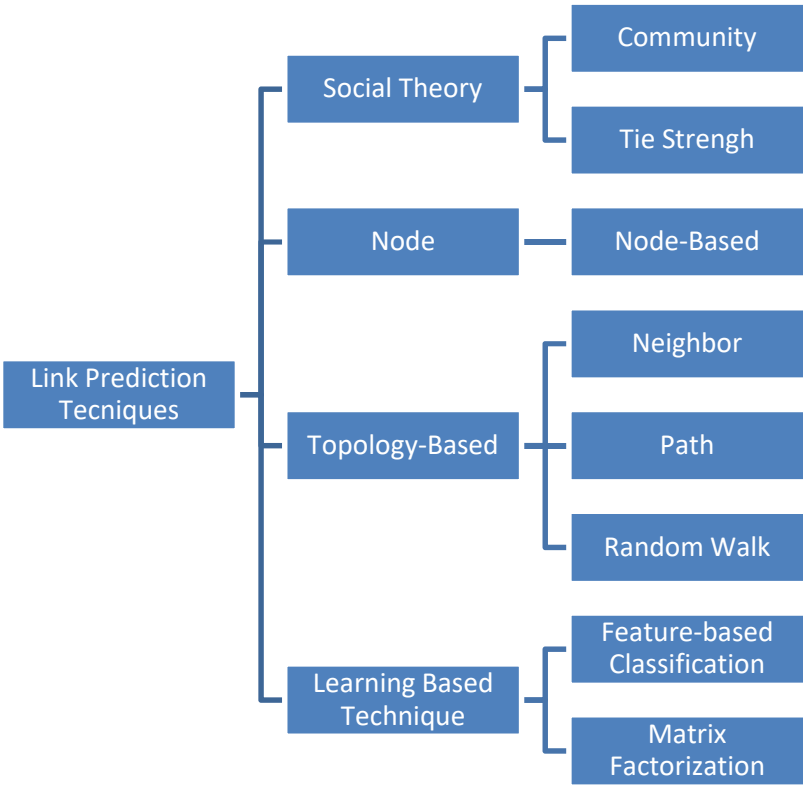



Figure 2.4 Link-Prediction Techniques Overview

**Topology-Based Metrics** is another very commonly used technique to solve link prediction problem I will include the detail explanation in sub-section 2.4. Moreover, **Social Theory techniques**, from my view, can be regarded as the 'ancestor' of the link prediction problem as a whole. Generally speaking, these techniques come from the classical social analysis methods, and some of the most brilliant ideas will be introduced in next subsection.

Last but not least, the **Learning-Based Methods** which is quite different from the other metrics. So far the both the node-based and topology based metrics are built to compute the similarities between node pairs and created a similarity list. Finally, the prediction can be given by ranking the list in descending order. However, the learning based method treats the link prediction problem in a slightly different way, and the major application such as recommendation system is highly related to this type of techniques. This kind of method sees link prediction as a binary classification problem [6, 84] hence models from machine learning such as classifier [110], collaborative filtering models are used to help to solve this problem. The basic idea is node pairs in a graph will be classified and correspond to an instance with features or can be regarded as 'labelled'. Then if a pair of nodes has high possibility to link will be

labelled as positive otherwise negative. In my view, this spectrum of techniques is evolved from the previous metrics and combined the strength of both nodes based and topology based methods. In this thesis, my research related to this spectrum of techniques will focus on its application emphasis on how to utilise the technique to improve the function of recommendation system and detail literature review will be introduced in sub-section 2.5.

## 2.2 Social Theory Based Metrics

The sociologist professor at Yale, Stanley Milgram [9, 16, 101] have established a theory called Six Degrees of Separation back in the 60s of last century, also known as the small world theory. This theory has derived a compelling vision as 'Given two strangers with proper tools for communication, somehow there will be certain relations and connections between them'. In the era of Web2.0, such theory has become the core idea for Social Networking Sites (SNS). It is indeed proved right that there is a common connection between two random people. However, in real life, many different factors will affect the level of relations between people such as families, relatives, friends and colleagues. Moreover, factors of other dimensions such as distance and time will also add on to the attributes of such connect.

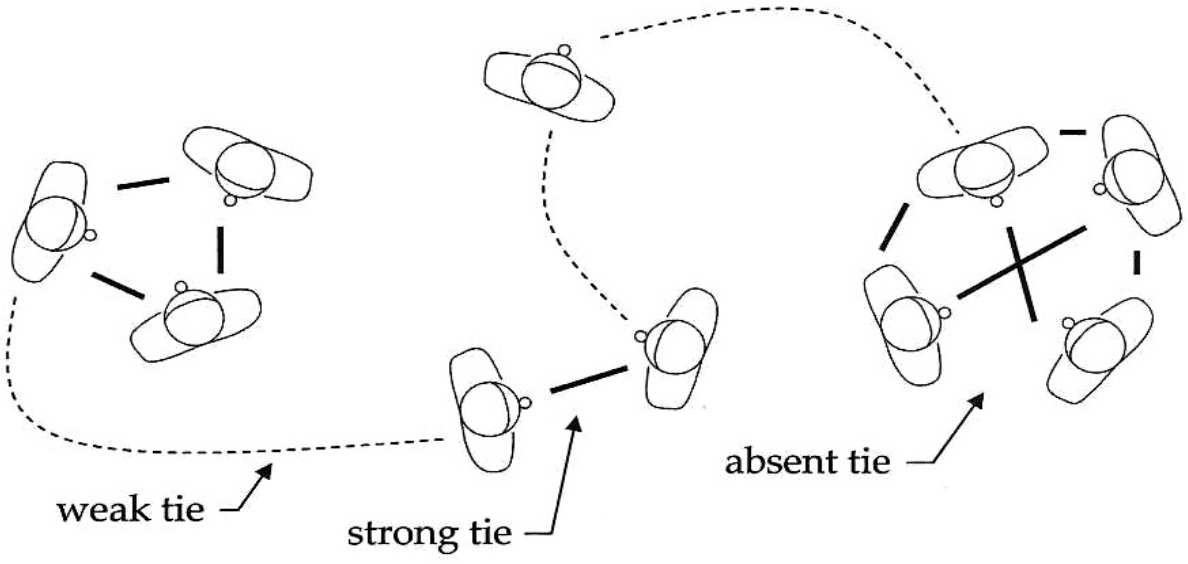



Figure 2.5 Strong ties and Weak ties

Mark Granovetter [45, 92], the sociologist professor at Stanford, put forward the famous idea 'the weak ties are stronger than strong ties when it comes to information proliferation' (in

his thesis “ties” indicates “links” between nodes or individuals). As mentioned, many factors contribute to affect connections between people. Mark distinguishes these differences into two groups: Strong Ties and Weak Ties where strong ties refer to connections between families, relatives and friends while weak ties refer to the other links in society, and these relationships seem fragile and powerless. However, when it comes to information proliferation, these weak ties unfold an underestimated power. The reason is that information within a strong tie are overlapped and blocked due to limited sources while weak ties have wider coverage and relationships from a different area (including physical distance or career fields.) with knowledge and information that are not familiar within our strong tie. Hence these weak-tie-friends have become a media between strong ties and so on. The famous social networking site LinkedIn is a very classic and successful example of using this theory to help customers to build up and largely expand their weak ties. Many have benefited from its large sources of industry partners.

Another basic concept of social networking comes from the anthropologist at Oxford, Robin Ian MacDonald Dunbar[46]. In order to predict a human’s social group size, he used the correlation observed for non-human primates and used a regression equation that provides to a human "mean group size" of 148 (casually rounded to 150) as known as the “Dunbar’s Number”. This figure suggested a cognitive limit to the number of people with whom that a normal people can maintain stable social relationships[46]. On the other words, only if the relationships are within this size can an individual know who each person is and how each person relates to every other person. Nowadays, such theory has been set as a standard for human resource management and social networking services. For example, SIM card memory for cell phone has a limit of 150 addresses; the old MSN set the default number of friends to 150. Last but not least, it is reported that the average number of friends on Facebook of the individual is around 130.

## 2.3 Node-based Metrics

When considering the link prediction problem, the most intuitive solution people can think of is calculating the similarity rate of the given two nodes. The idea is very intuitive: the more similar the nodes are in terms of their shared parameters and features the higher possibility that these two nodes can be linked. Interestingly, there is an old Chinese saying goes “birds of a feather flock together” which is entirely consistent with this idea. In fact, in social networks, people also tend to create relationships with people who share similar education background,

similar interests and so on. Generally speaking, this metrics is assigning a score between node (x, y) representing that the similarity score between node x and node y. The higher score means a higher probability that x and y may be linked in the future and vice versa. This way we can predict the links between nodes by ranking the similarity scores we calculated[122].

By using this method, first, we need to select the parameters of the nodes, in a practical social network, nodes usually have features (parameters) such as address, career, phone number, etc. As mentioned in the previous chapter, today there are much more abundant features in the online social networks thanks to the development of the Internet technology, and we can find more features like emails, personal interesting, research interest and publication recorded regarding academic, social networking and so on. All these information can be used for calculating the similarity between two nodes in a straight forward way. Hence text-based and string-based similarity metrics are usually used here since in most cases these node attribute values are in textual forms. The classic methods of this field can be found in some surveys [7, 79, 99]. I will not further review these two metrics as is not very relevant to my research focus since node-based metrics are mainly focused on how to select and collect enough in number as well as meaningful user attributes and features. In another word, this metrics is only useful if only we can obtain enough attributes in social networks.

## 2.4 Topology-based Metrics

While a node-based metrics requires many attributes to work, Topology-based Metrics that introduced in this section can work without any attributes of the nodes and edges. We named it topology-based methods because these type of methods utilise topological information rather than information of nodes or edges. There are many topology-based metrics proposed in the last few decades. Here, I will provide a general description of some of the most popular topology-based metrics in the realm of link prediction. Based on the characteristics of these metrics, normally they can be divided into three categories namely neighbor-based metrics, path-based metrics, and random-walk-based metrics.

Before listing out these classical methods I will first explain the standard notations in link prediction methods as shown in the table below:

Table 2.1 Common Notations Used in Link Prediction Methods

| | |
|---|---|
| Γ(x) | Set of Neighbours of x |
| Γ(y) | Set of Neighbours of y |
| Uppercase Letters | Adjacent matrix of the network |
| Lowercase Letters | Nodes in the social network |
| \|Γ(x)\| | Number of neighbours of node x |
| \|Γ(y)\| | Number of neighbours of node y |

### 2.4.1 Neighbor-Based Metrics

As mentioned, people tend to create new relationships with people that are closer to them in the real social network [131]. In node-based methods, we calculate the similarity rate by the given attributes. In neighbors-based metrics, apparently, we calculate the similarity rate by observing the common neighbours of the node pair. As a result, many neighbor-based metrics for link prediction have been proposed by researchers.

**Common Neighbours (CN)**

The beauty lies in the simplistic; CN metric has become the most widespread method used in link prediction problem largely due to its simplicity[104]. Many other methods are derived from CN. For a pair of nodes (x, y), the number of neighbours that both x and y have to interact with CN. The larger the amount of common neighbours the higher likelihood that x and y will be connected in the future.

$$Common\_neighbour(x, y) = \lfloor \Gamma(x) \cap \Gamma(y) \rfloor \quad (2.1)$$

Since common neighbour is not normalised, sometimes it can only reflects the relative similarities between a node pair. Therefore many other neighbor-based methods consider how to normalise the common neighbours as an improvement of the method as shown below.

**Jaccard Coefficient (JC)**

In order to solve the problem mentioned in CN metric, the Jaccard coefficient [130] normalises the size of common neighbours. It calculates the proportion of common neighbours in the total number of all their neighbours. And the pair with larger proportion have higher weight. This measure is defined as:

$$Jaccard\_coefficient(x,y) = \frac{\lfloor \Gamma(x) \cap \Gamma(y) \rfloor}{\lfloor \Gamma(x) \cup \Gamma(y) \rfloor} \tag{2.2}$$

**Salton Cosine Similarity (CS)**

Salton Cosine Similarity is a common cosine metric that measures the similarity between two nodes x and y [70]. The larger the cosine is, the higher likelihood the two nodes links. It is defined as:

$$\text{Salton CS}(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{\sqrt{|\Gamma(x)| \cdot |\Gamma(y)|}} \tag{2.3}$$

**Adamic-Adar Coefficient (AA)**

Originally the Adamic-Adar Coefficient was proposed by Adamic and Adar for computing similarity between two web pages [74, 157]. Social network analysis also wildely apply this method across different situations. Adamic-Adar is closely related to Jaccard's coefficient in terms of formulation. However, on the contrary to most common-neighbour based algorithms, AA defines that those who have fewer neighbours are weighted more heavily. It is defined as:

$$Adamic\_Adar(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log|\Gamma(z)|} \tag{2.4}$$

**Preferential Attachment (PA)**

Preferential Attachment [158] indicates that new links will be more likely to connect nodes that with a higher degree which means the more neighbours a node's neighbour has, the higher the chance that a new link will appear between the target node pair. It is defined as:

$$Preferential\_Attachement(x,y) = \left|\Gamma(x)\right| \times \left|\Gamma(y)\right| \tag{2.5}$$

### 2.4.2 Path-Based Metrics

Similar to neigh-based metrics another metrics that does not require attributes of nodes and edges is called path-based metrics. Obviously, rather than using the nodes and neighbour's information, these type of metrics calculates the paths between two nodes for computing the similarities of node pairs [121].

**Shortest Distance (SD)**

The similarity between two nodes can be defined by the shortest distance which means the shorter the path distance between two nodes the more similar they are. As shown:

$$\text{SD Score } (\mathrm{x},\mathrm{y}) = Length\ (Shorts\ (x,y)) \tag{2.6}$$

**Local Path (LP)**

Unlike the previous node-based metrics or the simple shortest distance method that only use the information and features of the nearest neighbours [69, 94], LP metric extends the use of information to the neighbours within three distances in length to the current node. Obviously, the shorter the path is, the more relevant the path will be, so there is an adjustment factor $\boldsymbol{\alpha}$ that applied in this method in order to provide more sensible result. It is worth mention that $\boldsymbol{\alpha}$ should

be a number $-\mathbf{1} \leq \boldsymbol{\alpha} \leq 1$ this metric's formula is represented as (2.7). Where, $\mathbf{A^2}$ and $\boldsymbol{A^3}$ denote adjacency matrices that having 2 length and 3 length distances to the nodes respectively.

$$\mathrm{LP} = \mathrm{A}^2 + \ \alpha\, A^3 \tag{2.7}$$

**Katz (KA)**

Compared to LP that considers paths up to 3 paths distance, KA takes all paths between two nodes into consideration [130]. Similar to CN metrics introduced above, in KA the longer the path is the less weight it has on final similarities. Thus as shown below the $\boldsymbol{paths^{l}_{x,y}}$ is the set of all paths from $\mathbf{x}$ to $\mathbf{y}$ with length$\boldsymbol{l}$, $\mathbf{0} \leq \boldsymbol{\beta} \leq \mathbf{1}$ is the decay factor used to limit the growth of length and the smaller $\boldsymbol{\beta}$ will cause smaller $\boldsymbol{l}$ , and lager similarity degree.

$$\mathrm{Katz}\,(\mathrm{x},\mathrm{y}) = \sum_{l=1}^{\infty} \beta^{l} \times \left|paths^{l}_{x,y}\right| = \beta A + \ \beta^2 A^2 \ + \ \beta^3 A^3 + \cdots \tag{2.8}$$

### *2.4.3 Random-Walk Based Metrics*

Random-Walk can also be the model for presenting social interaction between nodes in a social network [36, 38]. The destination of a random walker from a node can be denoted by the transition probabilities from a node to its neighbours. There are many link prediction metrics that calculates similarities between nodes based on random walk. This section will briefly introduce some of the most famous ones that are known to most researchers in this area[49]. The fundamental idea is: set the node x as a starting point and y as the ending point. We calculate the Hitting time or can be understood as 'steps' it took from x to y, the fewer steps are higher similarity x and y has.

**Hitting Time (HT) [49]**

This idea comes to a graph theory, for two vertices x and y in a graph, HT $(\mathrm{x},\mathrm{y})$ defines the expected number of steps required from a random walk from x to y. Shorter HT means that the

node pair has higher similarity score. Let $P = D_A^{-1}A$, where diagonal matrix $D_A$ of A has value $(D_A)_{i,i} = \sum_i A_{i,j}$ and $P_{i,j}$ is the probability of stepping on node **j** from node **i**.

$$HT(x,y) = 1 + \sum_{\omega \in \Gamma(x)} P_{x,\omega}\ HT(\omega, y) \tag{2.9}$$

**Commute Time (CT) [99]**

Due to HT metrics is non-symmetric; CT has developed to from optimising the metrics from a contrary perspective. CT counts the expected steps both from x to y and y to x and is defined as:

$$CT(x,y) = HT(x,y) + HT(y,x) = M\,(\, l^{\dagger}_{x,x} + l^{\dagger}_{y,y} - 2l^{\dagger}_{x,y}\,) \tag{2.10}$$

Where the Laplacian matrix $\boldsymbol{l} = \mathbf{D}_{\boldsymbol{A}} - \boldsymbol{A}$ has $\boldsymbol{l}_{\dagger}$ as the pseudo-inverse, **M** is the number of edges in a social network as a constant. $\boldsymbol{l}^{\dagger}_{\boldsymbol{x},\boldsymbol{x}}$ is the set of factors of matrix $\boldsymbol{L}_{\dagger}$. The less value of the CT the higher similarity rate dose the node pair has.

**Cosine Similarity Time (CST) [34]**

Based on $\boldsymbol{l}_{\dagger}$ the cosine similarity time metric is calculating similarity of two vectors and it can be defined as follows:

$$CST(x,\ y) = {l^{\dagger}_{x,y}} \Big/ \sqrt{l^{\dagger}_{x,x} l^{\dagger}_{y,y}} \tag{2.11}$$

**SimRank [67]**

This algorithm is a recursion-based method. The assumption is given that whether two nodes have connections is related to its neighbouring connections. In another word, if two nodes

are connected to similar nodes then this node pair are similar. There is a parameter γ that controls how fast the weight of connected nodes decrease as they get farther away from the original nodes.

$$\text{SimRank}(x,y) = \gamma \cdot \sum_{a\in\Gamma(x)} \sum_{b\in\Gamma(y)} SimRank(a,b) \Big/ |\Gamma(x)| \cdot |\Gamma(y)|$$

$$E = \{e|e \le x, b > or \ < a, y >, ba \in \Gamma(x), b = \Gamma(y)\} \tag{2.12}$$

Where $\boldsymbol{\gamma}$ $(\mathbf{0} < \boldsymbol{\gamma} \le \mathbf{1})$ is the decay factor of the recursion, E representing the neighboring node set of node x and y.

## 2.5 Learning-based Methods

As mentioned this spectrum of techniques combines both node-based and topology-based methods since the features of nodes considered in these methods consist of both similarity features from topology-based metrics and other specific social network features from node-based metrics such as textual information, domain knowledge and other attributes. Many learning-based link prediction methods have been proposed in recent years based on the features provided by the relatively classic methods introduced above regarding internal attributes and external information [41, 79]. Most of the learning based methods can be regarded as a typical feature-based classification problem [84, 116, 161].

### *2.5.1 Feature Based Classification*

Let $\mathbf{x}, \mathbf{y} \in \mathbf{V}$ be nodes in a graph $\mathbf{G}$ （$\mathbf{V}, \mathbf{E}$） and $\boldsymbol{l}^{(x,y)}$ be the label of the node pair instance$(\mathbf{x}, \mathbf{y})$. We can have:

$$l^{(x,y)} = \begin{cases} +1 \ if \ (x,y) \in E \\ -1 if \ (x,y) \notin E \end{cases}$$

Where a pair of nodes can be labelled as positive if there is a link connecting the nodes, otherwise, the pair is labelled as negative. As we can see this is a typical binary classification

problem and many supervised classification learning models can be used to solve it, just to name a few decision tree, support vector machines [23], Bayes, etc. While using this method, it is important to define, select and collect appropriate features from social networks. Thus we can use features provide by node-based, topology-based and social theory based metrics. However, all these features extracted are very sparse in terms of semantic description [85, 150]. Providing a unified way to describe the abundant and diverse academic information has been a debatable and challenge issue around the world. Newman [105, 106] firstly rose up the idea of analysing the structure of scientific collaboration network by calculating statistics about coauthors relationships. FRBR (Functional Requirements for Bibliographic Records) [76] is an entity-relationship model developed by the International Federation of Library Associations and Libraries (IFLA), which is an application model for describing bibliography records in the area of academic publication. Furthermore, Scholarly Works Application Profile (SWAP) created on top of FRBR as its semantic implementation further introduced a way of describing electronic publications such as peer-reviewed journal articles, work papers and theories, etc. On the other hand, FOAF [21, 155] which stands for Friend of a Friend is a metadata standard focus on describing people and those relationships among people that have become the basic element of a virtual community. Another widely accepted metadata standard is Dublin Core [142], which is a set of predefined properties for the description of documents in multi-disciplines. Finally, the MarcOnt ontology is also a unified bibliography proposed by Dabrowski [40] which is created based on analysis of a wide range of existing literature standards, including MARC21, ISBN, BibTex, FRBR, that explore the field of semantic description of academic literature.

### 2.5.2 Matrix Factorization

Link prediction problem can also be regarded as a matrix completion problem which the matrix factorization can be used to solve this issue [47, 100] . The graph we considered in link prediction problem can be factorised as $\mathbf{G} \approx \mathbf{L}(\mathbf{U} \wedge \mathbf{U}^{\boldsymbol{T}})$ for $\mathbf{U} \in \mathbb{R}^{\boldsymbol{n}\times\boldsymbol{k}}$, $\wedge \in \mathbb{R}^{\boldsymbol{k}\times\boldsymbol{k}}$ and link function $\mathbf{L}(\cdot)$, where $\boldsymbol{n}$ is the number of nodes and $\mathbf{k}$ is the number of latent features. Each node $\mathbf{x}$ has a corresponding latent vector $\boldsymbol{u_x} \in \mathbb{R}$ . And the predicted score of this mode for a pair node $(\boldsymbol{x}, \boldsymbol{y})$ is $\mathbf{L}(\boldsymbol{u_x^T} \wedge \boldsymbol{u_y})$.

### 2.5.3 Recommender System

Recommender systems are to assist users to find out suitable items by analysing users' preferences. The core part of recommender systems is recommendation algorithms that can be divided into mainly classified into content based (CB) methods [109, 128] and collaborative filtering (CF) methods[15, 64, 124, 146]. Content-based recommendation methods extract characteristic units of users and items from their profiles and then recommend suitable items that are similar in content to items the user has liked in the past, or matched to attributes of the user[15]. These methods are very similar to a common neighbour in node-based similarity calculation. In contrast, collaborative filtering methods recommend items based on the preferences of other similar users or items and have been extensively used in some famous commercial systems [64], such as ebay.com. There are two main disciplines of collaborative filtering: the neighbourhood methods and the model-based methods [124]. The former predicts a user's rating on a target item by the other users or items with high correlations. However, the latter uses the user-item matrix, in whole or in part, to train a prediction model.

**Collaborative Filtering**

As the major approach for a recommendation system, collaborative filtering has been most widely applied in different fields because of its advantage of relying on the user-item interaction history [10, 53, 64]. There are two types of collaborative filtering approaches: neighborhood-based and model-based. The difference between the two lies in how to use the user-item ratings. The former directly uses the stored ratings in the prediction. The latter uses these ratings to learn a predictive model.

Currently, Matrix factorization (MF) is one of the most popular model-based CF methods. MF techniques, including principal component analysis (PCA) [100], singular value decomposition (SVD) [134], Regularized Matrix Factorization (RMF) and latent Dirichlet allocation (LDA) [18], have been in particular well implementation to recommender systems. But these methods also encounter the data sparsity problem. To learn the characteristics of users/items, traditional MF techniques map both users and items into two low-rank user-specific [52]. And now, a large number of variants are proposed. For instance, Koren [78] proposed a methodology, named SVD++, to incorporate the SVD with neighbourhood information. Ma et al. [96] extended the RMF by integrating two social regularisation terms to constrain the matrix

factorization objective function under the assumption that friends with similar or dissimilar tastes are treated differently in the social network. Yelong Shen et al. [126] advanced a joint Personal and Social Latent factor (PSLF) model for the social recommendation. Santosh et al. [72]proposed an item-based method for generating top-N recommendations that learn the item-item similarity matrix as the product of two low-dimensional latent factor matrices. Chu-Xu Zhang et al. [159] considered the neighbours' impact on the interest of each user in the same LFS and proposed a recommendation model based on clustering of users (UCMF). Szwabe et al. [134] combined random indexing (RI) technique and SVD to describe content features of items.

## 2.6 Chapter Summary

This chapter has presented a big picture of the link prediction problem in terms of the problem statement, problem categories and problem approach. Many classical techniques related to the issue have been presented and introduced. They are all good starting points and also the footstone to my research. It is worth mentioning that the learning-based methods are relatively the newest domain of this study and also where many other fields of studies can combine with the research. Meanwhile many applications are derived from this domain such as the recommendation system which is the most common application of link prediction in the social network.

We are now in an information era and traditional social networks are running online, where traditional information systems are depending on the Internet to provide useful functions. However with all these rapidly growing amounts of information available on the Internet, how to let people efficiently find most useful information has attracts much attention from both academia and industry communities during the past decade. As an indispensable technique for solving the information overload problem as well as the primary application direction for link prediction methods, recommender systems have been applied to address the problem in different domains, such as product recommendation in Amazon.com, music recommendation at Yahoo Music, video recommendation at YouTube and even route recommendation for GPS navigation. So in this study, many of the research outcomes in this thesis are to either improve the current recommendation efficient and accuracy or establish a new model with innovative algorithms to create a new recommender system.

# Chapter 3

## The User relationship Analysis and Link Prediction in the Online Social Network

*Many real world domains can be well described by networks, where nodes represent individuals or agents and links denote the relations or interactions between nodes[94]. Particular examples such as Email System like Enron have 250,000 emails connecting over 28,000 people, Telephone Calls Network like AT&T that recorded 275 million calls each day among 350 million individuals. Last but not least Research Publications Networks interpreted as co-authorship events such as Cite Seer archive 730,000 papers with over 770,000 authors.*

*Social networks are very dynamic objects since new edges and nodes are added to the graph over the time[6]. Social network Analysis has become a hot topic in computer science and many other subjects. Nowadays Social Network Analysis heavily relies on link prediction since its application has been widely spread over different domains. In recent years, there has been a strong interest in the way in which the social network is represented in a graphical form and the method of predicting the topology and semantic measures using the similarity between the two nodes. This chapter introduces two algorithms for specific types of online social networks: online social networking. Experimental results show that these two algorithms can achieve better results than other algorithms.*

## 3.1 Introduction

A social network [7, 130]is a set of people or groups each of which has connections to some or all of the others. Today this network has been tremendously expanded over the Internet, and many online social network sites are growing at a remarkable speed. Much valuable information lies behind the screen. Social Network Analysis (SNA) has a history stretching back at least half a century. It focuses on the structure of relationships, ranging from casual acquaintance to close bonds, and measures formal or informal relationships to understand the connection and the structure of numerous nodes [147]. So today there is much more information out there than traditional social network used to have that awaits new technology to explore and reveal.

In this chapter, I introduce firstly a method that aims at analysing the user relationship and propose an algorithm called URSMF-User Relationship Strength Fusing Multiple Factors. The result of this algorithm can also be used to generate a visualised friendship graph showing the relationship strength in a more visible and sensible way. This is a fundamental step before further research into this area as analyse user relationship can provide a big picture of the friendship linking graph. In the experiment, we apply data from an academic online social network called SCHOLAT[1] (www.SCHOLAT.com), and the result positively indicates the validity of the algorithm.

The data I studied comes from an academic online social network and there are many interesting attributes in this particular type of online social network that attracts me to look deeper. A number of papers and authors are used to compute for the productivity. Further, the variation of key properties (average distance, diameter, clustering coefficient, and giant component) of co-authorship network presents the patterns of the collaboration of the conference over time. Then I focus on the problem of link prediction particularly in the context of evolving co-authorship expecting to find out those implicit ideas how does the co-authorship relation happen. Moreover, how can we accurately predict and recommend a new possible co-authorship relation to a user? A hybrid approach utilising time-varied weight information of links has been proposed, and experiments have shown that the link prediction algorithm based on time-varied weight can reach a better result.

## 3.2 Related Work

For the User Relationship Strength Analysis Algorithm I proposed, several works have been done. Xiang et al. [152] proposed an algorithm based on Trust Propagation Strategy called TP-URS and this method mainly utilised the structure feature of users' network. Yu et al.[156] proposed another method that calculates relationship strength by modelling users interactions over a period based on Hawkes Process and it is called as HP-URS. Both Zhao [166] and Khadangi [77] utilised users' interaction information and their profile features to calculate relationship strength after collecting various kinds of interaction data from the network. Generally speaking, current algorithms for calculating relationship strength has achieved some progress in this topic area. However, most of the current methods only take a single attribute into consideration, either relationship network structure or user interaction but never combine multiple attributes to apply in relationship strength calculation. On the other hand, user relationship strength in OSN is affected by multiple factors in reality, for example, user profile similarity degree, user friendship network structure similarity degree and user interaction intensity. All these factors are reflecting the relationship strength to some degree. Hence we proposed a new way to calculate the relationship strength called URSMF – User Relationship Strength Fusing Multiple Factors.

For the further link prediction algorithm I proposed the related works are mainly from classic competitors: there are several approaches to solving this link prediction problem. The topology-based approach is the most widespread one, including Common neighbours [104], Adamic/Adar [2], Preferential Attachment [158]. Liben-Nowell and Kleinberg [89] examine various topological features. They find that topological information is quite useful when compared to a random predictor. In addition, it presents good performance and is easy to implement. Researchers have also used Probabilistic Models to solve the link prediction problem. Taskar et al. [139] use discriminatively trained relational Markov networks to define a joint probabilistic model over the entire graph [121]. The trained model is used to collectively classify the test data. Kashima and Abe [74] propose a parameterized probabilistic model of network evolution and then use it for link prediction. They assume the network structure is in a stationary state and propose an EM algorithm to estimate model parameters. They report encouraging results on two small biological datasets. However, both collective classification and training global probabilistic models can be expensive to compute and typically do not scale well to medium and large-scale

networks [17, 93]. As link prediction is fundamentally a binary classification problem, it is natural to use the powerful binary classification models that have been developed in machine learning [53, 55]. The primary advantage of this approach is the ability to combine multiple attributes and unsupervised link predictors into one joint prediction model [95]. Some earlier link prediction approaches have also been applied to this type of data [144]. However, these previous methods have a drawback. These early works did not take into account the weight of the links created at different times in the past. Obviously, the old event is unlikely to be related to the recent event and decide the future. To overcome this limitation, we propose a hybrid approach that takes advantage of the time-varying weight of the link. In this study, we evaluated the validity of the proposed method using the bibliographic data set: DBLP. The results show that the link prediction algorithm based on time - varying weights can achieve better results.

### 3.3 User Relationship Strength in Online Social Networks

Online social networks (OSN) are thriving over the internet; large companies such as Facebook, Twitter and Weixin are attracting a large number of users, and the number is rising fast. Users in these OSN sites can build up their own friendship network and interact with their friends through this network by sharing, comment, click ‘Like’ or ‘Forward’ etc. During these kinds of interactions, users can always find new friends with the same interest and expand this network. This network has a very complicated structure, and a tremendous amount of user-created data is creating every day. Much of the implicit information behind these data is very valuable in research. One of the hottest research topics in this area is very intuitive called Mining User Relationship. The main research problem of this topic is how to effectively calculated ‘Relationship Strength’. This is a very fundamental problem to solve before step into other related topics in this area including Friend Recommendation [98], Community Discovery [165], and Information Diffusion [124], etc. An algorithm to calculate user relationship strength (RS) have been proposed called URSMF – User Relationship Strength Fusing Multiple Factors which integrate various factors that will affect user’s relationship strength and further practical experiments prove its validity.

### 3.3.1 Computation Framework of URSMF

URSMF took three types of factors into account and integrated them into the calculation namely the User Profile Similarity Degree, the User Friendship Network Structure Similarity Degree and the User Interaction Intensity. By considering all these factors, we need to first collect all the data we need including user's profile data, friendship network structure and history data of users' interaction. It is worth mentioning that for user interaction data we focus on four classic interaction data namely the sharing, comment, thumbs up (like) and forward.

After preprocessing the data collected we calculate the similarity degree of all three factors respectively: Sim_profile represents the similarity degree of the user profile, Sim_friendship represents network structure similarity degree, and at last the S_interaction is for interaction intensity. And finally, we come up with relationship strength (RS) by adding weights to these attributes.

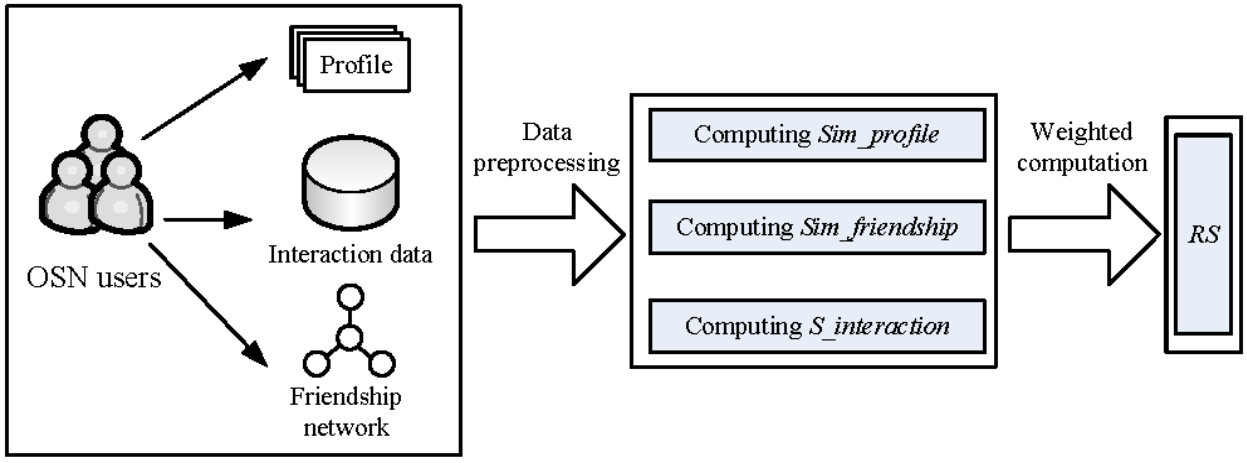



Figure 3.1 Computation framework of URSMF

### 3.3.2 Factors Similarity Calculation Detail

#### User Profile Similarity Degree

OSN user profile document always contains some opened personal information such as personal interests, education, career, life experience and other information. Since most of these user created data are genuine, then users with similar interest or life experience are highly likely

to be good friends in real life. So, the result of similarity degree for a user profile can reflect on the relationship strength of users.

Text Vector Space Model can be used to calculate the similarity degree of profile document. First, we exclude stop words in profile documents, and then we extract feature words and determine weights for them based on feature word dictionary $T = \{t_1, t_2, \ldots, t_m\}$. Given user $u_i$ and his profile document can be represented as a feature word vector $u_i = (w_{i1}, w_{i2}, \ldots, w_{im})$, where $w_{ip}$ represents the feature word's term frequency-inverse document frequency $t_p$ in the profile document of the user $u_i$ and $m$ is the length of the feature word dictionary. Finally, the definition of the calculation of user profile similarity degree $Sim\_profile(u_i, u_j)$ is defined below.

**Definition 1:**

User $u_i$ profile document is represented as $u_i = (w_{i1}, w_{i2}, \ldots, w_{im})$, and user $u_j$ profile document is represented as $u_j = (w_{j1}, w_{j2}, \ldots, w_{jm})$ and then the similarity degree of profile documents between the user $u_i$ and $u_j$ can be defined as:

$$Sim\_profile(u_i, u_j) = cos(u_i, u_j) = \frac{\sum_{k=1}^{m} w_{ik} \times w_{jk}}{\sqrt{\sum_{k=1}^{m} w^2_{ik}} \times \sqrt{\sum_{k=1}^{m} w^2_{jk}}} \quad (3.1)$$

The valid value range is [0,1] for $Sim\_profile(u_i, u_j)$ where 0 means completely different, and 1 means entirely the same.

### User Friendship Network Structure Similarity Degree

In real life, if two people have more common friends, there is a higher chance that they have a closer relationship. From the perspective of topological structure, if two users have more common linking points as neighbours then these two users have high similarity in friendship network structure. Therefore the similarity of user friendship network structure can also

represent the relationship strength of users. We can calculate this degree base on Jaccard's length calculation definition:

**Definition 2:**

Define the neighbouring nodes set as $N_i$ for user $u_i$ and neighbouring nodes set $N_j$ for user $u_j$; then the friendship network structure similarity degree can be defined as $Sim\_friendship(u_i,u_j)$:

$$Sim\_friendship(u_i,u_j) = Jaccard(N_i,N_j) = \frac{|N_i \quad N_j|}{|N_i \quad N_j|} \tag{3.2}$$

The valid value range is [0,1] for $Sim\_friendship(u_i,u_j)$ where 0 means completely different and 1 means entirely the same.

### User Interaction Intensity Degree

In a normal online social network, if two people have more interaction behaviours in terms of sharing each other things, leaving comment forwarding information, etc. the more likely they are to have strong relationship strength. The relationship strength can be measured by calculating user interaction intensity. However, all the interaction behaviours only make sense when both of the users react so we have designed a special algorithm as shown below:

**Definition 3:**

We have user $u_i$ and $u_j$, and we define different interaction means with S, L, C and F to represent 'sharing', 'thumbs up', 'comment' and 'forward' respectively. When the action comes from user $u_i$ to user $u_j$ we have $S_{ij}$, $L_{ij}$, $C_{ij}$ and $F_{ij}$ while the action comes from the other way we have $S_{ji}$, $L_{ji}$, $C_{ji}$ and $F_{ji}$. So we have the user interaction intensity degree $S\_interaction(u_i,u_j)$:

$$S\_interaction(u_i,u_j) = \frac{\min(S_{ij},S_{ji}) + \min(L_{ij},L_{ji}) + \min(C_{ij},C_{ji}) + \min(F_{ij},F_{ji})}{\max(S_{ij},S_{ji}) + \max(L_{ij},L_{ji}) + \max(C_{ij},C_{ji}) + \max(F_{ij},F_{ji})} \tag{3.3}$$

The valid value range is [0,1] for $S\_interaction(u_i,u_j)$ where 0 means weak interaction i ntensity and 1 means strong interaction intensity.

### User Relationship Strength

According to the definitions mentioned above, we have acquired the similarity of the user profile $Sim\_profile(u_i,u_j)$ , the similarity of user friendship network structure $Sim\_friendship(u_i,u_j)$ and the interaction intensity degree $S\_interaction(u_i,u_j)$. Based on these factors we worked out, we can finally start working on the relationship strength by adjusting the weight to a different element and coming up with the relationship strength $RS(u_i,u_j)$ for user $u_i$ and $u_j$ :

$$RS(u_i,u_j) = \alpha Sim\_profile(u_i,u_j) + \beta Sim\_friendship(u_i,u_j) + \lambda S\_interaction(u_i,u_j) \tag{3.4}$$

The valid value range for $\alpha$ , $\beta$ and $\lambda$is [0,1] and $\alpha + \beta + \lambda = 1$. In practice, the value of $\alpha$ , $\beta$ and $\lambda$can be adjusted according to real OSN data analysis.

## *3.3.3 Experiment Analysis and Application*

### Experiment Data and Evaluation Standard

URSMF has been applied to real OSN data to calculate the relationship strength to support the friend recommendation function, and the result turns out positive. The data set comes from an OSN website called SCHOLAT [1](www. SCHOLAT.com), we have collected 5000 users' related information including user personal profile (opened), friend lists and history interaction data. According to the timeline, we separated training dataset and testing dataset.

URSMF has been used upon training dataset to calculate pairwise relationship strength between every two users. Then randomly pick up 100 target users to apply ranking process by Top-K friend recommendation method based on RS and prosecute evaluation process. The evaluation standard indicator used including precision rate P, recall rate R and composite indicator F, and definition is shown as:

$$P = \frac{|L \quad L'|}{|L|} \tag{3.5}$$

$$R = \frac{|L \quad L'|}{|L'|} \tag{3.6}$$

$$F = \frac{2PR}{P+R} \tag{3.7}$$

$L$ represents the recommended friend list and $L^{'}$ stands for the friend list that target user accepted. Ten times iteration has been performed to acquire the average value of every indicator as a result of evaluation during the rest of the experiments.

### Adjustment Strategy of Parameter α, β and λ

The parameter $\alpha$ , $\beta$ and $\lambda$ have a significant impact on the calculation of RS, and also affect the performance of the friend recommendation system. There is a straightforward and efficient strategy to adjust its value to a reasonable range. First, we respectively assign value 1 to $\alpha$ , $\beta$ or $\lambda$ while the other two as 0 and calculate the results into three groups. Then we apply top-K friend recommendation method, and the value of these three parameters can be determined according to the average value of indicator F. By using this strategy, we have tested on the dataset from SCHOLAT, and the result is shown in the table below:

Table 3.1 Experiment Result

| α | β | λ | P | R | F |
|---|---|---|---|---|---|
| 1 | 0 | 0 | 0.21 | 0.17 | 0.19 |
| 0 | 1 | 0 | 0.26 | 0.19 | 0.22 |
| 0 | 0 | 1 | 0.23 | 0.15 | 0.18 |

According to the value of indicator F we have:

$$\alpha = 0.19/(0.19+0.22+0.18) = 0.32$$
$$\beta = 0.22/(0.19+0.22+0.18) = 0.37$$
$$\lambda = 0.18/(0.19+0.22+0.18) = 0.31$$

Apply the new value of $\alpha$ , $\beta$ and $\lambda$ into another experiment and the new result for evaluation shows in table 2 shows that after adjustment every evaluation indicators have been largely improved.

Table 3.2 Experiment Result after Parameters Adjusted

| α | β | λ | P | R | F |
|---|---|---|---|---|---|
| 0.32 | 0.37 | 0.31 | 0.39 | 0.32 | 0.35 |

## Experiment Result and Analysis

To testify the effectiveness of the URSMF, we have conducted experiments to compare our algorithm with TP-URS [152] and HP-URS [156]. It is worth mentioning that TP-URS only considers user friendship network structure as the factor while HP-URS only takes interaction factor into account. The parameter value for $\alpha$ , $\beta$ and $\lambda$used in URSMF are set to be 0.32、0.37 and 0.31 respectively. We have conducted Top-2, Top-4, Top-5, Top-8 and Top-10 recommendation performance test and the result of evaluation indicators are shown below:

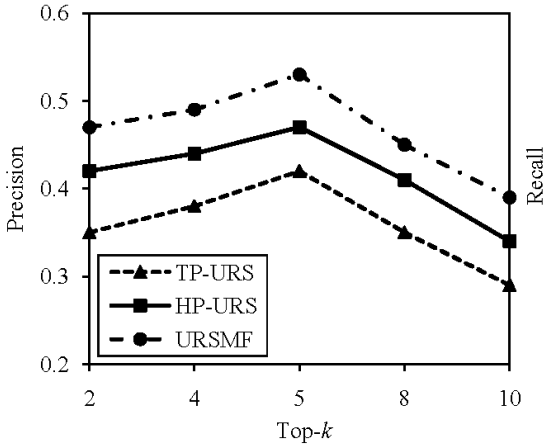



Figure 3.2 Precision Comparison

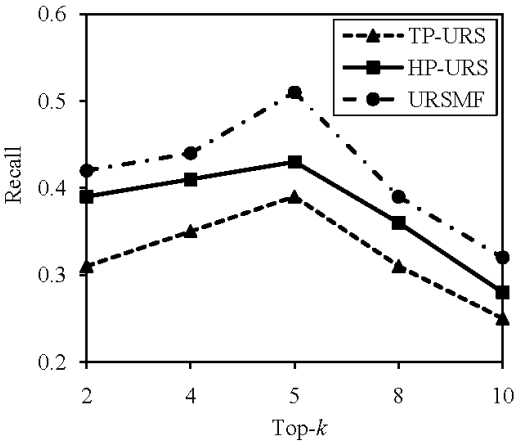



Figure 3.3 Recall Comparison

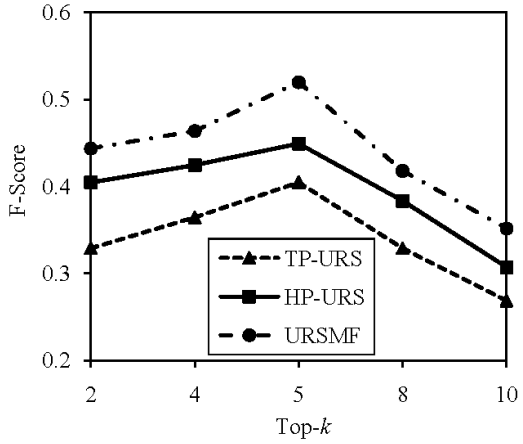



Figure 3.4 F-Score Comparison

Result figure 3.2 to 3.4 show that URSMF has the best recommendation performance. It is proven that methods that consider only one factor has its limitation on performance in terms of data loss and noises. URSMF on the other hand, considers three different factors simultaneously in a compatible way so that data noises and loses can be dramatically suppressed to obtained a more accurate relationship strength value improving the performance of friend recommendation system.

### Visualisation Application of User Relationship Strength

After successfully working out the relationship strength for our sample users in SCHOLAT we can quickly build a visualisation application to reveal the RS in a more sensible way as shown in figure 3.5. The link between user to their friends will be different based on the value of RS, the stronger the strength is, the shorter the link is and. Not only can this application provide the user with a more straightforward way to observe the relationship strength between his friends, but it can also structuralize the user friendship network regarding clarifying visualisation layout for future work.


Figure 3.5 Visualisation of Relationship Strength

## 3.4 Link Prediction Algorithm Based on Time Varied Weight

After analysing the user relationships of the academic online social network in the previous sections, we want to predict the structure and evolution of one of the most classic networks in this area of the co-authorship network.

### 3.4.1 General Procedure of Link Prediction from Network Analysis

Consider a network graph G, where $G=(V,E,W)$ that is a weighted network graph in which node $v_i \in V$, edge $(v_i,v_j) \in E, 1 \le i,j \le |V|$, $w_{ij} \in W$ represents the weight of the edge $(v_i,v_j)$. We can model the link prediction problem as a supervised classification task, where each data point corresponds to a pair of nodes in the network. That is, given some training intervals ($[t_0,t_0']$) and their link information, we train a model that can be used to predict the links will be formed in the test interval ($[t_1,t_1']$). To train the learning model, we need two aspects of information:

(1) Labelling information. We can use the link information of the training interval ($[t_0,t_0']$) to get the label. A more formal representation is as follows: for graph G, the nodes $v_i,v_j \in V$ $y^{\langle v_i,v_j\rangle}$ represents the label of the data point $\langle v_i,v_j\rangle$. We assume that the interaction between the two nodes is symmetric, thus $\langle v_i,v_j\rangle$ and $\langle v_j,v_i\rangle$ represents the same data point, $y^{\langle v_i,v_j\rangle}=y^{\langle v_j,v_i\rangle}$.

$$y^{\langle v_i,v_j\rangle}=\begin{cases}+1, if \langle v_i,v_j\rangle \in E\\ -1, if \langle v_i,v_j\rangle \notin E\end{cases}$$

By using the above labelling method for the training dataset, we create a classification model through which we can predict the unknown tags of the node pair $\langle v_i,v_j\rangle$ in graph G$[t_1,t_1']$, where $(v_i,v_j) \notin E$.

(2) Feature information. We turn graphs into features that are fit for machine learning. For each pair of nodes in the graph, we can calculate various properties of the graph, such as the shortest path, common neighbours. Through the values of these attributes, we can calculate the similarity between two nodes. The higher the similarity of two nodes is, the greater the possibility of a connection between them will form, and vice versa. Many attributes can be used to construct the feature vector, but the validity of the attributes depends on the characteristic of the networks to be predicted to some extent. We compute the values of these attributes into a

feature vector, $f_{ij}$ representing the feature vector of the node pair $(v_i, v_j)$. F denotes the complete set of the feature vectors: we have $F = \{f_{ij} \mid (v_i, v_j) \in V \times V\}$. Directed graphs, undirected graphs, and whether or not self-joins are allowed will cause the size of F to be different. In the network of co-authorship, both self-linking and directed edges are meaningless, so $|F| = \frac{|V| \cdot |V-1|}{2}$.

In general, we predict its label by learning the information contained in each feature vector $f_{ij}$, $y^{\langle v_i, v_j \rangle}$ is true if and only if a new connected link $(v_i, v_j)$ appears. We assume that the labels of some of the data points are already known. Using the set of labels $Y_r$ representing these labels, we train the supervised learning algorithm by using the known feature vector sets and the corresponding set of labels and the use the learned model to predict the corresponding set of labels for the remaining data points $Y_s$. So, the training set is defined as $F_r = \{f_{ij} \mid y_{ij} \in Y_r\}$ the test set is defined as $F_s = F - F_r$. The basic framework of link prediction that is constructed from the network structure is as follows:

1. Create a network.
2. Extract the node pair features from the network and construct the feature vectors.
3. Train the model by supervised learning algorithms using the captured feature vectors and labels' information.
4. Predict new links between nodes.

We first turn the network graph into the feature vectors according to the constructed network and then use the supervised learning algorithm to get a predicting model, which can predict the new link. In the above algorithm, feature vector selection is most important. We will discuss the issue as below.

### *3.4.2 Topological Characteristics Based on Time-Varying Weights*

Choosing an appropriate feature set is critical for any machine learning algorithm. For link prediction problems, each instance corresponds to a pair of nodes, and the label represents the

state of the current node pair. Therefore, the selected features should represent the similarity of the node pair to some extent. Now we introduce the feature vectors we have chosen.

In the existing research work, the graph-based topological structure is the most widely used and most important feature. In fact, many link prediction works only focus on topological characteristics. The biggest advantage of topology characteristics is that it can be used in any field in general, and can be calculated without any other knowledge. However, these approaches only consider the binary relations among nodes and neglect the weight of links between them.

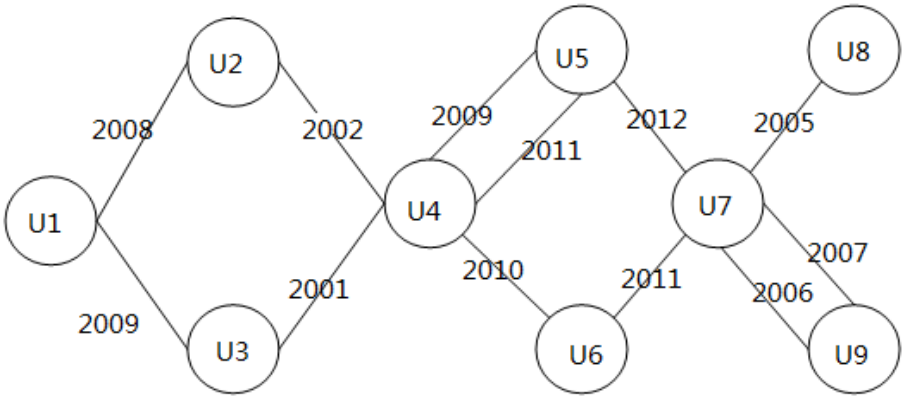



Figure 3.6 Co-authorship Network with time stamp

In the real cooperative network, we believe that the weight of the links between the two researchers is closely related to the number of papers they have cooperated and the time when they cooperated. It is believed that the longer the time point of cooperation between the two researchers is, the weaker the impact will be on future cooperation. In Figure 3.6, we consider the possibility of cooperation between U4 and U1 in the future less than the possibility of U4 and U7. Therefore, we propose a link-weighting based on time variation:

$$w^{t}(x,y)=\frac{1}{(t^{'}-t)+1}\alpha$$

Here, $w^{t}(x,y)$ represents a weight of cooperation occurs at time t at the nodes x, y. $t^{'}$ denotes the current time, t denotes the initial time, $\frac{1}{(t^{'}-t)+1}$ denotes the decay factor, and $\alpha$

denotes the basic weight, equaling to 1. The total weight of the nodes x, y is calculated as follows, and T represents the time point set when all the nodes x, y cooperate

$$w_T(x, y) = \sum_{t \in T} w^t$$

Give a simple undirected network G (V, E), where each edge $\mathbf{e_t} =< \mathbf{u}, \mathbf{v} > \in \mathbf{E}$ represents an interaction between u and v at a particular time t. For each pair of nodes, x, y ∈ V, we assign a score, $\mathbf{S_{xy}}$ according to a given similarity measure. Higher score means higher similarity between these two nodes, and vice versa. The link between the two nodes predicts that there will be a high degree of similarity in the future of the country's interaction. Thus, the appropriate similarity measure method of the computer is part of the most critical link prediction.

The topological based approach is the most widespread one. It also presents good performance and is easy to implement. As introduced in the last chapter there are some more famous topological similarity methods. Just to make it clear for comparison, I will briefly present them again below:

**Common Neighbours (CN) [106]** In common sense, two nodes, x and y, are more likely to form a link in the future if they have many common neighbours. Let $\Gamma(x)$ denote the set of neighbours of node x. The simplest measure of the neighbourhood overlap is the directed count:

$$Common\_neighbour(x, y) = |\Gamma(x) \cap \Gamma(y)|$$

**Jaccard Coefficient(JC)[130]** It measures the probability of two nodes being linked by calculating the ratio between the number of neighbours they share and the total number of distinct neighbours they have. It is defined as:

$$Jaccard\_coefficient(x, y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|}$$

**Adamic-Adar (AA).** Adamic and Adar [2] proposed a distance measure to decide when two people's personal home pages are strongly 'related'. In particular, they computed features of

the pages and defined the similarity between two pages x, y as follows $\sum_{\mathbf{z}} \frac{\mathbf{1}}{\mathbf{log(frequency(z))}}$ , where z is a feature shared by pages x, y. This refines the simple counting of common features by weighting rarer features more heavily. The similarity between nodes x and y can be computed by Equation 3:

$$Adamic\_Adar(x,y) = \sum_{z \in \Gamma(x) \ \ \Gamma(y)} \frac{1}{\log|\Gamma(z)|}$$

**Preferential Attachment (PA).** The PA measure assumes that the probability of a future link between two nodes is proportional to their degree. Barabasi et al. [13] is confirmed in the co-network node, this probability is the number of collaborators related to the product.Hence, it is defined as:

$$Preferential\_Attachement(x,y) = \left|\Gamma(x)\right| \times \left|\Gamma(y)\right|$$

As we can see that the similarity metrics mentioned above only consider the binary relations among nodes. They ignore the time dimension and frequency of link occurrences. However, in the real world, a link established at different time interval will have a very different effect on relationship development, in this case, the future co-authorship possibilities. Thus, time should be consider a varied weights in the calculation of the similarity in order to improve the accuracy of prediction. In the co-authorship network, we should not only consider the weight of the links about the number of co-authorized papers but also the coauthored time of those papers. Papers that co-authorized in a much earlier time interval should have lower likelihood and weight to predict the future collaboration relationship comparing to those that created closer to present time.

Based on the idea proposed above, we have built time-varied weighted CN, the time-varied weighted JC, the time-varied weighted AA and time-varied weighted PA denoted by tw-CN, tw-JC, tw-AA and tw-PA respectively as shown below:

$$s_{xy}^{tw-CN} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} w^t(x,z) + w^t(y,z)$$

$$s_{xy}^{tw-JC} = \frac{\sum_{Z\in\Gamma(x)\cap\Gamma(y)} w^t(x,z) + w^t(y,z)}{\sum_{z_x\in\Gamma(x)} w^t(x,z_x) + \sum_{z_y\in\Gamma(y)} w^t(x,z_y)}$$

$$s_{xy}^{tw-AA} = \sum_{z\in\Gamma(x)\cap\Gamma(y)} \frac{1}{\log \sum_{z\in\Gamma(x)\cap\Gamma(y)} w^t(x,z) + w^t(y,z)}$$

$$s_{xy}^{tw-PA} = \sum_{z_x\in\Gamma(x)} w^t(x,z_x) \times \sum_{z_x\Gamma(y)} w^t(y,z_y)$$

Here, $w^t(x,z)$ denotes the time-varied weight of link between nodes x and y, $\mathbf{t} \in \{\mathbf{t_1}, \mathbf{t_2}, \mathbf{t_3} \ldots \mathbf{t_n}\}$is a time series when x and y coauthored a paper. It is common sense that the effect of the link established at time $\mathbf{t_{n-1}}$ is lower than that of the link established at time $\mathbf{t_n}$.

Although the topology-based feature has a good effect on the link prediction, it also has a fatal flaw. From the definition of the above-mentioned topological features, we can see that the collaborators who anticipate the cooperation must have a common partner in the cooperative network. But in real life, many new collaborators do not have a co-author before. Therefore, it is not sufficient to predict the co-author relationship by only using the information provided by the topological feature, so we propose to add the location feature and semantic feature.

### *3.4.3 Semantic and Location Similarity Features*

Although topological features are general, they can be adapted to all social networks, but for the co-authorship network, semantic information from the two authors can help us to analyse the similarity among authors better. The higher the similarity of semantic information between the two authors, the closer the research interests of the two authors are, and more likely they will cooperate later. Here we calculate the similarity of the two authors by the similarity of keywords contained in the titles of the author published papers.

First of all, we need to extract keywords in the title of the paper. We found that there are many meaningless words and symbols in the title, for example, articles and conjunctions “a”, “the”, “and” and punctuation marks ":", "?", so we create a stop word phrase stopWords [] =

{ "a", "an", ..., "the", ...} and punctuation phrases speWords [] = { ":", "?", ...} matches each word in the title, removing words that do not relate to the semantics. After the pause words are processed, each paper is represented by the word vector p = {t1,t2,...tn}, and the vector of all the papers of an author makes up a vector AuthorTerm ={ t1,t2,...tn,..}. Next, we calculate the author's semantic similarity by the authorTerm vector. Hasan[6], just introduced a keyword matching degree measure that is such a simple definition and easy calculating method, compared with the only use of topological features, which give better and forecasting results. Wohlfarth[149], introduces two simple semantic information-based metrics: keyword matching number and common event number. Sachan[119], represents the author's research interest by using two semantic information, the author's title and abstract information. Bartal[14], also mentioned using social network analysis method to extract simple semantic information from the title of the paper to predict.

However, when calculating their semantic similarity, they simply compute the intersection of two sets of authorTerm or use Jaccard's coefficient. This method treats all the words equally, ignoring the importance of different words. Here we use the vector space model (VSM) proposed by Salton[120], to calculate the similarity. Each term in the vector space model has a corresponding weight, calculated by TF-IDF. TF represents the number of occurrences of the term in the AuthorTerm set, and the more the number of occurrences; the more likely it is to represent the author's research interest. IDF indicates the frequency of the term in all collections, the lower the frequency, the more it can distinguish the author's research interests. The weight of each term is equal to the product of the two values. Finally, we calculate the similarity of two vectors by the cosine similarity.

In real life, a person's location information can reflect his interests to a certain extent. Two strangers in the same location are more likely to become friends. Based on this fact, we propose to use the nearest location information to evaluate the similarity of the two scholars. In the network of collaborators, we use scholars' recent conferences to locate scholars. For example, if scholar A participates in SIGKDD and CHI meetings and scholar B participates in SIGMOD, SIGKDD and CSCW meetings, then the position information of scholar A and scholar B is denoted as $p_A$ = {SIGKDD, CHI}, $p_B$ = {SIGMOD, SIGKDD, CSCW} respectively. Two scholars with similar location information, indicating that they focus on the same research areas, have similar research interests and even have the opportunity to communicate face to face, which

will enhance their future cooperation possibilities. In the following, we introduce the step of calculating the similarity of position. First, we get all the meetings that scholars attend so that we can get the position information of the scholar. Then the Jaccard's coefficient is used to compare the similarity of two scholars' position information.

### 3.4.4 Dataset and Experiment

#### Dataset

In the experiment, four classic link prediction algorithms has been used to compare with the proposed algorithm in this chapter. A bibliographic dataset: DBLP (http://dblp.uni-trier.de/xml/) of information about different research publications in the field of computer science has been used to evaluate the examined algorithms. It lists the author's research publications by years, including international journals and conferences. The DBLP dataset is an XML file that can be downloaded from the DLBP website. Of which, dblp.xml is the data set we need; dblp.dtd is the format description file. To use the DBLP dataset, we first parse dblp.xml, which contains multiple types of bibliography, including article, proceedings, inproceedings, book, incollection, PhD thesis and master thesis.

Five years of data, from 2006 to 2010 of DBLP has been used. First 4 years were used as training (ET) and the last year as testing (EP). The datasets have been summarised in Table 3.3.

TABLE 3.3. THE DETAIL OF DBLP DATASET FROM 2006-2010

| number / year | Papers count | Authors count |
|---|---|---|
| 2006 | 82,739 | 131,703 |
| 2007 | 89,868 | 145,208 |
| 2008 | 95,485 | 155,960 |
| 2009 | 103,772 | 174,828 |
| 2010 | 103,504 | 175,831 |

The training set ET is treated as a known message, and the probe set EP is used for testing. It is not allowed to use any information in the probe set to predict. Obviously, ET p = 0. For each pair of non-directional connection nodes with at least one neighbour in ET, we generate the similarity of a pair of nodes according to the weight of the ET in the ET, and then calculate the final performance of each algorithm according to whether the link appears in the EP.

## Evaluation Standard

Before jumping into the experiments, let us explain some performance metrics for predictor evaluation. The predictor can predict positive (p) or negative (n) for the corresponding tag. In the positive case, if p is correct, the prediction is true positive (TP), otherwise false positive (FP). On the contrary, in the negative case, if the correct or false negative (FN) error, the forecast can be true negative (TN). Now we can define the measure recall as the proportion of TP predictions in all real tags. This recall will let us know how good it is to predict the future. It may also be useful to define the measure accuracy as the proportion of TP predictions for all positive predictions. Accuracy will help determine how p fit the entire data.

$$\textbf{precision} = \frac{|\boldsymbol{TP}|}{|\boldsymbol{TP}| + |\boldsymbol{FP}|}, \qquad \textbf{recall} = \frac{|\boldsymbol{TP}|}{|\boldsymbol{TP}| + |\boldsymbol{FN}|}$$

ROC curve takes false positive rate as the X axis, and true positive rate as the Y axis and both axis lengths are 1, thus forming a square two-dimensional ROC space. In this space, every data point (FPR, TPR) is plotted, and then connected them with the two endpoints (0,0) and (1,1), forming a ROC curve. Figure 10 shows an example of ROC curve.

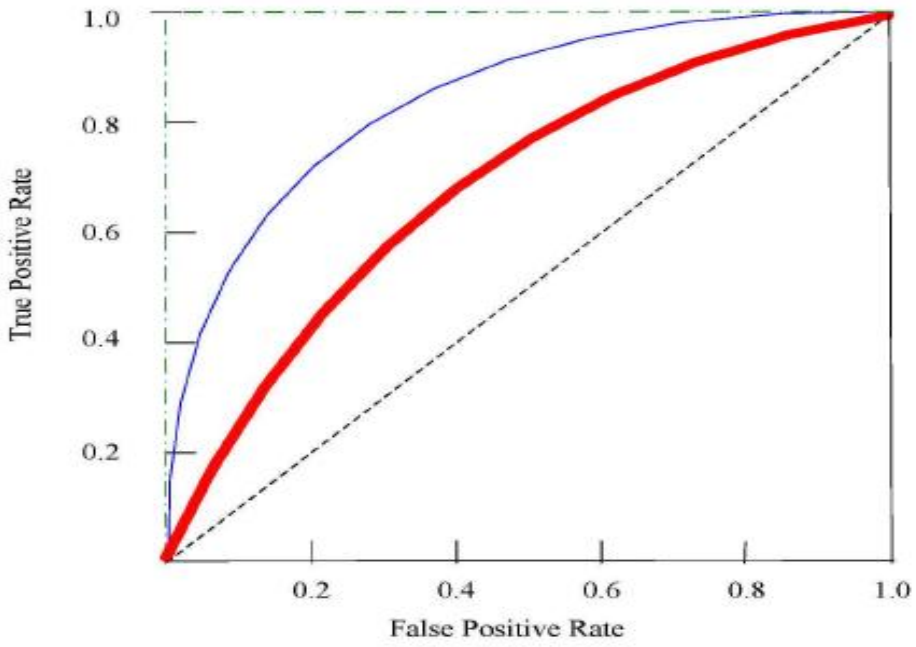



Figure 3.7: Example of AUC

Theoretically, the ROC curve should be a diagonal line from the origin (0,0) to the upper right point (1,1), that is the dotted line in the figure above if the experiment randomly predicts. The closer to the upper left corner the ROC curve is, the better the prediction performance of the algorithm. The performance of the predictor represented by the thin solid line ROC curve is better than that of the thick solid ROC curve. AUC is the area under the ROC curve, and the greater the value of AUC, the predictive performance of the predictor, is better. Corresponding to the ROC curve, when the predictor randomly predicts, the AUC value is 0.5 that is the area under the dotted line. The best ROC curve is the dot-dash line, and corresponding AUC value is 1. In general, the predicted ROC curve is located above the dashed line, and the farther away from the dotted line shows that, the better the prediction effect is. So usually AUC value ranges from 0.5 to 1.

Some link prediction methods have relatively high precisions for their top-K predictions when K is small because they are good at predicting the "easy" links. However, with the increase K, the accuracy will drop sharply. It seems that there is an additional evaluation index that can be used to measure the accuracy of the classifier without considering any truncation points. For this reason, we use a standard metric AUC, the area under the receiver operating characteristic (ROC) [159] to quantify the accuracy of the prediction algorithm. These curves can achieve true positive rates (TP) at all false positive rates (FP) by changing the threshold of the probability threshold or fraction. AUC is an important method of performance measurement, traditionally used for unbalanced classification.

## Experimental Results

From the previous analysis, we know that the accuracy of the feature's values between nodes directly affects the accuracy of link prediction, and that depends on network saturation and convergence. We believe that the network will become more and more saturated. We tested the effect of time interval on the unsupervised method since 2000, as shown in Figure 3.8:

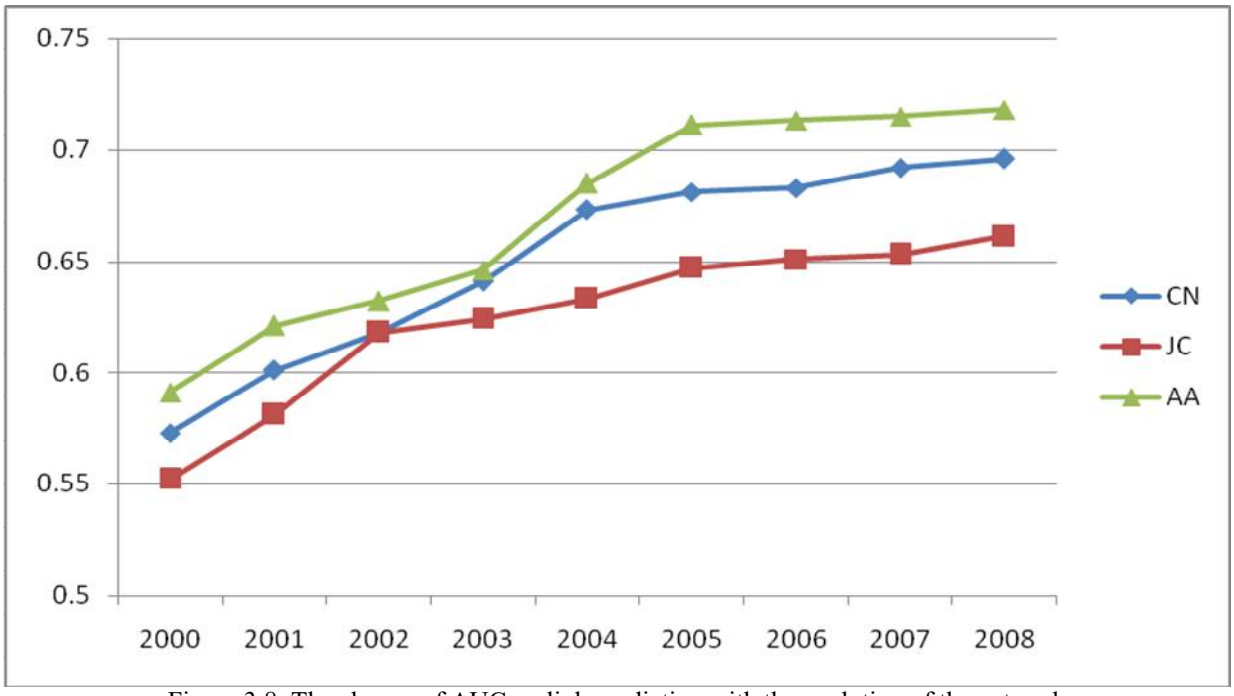


Figure 3.8: The change of AUC on link prediction with the evolution of the network

As the time interval increases, the predictive ability gradually increases. Among them, from 2000 to 2005, with the increase of time, the predicting ability is most progressive. From 2005 to 2008, the growth rate of predicting ability is slowing down. Considering the balance between computational efficiency and the ability to predict, in this experiment we used literature records at the time span from 2001 to 2006.

In the process of data analysis, we found that the cooperation between many authors only occurred once, that is, the weight between the two authors is only 1. For example, the two authors cooperate with the third author only once, there is no other cooperative paper between them, we believe that this is accidental in their cooperation, so the information for the future trend prediction is not helpful, and for this reason, we will remove the edges with weight 1. To construct the classifier, the data of the last six years (2000-2005) is selected as the training set, and the data of the last six years (2001-2006) is chosen as the test set. And extracts the feature values of the node pair from the first five years, and fetches the corresponding label from the last year. In the training set and test set, we found that some authors were active only in the previous five years and did not publish papers in the following year. On the contrary, some authors only published papers in the last year. This leads to a significant number of feature vectors and labels

cannot match, in order to avoid this situation, we only leave the data set is active for two time periods of the author.

TABLE 3.4 THE AUC VALUES FOR ALGORITHMS

| | AUC |
|---|---|
| PA | 0.5175 |
| JC | 0.5815 |
| CN | 0.6546 |
| AA | 0.7556 |
| tw-PA | 0.5531 |
| tw-JC | 0.6012 |
| tw-CN | 0.6901 |
| tw-AA | 0.7641 |

Table 3.4 contains the results of these experiments. We note that the best performance of the TW AA algorithm is AUC 0.7641, and all the mixing methods can achieve better results than the traditional method using time-varying weights. For example, the TW algorithm has a higher AUC than the CN algorithm; the TW JC algorithm has a higher AUC than the JC algorithm. Therefore, we believe that the time-based link prediction algorithm can also be applied to other methods.

## 3.5 Chapter Conclusions

Relationship Strength has attracted more and more attention recently, and it is paramount to analyse user relationship in the online social network. Most of the current methods only considered a single type of factor that is related to the relationship strength value thus the result is often limited and lack of accuracy. In this chapter, I proposed a method that combines three classic types of factors that are related to relationship strength. The major innovation of URSMF is that it builds a framework that can scientifically put weights to different factors. This allows the RS algorithm the ability to calculate the relationship value based on three factors simultaneously without data overload or data conflicts to occur. Finally, the experiment and evaluation results show that this method can improve the performance of friend recommendation system in online social networking sites. This chapter also introduced the link prediction

definition and algorithms in detail. And the due to the limitation in traditional link prediction for this particular domain, I proposed a new algorithm based on time-varied weight. The experimental results show that the prediction performance of our algorithm is superior to that of the traditional link prediction algorithm. There are many issues in future research about this problem. An important future work would be evaluating the generality of the method by testing it on various kinds of networks and comparing the predictive accuracy.

# Chapter 4

## Friend Recommendation Based On User-Interest-Tag in Online Social Network

*With the rapid growth of the number of users in the social network, how to recommend friends with same interests has become the focus of the current research. Therefore, I propose a hybrid friend recommendation algorithm based on 'user–item - tag graph ' and 'user personal interests'. Firstly, it calculates similarities between users by mass diffusion method in the tripartite graph. Secondly, it introduces the relationship between users and tag graph of users and detects communities in the tag graphs of users for representing topics of users' interests. Then the similarities between users are measured by Kullback-Leibler divergence according to their topics distributions. Finally, two kinds of similarities are integrated for user recommendation by the harmonic mean method. The experimental results on datasets of Delicious and Last.fm demonstrate that our method can effectively improve the accuracy of Top-N recommendation in terms of precision, recall and F1.*

## 4.1 Introduction

With the development of Web 2.0, the online social network (OSN) and Social Tagging Systems (STS) have been flourishing, such as Facebook, Twitter, Last.fm and Sina Weibo websites. Among them, how to recommend friends with the same hobbies has become one of the social network recommended research hotspots. At present, in most of the social networking sites, friend-recommendation algorithm is divided into two categories: one is to use the user's common interests to recommend friends, mainly based on users' information with the content-based collaborative recommendation; another one is based on the users' social graphs to recommend, to friends of friends and followees of followees. The two methods have limitations. The effect of former method will be affected when the user information is not complete, especially the newly registered users. At the same time, because someone who has many common friends with you probably already known to you, the latter method is useless to expand the circle of friends. And as the massive amounts of data grow, it is hard to capture the evolution of the users' interests by content.

As one of the important technologies of Web2.0, social tag [12, 88, 91, 169] allows users freely to label a variety of resources according to their own needs and understand. At present, a large number of tag-based Web sites have been the user's favour and achieved great success, such as YouTube, Delicious, Last.fm and Douban. In the Last.Fm music website, the song "Roll in the Deep" have been evaluated by 307,915 people, and 61 tags were marked, of which 5091 people played the "amazing voice" tag, and 5068 people hit the "best of 2011" tag.

In the socialisation tag, the tags and the frequency of usage of the tags can effectively reflect the user's interest preference and the user's preference for a certain kind of resource, so the tag is the important bridge between the user and the resources. Guy et al. [56] proposed a novel framework for the recommendation, which uses user's tag information to expand the user's interest and hobbies, so as to improve the recommendation accuracy of the item. Agarwal et al. [3] designed an adaptive similarity computation to learn individual preferences toward a particular set of attributes and incorporate effective missing data prediction algorithm for friend recommendation. Manca et al. [98] used diffusion kernels on the tag network to measure the similarity between pairs of tags and retrieved top k people sharing the most similar tags. Zhou et al. [170] built a User Recommendation (UserRec) framework for interest-based user recommendation.

However, these methods are inadequate to capture the user-item relationship. Ming-Sheng et al. [124] proposed a collaborative filtering algorithm with diffusion-based similarity on Tripartite Graphs which integrated user preferences of both collected items and used tags for the recommendation. But DBTCF cannot model users' personal interests via tags at a high level. Therefore, in this work, to capture the user-item-tag relationship, a hybrid collaborative filtering recommendation algorithm by combining the diffusion on the user-item-tag graph and user personal interest model (UITGCF) for friend recommendations is provided.

## 4.2 Related Work

### *4.2.1 DTGCF*

The DTGCF [124] reduces this user-item-tag tripartite graph into two pair correlations: user-object and user-tag, and then utilises user–item topology and user-tag topology with resource allocation to get the similarities among users. In the user-item bipartite graph, the resource-allocation process consists of two steps: first from users to items, then back to users. And in the user-tag bipartite graph, the first step is from users to tags.

The user u distributes the resource equally to all the items he/she has collected. So, the resource that item gets from user $u_i$:

$$r_{iu} = \frac{\alpha_{ui}}{k(u)} \tag{4.1}$$

where k(u) is the degree of u in the user-item bipartite graph, and $a_{ui}$ is 1, which means that the user u has purchased or commented on the item i. Similarly, the resource t obtained from the user u is $r_{ut}$:

$$r_{tu} = \frac{\alpha'_{ut}}{k'(u)} \tag{4.2}$$

where $k'(u)$ is the degree of item i in the user-tag bipartite graph. $a'_{ut}$ is 1, which means that the user u uses tag t.

Each item distributes its resource equally to its neighbouring users with equal probability. Then the resources flow back to users. Thus, the item-based similarity between u and v with the target user u is:

$$s_{uv} = \frac{1}{k(v)} \sum_{i=1}^{n} \frac{\alpha_{ui} \alpha_{vi}}{k(i)} \tag{4.3}$$

where k(i) is the degree of item i in the user-object bipartite graph.

Analogously, each tag redistributes the received resource to all its neighbouring users. Thus, the tag-based similarity between user u and v is:

$$s'_{uv} = \frac{1}{k'(v)} \sum_{t=1}^{r} \frac{\alpha'_{ut} \alpha'_{vt}}{k'(t)} \tag{4.4}$$

where k′(t) are the degrees of tag t in the user-tag bipartite graph.

Finally, the above two diffusion-based similarities are integrated by the simplest way to get better recommendations. The resource allocation process of the DTGCF is formulated as:

$$sim_{DTG}(u,v) = \lambda s_{uv} + (1-\lambda) s'_{uv} \tag{4.5}$$

### *4.2.2 Modeling Base on User-Interest*

At present, there is no clear definition of user interest modelling. Most of the literature [132, 148, 164] thinks that user interest modelling is the collection and analysis of user's personal information, behaviour and other information to form the user's personal interest model. In the social labelling system, users are free to use the label to label the project, while the different labels also express the user's own diverse interests. So you can use the label to the user interest modelling and applied to the Collaborative Recommendation, so as to improve the recommended effect [68, 90, 91]. However, most of the tag-based collaborative recommendation algorithms are mainly two kinds; one is to use the number of users or items with co-occurrence tags to calculate the similarity of users or items [57, 160] but will ignore the label itself has the semantics and tags.

And the other is to use LDA and other models to model the interest of users. But when the number of labels is sparse, it will affect the efficiency of the model. Therefore, Zhou et al. [170] proposed a novel user recommendation framework Recommendation, UserRec, the user label map for community discovery to build user interest model, the use of KL distance to calculate the similarity between users.

Although the UserRec framework can model user interest based on tags, it lacks user and project considerations, whereas the DTGCF algorithm takes into account the relationship between user-item and user-tag, ignoring the co-occurrence and semantic problems of tags, resulting in different degrees of information loss. To solve the above problems, we first calculate the similarity between users by using the "user-item-tag" tripartite graph recommendation algorithm. At the same time, we borrow UserRec user interest modelling method and increase the tag processing. The user's interest tags are constructed to explore the user's interest, and the KL distance is used to calculate the similarity between the users according to the distribution of user's topic. Finally, the two groups of results are merged using the harmonic mean to get the final user similarity.

## 4.3 UITGCF Method

### *4.3.1 Tag Preprocessing*

In collaborative tagging systems, users can freely annotate tags to different items and have no need of specific skills to participate. At the same time, tags can contain abstracted content of items with personalised preferences. So, tags have been applied in different systems, such as Delicious, Last.Fm, YouTube and Movielen, etc. However, using the raw tags is not very meaningful, as tags are words or phrases that users can freely add. There are spelling errors and meaningless characters, such as //at, ???##, etc. Thus, it is necessary to preprocess tags. Firstly, we preprocess the tags with symbol regularisation. If a tag matches regular expression R = [a-z, A-Z]{2,}, it can be reserved. Furthermore, stemming, stop words removal are applied to all of these data sets.

### *4.3.2 Tag-Based User Interest Modelling*

In social tagging systems, user's interests can be reflected in their tagging activities. And Tag co-occurrence can be used to characterise and capture user's interests. Hence, we can utilise

the tag co-occurrence network to construct users' interest model [88]. The first step is to construct an undirected weighted tag co-occurrence graph for each user. Let UG = (T, E, W) be an undirected weighted users' tag-graph, where T is a set of tags. E represents a set of edges. The edge between two tags reflects that the user annotates the same resource with two different tags. W is the set of weight of edges, w (tk,tm) ∈ W represents the number of times that tag k occurred together with tag m within the same item.

In the second step, we adopt Louvain method to community detection in networks [5]. The Louvain method is a simple, efficient and easy-to-implement method for identifying communities in large networks and finds high modularity partitions of large networks in a short time. After community detection, tag co-occurrence graph of a user would be divided into several communities. Each community, which is represented by a set of tags used by the user, indicates one topic of the user. The set of topics of all the users is named C here. Given a target user u, we define all the topics of a user u, UC(u), as:

$$UC(u) = \{c_m^u \,|\, c_m^u \ \ is\ \ a\ \ topic\ \ of\ \ the\ \ user\ \ u,\ \ c_m^u \in C\}$$

where $c_m^u$ is a topic of the user u. $c_m^u$ ={tk | tk is a tag belonging to the corresponding community of topic $c_m^u$ by the Louvain algorithm, tk∈T},

Figure 4.1 demonstrates the communities of user 243 from Last.fm dataset. The size of a node is proportional to the number of individuals in the corresponding community. After the proposed two steps, the user's interest can be represented with several topics, which consist of one or more tags. Table 4.1 shows the topics of

Table 4.1 Topics of User 243

| U243 | tag |
|---|---|
| Community 1 | 66,67,69,1,4,65,68,5,3 |
| Community 2 | 89,30,84,85,88,70,76,73,8,6,87 |
| Community 3 | 35,150,159,151,39,2,14,7,146,152,148,149,160,241,33,103,22 |

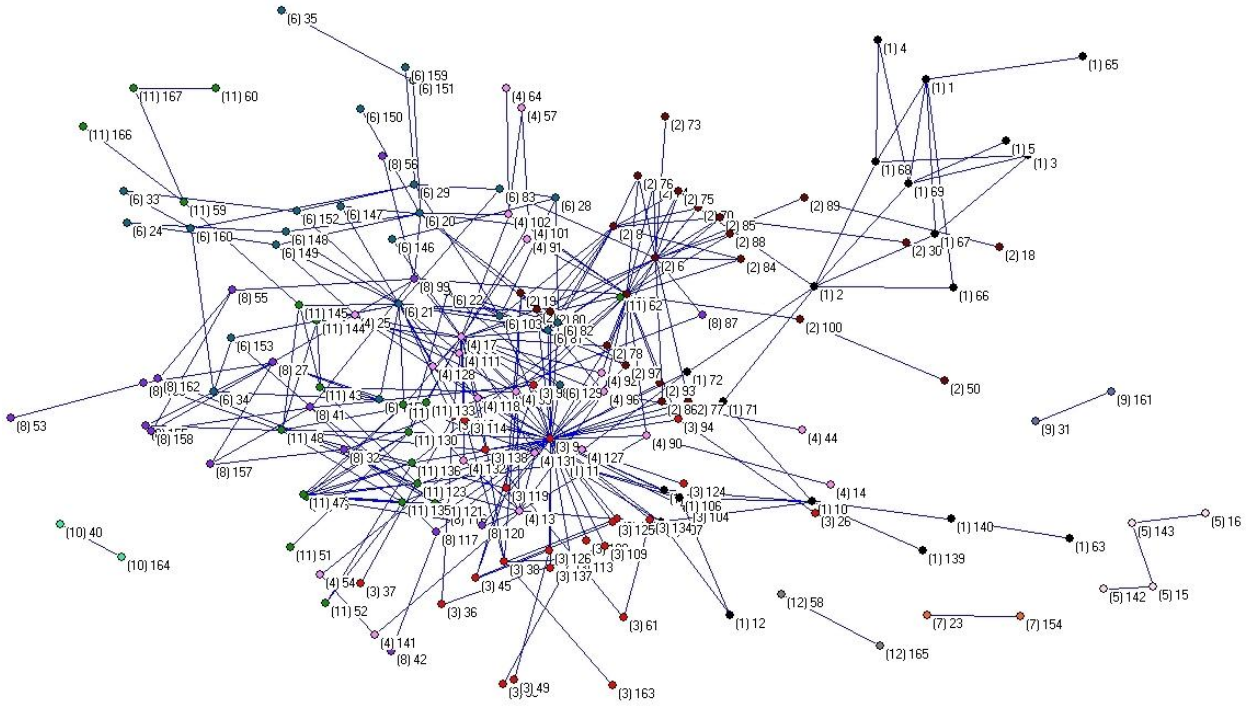


Figure 4.1 the Communities of user 243 from Last.fm dataset

### *4.3.3 Integrated User Similarity*

Based on user's interest model, we further propose hybrid user recommendation algorithm by integrating the diffusion on user-item-tag graph and users' personal interests in three-stage. First, the topics of each user can be represented by a discrete random variable. We introduce a probability value to measure the impact of each topic to a user. The impact of a topic $c_m^u$ to a user u is denoted as:

$$TN(u, c_m^u) = \sum_{t_k \in UC(u)} N(t_k, u, c_m^u) \tag{4.6}$$

Where N(tk, u, cum)is the number of times tag, this used by user u, where tk∈S(u) and tk∈cum. After defining the impact of a topic to a user, we define the total impacts of all the topics on a user in Eq. (4.6).

$$\Pr(u_i, c_m^{u_i}) = \frac{TN(u_i, c_m^{u_i})}{TTN(u_i)} \tag{4.7}$$

Where $TTN(u) = \sum_{c_m^u \in UC(u)} TN(u, c_m^u)$ represents the numbers of times that tags used by the user in all topics.

Then, according to users' topic distributions, the similarity between two users can be calculated by Kullback-Leibler divergence (KL-divergence) that is a measure between two probability distributions. The similarity between a pair of users can be measured by Eq (4.7).

$$KL(u|v) = \sum_{c_m^u \in UC(u)} \Pr(u, c_m^u) \log \frac{\Pr(u, c_m^u)}{\Pr(v, c_m^v)} \tag{4.8}$$

Finally, because the similarity of users' topic distributions lacks considering user-item relation, we need to integrate the similarity with another users' similarity from DTGCF method. So, we adapt to use the harmonic mean to integrate the two different user similarities, that is, to define the final user similarity:

$$sim(u,v) = \begin{cases} \dfrac{2 \cdot sim_{DTG}(u,v) \cdot sim_{UIG}(u,v)}{sim_{DTG}(u,v) + sim_{UIG}(u,v)} & \text{sim}_{UIG}(u,v) \neq 0 \\ sim_{DTG}(u,v) & \text{sim}_{UIG}(u,v) = 0 \end{cases} \tag{4.9}$$

Where $S_{uv}^{*}$ is the user similarity from DTGCF method. To facilitate the calculation, we normalise the $KL(u|v)$ by Eq (4.9):

$$sim_{UIG}(u, v) = \exp(-KL(u|v)) \tag{4.10}$$

We then sort all users who are not the friends of user u in the descending order of their similarity scores, and the top-k users will be recommended to u.

## 4.4 Experiments

In this section, we report the results obtained from experiments conducted to compare the DUITCF with existing recommendation methods by using well-known benchmark data sets hetrec2011-delicious-2k [44] and hetrec2011-last.fm-2k [82].

### *4.4.1 Dataset and Description*

We use two real life data sets, Delicious and last.fm, to evaluate the effectiveness of our recommendation algorithm. Two data sets are released in the framework of HetRec 2011 [133].Table 4.2 shows the description of two datasets.

Table 4.2 Original Dataset Description

| Dataset | users | items | tags | user-user | user-tag-item | user-items |
|---|---|---|---|---|---|---|
| Delicous | 1857 | 69226 | 53388 | 15328 | 437593 | 104799 |
| Last.fm | 1892 | 17362 | 11946 | 25434 | 186479 | 92834 |

### *4.4.2 Evaluation Metrics*

We employ two widely used accuracy metrics, Precision, Recall and F1, to investigate the prediction quality of our proposed UITGCF model in comparison with other counterparts. Precision, Recall and F1 are defined as:

$$\text{Precision} = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |R(u)|} \tag{4.11}$$

$$\text{Recall} = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |T(u)|} \tag{4.12}$$

$$\text{F1} = \frac{2*\text{Precision}*\text{Recall}}{\text{Precision}+\text{Recall}} \tag{4.13}$$

Where R(u) represents the recommendation list for user u. T(u) is the set of items that u has collected in the test set. Precision and Recall are standard metrics in Top-N recommendation (N ranging over 5 to 100 in our experiment). As a single value, F1 combines both the precision and recall measures with equal weight.

### 4.4.3 Comparisons

To evaluate the performance of our proposed UITGCF, we will compare the following different methods described in this paper:

1) UserRec [170]: it is an efficient two-phase UserRec framework for users' interest modelling and interest-based user recommendation.
2) CosTag [3]: it uses Jaccard-similarity to measure the similarity between two users based on tags.
3) DTGCF [124]: it calculates two of kinds of similarities between users by using a diffusion-based process for the recommendation.

### 4.4.4 Experiment Results

#### The result of the tag-graph based community detection method

We generate an undirected weighted tag-graph for each user from Delicious and Last.fm dataset, then adopt Louvain algorithm to detect communities on user's tag-graph. The results are shown in Table 3.

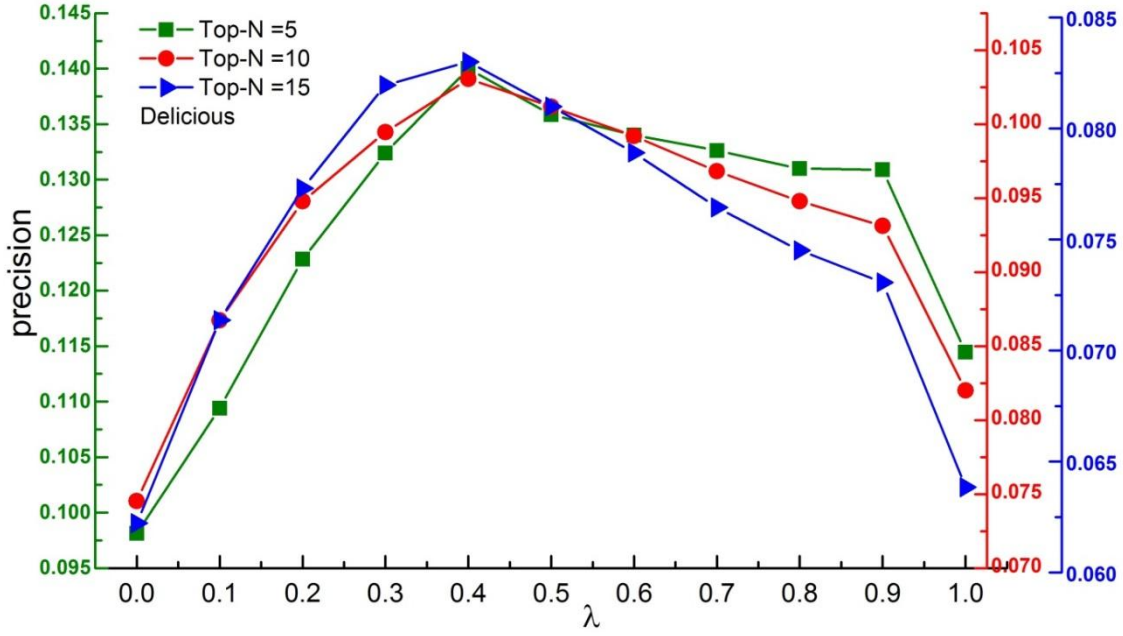
Table 4.3 Dataset Statistics

| statistics | Last.fm | Delicious |
|---|---|---|
| #Averagecommunities per user | 2.84773 | 11.2933 |
| #Average tags per community | 6.59644 | 11.0014 |

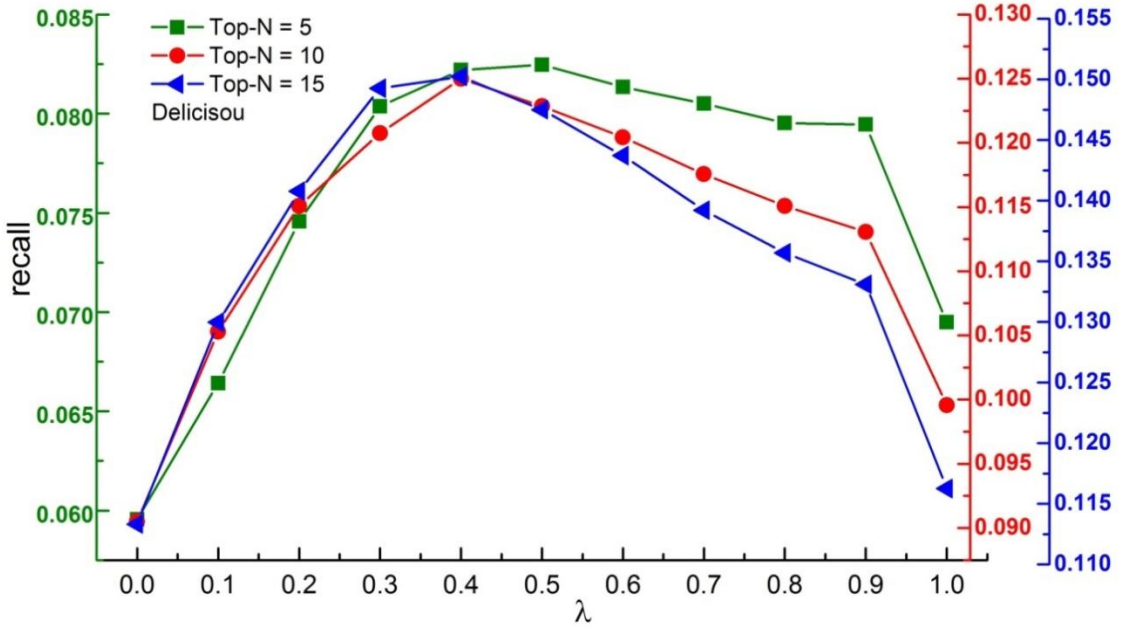
#### Impact of the Parameter $\lambda$

In DTGCF, as a tunable parameter, λ determines how much tag information should incorporate diffusion-based similarity in user-item bipartite network .λ=0 and λ=1 correspond to the cases for pure user-object and user-tag diffusions, respectively. To test the effect of the parameter λ, we conduct experiments by setting λ from 0 to 1 on different N. From the results shown in Figure 4.2 and Figure 4.3, we can observe that the value of λ impacts the recommendation results importantly. As λ increases, the Precision and Recall values increase at first, but when λ goes up to a certain threshold like 0.4 on the Delicious dataset, the Precision

and Recall values decrease with further increase of the value of λ. Thus, the best threshold of parameter λ is 0.4 on the Delicious dataset, and 0.5 is the best threshold for Last.fm dataset.

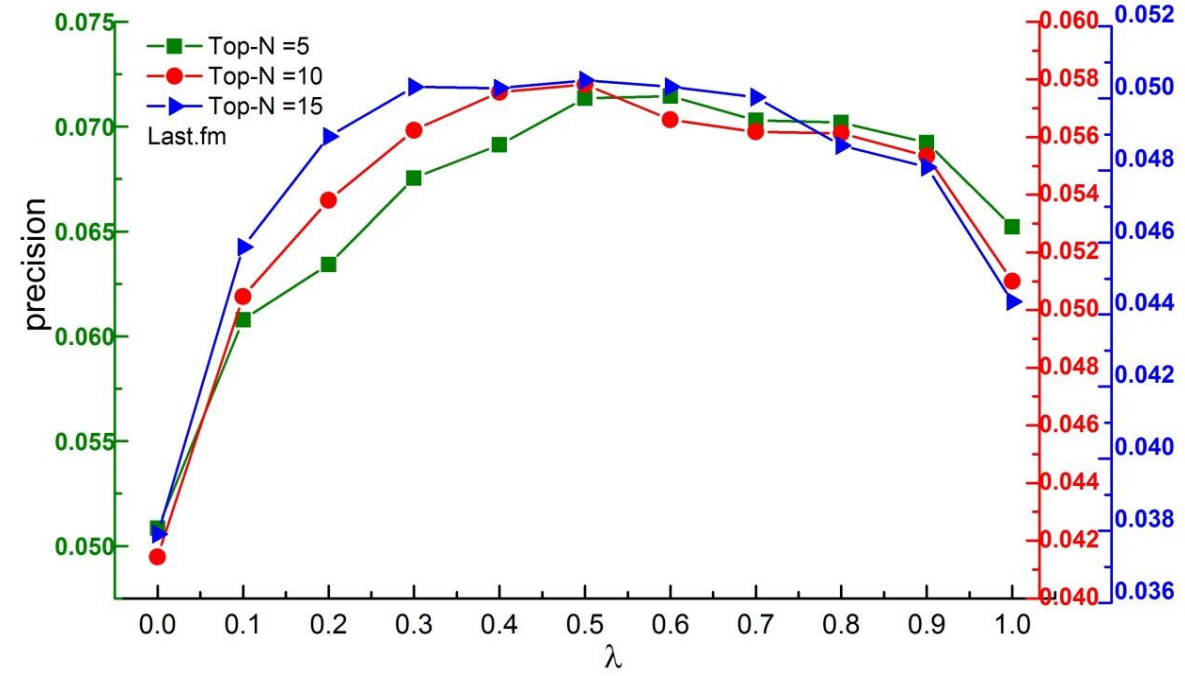



(a) λ versus Precision

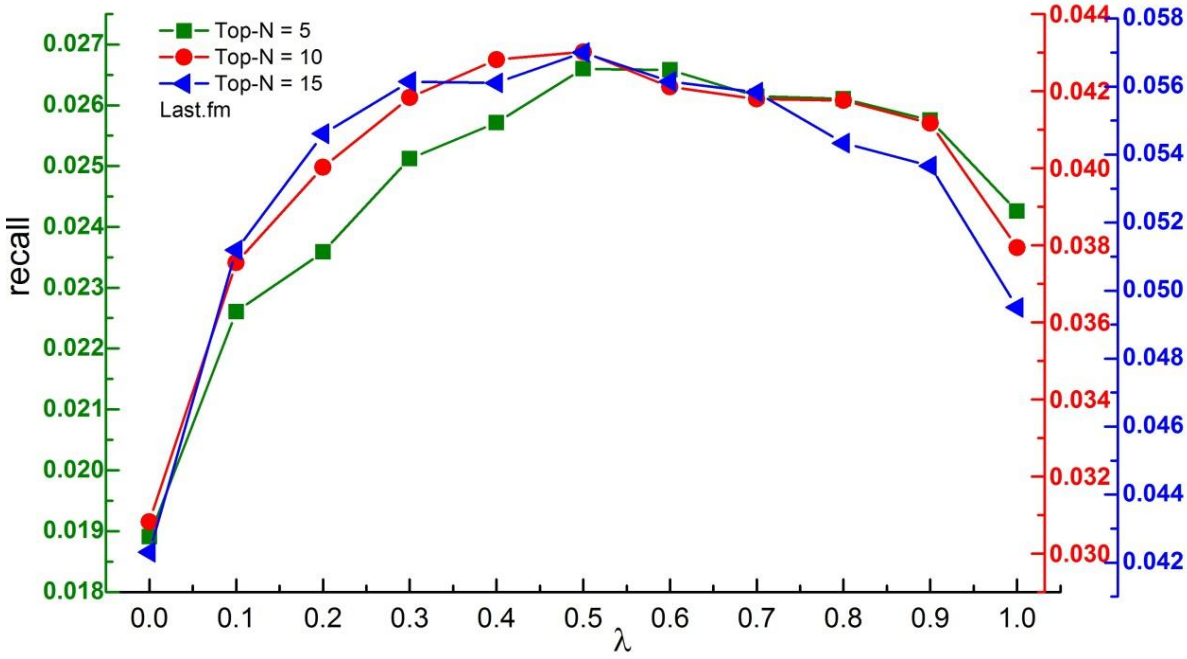



(b) λ versus Recall

Figure 4.2 Impact of Parameter λ on Delicious




(a) λ versus Precision

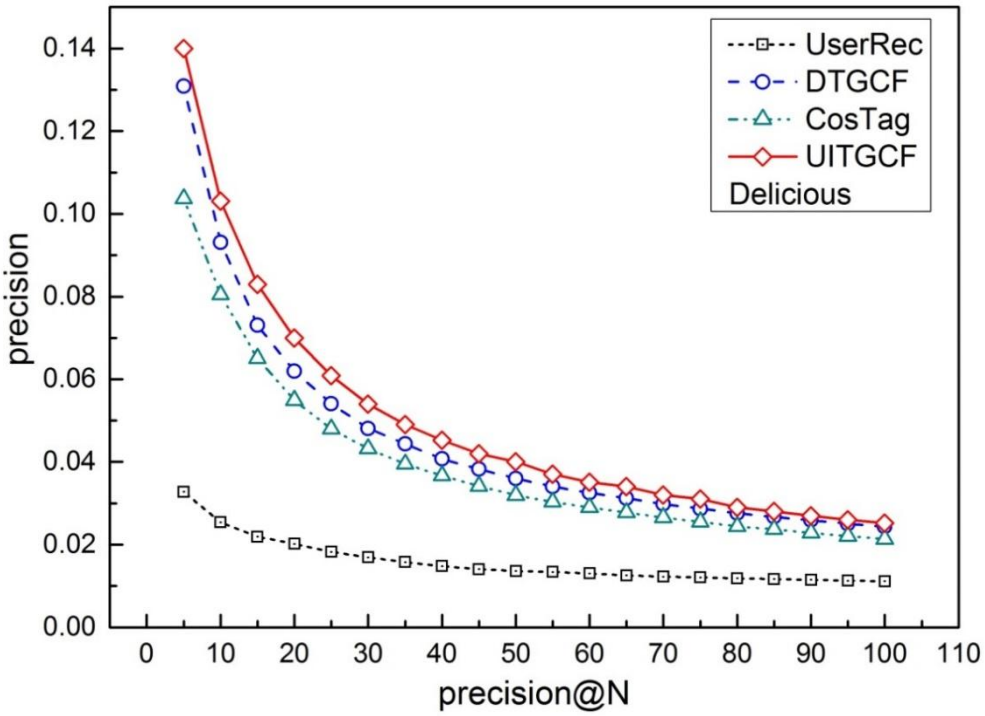



(b) λ versus Recall, Last.fm

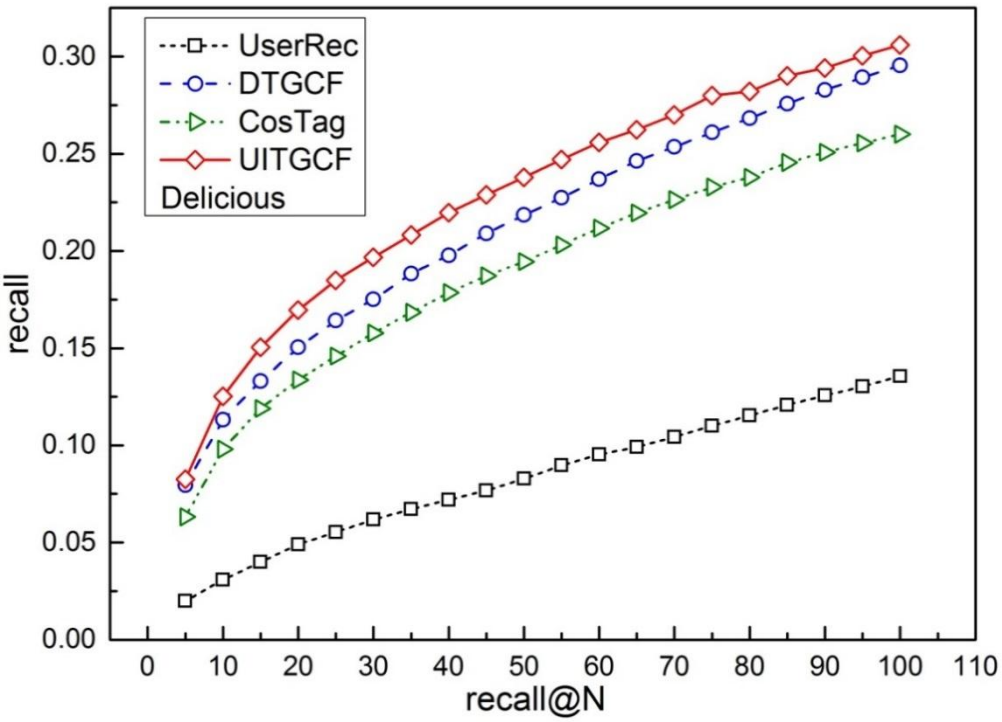
Figure 4.3 Impact of Parameter λ on Last.fm

## Comparisons with other methods

We apply the UITGCF and the other methods on the two real-world data sets to investigate their predictive performance in terms of accuracy. As shown in Figure 4.4 and Figure 4.5, UITGCF outperforms other methods in the Delicious dataset and Last.fm dataset. With respect to the Recall accuracy, when N is 5, 20 and 50, UITGCF outperforms DTGCF by 6.95%, 13.07% and 11.14% on the Delicious dataset, respectively. And on Last.fm dataset, UITGCF outperforms DTGCF by 18.6%, 9.29% and 7.59%, respectively. With respect to the Recall accuracy, when N is 50, 75 and 100, our method outperforms DTGCF by 8.89%, 7.29% and 5.08% on Delicious data set. On Last.fm dataset, UITGCF outperforms DTGCF by 7.53%, 5.73% and 6.69%, respectively. We can see that UITGCF does improve the accuracy of the top-N recommendation list.

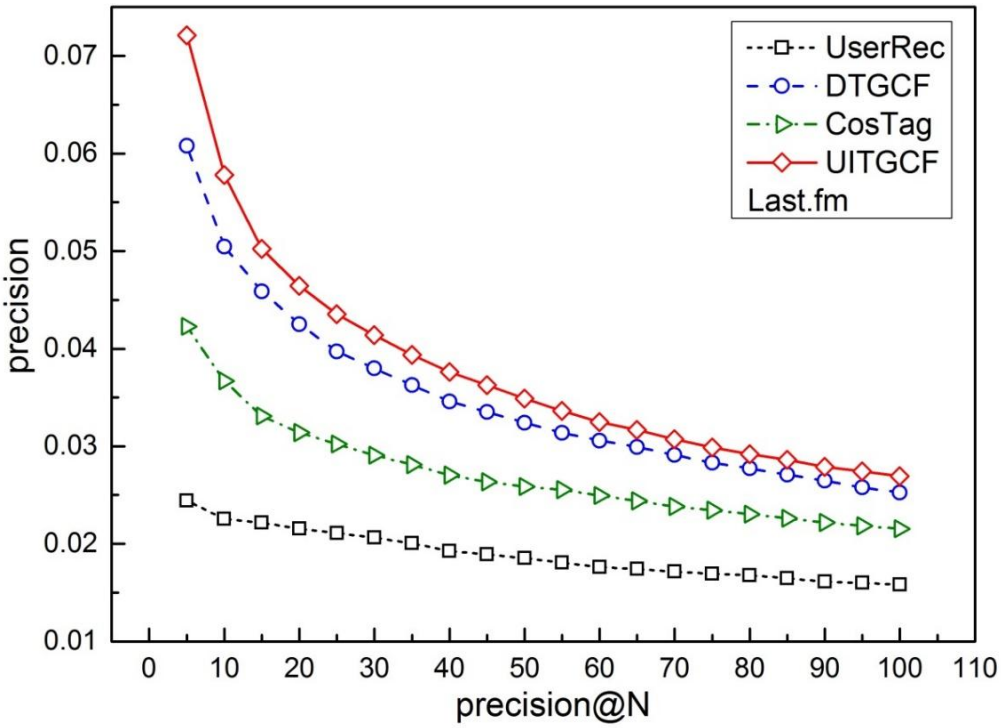



(a) Precision at N, Delicious

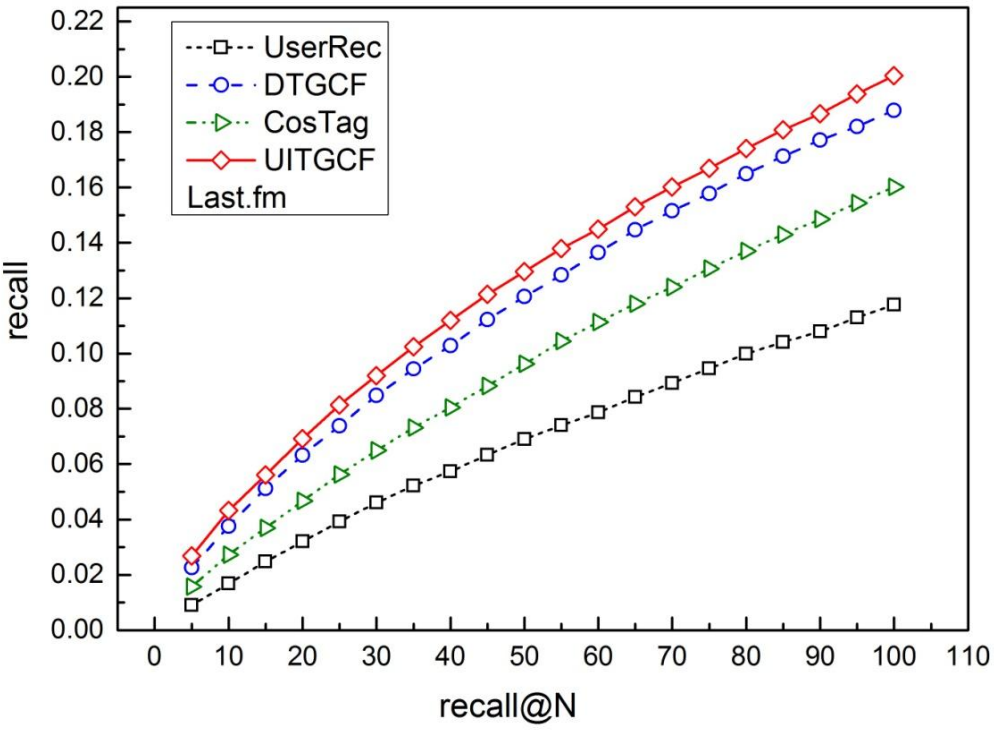



(b) Recall at N, Delicious

Figure 4.4 Comparisons of Precision and Recall on Delicious



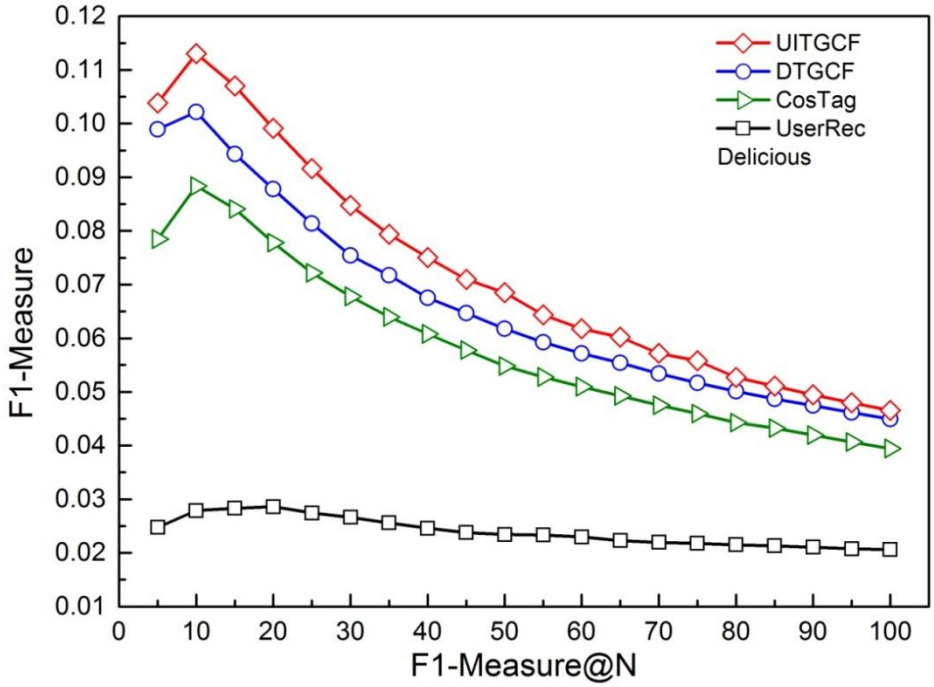

(a) Precision N,Last. fm



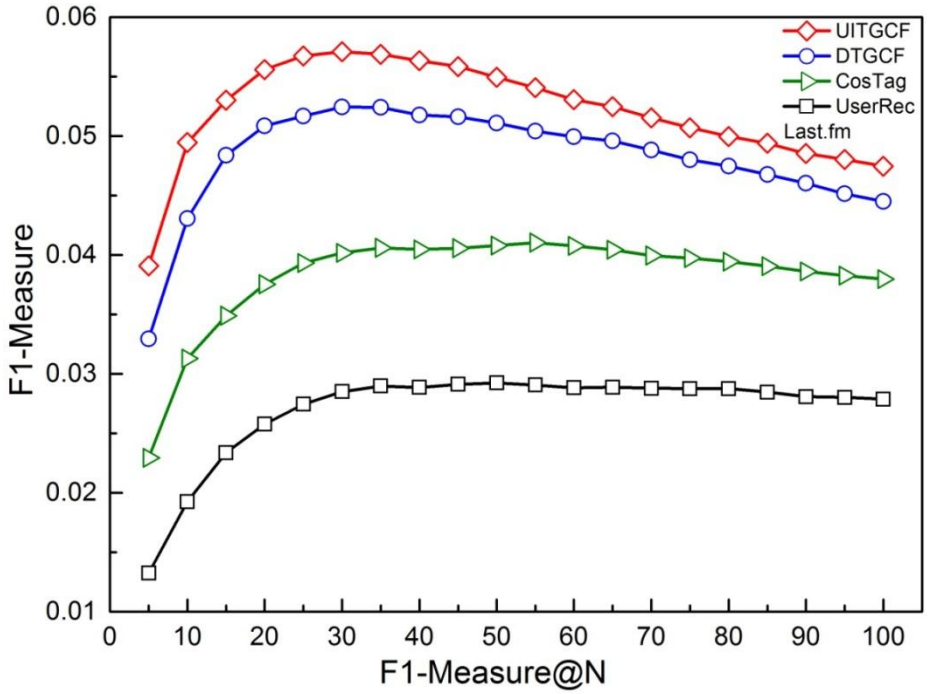

(b) Recall at N, Last.fm

Figure 4.5 Comparisons of Precision and Recall on Last.fm

Furthermore, we adopt F-measure to evaluate our algorithms. As shown in Fig.5, the performance of UITGCF is better than the other methods at any length of recommendation.



(a) F1-Measure@N, Delicious




(b) F1-Measure@N, Last.fm

Figure 4.6 Comparisons of F1 on two datasets

## 4.5 Application

From the above experiment, we are confident to say that the UITGCF has good recommendation quality. Hence we apply it to our live running academic service platform – SCHOLAT.com to help to support and to improve the friend recommendation function in SCHOLAT, users have information such as publications, research projects, teaching experience and related friends or collaboration partners, etc. We used this information like tags and modify $\text{sim}\,\text{DTG}\,(u, v)$ to utilise Text Vector Space Model (TVSM) to calculate the similarity rate between user u and user v.

A document or documents can be represented with a vector space by the terms occurring in the document with a weight for each term.

$$W_{i,j} = f_{i,j} \cdot \log \frac{N}{n_i} \tag{4.14}$$

where $f_{ij}$ is the term frequency of term T in document d (a local parameter), N is the total number of documents and $n_i$ is the number of documents containing the term. Wi,j represents the weight of i-th term in document j.

$$sim_{profile}(u,v) = \frac{\sum_{k=1}^{|T|} w_{k,u} \times w_{k,v}}{\sqrt{\sum_{k=1}^{|T|} w_{k,u}^2} \times \sqrt{\sum_{k=1}^{|T|} w_{k,v}^2}} \tag{4.15}$$

$$sim(u,v) = \begin{cases} \dfrac{2 \cdot sim_{profile}(u,v) \cdot sim_{UIG}(u,v)}{sim_{profile}(u,v) + sim_{UIG}(u,v)} & \text{sim}_{UIG}(u,v) \neq 0 \\ sim_{profile}(u,v) & \text{sim}_{UIG}(u,v) = 0 \end{cases} \tag{4.16}$$

Finally, the users are ranked by the similarities and select the top k users for the recommendation. We randomly choose users from the database to make a recommendation.

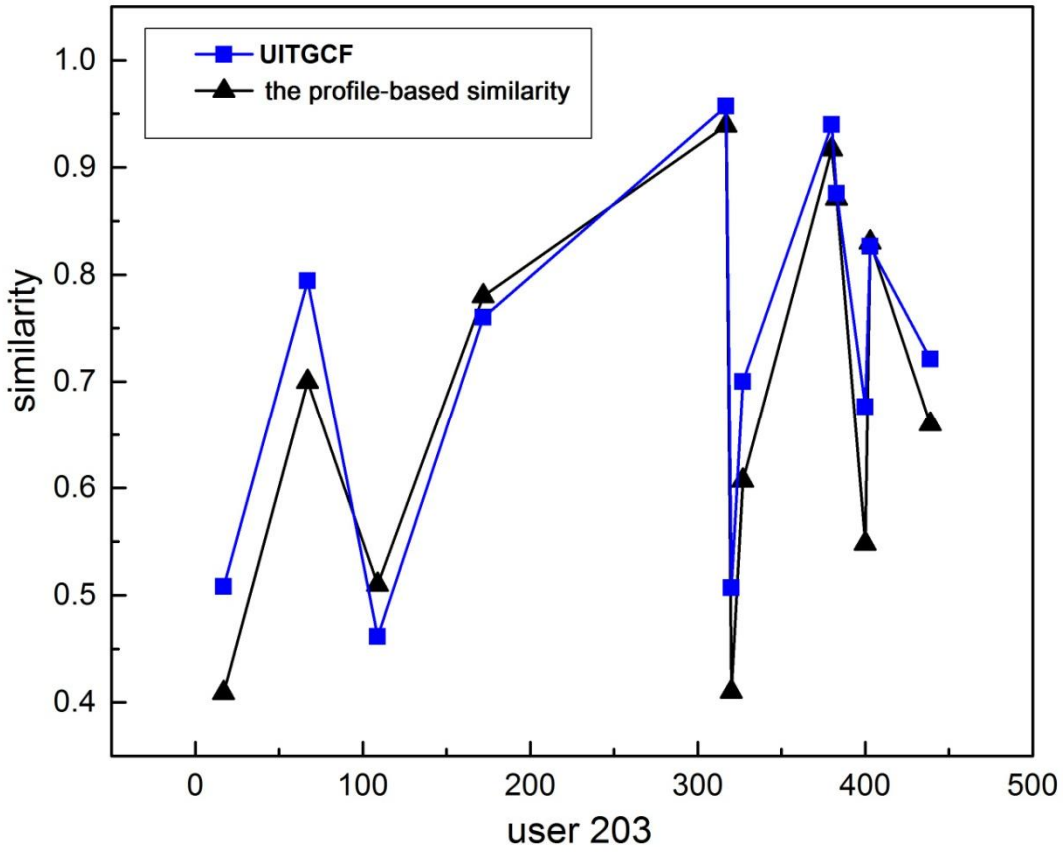



Figure 4.7 User 203 Result Analysis

As shown in the graph, we have analysed 203 users of our system. User 66 has a great influence in the research field. The similarity between user 203 and user 66 is improved by our method. Comparing with other recommendation method based on user's personal information, our method can provide considerablely more reliable results

## 4.6 Chapter Conclusion

In this chapter, a hybrid collaborative filtering recommendation algorithm by utilising the diffusion on the user-item-tag graph and user personal interests is presented. A distinct feature of our model is that it integrates similarities between users from mass diffusion method in tripartite graph and similarities of users from interest-based user recommendation in social tagging systems. Experiments on two datasets Delicious and Last.fm show that our method can improve the accuracy of Top-N recommendation.

# Chapter 5

## Sparse Ranking Model Adaptation for Cross-Domain Learning to Rank

*Cross-domain learning to rank problem has become a hot topic in transfer learning and learning to rank communities. In this problem, there is scarce human-labeled data in the target domain, but sufficient labelled data exists in a related domain named as source domain. In order to obtain an effective ranking model for the target domain, various ranking adaptation frameworks are proposed to learn ranking models with the help of the labelled data in the source domain [135]. In this thesis, I propose a sparse model adaptation framework, which utilises* $\ell_1$ *Regularisation to transfer the most confident prior knowledge from the source domain to the target domain. Due to the sparsity-inducing property of the* $\ell_1$ *Regularisation, the framework can reduce the negative effects of the feature gap between source domain data and target domain data. However, the optimisation problem formulated by the framework is non-differentiable. It is difficult to obtain the solution by the most popular methods. To address this problem, we design an efficient algorithm from the primal-dual perspective. Empirical evaluation over LETOR benchmark data [118]collections validates that the proposed algorithm can significantly improve the accuracy of the ranking prediction.*

## 5.1 Introduction

Learning to rank has attracted a lot of attention in recent years and many algorithms have been proposed [80, 81, 108, 153] as supervised learning methods, which require sufficient labelled data to train precise ranking models. However, in some applications, it is impractical to collect sufficient labelled data, while there is plenty of labelled data in a related domain named as source domain. Hence it results in a new problem named cross-domain learning to rank problem, which is to utilise the labelled data in the source domain to improve the ranking accuracy in the target domain. Recently, there are several ranking adaptation methods proposed to solve the cross-domain learning to rank problem [28, 112]. TransRank [32] chooses the k-best queries from the source domain for each target query to train a ranking model. CLRank [31] proposes a feature-level adaptation methods for cross-domain learning to rank. RA-SVM [102, 154] utilises the model parameter learned from the source domain to refine the target ranking model in the learning process.

Almost all the work ignores the negative effects of the noises brought from the poor features. When some features are noisy or redundant, it will result in poor generalisation performance. Sparse models [58, 80, 81] has emerged as a successful mechanism to solve this problem. It adopts the $\ell 1$ regularisation to choose the most relevant features. For example, FenchelRank [80] is a sparse ranking method with the $\ell 1$ constrain.

Sparse models are also desirable for cross-domain learning to rank. They offer several advantages: (1) Sparse model can avoid the poor performance when there are many redundant/noisy features in the target domain, which has been shown in the previous work [80]. Since there are very few labelled data in the target domain, it cannot learn a good model for all the features. In such case, the sparse model is a natural candidate since only a few parameters are needed to learn. (2) The sparse model can help to model the difference between the source and target's training weights. Suppose that we have a learned model from the source domain, it is reasonable to assume that most of the ranking weights in the source domain are similar to that of the target domain. While, due to the inconsistent joint distributions of feature and relevance, source model cannot be applied directly in the target domain. Since $\ell 1$norm has been observed that it can encourage many parameters to be zeros and only few parameters have larger values, it is a good choice to model the relationship between target model and source model.

Based on the above discussion, a sparse ranking method has been proposed to resolve the cross-domain learning to rank problem, which utilises $\ell 1$ regularisation to transfer the most confident prior knowledge of the source domain to target domain. We first formulate the sparse cross-domain learning to rank problem into a convex optimisation problem by combining a pairwise ranking loss and two sparse, inducing $\ell 1$ norms. One is for the target model. Another is to measure the difference between the source model and the target model. The optimisation problem is difficult to optimise since the two $\ell 1$ norms are non-differentiable. We propose a novel learning algorithm for this optimisation problem from the primal-dual perspective. Furthermore, we prove that, after T iterations, our algorithm can obtain a solution with the desired tolerance optimisation error $\epsilon = (1/T)$. Finally, our experimental results on several algorithms verify that, the proposed algorithm can significantly improve the accuracy of the ranking prediction in the data-scarce target domain.

This chapter is organised as follows. Section 2 gives a brief introduction to the problem definition and some specific related work of this chapter. Section 3 presents the algorithm overview in terms of the notations will be used throughout this chapter. Section 4 describes the general framework of the cross-domain learning to rank problem including the problem statement. Section 5 first introduces the designing principle of the proposed algorithm, then shows the details of the algorithm and makes a convergence analysis of the algorithm. Finally, Section 6 presents the experimental details and concludes this chapter.

## 5.2 Related Work

### *5.2.1 Learning to Rank*

Learning to rank is a kind of machine learning technique used to solve ranking problems. It works at the following stages. In training stage, a ranking model is learned from the labelled training data, which is composed of query-document pairs with relevance judgments labelled by human beings. In the testing stage, when a new query with some unlabeled documents is provided, the trained ranking model predicts the relevance degrees of these new query-document pairs and returns a ranked list of the documents according to their predicted relevance degrees.

Existing learning-to-rank algorithms are mainly categorised into three groups: the pointwise, pairwise and listwise approaches. The pointwise methods define the loss function based on individual documents, and cast ranking problems as regression or classification based

problems [39, 74]. For example, McRank [86] casts the ranking problem as multiple classification problems. The pointwise methods directly adopt the existing methods for ranking and do not consider the orders of document lists. The pairwise methods are proposed to deal this problem [75], which do not directly depend on the relevance label of each document but reduce ranking problems to classification problems based on the relative orders of document pairs. RankBoost [50], RankingSVM [29, 63] and RankNet [24] are the well-known pairwise algorithms. RankBoost is the extension of the Ad-aBoost algorithm, and it learns weak ranker based on the distribution defined on document pairs, while the pairwise methods only consider the relationships of each document pair. Further, the listwise approaches are proposed to consider the order of all ranking lists. The listwise approaches take document lists as instances and consider the ranking problems based on the list of all participating documents [113]. One of the representative works is LambdaRank [25], which directly optimises Information-Retrieval evaluation measures. LambdaRank does not search the smooth approximation of the IR evaluation measures but defines the smooth approximation gradient of the target cost function. The λ-gradient is proposed to specify rules that how the rank order of documents should change.

Another related work is FenchelRank [80], which is an efficient primal-dual framework for sparse learning to rank problems. It verifies that the property of the sparsity naturally chooses the most useful features to construct a ranking model. The experimental results show that FenchelRank can significantly improve the ranking accuracies. In this chapter, the primal-dual framework is used to optimise our problem.

### *5.2.2 Cross-Domain Learning to Rank*

How to transfer the knowledge of the source domain to target domain has become a hot research topic in recent years. For example, a newly born vertical search engine lacks labelled query-product pairs to train a precise ranking model, while the labelled query-document pairs from the other vertical search engine are sufficient. However, due to the inconsistent joint distributions of feature and relevance, source domain data cannot be applied directly to the training data in the target domain. It results in a new problem named cross-domain learning to rank problem.

Many cross-domain learning to rank algorithms belong to data adaptation, which utilises the labelled data from source domains directly. Instance-level transfer learning and feature-level

transfer learning are two widely used methods. The instance level chooses the most related source domain data to the target domain. For instance, TransRank [32] chooses the k-best queries from the source domain and puts them into target domain data for learning a ranking model. Cai et.al [27] shows that query weighting is more important than document weighting, and two query weight schemes are proposed. The feature-level is to find a low-dimensional feature representation which is shared by both target domain and source domain. Typical algorithms include CLRank [31], which proposes using linear combinations of the original features to find the shared features, and shows that feature-level is a special case of multitask learning.

Another ranking adaptation approach is a model adaptation. Instead of directly utilising the labelled data from the source domain, it adapts an existing ranking model to the target domain. Geng et.al [54] proposes a regularisation based algorithm, which utilises the source ranking model as the prior information. PairwiseTrada [11] adapts multi ranking models into the target pairwise preference data. In this chapter, a sparse ranking model adaptation for cross-domain learning to rank has been proposed.

## 5.3 Algorithm Overview

Utilising source domain data effectively is the key issue in cross-domain learning to rank. In this chapter, a sparse ranking adaptation framework has been proposed to keep the usage of the source domain data away from the noisy features. However, the objective function of the framework, which is made up of ranking loss and the sparse inducing norms, i.e. $\boldsymbol{\ell_1}$ norms, is non-smooth and hard to optimize. We analyzed the optimization problem from the primal dual perspective, and design a convergence provable algorithm to solve the problem. We prove that, after **T** iterations, our proposed algorithm is guaranteed to obtain a solution with desired tolerant optimization error $\boldsymbol{\epsilon} = \boldsymbol{O}(\frac{\mathbf{1}}{\boldsymbol{T}})$. In addition, experiments over benchmark datasets show that the proposed algorithm achieves state-of-the-art performance. In the future, we plan to further study the sparsity-inducing method for other scenarios of transfer learning to rank, such as multi-task ranking and cross domain learning to rank over different feature sets.

### *5.3.1 Notations*

This section provides some notations used in this chapter. Suppose that the target domain data are given by $S = \{(q_i, X_i, Y_i)\}_{i=1}^{m}$, where $X_i \in R^d$ is a document, $Y_i$ is the relevance judgment

and $q_i$ denotes the query. Let $(X_j, X_k)$ denotes an ordered documents pair, where $X_j$ and $X_k$are all in the same query and $X_j$ should be more relevant than the $X_k$. Let the number of all the ordered pairs as p. Assume that the pair $(X_j, X_k)$ is located at the $l$ position of all the document pairs. Let $\Phi$ be a matrix of size $p \times d$, where the $l^{th}$ row of the matrix is $(X_j - X_k)$. We also denote the Fenchel primal problem and dual problem as $\Psi(\sigma)$ and $\Gamma(\omega)$ respectively, where $\sigma \in R^d$ and $\omega \in R^d$ Let $\rho$ be the radius of $\ell_1$-ball.

Since the number of labelled target domain data is too small, the goal of cross-domain learning to rank is to learn a ranking function $f: R^d \rightarrow R$ with the help of source domain data. In this paper, we consider transferring the knowledge from the ranking model $\omega_c$, which is obtained by learning to rank algorithms in the source domain data. The notions are summarized in Table 5.1

Table 5.1 List of notations

| Notation | Meaning |
|---|---|
| $\boldsymbol{S = \{(q_i, X_i, Y_i)\}_{i=1}^m}$ | training set |
| $\boldsymbol{d}$ | the dimension of data |
| $\boldsymbol{p}$ | number of the ordered pairs |
| $\boldsymbol{\omega_c}$ | ranking model in the source domain |
| $\boldsymbol{\rho}$ | the radius of $\ell_1$-ball: $\|\omega\|_1 \leq \rho$ and $\|\omega - \omega_c\|_1 \leq \rho$ |
| $\boldsymbol{\Phi}$ | matrix in $\mathbb{R}^{p \times d}$ that contains the pairwise information |
| $\boldsymbol{I_C(\omega)}$ | $I_C(\omega) = 0$ if $\omega \in C$, otherwise $I_C(\omega) = \infty$ |

## 5.4 Problem Statement

A general framework of learning to rank is

$$\min_{\omega} \sum_{i=1}^{m} L(X_i, Y_i; \omega) + \lambda\Omega(\omega) \tag{5.1}$$

where $\boldsymbol{L(X_i, Y_i; \omega)}$ is the ranking loss function and $\boldsymbol{\lambda}$ is a parameter to control the trade-off between training error and model complexity.

To utilise the knowledge in the source domain, model adaptation has shown great success. It is reasonable to assume that the target model $\boldsymbol{\omega}$ and the source model $\boldsymbol{\omega_c}$ should have similar shapes in the function space. For example, the Ranking Adaptation SVM (RA-SVM) models the similarity as:

$$\min_{\omega} \sum_{i=1}^{m} L(X_i, Y_i; \omega) + \lambda\left[ \delta/2\|\omega\|_2^2 + (1-\delta)/2\|\omega - \omega_c\|_2^2\right] \tag{5.2}$$

A new adaptation regularisation term $\boldsymbol{\|\omega - \omega_c\|_2^2}$ is added into the objective function, which make sure the distance between the target model and source model are close.

RA_SVM focuses on the optimisation of $\boldsymbol{\ell_2}$ norm adaptation regularization. In contrast, we adopt $\boldsymbol{\|\omega\|_1 + \|\omega - \omega_c\|_1}$ as the regularizer. Sparse regularization has been proved to be effective in many applications. In this paper, interests lie in the sparse adaptation regularization. By replacing the $\boldsymbol{\ell_2}$ norm distance, the optimization problem can be stated as:

$$\min_{\omega} \sum_{i=1}^{m} L(X_i, Y_i; \omega) \qquad s.t.\ \|\omega\|_1 \le \rho_1; \|\omega - c_c\|_1 \le \rho_2 \tag{5.3}$$

It is well known that $\boldsymbol{\ell_1}$ norm usually leads to a sparse solution while $\boldsymbol{\ell_2}$ norrm does not. Hence the optimal solution of E.q (5.2) and E.q (5.3) are expected to be different:

- We obtain a sparse model while RA_SVM has a dense model.
- We only require most of the ranking weights in the source domain are similar to that of the target domain and a few of them can be far away. While RA_SVM requires all the ranking weights between the source model and target model are close.

The above two observations show that when there are a lot of noise in the data and large different distribution between two models, our method shows superior to RA_SVM. The set $\{\boldsymbol{\omega}|\|\boldsymbol{\omega}\|_1 \leq \boldsymbol{\rho_1} \cap \|\boldsymbol{\omega} - \boldsymbol{\omega_c}\|_1 \leq \boldsymbol{\rho_2}\}$ should not be empty so that the solution of the problem 3 can be obtained. We set $\boldsymbol{\rho_1} = \boldsymbol{\rho_2} = \boldsymbol{\rho}$ for simplicity in this paper, and in order to analyze the optimization problem from the Fenchel primal dual view, we select the square hinge loss as the loss function and rewrite the Eq.(5.3) as

$$\max_{\omega} \Gamma(\omega) = \max_{\omega} - \frac{\rho^2}{p} \sum_{i=1}^{p} max\left(0, \frac{1}{\rho} - (\Phi\omega)_i\right)^2 I_{\|\omega\|_1 \leq 1}(\omega) - I_{\|\omega - \omega_c\|_1 \leq 1}(\omega) \tag{5.4}$$

The primal problem of Eq.(5.4) can be written as

$$min\, \Psi(\sigma) = \min_{\sigma \geq 0} \frac{p}{4\rho^2} \|\sigma\|_2^2 - \frac{1}{\rho}\|\sigma\|_1 + \|\Phi^T\sigma\|_\infty + \frac{1}{2}\langle \Phi^T\sigma | \omega_c \rangle \tag{5.5}$$

More details about Fenchel primal and dual are presented in 5.4.1.

### *5.4.1 Algorithm Analysis Statement*

Some concepts and lemmas which will be used for the analysis of our algorithm are presented here. Fenchel primal-dual view of sparse cross-domain learning to rank

A function $f$ is convex if for all $\theta_1, \theta_2 \in dom(f)$, the domain of $f$ is denoted as $dom(f)$. $0 \leq \delta \leq 1$ and $f(\delta\theta_1 + (1-\delta)\theta_2 \leq \delta f(\theta_1) + (1-\delta) f(\theta_2)$. A vector $\phi$ is a sub-gradient of a function $f$ at $x$ if

$$\forall \theta_2, \langle \phi, \theta_1 - \theta_2 \rangle \leq f(\theta_1) - f(\theta_2) \tag{5.6}$$

We define the set of subgradients of function $f$ at $\theta_1$ as $\partial f(\theta_1)$. Note that if $f$ is convex and differentiable at $\theta_1$, and then $\partial f(\theta_1)$ consists of only a single vector, the gradient of $f$ at $\theta_1$ is denoted by $\nabla f(\theta_1)$.An important concept used in this paper is the Fenchel conjugate.

**Definition 1:** The Fenchel conjugate of a function f is defined as:

$$f^*(\theta) = \max_{\theta_1 \in dom(f),} \left(\langle \theta, \theta_1 \rangle - f(\theta_1)\right) \tag{5.7}$$

Note that if $f(\theta_1)$ is a convex and closed function, then the Fenchel conjugate of $f^*$is f itself.

**Lemma 1:** *(Proposition 3.3.4[20], Fenchel-Young inequality) Any points* $\theta_1 : \theta_2 \in dom(h)$ *and* $\theta_2 : \theta_2 \in dom(h^*)$ *satisfy the inequality:*

$$h(\theta_1) + h^*(\theta) \geq \langle \theta_1, \theta_2 \rangle \tag{5.8}$$

*Equality holds if and only if* $\theta_2 \in \partial h(\theta_1)$

**Lemma 2** *(Theorem 3.1.8[20]) If the function* $f : \mathcal{R}^p \to (-\infty, +\infty]$ *is convex, then any points* $\theta_1$*in core (*$dom(f)$*) and any direction* $\theta_2$*in* $\mathcal{R}^p$*satisfy*

$$f'(\theta_1; \theta_2) = max\{\langle \phi, \theta_2 \rangle | \phi \in \partial f(\theta_1)\} \tag{5.9}$$

*In particular, the sub-differential* $\partial f(\theta_1)$*is noneempty.*

Note that the core of a set C(written $core(dom(f))$) is the set of points $\theta_1$in $dom(f)$ such that for any direction $\theta_2$ in $\mathcal{R}^p$, $\theta_1 + \delta\theta_2$lies in $dom(f)$ for all small real $\delta$.

**Lemma 3** *(Lemma 3.2.6[20]) If the function* $h : E \to (-\infty, +\infty]$ *is convex and some point x in core (*$dom\ h$*) satisfies* $h(x) > -\infty$*, then* $h$ *never takes the value* $-\infty$

The next theorem is an important property for interpretation of the multi-Fenchel primal-dual framework and plays an important role in our analysis.

**Theorem 1** *For function* $f : \mathcal{R}^p \to (-\infty, +\infty]$, $g_i : \mathcal{R}^d \to (-\infty, +\infty]$, $i = 1, \dots, z$ *be a closed and convex functions.* $\Phi$ *is a* $\mathbb{R}^{p \times d}$ *matrix, then*

$$\sup_{\omega} -\mathrm{f}^{*}(-\Phi\omega) - \textstyle\sum_{\mathrm{i}=1}^{\mathrm{z}} \mathrm{g}_{\mathrm{i}}^{*}(\omega) \leq \inf_{\sigma} \mathrm{f}(\sigma) + \sum_{\mathrm{i}=1}^{\mathrm{z}} \mathrm{g}_{\mathrm{i}}\ (\Phi^{\mathrm{T}}\sigma/z)$$

*The equality holds when*

$$0 \in \bigcap_{i=1}^{z}(core(zdom(g_i) - \Phi^T dom(f)))$$

*Proof.* The weak duality inequality holds immediately from the Fenchel-Yong inequality (Lemma 1). To prove the equality, we define a function $\psi: \mathcal{R}^{\mathrm{d}} \rightarrow (-\infty, +\infty]$ by

$$\psi(v) = \inf_{\sigma \in \mathcal{R}^p} \left\{ f(\sigma) + \textstyle\sum_{i=1}^{z} g\left(\frac{\Phi^T\sigma + v}{z}\right)\right\}$$

It is easy to check $\psi$ is convex and $dom(\psi) = zdom(g) - \Phi^T dom(f)$. If the condition$0 \in \bigcap_{i=1}^{z}(core(zdom(g_i) - \Phi^T dom(f)))$ holds and $(\sigma) + \sum_{i=1}^{z} g\left(\frac{\Phi^T\sigma + v}{z}\right)$ is finite, then it can observed from Lemma 2 and Lemma 3 that there exists a sub-gradient $-\omega \in \partial\, \psi(0)$.

We have:

$$\psi(0) \leq\ \psi(v) + \langle -\omega, v\rangle, for\ all\ v\ \in R^d, \leq\ \ f(\sigma) + \sum_{i=1}^{z} g_i\left(\frac{\Phi^T\sigma + v}{z}\right) + \langle -\omega, v\rangle, for\ all\ v$$

$$\in R^d, \sigma \in R^p\ = \{f(\sigma) - \langle -\Phi\omega, \sigma\rangle\} + \sum_{i=1}^{z} g_i\left\{g_i\left(\frac{\Phi^T\sigma + v}{z}\right) - \langle -\omega, \frac{\Phi^T\sigma + v}{z}\rangle\right\}$$

Taking the infimum over all points v, and then over all points **σ** gives the inequalities

$$\psi(0) \leq -f^*(-\Phi\omega) - \textstyle\sum_{i=1}^{z} g_i^*(\omega) \leq \sup_{\omega} - f^*(-\Phi\omega) - \sum_{i=1}^{z} g_i^*(\omega) \leq \inf_{\sigma} f(\sigma) +$$

$$\textstyle\sum_{i=1}^{z} g_i\ \left(\frac{\Phi^T\sigma}{z}\right) =\ \psi(0).$$

Thus $$\sup_{\omega} - f^*(-\Phi\omega) - \textstyle\sum_{i=1}^{z} g_i^*(\omega) = \inf_{\sigma} f(d) + \sum_{i=1}^{z} g_i\left(\frac{\Phi^T\sigma}{z}\right)$$

When

$$0 \in \bigcap_{i=1}^{z}(core(zdom(g_i) - \Phi^T dom(f)))$$

When $\boldsymbol{z = 2}$, Theorem 1 can be stated as

**Corollary 1** *Let* $f: R^p(-\infty, +\infty], g_1, g_2: R^d(-\infty, +\infty]$ *be closed and convex functions* $\Phi$ *is a* $\mathbb{R}^{p\times m}$*matrix, then*

$$\sup_{\omega} -f^*(-\Phi\omega) - g_1^*(\omega) - g_2^*(\omega) \le \inf_{\sigma} f(\sigma) + g_1(\Phi^T\sigma/2) + g_2(\Phi^T\sigma/2) \quad (5.10)$$

*The equality holds when*

$$0 \in \left(core\left(2dom(g_1) - \Phi^T dom(f)\right)\right) \cap \left(core\left(2dom(g_2) - \Phi^T dom(f)\right)\right)$$

*When* $z = 1$*, Theorem 1 can be stated as*

**Corollary 2** *Let* $f: R^p(-\infty, +\infty], g_1: R^d(-\infty, +\infty]$ *be closed and convex functions.* $\Phi$ *is a* $\mathbb{R}^{p\times m}$*matrix, then*

$$\sup_{\omega} -f^*(-\Phi\omega) - g_1^*(\omega) \le \inf_{\sigma} f(\sigma) + g_1(\Phi^T\sigma) \quad (5.11)$$

*The equality holds when*

$$0 \in \left(core\left(dom(g_1) - \Phi^T dom(f)\right)\right)$$

Through **Corollary 1**, we can get the dual presentation of the problem (5.4) in the following manner. We denote $\mathbf{g_1^*(\omega)} = \mathrm{I}_{\mathrm{C}_1}(\omega)$ and $\mathrm{g}_2^*(\omega) = \mathrm{I}_{\mathrm{C}_2}(\omega)$ as the regularization terms of Equation (5.4), where $C_1 = \{\omega | \|\omega\|_1 \le 1\}$, $C_2 = \{\omega | \|\omega - \omega_c\|_1 \le 1\}$.

Denoting

$$f^*(x) = \frac{\rho^2}{p}(max(0, 1 + x))^2 = \frac{\rho^2}{p}|1 + x|_+^2,$$

We have

$$f^*(-\Phi\omega) = \frac{\rho^2}{p}\sum_{i=1}^{p} max(0,1-(\Phi\omega)_i)^2,$$

Which is the loss term in (4).

The problem (4) can be rewritten in its dual form:

$$\max_{\omega} \Gamma(\omega) = \max_{\omega} -f^*(-\Phi\omega) - g_1^*(\omega) - g_2^*(\omega) \tag{5.12}$$

Accordingly, we can also define the primal problem of Eq. (4). The following theorems state the definitions of $f(\sigma), g_1(\Phi^T\sigma)$ and $g_2(\Phi^T\sigma)$.

**Theorem 2** *(Lemma 3 [80]) Let* $f^*(-\Phi\omega) = \sum_{i=1}^{p}\frac{\rho^2}{p}max(0,\frac{1}{\rho}-(\Phi\omega)_i)^2,$

*then the Fenchel conjugate of*

$f^*(-\Phi\omega)$ is $f(\sigma) = \sum_{i=1}^{p}(\frac{p}{4\rho^2}\sigma_i^2 - \frac{1}{\rho}\sigma_i + I_{\sigma_i\geq 0}(\sigma_i))$.

**Theorem 3** *The Fenchel conjugates of* $g_1^*(\omega) = I_{\|\omega\|_1\leq 1}(\omega)$ *and* $g_2^*(\omega) = I_{\|\omega-\omega_c\|_1\leq 1}(\omega)$ *are* $g_1(\Phi^T\sigma/2) = \frac{1}{2}\|\Phi^T\sigma\|_\infty$ *and* $g_2(\Phi^T\sigma/2) = \frac{1}{2}\|\Phi^T\sigma\|_\infty + \frac{1}{2}\langle\Phi^T\sigma,\omega_c\rangle$ *respectively.*

*Proof.* The Fenchel conjugate of $g_1^*(\omega)$ is

$$\begin{aligned} g_1(\Phi^T\sigma/2) &= \max_{\omega}\langle\Phi^T\sigma/2\,,\omega\rangle - g_1^*(\omega) \\ &= \max_{\|\omega\|_1\leq 1}\langle\Phi^T\sigma/2\,,\omega\rangle \\ &= \frac{1}{2}\max_i|\Phi^T\sigma_i| \\ &= \frac{1}{2}\|\Phi^T\sigma\|_\infty \end{aligned}$$

And the Fenchel conjugate of $g_2^*(\omega)$ is

$$g_2(\Phi^T\sigma) = \max_{\omega}\langle\Phi^T\sigma/2\,,\omega\rangle - g_2^*(\omega)$$

$$
\begin{aligned}
&= \max_{\|\omega-\omega_c\|_1 \le 1} \langle \Phi^T \sigma/2 \,, \omega \rangle \\
&= \max_{\|t\|_1 \le 1} \langle \Phi^T \sigma/2 \,, t \rangle + \langle \Phi^T \sigma/2 \,, \omega_c \rangle \\
&= \max_i |(\Phi^T \sigma/2)_i| + \langle \Phi^T \sigma/2 \,, \omega_c \rangle \\
&= \frac{1}{2} \|\Phi^T \sigma\|_\infty + \langle \Phi^T \sigma, \omega_c \rangle.
\end{aligned}
$$

Combining Theorem 2 and Theorem 3, the primal problem of Eq. (4) can be written as

$$
min\, \Psi(\sigma) = min\, f(\sigma) + \; g_1(\Phi^T \sigma/2) + \; g_2(\Phi^T \sigma/2) = \min_{\sigma \ge 0} \frac{p}{4\rho^2} \|\sigma\|_2^2 - \frac{1}{\rho} \|\sigma\|_1 + \|\Phi^T \sigma\|_\infty + \frac{1}{2} \langle \Phi^T \sigma \, , \omega_c \rangle \quad (5.13)
$$

According to Theorem 1, the Fenchel primal objective is always the upper bound of the dual's:

$$
max\, \Gamma(\omega) \le min\, \Psi(\sigma) \quad (5.14)
$$

We analyse the convergence properties of the proposed algorithm here. We set $\omega^* = \underset{\omega}{argmax}\, \Gamma\,(\omega)$ as the best solution of the problem (4). At each iteration$t$, we denote $\epsilon_t = \Gamma\,(\omega^*) - \Gamma\,(\omega_t)$ as the difference between the best solution and the solution obtained at iteration$t$. Theorem 4 establishes the stopping criterion for our algorithm.

**Theorem 4:** For all t, we have $\epsilon_t \le \|\Phi^T \sigma_t\|_\infty + \langle \sigma_t, -\Phi \omega_t \rangle$. **Theorem 5** establishes the upper bound of required iterations to obtain a **ϵ**-accurate solution.

**Theorem 5**: The TFRank algorithm terminates after at most $16\,\rho^2/\epsilon - 1$ iterations and returns a ϵ-accurate solution, where $\rho^2 \ge 0.125$.*Proof.* Since$\Gamma\,(\omega_{t+1}) > \Gamma\,(\omega_{t+\frac{1}{2}})$, we have

$$
\begin{aligned}
\epsilon_t - \epsilon_{t+1} \quad &= \big(\Gamma\,(\omega^*) - \Gamma\,(\omega_t)\big) - \big(\Gamma\,(\omega^*) - \Gamma\,(\omega_{t+1})\big) \\
&= \Gamma\,(\omega_{t+1}) - \Gamma\,(\omega_w) \\
&\ge \Gamma\,\left(\omega_{t+\frac{1}{2}}\right) - \Gamma\,(\omega_w)
\end{aligned}
$$

$$= \epsilon_t - \epsilon_{t+\frac{1}{2}} \tag{5.15}$$

The proof of the convergence rate about FenchelRank (Theorem 1 [80]) shows that

$$\epsilon_t - \epsilon_{t+\frac{1}{2}} \geq \epsilon_t^2 / 16\rho^2 \tag{5.16}$$

Combining equation (5.15) and equation (5.16), we can obtain

$$\epsilon_t - \epsilon_{t+1} \geq \epsilon_t^2 / 16\rho^2 \tag{5.17}$$

According to Eq. (19), the result $\epsilon_t \leq \frac{16\rho^2}{t+1}$ holds from the proof of the convergence rate about FenchelRank [80]. In summary, when the iteration $t \geq \frac{16\rho^2}{c} - 1$, TFRank algorithm can obtain a $\boldsymbol{\epsilon}$-accurate solution. This concludes our proof.

## 5.5 Optimisation Process

In this section, primal-dual methods have been employed to solve the optimisation problem (5.4). Primal-dual methods are based on the theory of Fenchel Duality (Corollary 1), which shows that the primal objective is always the upper bound of the dual's. FenchelRank follows the genetic algorithmic framework to solve the ranking optimisation problems of the form: $min\, f^*(-\Phi\omega) + g^*(\omega)$ where $f^*$ is a convex loss function and $g^*$ is the regularization term, such as the $\ell_1$ constrain $\|\omega\|_1 \leq 1$. In this paper, we extend it to a more general form: $\boldsymbol{min\, f^*(-\Phi\omega) + \sum_{i=1}^{z} g_i^*(\omega)}$ We show that this general form also satisfies the theory of Fenchel Duality. The detail can be found in Theorem 1 and Appendix A.

The proposed sparse cross-domain learning to rank algorithm referred as TFRank, is described in Algorithm1.

Algorithm 1 TFRank algorithm

Input: pairwise data matrix $\boldsymbol{\Phi}$, desired accuracy $\boldsymbol{\epsilon}$, maximal iteration number $\boldsymbol{T}$, the radius $\boldsymbol{\rho}$ of $\boldsymbol{\ell_1}$ ball, the source model $\boldsymbol{\omega_c}$.

Output: linear ranking predictor $\boldsymbol{\omega}$.

Initialize: $\boldsymbol{\omega_1 = 0_d}$, $\boldsymbol{\omega_c' = \omega_c}$

For $\boldsymbol{t = 1, \dots, T}$

(1)Check the early stopping criterion

Let $\boldsymbol{\sigma_t = \partial f * (-\Phi \omega_t)}$

If $\|\boldsymbol{\Phi \hat{} T\ \sigma\_t}\|_\infty + \langle \boldsymbol{\sigma_t, -\Phi \omega_t} \rangle \leq \boldsymbol{\epsilon}$

return $\boldsymbol{\omega_t}$ as ranking predictor $\boldsymbol{\omega}$

End If

(2)Greedily choose a feature to update

Choose

$$\boldsymbol{j_t = \underset{j}{argmax} \left|(\Phi^T \sigma_t)_j\right|}$$

Finding

$$\boldsymbol{\mu_t = \underset{0 \leq \mu \leq 1, \|\omega_{t+\frac{1}{2}} - \omega_c\|_1 \leq 1}{argmax} \Gamma\left((1-\mu)\omega_t + \mu sign\left((\Phi^T \sigma_t)_{j_t}\right) e^{j_t}\right)}$$

where $\boldsymbol{\omega_{t+\frac{1}{2}} = (1-\mu_t)\omega_t + \mu_t sign\left((\Phi^T \sigma_t)_{j_t}\right) e^{j_t}}$

(3)Select a new feature according to the source model

Update $\boldsymbol{\sigma_{t+\frac{1}{2}} = \partial f^* (-\Phi \omega_{t+\frac{1}{2}})}$

Choose

$$\boldsymbol{j_{t+\frac{1}{2}} = \underset{j}{argmax} \left|(\Phi^T \sigma_{t+\frac{1}{2}})_j \times (\omega_c')_j\right|}$$

Finding

$$\boldsymbol{\mu_{t+\frac{1}{2}} = \underset{0 \leq \mu \leq 1}{argmax} \Gamma\left((1-\mu)\omega_{t+\frac{1}{2}} + \mu(\omega_c')_{j_{t+\frac{1}{2}}} e^{j_{t+\frac{1}{2}}}\right)}$$

where

$$\boldsymbol{\omega_{t+1} = \left(1 - \mu_{t+\frac{1}{2}}\right)\omega_{t+\frac{1}{2}} + \mu_{t+\frac{1}{2}} (\omega_c')_{j_{t+\frac{1}{2}}} e^{j_{t+\frac{1}{2}}}}$$

If $\left|(\omega_c')_{j_{t+\frac{1}{2}}}\right| > 0$

Let $\omega_c' = \omega_c' - \mu_{t+\frac{1}{2}}(\omega_c')_{j_{t+\frac{1}{2}}} e^{j_{t+\frac{1}{2}}}$

End If

End For

The input of the algorithm includes a pairwise data matrix $\mathbf{\Phi}$, a desired accuracy $\boldsymbol{\epsilon}$, the number of iterations $\mathbf{T}$ , the radius of $\boldsymbol{\ell_1}$-ball $\boldsymbol{\rho}$ and the source model $\boldsymbol{\omega_c}$. The dual solution $\boldsymbol{\omega}$ is initialized to be $\mathbf{0_d}$, and the algorithm stops when the desired accuracy $\boldsymbol{\epsilon}$ is met or the maximal iteration number $\mathbf{T}$ reached.

In the iteration: two variables, dual solution $\boldsymbol{\omega}$ and primal solution$\boldsymbol{\sigma}$, are updated to find the solution. Specifically, our algorithm has three main steps: (1) Check the early stopping criterion. (2) Greedily choose a feature to update. (3) Select a new feature according to the source model. Note that Step (1) is an identical step in the FenchelRank algorithm, and Step (2) is similar to the FenchelRank algorithm, except that we also require the new constraint $\|\boldsymbol{\omega} - \boldsymbol{\omega_c}\|_1$is less than 1. Step (3) is to transfer the most confident prior knowledge from the source domain. If the feature in the source domain is important, it should have the high probability to be selected in the target domain. We will discuss these issues in the following subsections, respectively.

### *5.5.1 Checking the early stopping criterion*

Theorem 4 establishes the stopping criterion for our algorithm. It shows that the difference between the best solution and the solution obtained at iteration $\mathbf{t}$ is less than $\|\mathbf{\Phi^T \sigma_t}\|_\infty + \langle \boldsymbol{\sigma_t}, -\mathbf{\Phi \omega_t} \rangle$.Hence, if$\|\mathbf{\Phi^T \sigma_t}\|_\infty + \langle \boldsymbol{\sigma_t}, -\mathbf{\Phi \omega_t} \rangle < \boldsymbol{\epsilon}$ we obtain a $\boldsymbol{\epsilon}$-accuracy solution. This is also verified in the lemma 11 of [123].

### *5.5.2 Greedily choose a feature to update*

In this subsection, we describe how to choose a feature and compute an appropriate step size to update the weight of this feature. At iteration, given the dual variables $\boldsymbol{\omega}$, we compute the primal vector $\boldsymbol{\sigma}$ as following

$$\sigma_t^{(i)} = \partial f^*((-\Phi\omega_t)_i) = \begin{cases} \frac{2\rho^2}{p}\left(\frac{1}{\rho} - (\Phi\omega_t)_i\right) & \text{if } \frac{1}{\rho} - (\Phi\omega_t)_i \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{5.18}$$

Where $\sigma_t^{(i)}$ denotes the $i^{th}$ coordinate of $\sigma_t$. Since $\sigma_t = \partial f^*(-\Phi\omega_t)$, we have $f^*(-\Phi\omega_t) + f(\sigma_t) = \langle \sigma_t, -\Phi\omega_t \rangle$ (Lemma 1). Then, the algorithm selects a feature $j_t$, which has the largest absolute value of $(\Phi^T\sigma_t)_j$ as a weak learner. This step is a common way to obtain a weak learner in boosting algorithms [51], which choose the most violated feature to update.

Given the selected feature, we set $\omega_{t+\frac{1}{2}}$ to be the convex combination of $\omega_t$ and the selected feature

$$\omega_{t+\frac{1}{2}} = (1-\mu_t)\omega_t + \mu_t sign\left((\Phi^T\sigma_t)_{j_t}\right)e^{j_t} \tag{5.19}$$

Where the sign function $sign(x) = 1$ if $x \geq 0$ and otherwise $sign(x) = -1$, and $e^i$ denotes the vector with all zeros except the $i^{th}$ element is 1.

The coefficient $\mu_t$ is calculated so as to maximize the increase of

$$\Gamma\left(\omega_{t+\frac{1}{2}}\right) - \Gamma(\omega_t) = \Gamma(1-\mu_t)\omega_t + \mu_t sign\left((\Phi^T\sigma_t)_{j_t}\right)e^{j_t}) - \Gamma(\omega_t)$$

Denoting

$$a_t = \Phi((sign\left(\left(\Phi^T\sigma_t\right)_{j_t}\right)e^{j_t} - \omega_t)$$

and

$$b_t = \frac{1}{\rho} - \Phi\omega_t$$

The problem can be simplified as

$$\mu_t = \underset{0\leq\mu\leq 1, \|\omega_{t+\frac{1}{2}} - \omega_c|_1 \leq 1}{argmin} (b_t - a_t\mu_t)_+^2$$

And it can be solved analytically, which has been shown in FenchelRank.

### *5.5.3 Select a new feature according to the source model*

In the following updated step, we select the feature with the help of the model parameter learned in the source domain. We set $\sigma_{t+\frac{1}{2}} = \partial f * (-\Phi\omega_{t+\frac{1}{2}})$. The feature employs the best-

weighted edge by $j_{t+\frac{1}{2}} = \underset{j}{argmax} \left|(\Phi^T \sigma_{t+\frac{1}{2}})_j \times (\omega_c')_j\right|$. Note that the large value in source model will have the largest change to be selected. After that, we use the similar method to update $\omega_{t+1} = \left(1 - \mu_{t+\frac{1}{2}}\right)\omega_{t+\frac{1}{2}} + \mu_{t+\frac{1}{2}}(\omega_c')_{j_{t+\frac{1}{2}}} e^{j_{t+\frac{1}{2}}}$ , and $\mu_{t+\frac{1}{2}} = \underset{0\leq\mu\leq1}{argmax}\, \Gamma\left((1-\mu)\omega_{t+\frac{1}{2}} + \mu(\omega_c')_{j_{t+\frac{1}{2}}} e^{j_{t+\frac{1}{2}}}\right)$ respectively. Finally, $\omega_c'$ is adjusted to weaken the impact of the selected element$(\omega_c')_{j_{t+\frac{1}{2}}}$. This strategy guarantees that the ranking model $\omega_t$ can approach $\omega_c$ in each round of the training process.

We show that the above update rule can satisfy the constraint of $\|\boldsymbol{\omega_t}\|_1 \leq \mathbf{1}$ and $\|\boldsymbol{\omega_t} - \boldsymbol{\omega_c}\|_1 \leq \mathbf{1}$. Since we initialize the weight vector $\boldsymbol{\omega_1}$ to be the zero vector and restrict the range of coefficient $\boldsymbol{\mu_t}$ to be in$[\mathbf{0}, \mathbf{1}]$, it is easy to verify that for any $\mathbf{t}$,

$$\begin{aligned}\left\|\boldsymbol{\omega}_{\boldsymbol{t+\frac{1}{2}}}\right\|_{\mathbf{1}} &= \left\|(\mathbf{1}-\boldsymbol{\mu_t})\boldsymbol{\omega_t} + \boldsymbol{\mu_t}\boldsymbol{e^i}\right\| \\ &\leq (\mathbf{1}-\boldsymbol{\mu_t})\|\boldsymbol{\omega_t}\| + \boldsymbol{\mu_t}\|\boldsymbol{e^i}\| \\ &\leq \mathbf{1}\end{aligned}$$

Since $\|\boldsymbol{\omega_c}\|_1 \leq \mathbf{1}$, we have

$$\begin{aligned}\|\boldsymbol{\omega_{t+1}}\|_1 &= \left\|\left(\mathbf{1}-\boldsymbol{\mu}_{\boldsymbol{t+\frac{1}{2}}}\right)\boldsymbol{\omega}_{\boldsymbol{t+\frac{1}{2}}} + \boldsymbol{\mu}_{\boldsymbol{t+\frac{1}{2}}}(\boldsymbol{\omega_c'})_{\boldsymbol{j}_{\boldsymbol{t+\frac{1}{2}}}} \boldsymbol{e}^{\boldsymbol{j}_{\boldsymbol{t+\frac{1}{2}}}}\right\|_{\mathbf{1}} \\ &\leq \left(\mathbf{1}-\boldsymbol{\mu}_{\boldsymbol{t+\frac{1}{2}}}\right) + \boldsymbol{\mu}_{\boldsymbol{t+\frac{1}{2}}} \\ &= \mathbf{1}\end{aligned}$$

Furthermore,

$$\begin{aligned}\|\boldsymbol{\omega_{t+1}} - \boldsymbol{\omega_c}\|_1 &\leq \left(\mathbf{1}-\boldsymbol{\mu}_{\boldsymbol{t+\frac{1}{2}}}\right)\left\|\boldsymbol{\omega}_{\boldsymbol{t+\frac{1}{2}}} - \boldsymbol{\omega_c}\right\|_{\mathbf{1}} \\ &\quad + \boldsymbol{\mu}_{\boldsymbol{t+\frac{1}{2}}}\left\|(\boldsymbol{\omega_c'})_{\boldsymbol{j}_{\boldsymbol{t+\frac{1}{2}}}} \boldsymbol{e}^{\boldsymbol{j}_{\boldsymbol{t+\frac{1}{2}}}} - \boldsymbol{\omega_c}\right\|_{\mathbf{1}} \\ &\leq \left(\mathbf{1}-\mu_{t+\frac{1}{2}}\right) + \mu_{t+\frac{1}{2}} \\ &= 1\end{aligned}$$

Thus $\|\omega_t\|_1 \leq 1$ and $\|\omega_t - \omega_c\|_1 \leq 1$ hold at each iteration $t$.

### 5.5.4 Discussion

Compared to the FenchelRank algorithm, our innovations are making the following differences. 1) FenchelRank utilised Fenchel-dual inequality as a tool to explain the upper bound of the algorithm. However, Fenchel-dual inequality only has two terms inequality, which is not useful for our proposed framework with three terms. Here, we propose a more general form of Fenchel-dual inequality, the correctness of which is proved by us in this paper. Based on the new Fenchel-dual inequality, we confirm the upper bound of our proposed algorithm. 2) Since an additional term learned from the source domain leads to the framework, the proposed algorithm has to be adjusted against FenchelRank. Accordingly, the convergence rate of our proposed algorithm needs to be analysed.

## 5.6 Experiment and Chapter Conclusion

In this section, we evaluate the effectiveness of the proposed algorithm on several public benchmark datasets.

### 5.6.1 Datasets

We conduct our experiments on LETOR 3.0 and LETOR 4.0 data collections [118], which are both the benchmark datasets for the research on learning to rank.

Letor 3.0 contains seven datasets: TD2003, TD2004, HP2003, HP2004, NP2003, NP2004 and OHSUMED. The former six datasets are extracted from the TREC 2003 and TREC 2004 web track, in which there are three search tasks: topic distillation (TD2003, NP2004), homepage finding (HP2003, HP2004) and named page finding (NP2003, NP2004). Hence, we use TD2003 (HP2003, NP2003) as the source domain dataset of the target domain dataset TD2004 (HP2004, NP2004) respectively. In all the six datasets, there are 64 features extracted from the. Gov collection, covering from a wide range including low lever features and high-level features. Also, there are two levels of relevance (irrelevant or relevant) for each query-document pair. OHSUMED is not selected because it does not have related domain dataset.

Letor 4.0 contains two datasets: MQ2007 and MQ2008, which are extracted from Million Query tracks of TREC 2007 and TREC 2008. There are about 1700 queries in MQ2007 and about 800 queries in MQ2008, where the associated documents are constructed in 46 features with three relevance degrees (irrelevant, partially relevant and highly relevant). We use MQ2007 as the source domain dataset, and treat MQ2008 as the target dataset. The details of the datasets are shown in tables below.

Table 5.2 the information of the original datasets used in the experiments

| Dataset | # Query | # Documents | Relevance Degree | Dimension |
|---|---|---|---|---|
| TD2003 | 50 | 49058 | 2 | 64 |
| TD2004 | 75 | 74146 | 2 | 64 |
| HP2003 | 150 | 147606 | 2 | 64 |
| HP2004 | 75 | 74409 | 2 | 64 |
| NP2003 | 150 | 148657 | 2 | 64 |
| NP2004 | 75 | 73834 | 2 | 64 |
| MQ2007 | 1700 | 69623 | 3 | 46 |
| MQ2008 | 800 | 15211 | 3 | 46 |

Table 5.3 Source Domain and Target Domain

| Source Domain | Target Domain |
|---|---|
| TD2003 | TD2004 |
| HP2003 | HP2004 |
| NP2003 | NP2004 |
| MQ2007 | MQ2008 |

### *5.6.2 Evaluation Measures*

In order to evaluate the ranking performance of our method, we use Mean Average Precision (MAP) as an evaluation measures. MAP is a standard evaluation measure commonly used for binary relevance judgments in Information Retrieval. It is the mean of average precisions over all queries, so we just need to present the definition of average precision as follows:

$$AP = \frac{\sum_{i=1}^{n} Pati \times rel(i)}{N_{rel}}$$

Where $\mathrm{Pati} = \frac{\sum_{j=1}^{i} \mathrm{rel(j)}}{i}$, $\mathrm{N_{rel}}$ is the number of relevant documents and $\mathrm{rel(i)}$ is an indicator function. If document i is relevant, then $\mathrm{rel(i)} = 1$, or else $\mathrm{rel(i)} = 0$.

NDCG is another widely used evaluation metric for information Retrieval. Unlike MAP, NDCG can deal with the datasets with multi-levels of relevance judgments. The NDCG value at position i for a query can be written as follows:

$$NDCGati = \frac{1}{Z_i} \sum_{j=1}^{i} \frac{2^{\rho(j)} - 1}{log(1+j)}.$$

Where $\rho(j)$ is the relevance grade of the jth document, and $Z_i$ is such a normalisation constant that the NDCG value for a perfect ranking is 1. Overall, NDCGati reflects the ranking accuracy at the top i positions of a ranking list.

### *5.6.3 Experimental Results*

Six algorithms are implemented to demonstrate the effectiveness of TFRank, which have been shown in the following:

- TFRank_s: this only uses the source domain datasets.
- TFRank_t: this only uses the few-labeled data in target datasets.
- TFRank_mix: this directly utilises the data in both source domain and target domain.
- CLRank_inst: this is a data adaptation based instance weighting algorithm.
- RA_SVM: this is a dense model adaptation algorithm.
- TFRank: this is our sparse cross-domain learning to rank solution,

Noticing that TFRank_s, TFRank_t and TFRank_mix are all implemented by the TFRank algorithm for fair comparison. CLRank_inst is data adaptation method, and RA_SVM is model adaptation algorithm.

Since the cross-domain learning to rank is a data-starved problem, the number of the labelled queries selected from data pool should be small. The situations of utilising 5 and 10 labelled queries from the training data pool are shown averaged over the ten randomly repeated rounds. In each round, the training queries in target domain are selected randomly. The parameter ρ is chosen in the set $\{1, 2, 10, 20, 100, 200\}$ by cross-validation.

Table 5.4 Comparison on MAP values (5 queries selected in target domain)

| Source Domain | TD2003 | HP2003 | NP2003 | MQ2007 |
|---|---|---|---|---|
| Target Domain | TD2004 | HP2004 | NP2004 | MQ2008 |
| TFRank_s | 0.1798 | 0.6180 | 0.6499 | 0.4510 |
| TFRank_t | 0.1767 | 0.6193 | 0.6063 | 0.4350 |
| TFRank_mix | 0.1802 | 0.6596 | 0.6413 | 0.4687 |
| CLRank_inst | 0.1927 | 0.6738 | 0.6612 | 0.4721 |
| RA_SVM | 0.2023 | 0.6801 | 0.6597 | 0.4783 |
| TFRank | 0.2029 | 0.6904 | 0.6669 | 0.4853 |

Table 5.5 Comparison on MAP values (10 queries selected in target domain)

| Source Domain | TD2003 | HP2003 | NP2003 | MQ2007 |
|---|---|---|---|---|
| Target Domain | TD2004 | HP2004 | NP2004 | MQ2008 |
| TFRank_s | 0.1798 | 0.6180 | 0.6499 | 0.4510 |
| TFRank_t | 0.1940 | 0.6708 | 0.6541 | 0.4681 |
| TFRank_mix | 0.1885 | 0.6842 | 0.6580 | 0.4747 |
| CLRank_inst | 0.2010 | 0.6838 | 0.6695 | 0.4750 |
| RA_SVM | 0.2043 | 0.6850 | 0.6675 | 0.4789 |
| TFRank | 0.2126 | 0.6966 | 0.6749 | 0.4894 |

Table 5.4 and Table 5.5 list the MAP performance comparisons among all the algorithms. Fig. 5.1, Fig. 5.2, Fig. 5.3 and Fig. 5.4 reports the NDCG values of these algorithms. (1) Compared to TFRank_s, TFRank_t and TFRank_mix three non-ranking-adaptation algorithms, TFRank shows significant ranking performance gain. When 5 queries are selected in the target domain, TFRank has a 12% to 22% increase over TFRank_s, a 12% to 20% increase over TFRank_t, and a 8% to 16% increase over TFRank_mix with respect to NDCG and MAP on TD2004 datasets. When 10 queries are selected, TFRank shows a 12% to 18% increase over TFRank_s, a 2% to 7% increase over TFRank_t, and a 1% to 6% increase over TFRank_mix on HP2004 datasets. (2) Compared to CLRank_inst and RA_SVM two ranking adaptation algorithms, TFRank also achieves competitive performance. For example, on MQ2007/MQ2008 datasets, the value of MAP of TFRank is 0.4853/0.4789 of the second best algorithm, which indicates a 1.4/2.2% increase.

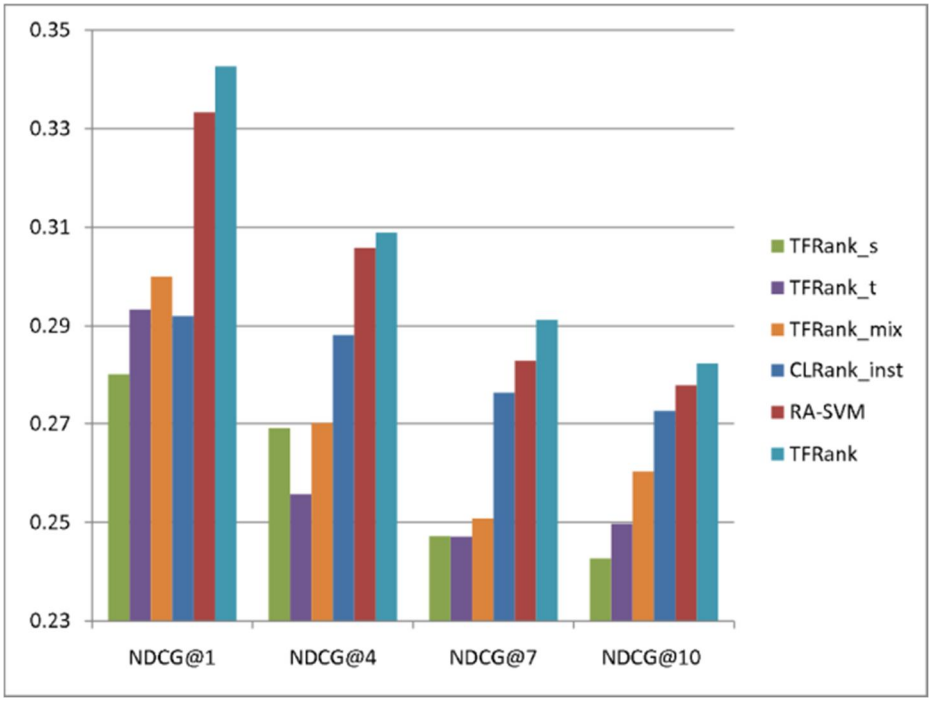



Figure 5.1 (a)

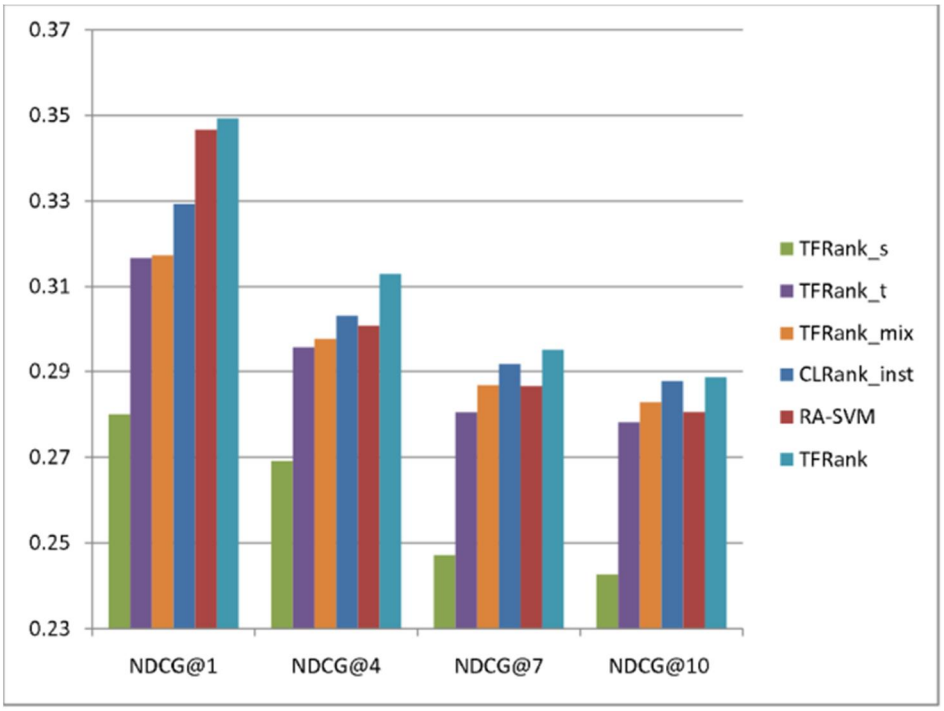



Figure 5.1 (b)

Figure 5.1 (a) reports NDCG values for 5 queries on TD2004 dataset and (b) reports NDCG values for 10 queries on TD2004 dataset

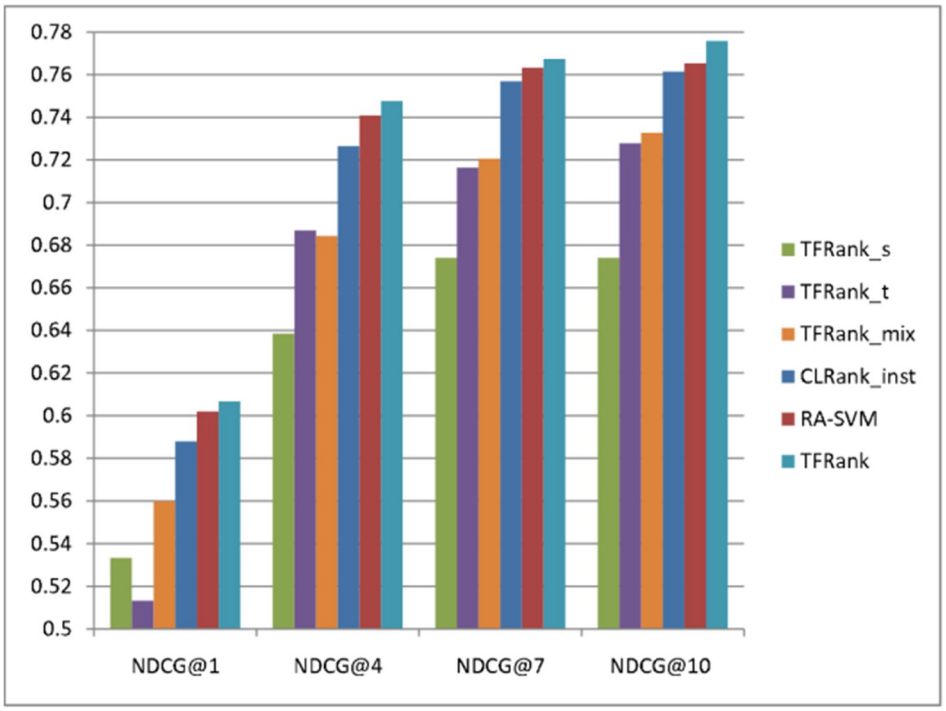



Figure 5.2 (a)

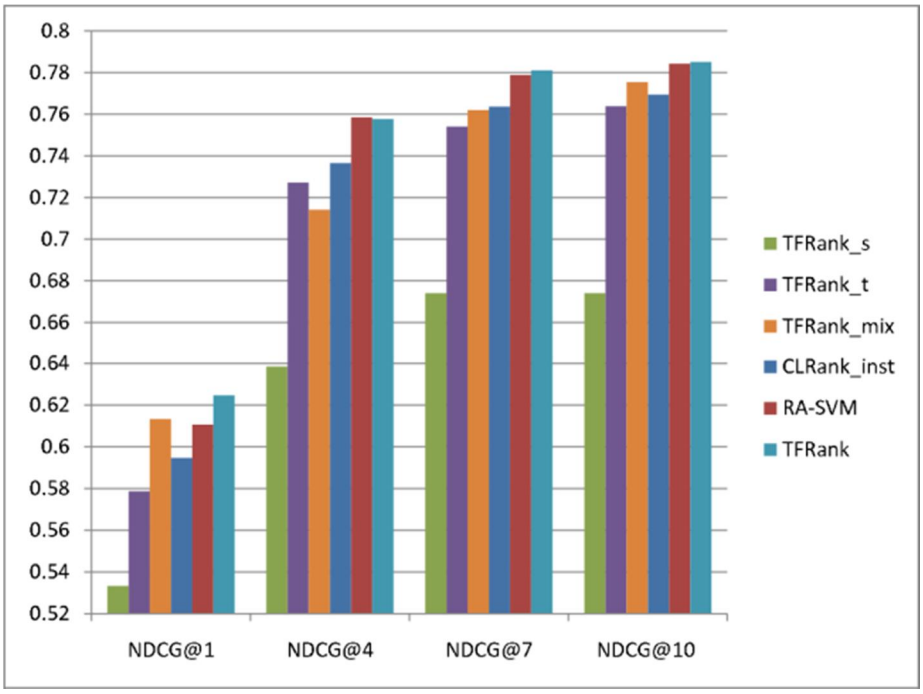



Figure 5.2 (b)

Figure 5.2 (a) reports NDCG values for 5 queries on HP2004 dataset and (b) reports NDCG values for 10 queries on HP2004 dataset.

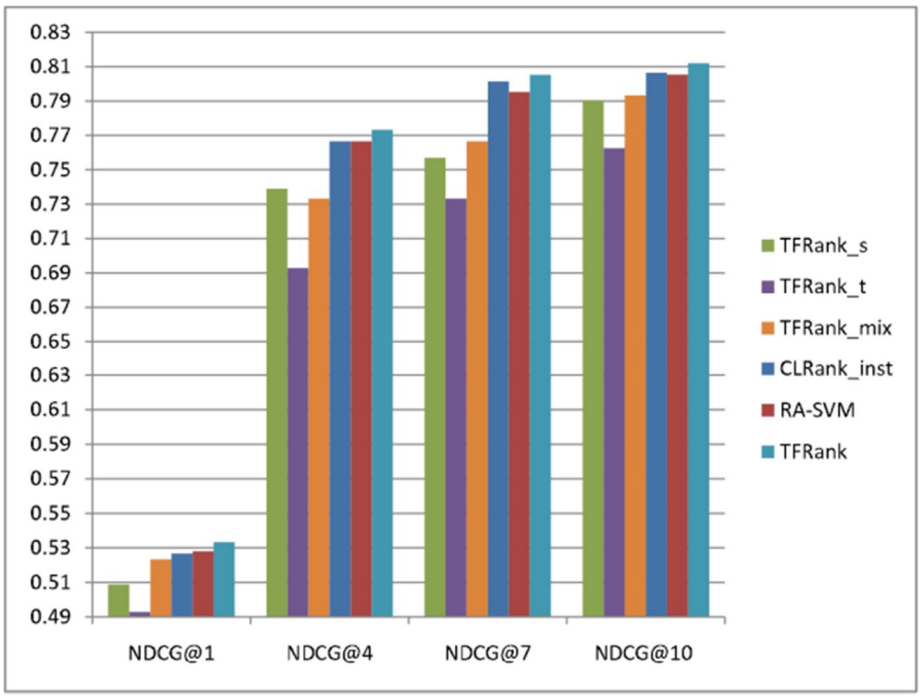



Figure 5.3 (a)

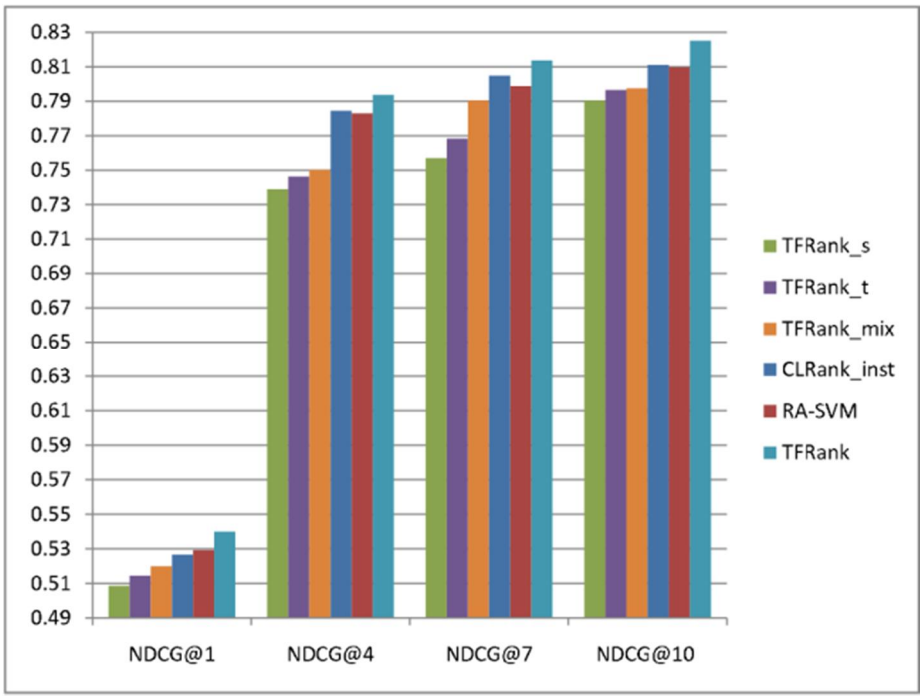



Figure 5.3 (b)

Figure 5.3 (a) reports NDCG values for 5 queries on NP2004 dataset and (b) reports NDCG values for 10 queries on NP2004 dataset.

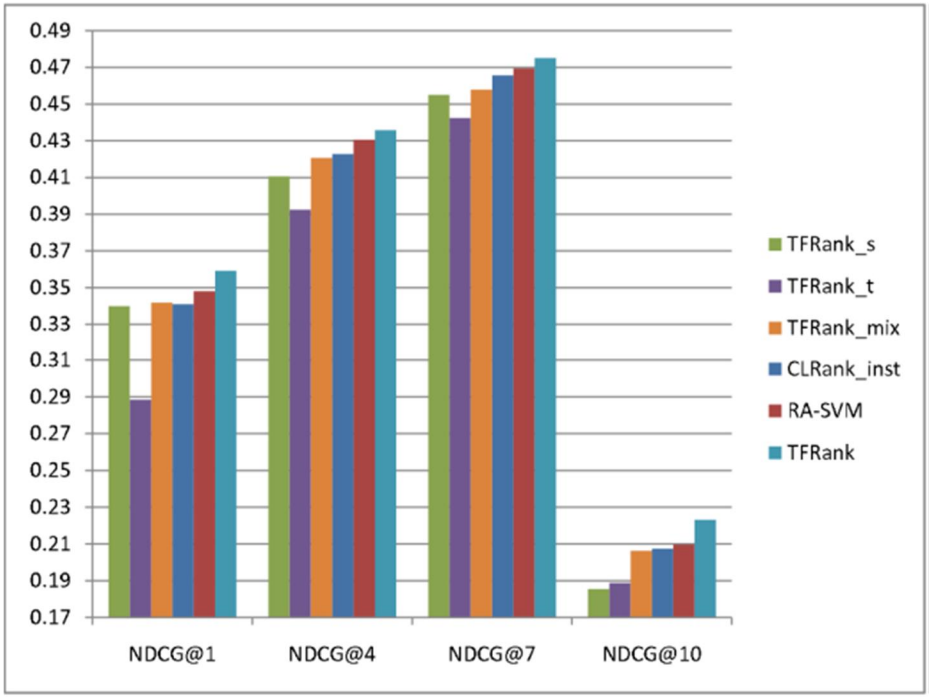



Figure 5.4 (a)

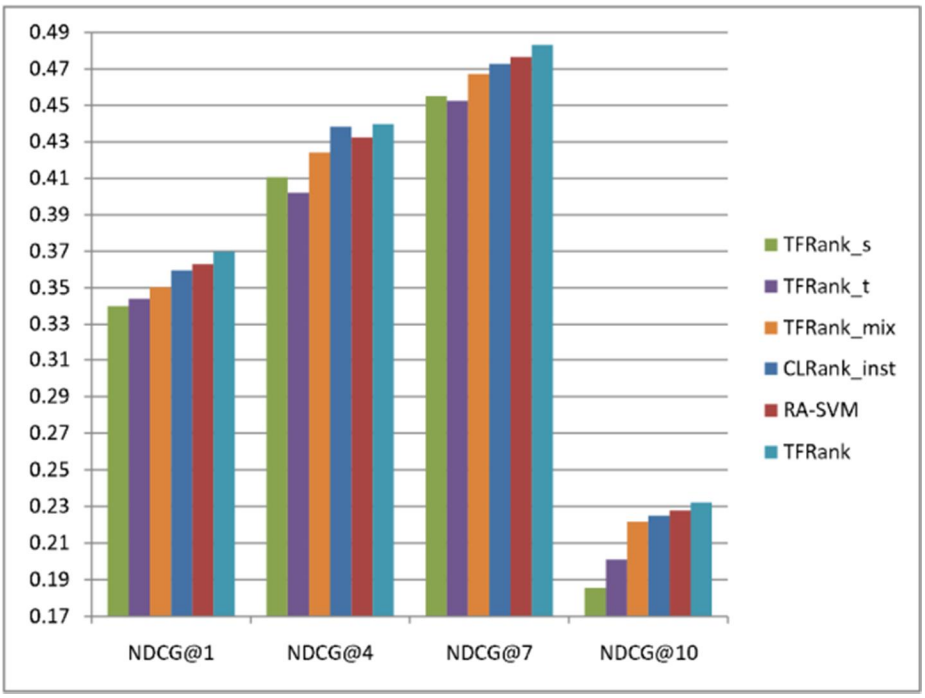



Figure 5.4 (b)

Figure 5.4 (a) reports NDCG values for 5 queries on the MQ2008 dataset and (b) reports NDCG values for 10 queries on the MQ2008 dataset.

We also conduct the *t*-test on the improvement of TFRank over the other best baselines. The result shows that TFRank has significant improvement than the TFRank_s, TFRank_t and TFRank_mix three not ranking adaptation algorithms and achieves competitive performance compared to CLRank_inst and RA_SVM. For example, the p-value of TFRank and TFRank_s is 0.0275 with respect to MAP on TD2004 datasets, and the p-value of TFRank and CLRank_inst is 0.2871.

From the results, we can see that our sparse model adaptation algorithm TFRank outperforms the dense model adaptation algorithm RA_SVM. Our method gains sparse results. This is the main reason why our method performs better than RA_SVM, because TD2004, HP2004, MQ2008 and NP2004 have been shown to contain lots of noisy features in previous works [80, 81].

Demonstrated results tell us that: the sparseness in ranking adaptation algorithms can help to improve the ranking performance, and TFRank can achieve the state-of-art performance, and TFRank can achieve the state-of-art performance among cross-domain learning to rank algorithms.

# Chapter 6

## SCHOLAT: An Application in Academia Social Network

*This chapter presents a network platform called SCHOLAT [1] (http://SCHOLAT.com/). The platform is a scholar-centered social network designed to form an academic community that helps scholars to establish contacts. Scholat takes advantage of many of my doctor's work during the period. During this time, I also provided two new professional services that are very useful for researchers, namely XPSearch and XSRecom. Furthermore, XPSearch is a search service that offers the same name as the author of a vertical paper or other publication. XSRecom uses the theme community approach to provide users with a list of "referrals" that can help them find potential partners who share the same interest in research and may be interested in building partnerships. XPSearch and XSRecom are built to improve search efficiency in terms of both papers search and people search so as to provide more partnership opportunities for researchers.*

## 6.1 Introduction

Nowadays, the Social Networking Sites (SNS) have become more and more important in our daily life. Because of the improving of WEB 2.0 technology, many different SNSs have proliferated over the past few years, such as Facebook , LinkedIn and Tweeter. Those platforms are useful for users in different fields. However, many professionals, particularly scholars, still find themselves being flooded with too much information. More importantly, they expect SNSs to provide them not only the chance to know new people or share moments with family and friends but also provide them with some particular academic services for supporting research and expanding their academia networks. Thus, we propose an innovative scholar oriented social network (SOSN) platform called SCHOLAT.com. In addition to providing normal social network functions, such as personal space, SCHOLAT provides two creative and useful services for scholars, namely XPSearch and XSRecom, which will be emphatically introduced in section 6.5 and 6.6 of this chapter:

XPSearch: This service provides vertical search of research papers with author name disambiguation. The user can not only enter the keyword search article but also search for an author's name and retrieve the visualization of the disambiguation results of all his / her publications. We use information from adjacent pages to improve the performance of name disambiguation.

XSRecom: The service provides the user with a list of recommended scholars from a community based on the combination of user links and content information. This tool can help users to find potential partners who share the same interest in research and may be interested in building partnerships. The novelty of XSRecom is that we put user links and content information so that scholars recommend more appropriate and accurate.

This chapter is organised as follows: Section 6.1 will briefly explain the motivation that encourages us to build this platform and will introduce the system architecture in detail. Section 6.2 will introduce a creative semantic description method used in this system. The next two sections 6.3and 6.4 will be the core sections that introduces the two features of this system mentioned above. Section 6.5 will concludes this chapter and presents the future expectation for this work.

### 6.1.1 Project Motivation

The major motivation for us to build SCHOLAT is that there is a significant need for a dedicated online service particularly for researchers in the era of Web2.0. From my survey, many universities, colleges and institutes in China and many all over the world only have simple and static web pages presenting information about their academic staffs. As the information updating processes of these websites are strictly under the management of the IT service's personnel of the university, much of the information presented on these websites is outdated. Even worse, they can only update information for the website based on the information provided by different schools. Generally speaking, academic staffs have no access to modify their own information in university websites.

Unfortunately, timeliness happens to be one of the most important criteria in academic projects, since research projects are usually limited by a specific funding/budget that often comes with a constrained timeline. Thus it rather inconveniences for researchers to find research partners from outdated web pages, not to mention of building up cooperative relations with them promptly. There are massive demands for the solution to solve this issue and two innovative functions have been built to help to solve it, which will be introduced in chapter 6.3.

### 6.1.2 System Architecture Overview

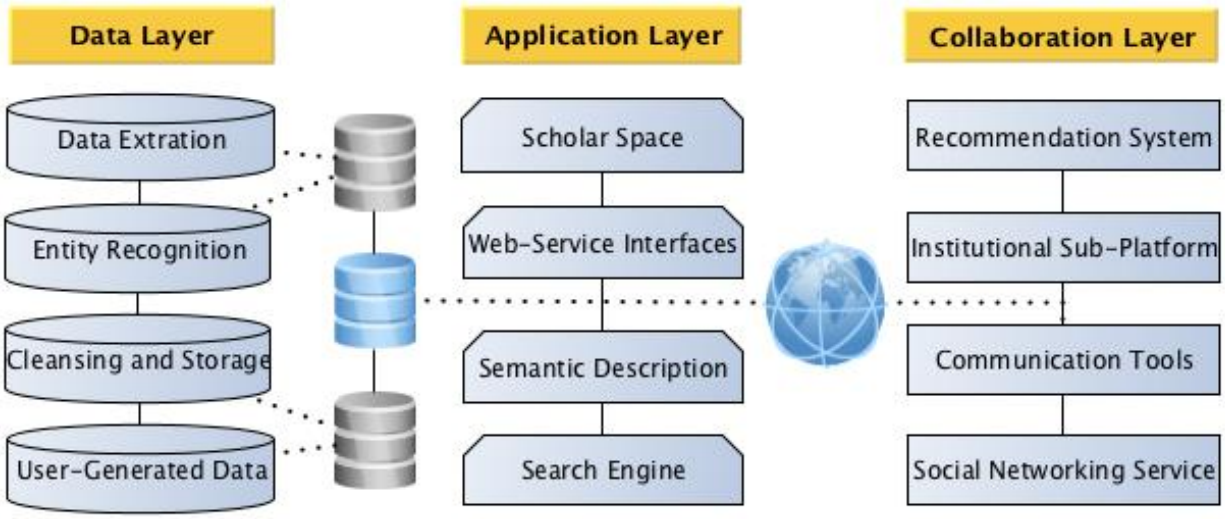



Figure 6.1 SCHOLAT Layered Architecture

As shown in figure 6.1, SCHOLAT is built on a layered architecture which consists of the data layer, the application layer and the collaboration layer. In the data layer, initial academic information was extracted mainly from the Internet by crawlers we deployed and then cleansed for preparing the next Named-Entity Recognition process and finally stored in distributed data stores. Also, user generated data is accumulated while more users activities are takeing place in SCHOLAT. With those processed information prepared in the data layer, the application layer provides users with some basic services, such as: an online personal academic space for managing personal profiles; a semantic description service that helps people to gain better understanding of the other's research works; a powerful search engine dedicated for searching academic information, and last but not least, Web-Service Interfaces which allows SCHOLAT to provide information service to other organizational systems. The collaboration layer provides more services based on academic relationships between researchers including, but not limited to 1) a recommendation system for academic work and people; 2) on-site communication tools such as on-site mail and instant messaging; 3) institutional sub-platforms which provides a team-based collaborative space allowing a research group or even a whole research center to set-up a virtual community for organizing activities, managing people and publishing information.

### *Data Layer*

The data layer is the fundamental layer where our vision to build "Scholar Oriented Social Network (SOSN)" takes shape. Workflow of Data Layer shows in figure 6.2:

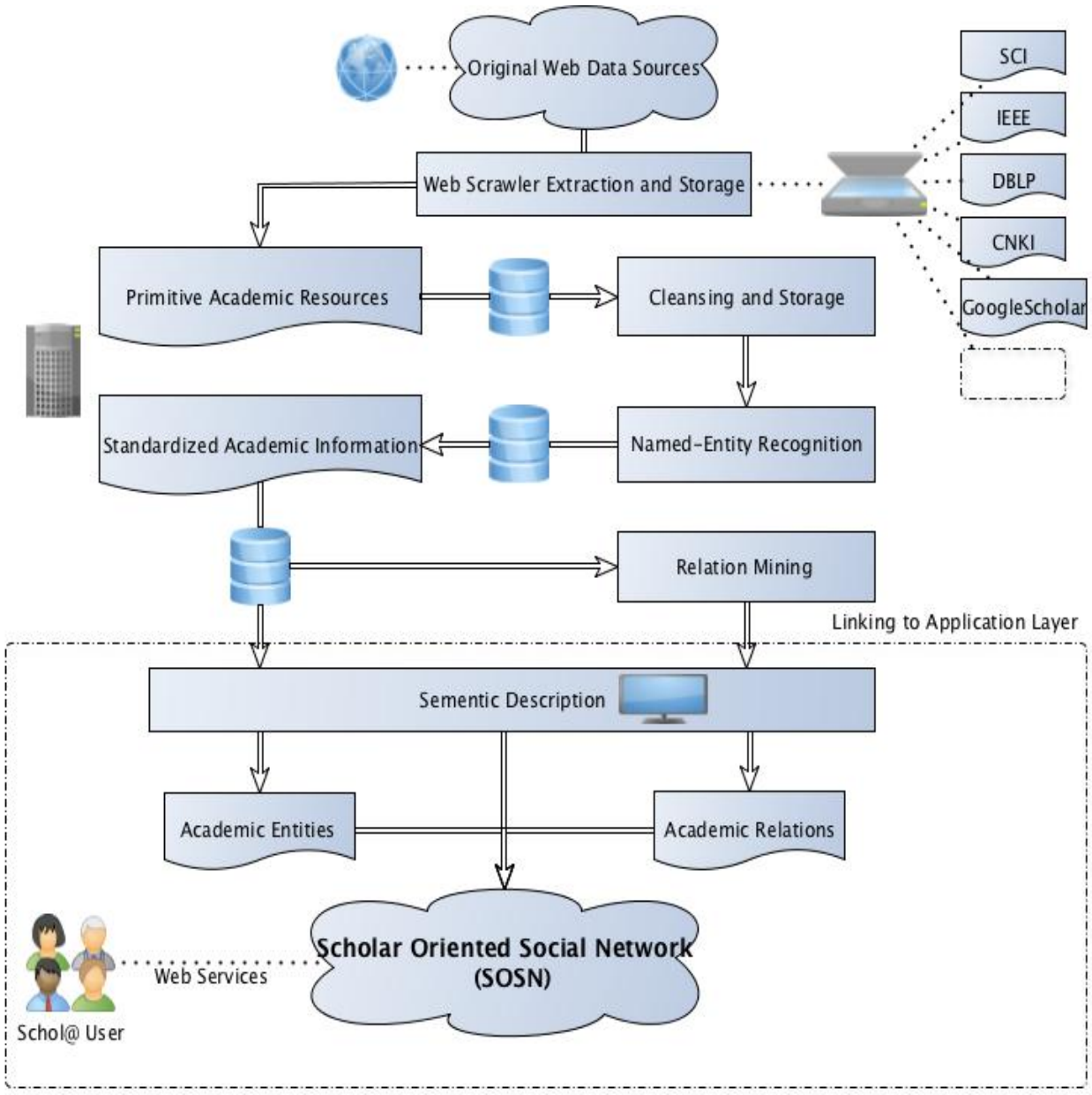



Figure 6.2 Workflow of Data Layer

### First Layer: Data Collection, Cleansing and Storage

The data collected in SCHOLAT mainly come from two ways: the user generated data and data crawled from the Internet. The user generated data accumulates as the users use SCHOLAT. These data can be divided into three categories: the user profile data, the team/institution data, and the user relationship data. They are very valuable and are the core assets of SCHOLAT, are also the data that I have used in Chapter 3 and Chapter 4. Meanwhile, we have designed two types of crawlers to fetch open-source academic information from the

Internet. After collecting and cleansing, these primitive academic resources are standardised into an acceptable and meaningful form and store into our distributed databases. Then they can be used in the XPSearch algorithm to support SCHOLAT search engine and provide further services for the registered users in SCHOLAT, such as paper sharing and push publication information, etc. When the crawler fetches pages from the academic sites, it does not need to fetch and store everything it meets. With the well-designed structure, the crawler knows exactly how the academic information is displayed on the page, so it can directly get the academic information itself without fetching other abundant content. Therefore, the data cleansing job for the traditional crawlers is completed in the data collecting phase, which significantly simplifies the following works. Using these crawlers, we collect about 90 million academic items, including papers, books, journals, conferences proceedings, etc. Storing so much data is a great challenge for the traditional RDBMS. Here we adopt a famous open source NoSQL database, HBase, to store the big data. Detailed information will be provided in section 6.3.2 of this chapter.

### *Second Layer: Entity Recognition and Relation Mining*

Entity Recognition and Relation Mining is a fundamental research area in providing academic information service. Accurately and efficiently recognising entities is the most fundamental process before analysing entities' relationships, which in our case is to obtain academic relationships. Thanks to the development of our search engine, nowadays Entity Recognition has been intelligently integrated into the process of data collection and cleansing. Hence we will focus more on Mining Relations in this layer.

The academic community is a complicated social network. Within this network, there are many entities such as scholars, research project, academic achievement, research fields and research teams. A relation between two entities can be regarded as a certain connection within a particular period and space [60]. For example, there is 'co-authors relation' between scholars, and there is 'belong-to relation' between research team and scholars. In our Scholar Oriented Social Network, we mainly focus on scholar oriented relationships. To avoid system redundancy we found out many complicated relationships can be calculated from three basic relations: 'possession relation' between authors and papers; 'originate relation' between papers and journals; 'reference relation' between papers. Hence we pick up three types of entities of a

scholar: Author (Ai), Paper (Pi) and Journal (Ji) and we can denote 'possession relation' as AP, 'originate relation' as PJ and 'reference relation' as PP. Then we can calculate other relations just to name a few:

- **Co-author relation:** CoA = $A_iP*A_jP$, where $CoA_{ij}$ indicates how many times does author $A_i$ and $A_j$ work together, and when i = j, $CoA_{ij}$ means number of the paper belongs to author $A_i$
- **Author-Reference relation:** ArP = AP*PP, where $AcP_{ij}$ indicates how many times does author Ai referenced paper $P_{j.}$
- **Author-Journal publishing relation:** AJ = AP*PJ, where $AJ_{ij}$ indicates how many times does author Ai publish paper onto journal $J_{j.y}$

### *Semantic Description*

Semantic Description is a key process that links between data layer and application layer and also one of the major components of the application layer. We will expand on the details presented in Section 6.2 of this Chapter. In brief, we have innovatively created an ontology called SOSN Ontology which provides a unified meta-data descriptions standard to describe academic information from multiple sources with different structures by using semantic web technologies and existing metadata standards.

## *Application Layer*

The Application layer is the second layer of our platform which provides basic services to our users. Moreover, some of our most innovative and creative methods have been implemented into practice in this layer including SOSN Ontology and Academic Search Engine.

### *SCHOLAT Personal Space*

A personal space has been provided to our users to manage their personal, academic information as a digital profile that can be accessed in any network environment. As we can see from figure 6.3, different from general social networking sites, we are focusing on presenting academic information such as papers, books and projects.

As mentioned in section 2, every user in SCHOLAT will obtain an easy-to-remember, fixed domain name for the personal link that makes it convenient for users to share their homepages to others (http://www.SCHOLAT.com/aarontang where aarontang is a customised name). This

homepage not only provides users with an online-storage space to store profiles, papers and other documents which can be downloaded to any terminal devices that can be accessed via the Internet but also founds the basic data set for further collaboration purpose. Users may experience more functions such as sharing papers, once connections have been established with other researchers or joining in research teams and groups. Personal space is an interface for users to meet up with potential research partners similar to how we dress up in real life. We also provide many customised formats for displaying homepage that is to best present the content regarding different languages such as Chinese characters.

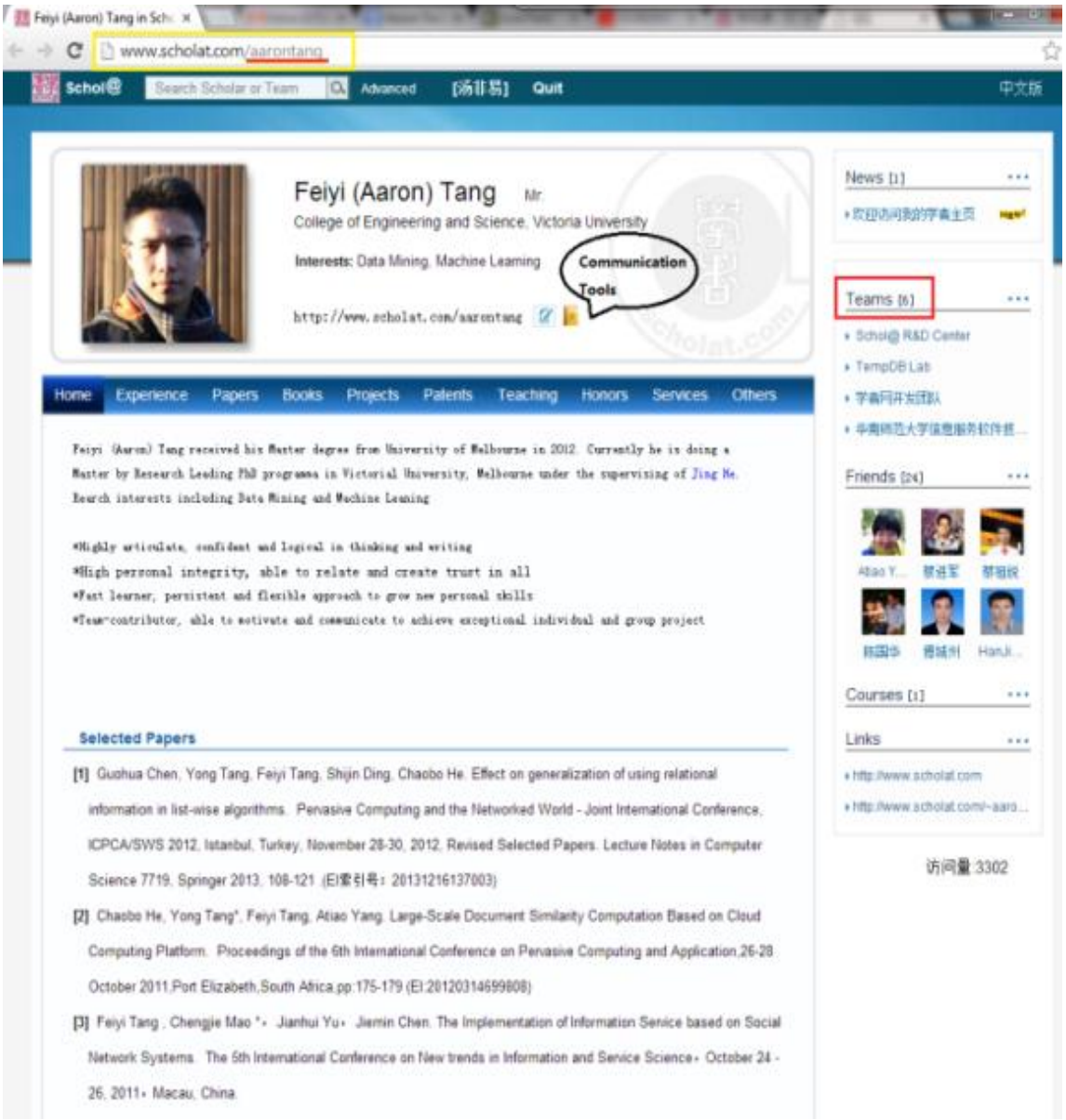



Figure 6.3 Example of a Personal Homepage

*Web-Service Interface*


Figure 6.4 SCNU's staff information service based on SCHOLAT

Web service interfaces is a service that aims to help researchers or research institutions to develop their own applications based on SCHOLAT. The idea is that our users (including institutions) can customise their homepages at their official institutional websites in a way that information displayed on those websites are extracted from SCHOLAT by using related web service interfaces. Thus on one hand, researchers can keep their information on official websites accurate and updated promptly; on the other hand, the institution can reduce cost on maintaining their staff's information in a long term manner. An example web page of a staff's information displayed officially on South China Normal University (SCNU) has been provided in Figure 6.4. As shown in the address bar of the web page the 'scnu.edu.cn' address indicates that it comes from the genuine official website of SCNU. The second-level domain name 'scholar' implies that it's an independent system for SCNU, whose content comes from the web services provided

by SCHOLAT. We have developed various web service interfaces to satisfy different needs of gaining different information from SCHOLAT and every interface is implemented using lightweight REST (Representational State Transfer) protocol so that it can be accessed by crossing applications through the URL address. Detailed descriptions of some classical web service interfaces are shown in Table 6.1:

Table 6.1 Web service interfaces

| Type | Name | Address |
|---|---|---|
| Personal Information Service | Profile | rest/JUserProfile/account/callback |
| | News Bulletin | rest/JPostMessage/account/callback |
| | Work Experience | rest/JWorkExperience/account/callback |
| | Scholar Title | rest/JScholarTitle/account/callback |
| | Education Background | rest/JEducation/account/callback |
| | Honor | rest/JHonor/account/callback |
| Academic Resource Service | Publication | rest/JPublication/account/callback |
| | Paper | rest/JPaper/account/callback |
| | Patent | rest/JPatent/account/callback |
| | Project | rest/JProject/account/callback |
| Social Network Service | Friends | rest/JFriends/account/callback |
| | Vistors | rest/JVistors/account/callback |
| | Friends Message | rest/JFriendsMessages/account/callback |
| | Social Tags | rest/JScholarTag/account/callback |

### *Search Engine*

SCHOLAT's Search Engine takes a very important role in the platform, as its function extended from data layer to application layer and also provided back-end services to

Collaboration Layer as well. More detailed description will be provided in Section 6.5 of this chapter.

### *Collaboration Layer*

Collaboration Layer is an upper layer where services provided here are based on established relationships. Hence, as the layer's name indicated, provided services are mainly for collaboration purpose.

#### *Friend Recommendation System*

Whether users can establish collaborative relationships successfully are largely depends on whether they can meet up with resourceful friends. In our case, whether they are sharing same research interests. In many social networking sites, we are experiencing information overload from random friend recommendations since basically those systems are just pushing anyone who may only share little elements in our life onto our screen. And the result is we need to skip most of the recommended people by scrolling down and down to the end of the list to see only a few are added, and the truly interested friends still needed to be added manually. Worse, many friends added in this way may just be acquaintances with little interactions. Bearing our purpose in mind, we provide feedback interfaces that adopt user generated data to help us improve our recommendation quality as a long-term strategy. On the other hand, we also adopted various strategies and considered elements in multidimensional ways to make sure our recommendation really helps our users to gain resourceful and active research partners. More specific introduction will be provided in Section 6.4 of this Chapter.

#### *Institutional Sub-Platforms*

Different from basic teams that form within SCHOLAT's main site or Web-Service that provide information services within other organisational websites, Institutional Sub-Platforms is a creative service-platform that hosts smaller institutions or research centres to help them manage their people, activities and projects.It aim is to help them reduce cost from developing and maintaining their own independent websites. More importantly, these research institutions can also enjoy advantages to be the institutional partner of SCHOLAT to get closer to a larger and more resourceful academic community. Below is an example of the sub-platform for Research Center of Software Technology for Information Service. As shown in figure 6.5,

similar to scholar's personal space, our institutional partner will also obtain an easy-to-remember and fixed domain name for their sub-platform (scnucs.scholat.com where 'scnucs' is the customised name for the institution). Moreover, as an institutional partner to us, all their personnel are registered automatically and can use SCHOLAT account to log in to the service.


Figure 6.5 Institutional Sub-Platforms

### *Real-Time Message Pushing Service*

As more and more researchers register into SCHOLAT, there are more research activities running and much more information floating among users and teams and institutional platforms.

Other than publishing this information on the homepage of SCHOLAT which still requires users to explore manually, a real-time message pushing service is provided to our users to enable them to know the latest news and events happening around them whenever they are online. Message pushing service is a very small but useful component in SCHOLAT and plays a very important role in improving user viscosity. Its main function is to firstly monitor various events and publications generated by user, teams or institutions such as announcements, friend requests, team invitations, friend's dynamics activities, messages from onsite emails and instant communication tool. and then instantly push them onto users' screen.

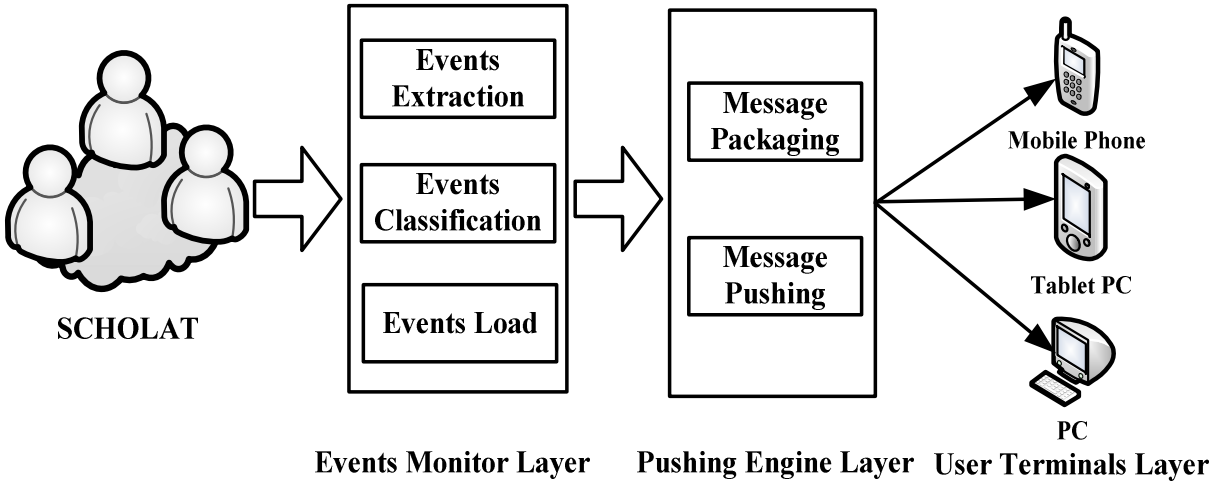



Figure 6.6 Framework of Message-Pushing Service

As shown in figure 6.6, the framework comprises three layers: events monitor layer, pushing engine layer and user terminals layer. In the events monitor layer, information of various events are being extracted and classified into many categories, such as updating profile, sharing papers and so on and then events load module stores these classified events in the events server from which pushing engine can fetch the latest events. In the next layer, information of classified events is being modified regarding summarising and packaging. Event information is summarised into short briefs and then packaged with some required information such as terminals address and priority level. Finally, these standardised briefs are further pushed into the next layer to reach multiple types of terminals, such as PC and mobile device. Corresponding pushing example is shown in Figure 6.8.

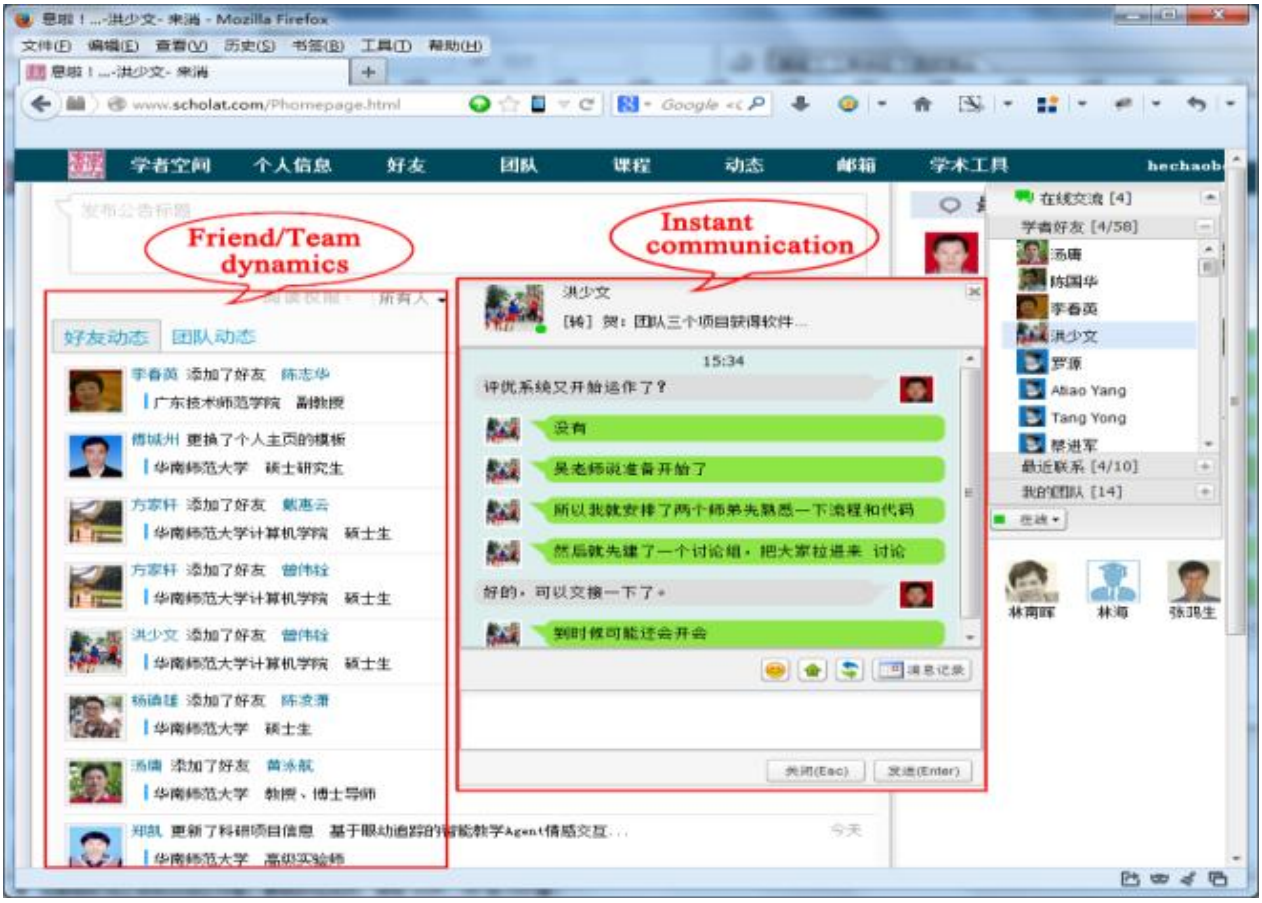



Figure 6.7 Pushing message to PC Web Terminal

## 6.2 SCHOLAT SOSN Ontology

A certain process is required to enable information exchanging and sharing between systems on the Internet, which is Semantic Description. Such process can be executed by markup languages such as Resource Description Framework (RDF).

### 6.2.1 Related Works and Referred Metadata Standards

Providing a unified way to describe the abundant and diverse academic information has been a debatable and challenging issue around the world. Newman [104-106] firstly raised the idea of analysing the structure of scientific collaboration network by calculating statistics about coauthors relationships. Functional Requirements for Bibliographic Records (FRBR), which is an entity-relationship model developed by the International Federation of Library Associations and Libraries (IFLA), is an application model for describing bibliography records in the area of

academic publication. Furthermore, Scholarly Works Application Profile (SWAP) created on top of FRBR as its semantic implementation further introduced a way of describing electronic publications such as peer-reviewed journal articles, work papers and theories. On the other hand, FOAF [21, 155] which stands for Friend of a Friend is a metadata standard focus on describing people and those relationships among people that have become the basic element of a virtual community. Another widely accepted metadata standard, Dublin Core [127], is a set of predefined properties for the description of documents in multi-disciplines. Finally, the MarcOnt ontology is also a unified bibliography proposed by Dabrowski [40] which is created based on analysis on a wide range of existing literature standards, including MARC21, ISBN, BibTex, FRBR. that first explores the field of semantic description of academic literature. Based on these experiences and semantic web technologies mentioned above, we have innovatively created an SOSN Ontology dedicated for describing academic information which has been implemented into SCHOLAT and has finally solved the issue of inconsistent academic information.

### *6.2.2 SOSN Ontology Model and Structure*

By analysing those function's characteristics and features of Data Layer and Application Layer we have created an SOSN Ontology Model as shown in Figure 6.8, and we have established the most basic vocabularies in SOSN Ontology as Scholar, Academic Work and Academic Team.

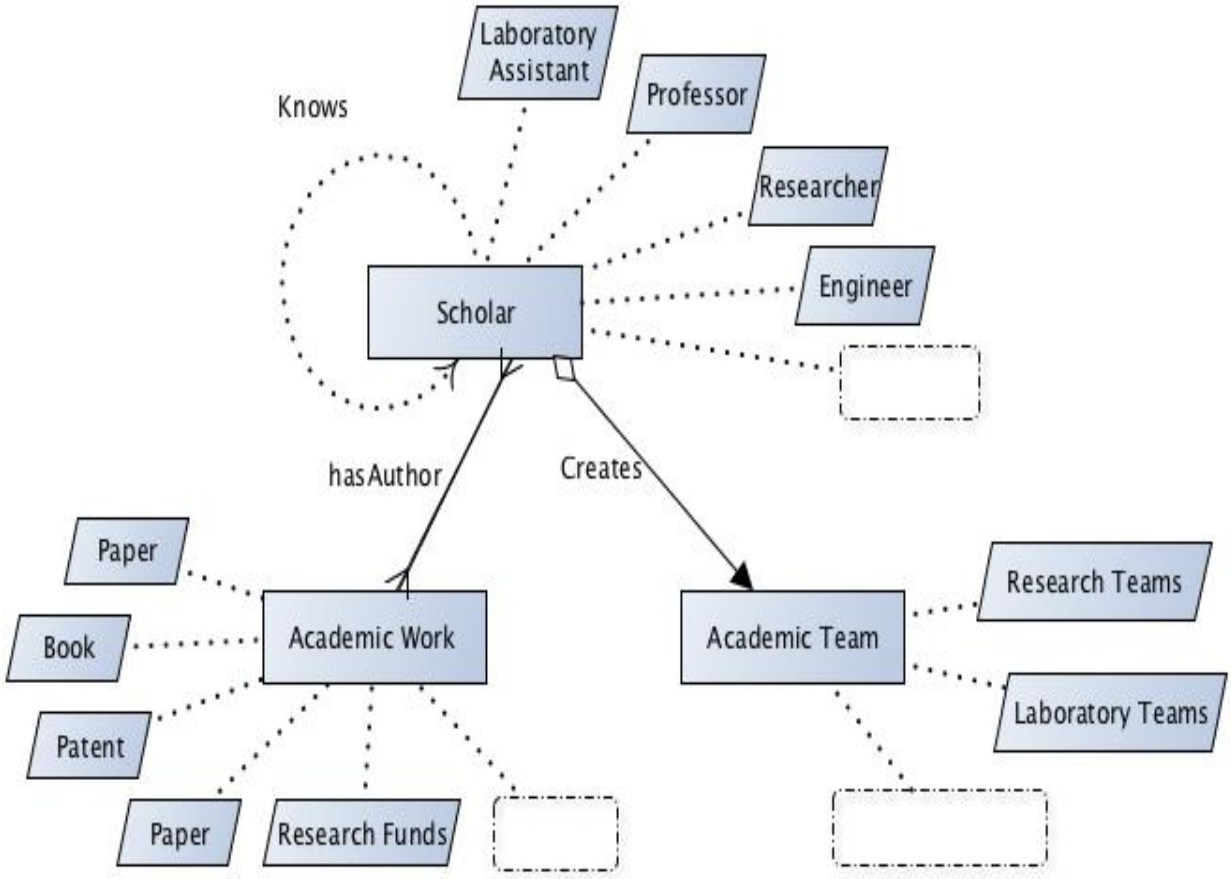



Figure 6.8 SOSN Ontology Model

Semantic Description of SOSN is based on the existing metadata, and here we introduce the structure by defining Classes and some of the key properties of SOSN ontology in Table 5.

**Scholar:** this class including all peoples that works in the scientific research field of SOSN including Professor, Engineer, Researchers, Scholars, and Laboratory Assistant, etc. The Scholar class is primarily defined in FOAF.

**Academic Team:** this class represents all scientific research teams in SOSN, including Research Laboratory, Research Teams and any Research Group that sharing common research interests and usually composed with properties of Scholar Class that have established connections.

**Academic Work:** this class represents all academic resources in SOSN, including papers, journal articles, books, scientific projects, reports, conference proceedings, research funds and patent, etc.

Table 6.2 Key properties of SOSN ontology main classes

| Class Name | Properties |
|---|---|
| Scholar | user account, surname, first name, gender, birthday, avatar, workunit, telephone, address, degree, research interest, academic title, mailbox, homepage, qqChatID, msnChatID, favouriteSite, hobby, knows,publications, hasRole, cites, creates, hasCommonField, hasCommonTeam, hasCo-author, hasInstructor, hasStudents |
| Academic Team | team account, team name, team creator, administrators, createDate, teamMembers, teamPublications, introduction |
| Academic Work | title,authors, abstract, keywords, contents,, source, area, chapter,issue, volumes, issueDate, publisher, copyright, project, projectType, accessRole |

## 6.3 SCHOLAT Search Engine: XPSearch

### 6.3.1. Overview

Academic search is a vital activity for researchers. In SCHOLAT, we design an academic search engine that provides publication search and scholar search services to users. We implement a crawler based on Nutch {http://nutch.apache.org} to fetch different kinds of academic websites accordingly. As a result of the continuous and iterative crawling, the SCHOLAT search engine has now indexed more than one hundred million citation records, including books, journals and conferences. The XPSearch service is implemented for users to search publications they needed. In particular, this tool integrates an important function: author name disambiguation.

Given an author name, most academic search engines just give the complete list of publications with the same name without further processing. Researchers may experience difficulty in focusing on the exact author in which they are interested. Though DBLP {http://dblp.uni-trier.de/} provides name disambiguation service, but it works only for few people with the same name. Many research works have been conducted in recent years to address

this problem. They can be divided into three categories: Classification Methods [59], Clustering Methods [61] and Probabilistic Models Method [62].

According to previous research, the co-authorship plays a very important role in disambiguating authors [48, 73]. And can be easily extracted from the co-author list in the publication. In this section, we present a co-authorship based model to solve the problem. The details are given in the following sections.

### 6.3.2 Data Collection Method

#### *Nutch Crawler*

Nutch is a well-known open source web crawler. It's a top project in the Apache Software Foundation. Here we adopt Nutch as the academic information crawler for SCHOLAT search engine.Because Nutch provides a flexible plugin system. Our main work is to develop a plugin for Nutch, which extends the extension point HtmlParserFilter. When the Nutch crawler fetches a web page, the plugin we developed will be started to parse the web page and extract the academic information it contains. This process is described in the following steps:

1. Get the URL of the fetched web page.
2. Construct an academic information extractor according to URL's regex pattern.
3. Use the extractor to extract academic information from the web page.
4. Store and index the retrieved academic information.

In step 2 of this algorithm, the academic information extractor will be constructed according to the URL of the fetched page, and then it will be used to analyse the content of the web page and retrieve the desired academic information. In many web sites, the contents of their page are generated dynamically, but the structure is often same with each other. In this case, their URLs often obey some rules that can be expressed by the regex expression. These pages with the same structure can be processed by the same academic information extractor. We write the correspondence between the URL pattern and academic information retriever in a configuration file. When a web page is fetched, the extractor can be constructed accordingly.

The pages with different structures should be processed by different scholar information extractors. Writing an extractor is complex and time-consuming. However, different extractors

have similar functions and structures. To make the development of different extractors easier and more robust, we design a language to describe the attributes and extracting process for the extractors, the Scholar Information Extractor language, or SIEL for short. The following is an example extractor file written in SIEL.

```
@package: com.scholat.fetch
@class: WanfangJournalExtractor
@desc: 万方期刊
@itemDesc:{authors}. {title}. {source}, {year},{vol}({issue}).
@type: JournalPaper
@lang: zh_CN


detailDiv := div{class:detail_div}
title <= $detailDiv->h1
english_title <= $detailDiv->h2
PDF <= $detailDiv->a{class:downloadft}.link
abstract <= $detailDiv->dl{class:abstract_dl}->dd
fields := $detailDiv->dl{id:perildical_dl}->dd[*]
author <= $fields[0]->a[*].linktext
organization <= $fields[1]
source <= $fields[2]->a.linktext
journal <= $fields[3]
yearVolIssue := $fields[4]
[[
String year=yearVolIssue.substring(0,4);
String vol=yearVolIssue.substring(6,yearVolIssue.indexOf('('));
String issue=yearVolIssue.substring(yearVolIssue.indexOf('(')+1,
yearVolIssue.indexOf(')'));
item.setYear(year);
item.setVol(vol);
item.setIssue(issue);
]]
```

Figure 6.9 an example SIEL file

The SIEL file is composed of two parts: the declaration part and the main body part, and the two parts are separated by blank lines. There are five different statements:

- Declare Statement: declares the attributes of the extractor, such as the package folder, the class name of the extractor, etc.
- Definition Statement defines a variable that can be used in the following statements.

- Storage Statement gets the property of the located node and store it into the specified field of the academic information.
- Remark Statement: provides a way to write remarks in the SIEL file.
- Embedding Statement: enables users to embed Java statements into the SIEL file.

Among these five kinds of statements in SIEL, the definition statement and storage statement are the most important and commonly used. They share a similar structure: node locating chain. The difference is that in the storage statement, a property of the located node (or node array) should be appended to the end of the node locating chain. If no property is specified, then a default property text will be used.

A node locating chain is a chain of node locating symbols. The node locating symbol can locate a node in the DOM tree with the specified tag type and attributes. The nodes are connected with symbol '->', which means that they are of parent-children relationship. Then using this chain, we can locate the node we wanted from top to down in the DOM tree.

We design a parser to parse the SIEL file and translate it into a Java class. Each statement is separately parsed into a Java statement block, and after all the statement is parsed, these Java statement blocks can be combined into a complete Java class.

### *Browser Crawler*

With the development of web technologies, now more and more web sites are constructed via the Ajax technology. In these sites, the complete content of the web page is not downloaded at the initial loading time. Instead, just a basic page is shown, and extra content is loaded only if the user is interested. In this case, the traditional crawler like Notch can not get the complete content given the page URL only. The JavaScript of the web page should also be parsed, and the user behaviour should be imitated to get the desired content, which is very costly and inconvenient. To tackle these challenges, we design a new kind of crawler, which runs as a plugin for the web browser. The architecture is described in the following figure.

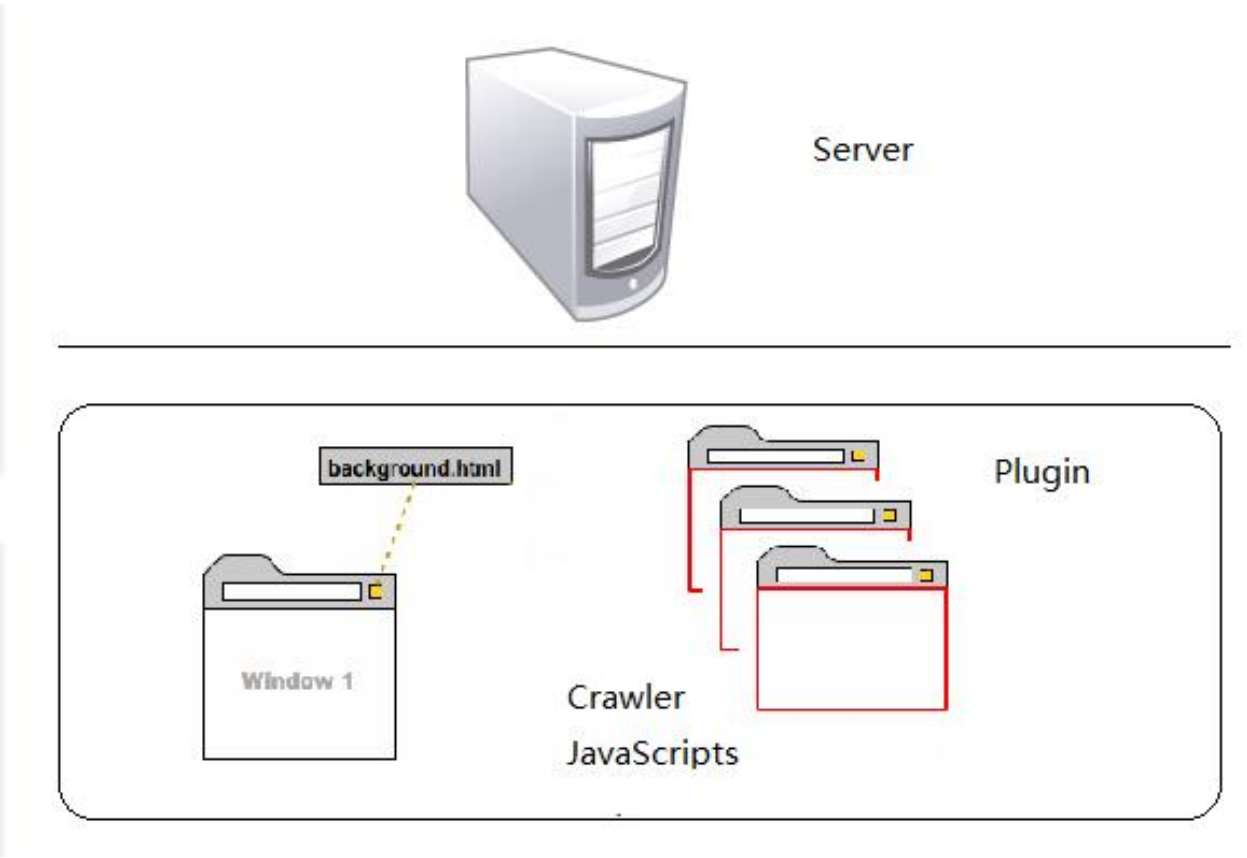



Figure 6.10 Architecture for browser crawler

As shown in the figure 6.10, the browser crawler I designed is divided into two parts, the server side and the plugin. And the plugin is future divided into two components, namely the background page and the content JavaScript. When the Chrome browser starts, the plugin's background page is initiated in the background.

When the plugin gets started, it will ask the server for a crawling task using Ajax. Once getting a task, it will open a tab in the browser and go to the start page for that task. When the page starts to load, the content scripts of our plugin will begin to run. The task of the content scripts is very simple: it does nothing but loads the real crawler javascript from the server, and when the real code gets ready, it will execute it using the eval function. When the academic information is fetched, it will be transferred to the background page temporally and from there sent to the server in a batch.

It has many advantages in this design compared with the traditional crawler:

1) The browser crawler is a truly distributed system and it has the ability to run across the world: all it needs is to connect to the server. Thus it is very to scale.

2) What you see is what you get: the crawler runs on the web page and gets the academic information just like a real human does. And it can imitate a real user's behaviour to get the desired content.

3) Easy to observe: if there is something wrong with the crawler, you can know it at the first time.

### 6.3.3 Author Name Disambiguation

Researchers are not able to carry out academic research with a variety of digital libraries or academic search engines, such as Google scholars and Microsoft academic search. When using a digital library or an academic search engine, the author frequently encounters ambiguity in his or her name. When we lock the author's name, these systems usually return a full list of publications. It requires a lot of queries to do this search [42, 43, 97]. It has the following negative effect: 1) normal, focus on the precise author he actually interest is very inconvenient; 2) research manager, determine the author's achievement is very difficult, so it may cause confusion when making such as promotion or research grants to decide; 3) it cannot establish the author list of publications, which cannot conduct further semantic analysis, such as determining the academic performance, the author summed up the research interest, or from his team, which in the current academic circles is very useful, social network sites. Therefore, it is very important to solve the ambiguity problem of the author [140, 145].

#### *Overview*

In recent years, a lot of research work has been carried out on this issue. They can be divided into three categories:

**Classification-Methods [59].** The so-called classified classification, simply speaking, is based on the characteristics or attributes of the text, divided into existing categories. For example, in Natural Language Processing NLP, the text classification we often refer to is a classification problem, and the general pattern classification methods can be applied to text categorization. Including the commonly used classification algorithms, decision tree

classification, Bias simple classification algorithm (native Bayesian classifier), based on support vector machine (SVM) classifier, neural network method, k- nearest neighbor (kNN), fuzzy classification method etc..

Classification, as a supervised learning method, requires that the information of each category must be known in advance and that all the items to be classified to have a category corresponding to them. But most of the time these conditions are not met, especially when dealing with massive data,. If the data meet the requirements of classification algorithm through the pretreatment, then the cost is very large, so consider using clustering algorithm.

**Clustering-Methods [61].** Clustering analysis is an important human behavior. As early as when they are a child, a person learns how to distinguish cats, dogs, animals and plants by constantly improving the clustering patterns in their subconscious. At present, it has been widely studied and successfully applied in many fields, such as pattern recognition, data analysis, image processing, market research, customer segmentation, Web document classification. Clustering is in accordance with a specific standard such as distance criterion. A data set is divided into different classes or clusters, the similarity of the data objects in the same cluster as big as possible, at the same time, data objects are not in the same cluster in the sex is as large as possible. After clustering, the same kind of data can be clustered together as much as possible, and different data can be separated as much as possible.

**Probabilistic-Models-Method**. The dependencies between actual authors and the author references are depicted in a probabilistic model, such as HMRF [162] or Naive Bayes [60], and then iteratively calculate the parameters in the model using EM or Gibbs Similarity methods [30]. Given a query string of a user, there is a collection of all relevant documents relative to that string. We consider such a collection as an ideal result document set, and we can easily get the result document after giving the ideal result set.In this way, we can treat query processing as the processing of ideal result document set attributes. The problem is that we don't know exactly what these attributes are, and what we know is that there are index terms to represent them. Since these attributes are not visible during the query, it is necessary to estimate these properties at the initial stage. This initial phase estimate allows us to return an ideal result set for the first retrieved document set and produce a preliminary probability description.

### *System Architecture*

The six degree of separation is a famous theory in social networks. It illustrates the potential strength of a friend. Co-author is also a friendship, and we want to connect the author into a complete network. Then we can check the distance between the other through the network, which can be used to eliminate the ambiguity of the author. We put forward a new idea to build a creative network. We divide it into two stages. The author of the co-authored paper can be regarded as a member of the academic community. First, we gather these circles into atomic groups and then connect the clusters together.

### *Building of the Atom Clusters*

The atomic cluster means that we have a high degree of confidence that the name reference in it refers to the same actual author [163]. We can use the word "name reference" and "paper" according to the context. The basic idea is that we first have high confidence clustering for authors, and hope that these clusters can play a guiding role in the following clusters. In order to maintain the high accuracy of the cluster, we used the common author characteristics. We calculated the common co-author name differentiation and it was checked with the THRESHOLD. Only when the total name difference is greater than the THRESHOLD, we will add the name reference to the Atom cluster. After evaluating, we set the THRESHOLD to 0.1 in the system. Due to the initial setup, the initial atomic cluster may have many fragments. We first construct a cluster and then iteratively compare each pair. If the similarity exceeds the THRESHOLD, then we merge the pair. If there is no merge, the algorithm is completed. We use federated search sets to keep efficiency. The Pseudo-code is illustrated in Figure 12.

```
Input: A list of papers P which all share same author name
Output: A list of atomic cluster list C
Create an empty atomic cluster list c={Φ} ;
for each paper p_i in P do
    create an atom cluster c_i and add it to C;
end
while true do
    for each pair of atom clusters c_i and c_j do
        if similarity between c_i and c_j > THRESHOLD then
            merge c_i and c_j;
        end
    end
    if atom cluster list doesn't change then
        break;
    end
end
return C;
```

Figure 6.11 Pseudo-codes for Atom Cluster Construction

Because each name has a series of clusters, we only use direct co-authored. It can maintain high precision, but the disadvantage is to tend to split an actual author into several clusters. Therefore, it is necessary to connect these atomic groups. Some authors co-authored a paper, each author corresponding to a specific atomic cluster, author name. Then we can conclude that these atomic groups are interconnected by this piece of paper.

The association information can be used to further improve the cluster aggregation. For example, for the author named $a$, it has atomic clusters such as $A_a^1$, $A_a^2$,..., $A_a^k$.

If $A_a^i$ and $A_b^j$ are associated with other author name's atomic cluster $A_b^k$, then we can infer that $A_a^i$ and $A_a^j$ corresponding to the same cluster, and we can merge the two-atom clusters.

### 6.3.4 Experiment for XPSearch performance

Some comparative experiments to evaluate the performance of XPSearch have been carried out. A machine powered by an Intel Core(TM)2 Duo CPU (2.93GHz) with 4GB RAM, running

Windows 7 is used to carry out the experiment. Each experiment has run five times, and the average is reported here. XPSearch is coded in Java.

### Dataset

Due to the lack of standard dataset in the field of author name disambiguation, we construct dataset ourselves. The amount of authors and papers indexed in SCHOLAT is very huge, for example, the name of "Wei Wang" appears more than 20,000 times. Therefore it's impractical to label for all of these papers. We randomly select ten authors registered in our site SCHOLAT. These authors list their publications in their homepages provided by SCHOLAT. Therefore we can gather the ground-truth dataset easily. Table 6.3 lists the authors we selected.

Table 6.3 Dataset

| Author Name | #Actual Authors | #Publications |
|---|---|---|
| Biqing Zeng | 1 | 5 |
| Yuhui Deng | 1 | 7 |
| Yuncheng Jiang | 1 | 35 |
| Hai Jin | 2 | 96 |
| Yifu Jin | 1 | 3 |
| Ronghua Luo | 2 | 4 |
| Wei Qiu | 1 | 2 |
| Yong Tang | 1 | 45 |
| Tao Wu | 21 | 36 |
| Xiangyun Xie | 1 | 2 |

### Experiment Results.

Like many other systems, we use the metrics of precision and recall to evaluate the performance of our proposed algorithm. The results are illustrated in figure 6.12 and figure 6.13 respectively. As we can see, both of the precision and recall are very high, and the average of the precision and recall touches 0.946 and 0.651 respectively, which is satisfying.

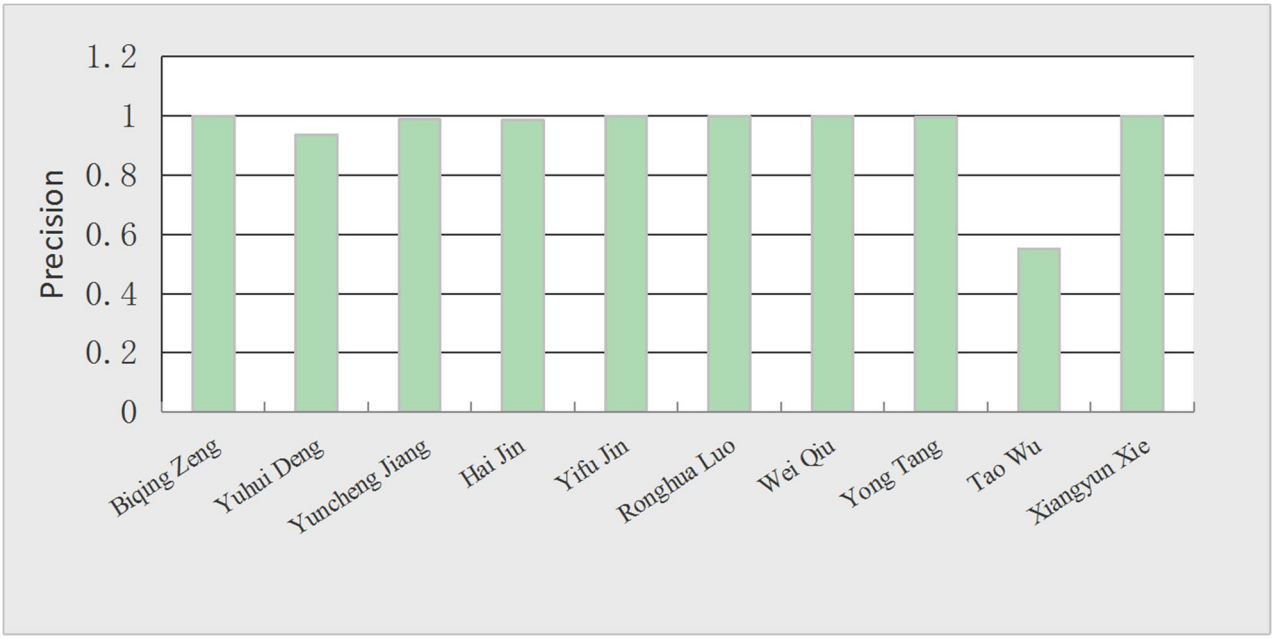



Figure 6.12 Precision of Author Disambiguation Algorithm

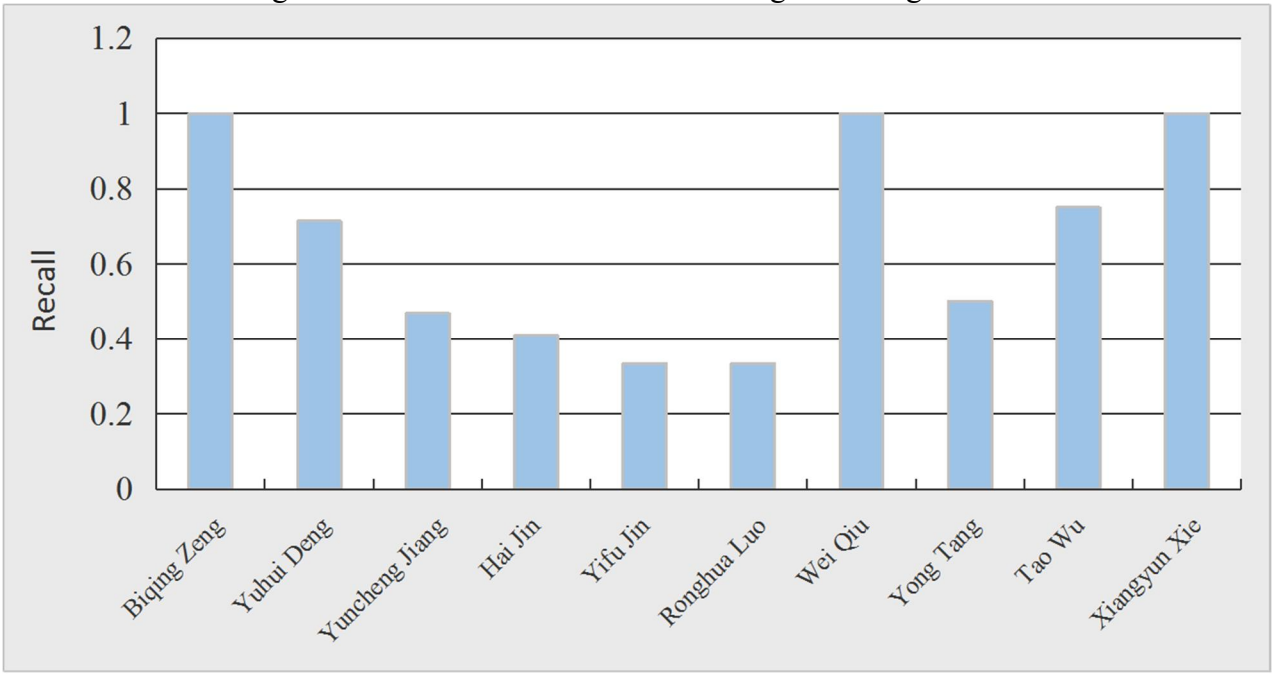



Figure 6.13 Recall of Author Disambiguation Algorithm

### 6.3.5 Demonstration

Figure 6.14 shows the disambiguation result of "Wei Wang" when users search a scholar name in SCHOLAT. Unlike other systems, SCHOLAT first displayed the authors with the same name. In this case, we have two scholars called "Wang Wei", there are two circles, by the "Wang Wei" co-author. When a user clicks a circle, a chart with additional information is displayed. The

user can click on the scholar's portrait to get more information such as his / her research interest, contact and much other information. Or click the list of edges in the chart to retrieve the collaborators and "Wang Wei". Through this design, users can more easily find the scholars and publications they are interested in.

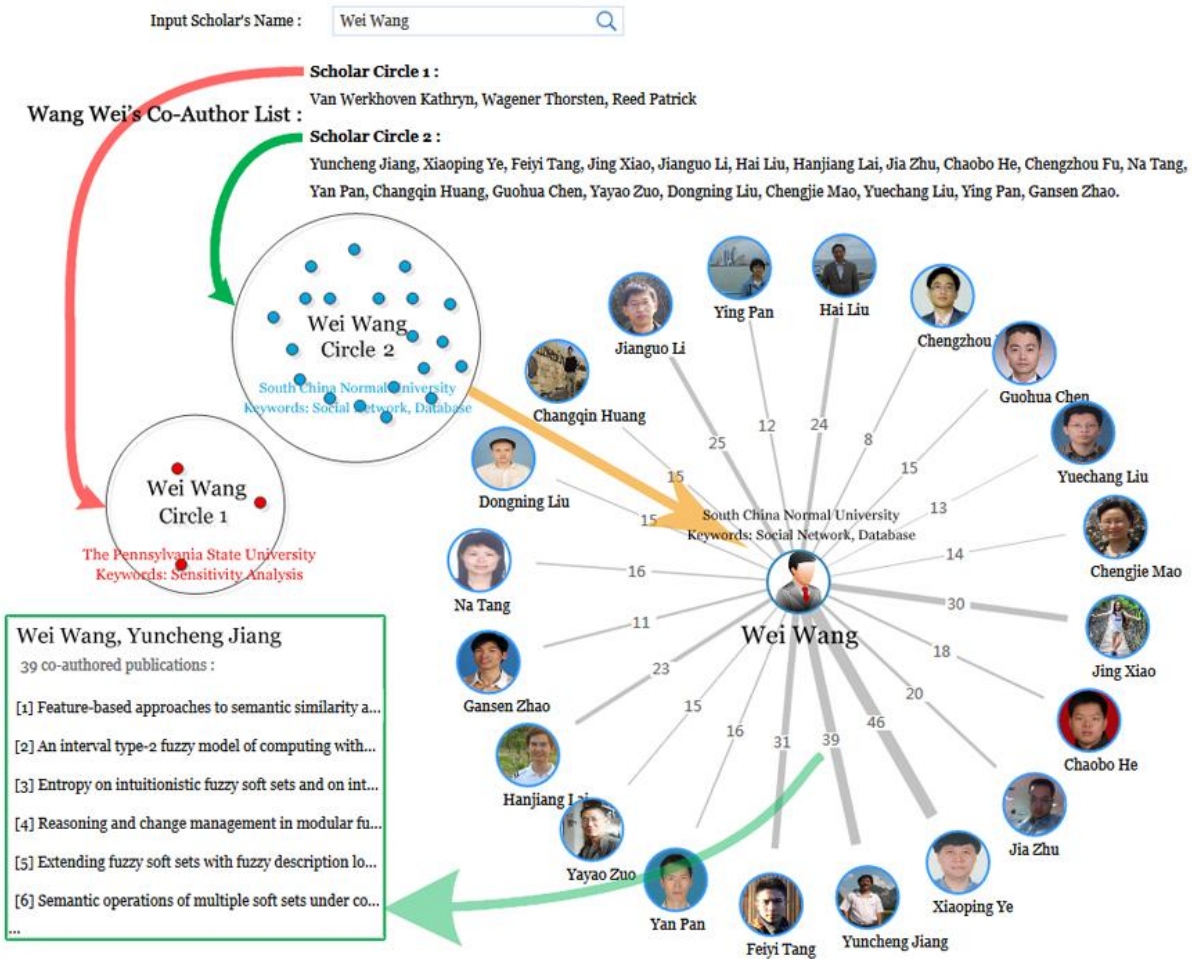



Figure 6.14 Author Disambiguation Result

## 6.4 SCHOLAT Recommendation System: XSRecom

SCHOLAT provides users with a friend recommendation service which can help our users to find potential research partners more accurately and efficiently in terms of higher acceptance rate of recommended friends and faster match up of friends. Generally speaking, we want to make sure that every people we recommend must be in his/her interests and can further help one extending social circles and academic resources.

The overall framework of XSRecom is shown in figure 6.15. It consists of three main stages: (1) The first stage constructs the user link matrix and the content feature matrix after data extraction and preprocessing. (2) The second stage exploits the thematic community by combining the Nonnegative Matrix Factorization (NMF) model and identifies the user members based on their intensity allocation for a given community. (3) The third stage calculates the paired user similarity for each community, generating a list of candidate friends. Then we will these candidate lists to get each target user to recommend a friend. Before we describe the details of each phase, in Table 6.4 we summarise the major notations used in the following sections.

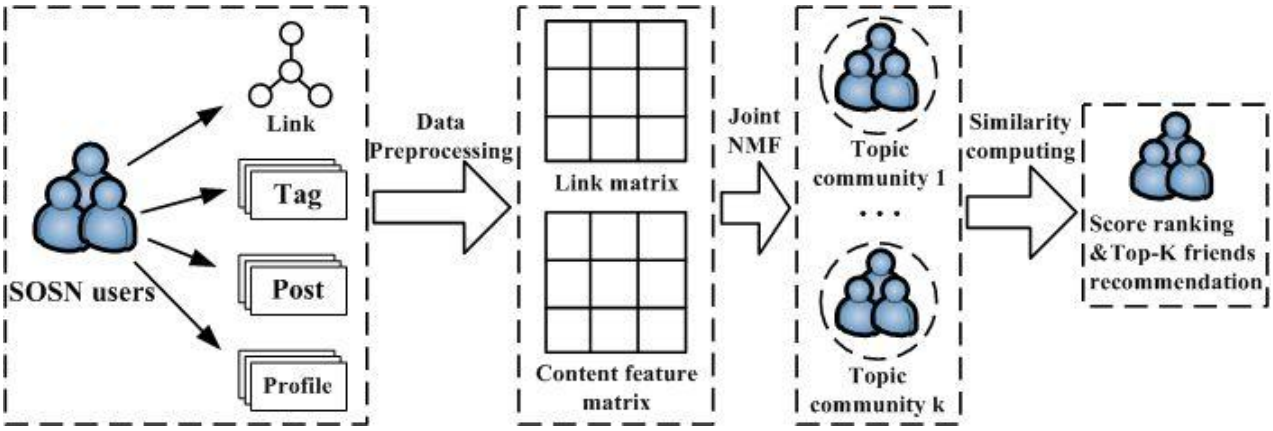


Figure 6.15 XSRecom Framework

Table 6.4 Notation used in XPRecom

| Notation | Description |
|---|---|
| $U=\{u_1,u_2,...,u_n\}$ | The users set |
| $e_{ij}$ | A link edge named e from $u_i$ to $u_j$ |
| $A=\{a_1,a_2,...,a_m\}$ | The content features set |
| $X=[x_{ij}]^{n\times n}\in R_+^{n\times n}$ | The user link matrix |
| $Y=[y_{ij}]^{m\times n}\in R_+^{m\times n}$ | The matrix of content feature-user |
| $r$ | The count of topic community |
| $\lambda$ | The coefficient of regularisation |

### 6.4.1 Topic Community Mining Using Joint NMF

SOSN includes links and content information, which can be denoted as $SOSN = \{U, E, A, X, Y\}$. Without loss of generality, we model the corresponding OSN link graph as the directed unweighted graph. That is to say, for $\forall x_{ij} \in X$, if $e_{ij} \in E$, then $x_{ij}$=1,else $x_{ij}$=0.

After filtering the stop word, we use the TF-IDF (term frequency-inverse document frequency) for each word as the eigenvalue for each user. That is, for $\forall y_{ij} \in Y$, we compute TF-IDF of feature $a_i \in u_j$ content information text as its value. Then we can get two nonnegative matrices: they are suitable for factorization using NMF, respectively.While using links and content information to tap the topic community, we fuse $X$ and $Y$ into the following joint NMF model:

$$\begin{aligned} &minJ(H,S,W) = \frac{1}{2}\{\alpha \| X - HSH^T \|_F^2 + (1-\alpha)\| Y - WH^T \|_F^2 + \lambda(\| H \|_F^2 + \| S \|_F^2 + \| W \|_F^2)\} \\ &s.t. H \geq 0, S \geq 0, W \geq 0 \end{aligned} \tag{6.1}$$

Where $H \in R_+^{nr}$, $S \in R_+^{rr}$, and $W \in R_+^{mr}$ are the topic community indicator, the topic community internal-strength indicator, the topic word affiliation indicator matrices, respectively. $\alpha \in [0,1]$, which is a hyper-parameter that controls the importance of each factorization. The minimization of $J$ is a typical constraint optimisation problem that can be solved by using the iterative optimisation solution method. We can export the following $h_{ij} \in H$, $S_{pq} \in S$, and $w_{ab} \in W$ multiplication rules:

$$h_{ij} = \frac{h_{ij}[\alpha(XHS^T + X^T HS) + (1-\alpha)Y^T W]_{ij}}{[\alpha H(SH^T HS^T + S^T H^T HS + (1-\alpha)W^T W + \lambda)]_{ij}} \tag{6.2}$$

$$s_{pq} = s_{pq} \frac{[\alpha H^T XH]_{pq}}{[\alpha H^T HSH^T H + \lambda S]_{pq}} \tag{6.3}$$

$$w_{ab} = w_{ab} \frac{[(1-\alpha)YH]_{ab}}{(1-\alpha)[WH^T H + \lambda W]_{ab}} \quad (6.4)$$

Under the above iterative update rule, the objective function $J$ does not increase, which ensures the convergence of the iteration. When the objective function $J$ converges, we can obtain the local optimal solution for $H$, $S$ and $W$. We can identify each user's community member based on $H$.. That is to say, $\forall h_{ij} \in H$ represents the strength of $u_i$ belonging to a community $c_j$, and we can obtain the result of academic social community discovery based on assign each user to his maximum membership strength community.

### 6.4.2 Scholar Recommendation Based on Topic Community

After discovering the theme community, the next stage will generate candidate friends from these communities as recommendations. Users in the same theme community share more similar links and content features, so they have a better representation than people outside the community, with similar interests that can be preferred as candidate candidates for the target audience. You can use the community-based similarity measure to calculate the grade score for each candidate friend. Namely, given the candidate user $u_i$ and the target user $u_j$, the rank score $score(i, j)$ can be computed as follows:

$$score(i, j) = [HH^T]_{ij} \quad (6.5)$$

Where $H$ is derived by Algorithm 2. Finally, we sort the scores and output the Top-K friends list to recommend to $u_j$.

### 6.4.3 Experiment for XPRecom performance

Some comparative experiments to evaluate the performance of XPRecom has been carried out. A machine powered by an Intel Core(TM)2 Duo CPU (2.93GHz) with 4GB RAM, running Windows 7 is used to carry out the experiment. Each experiment has run five times, and the average is reported here. XPRecom is coded in Java.

#### Comparative Methods and Experimental Datasets

We compare the performance of XSRecom with the following methods:

**1. Friends-of-Friends (FoF)** This is a classic link-based method that is often used by OSNs. If a particular user and target user have many common friends, then he / she may also be interested in becoming a target user's friend. So we recommend the most common friends of the user to the target user.

**2. Profile-based (PB) [5].** This is a classic link-based method that is often used by OSNs. If a particular user and target user have many common friends, then he / she may also be interested in becoming a target user's friend. So we recommend the most common friends of the user to the target user.

We use three real world OSN datasets for our experiments. The first one is from LinkedIn. Secondly, from Sina microblogging, which is the most popular microblogging system in China. The third is short, which is a popular movie comment online service network. All the data includes public friendship or follow the links, tags, profiles and jobs to crawl users. The statistics of these datasets after preprocessing is as shown in Table 6.5, where

$$Sparisity(X) = 1 - |E| / (|U| \times |U|)$$

and

$$Sparisity(Y) = 1 - |\{y_{ij} \mid y_{ij} \in Y \wedge y_{ij} \neq 0\}| / (|U| \times |A|)$$

From Table 6.4 we can see that $X$ and $Y$ are all extremely sparse matrices.

Table 6.5 Statistics of datasets.

| Statistics | LinkedIn | Weibo | Flixster |
|---|---|---|---|
| \|U\| | 183,647 | 234,832 | 126,936 |
| \|E\| | 21,710,923 | 33,231,784 | 14,271,458 |
| \|A\| | 70,631 | 94,537 | 60,315 |
| Sparsity(X) | 99.90% | 99.90% | 99.90% |
| Sparsity(Y) | 99.90% | 99.90% | 99.90% |

## Experimental Results

In order to evaluate the performance of the above recommended methods, we chose the conversion rate (CR), accuracy and recall rate in [165] as our evaluation index. In the recommended system, CR is a widely used measurement method used to assess whether the user

at least get a good recommendation. CR is a rough metric, but accuracy and recall are accurate metrics. In our next experiment, for each data set, we randomly selected 10 users per user as the test data set and the rest as the training data set. We calculate the CRS, precision, and recall rate for each method by calculating the average of each test user. At the same time, each experiment is repeated five times and the average of each measure is selected as the final result.

We compared the performance of the various methods recommended the Top-2, Top-4, Top-5, Top-8 and Top-10 friends, respectively. Figure 6.16 to Figure 6.18, which show the comparison results of various evaluation metrics on these three datasets. From the results, it is clear that the hybrid methods (TCB and RFG) are superior to the link-based (FOF) and content (PB) methods. This further validates that friend recommendations are more effective by combining user links with content information. In the two mixing methods, we propose a TCB method that is superior to the RFG method. This is because the need to calculate the build properties enhances the similarity of the paired network users, based on the $X$ and $Y$ matrices, but in Table 2, $X$ and $Y$ in the three data sets are very sparse, it is easy to produce the user similarity calculation error, and ultimately affect the recommended performance of the model.

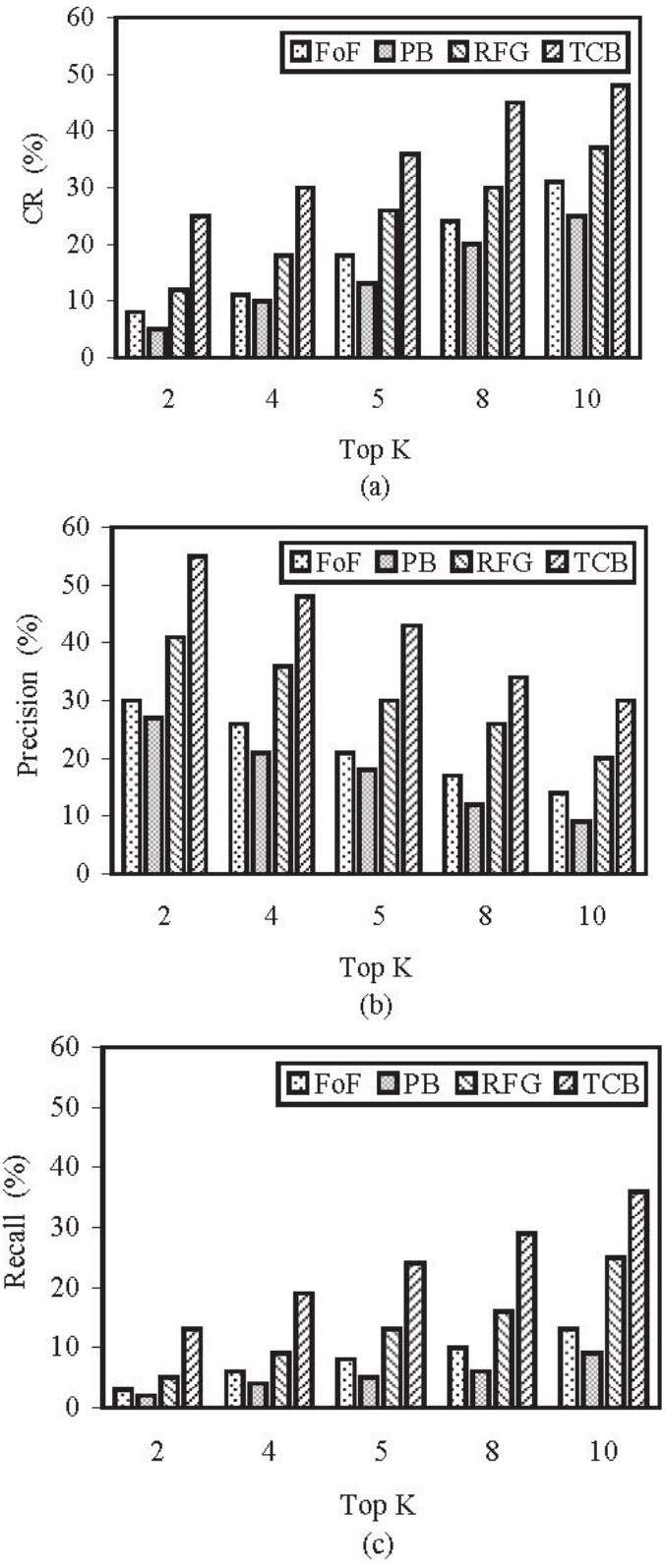
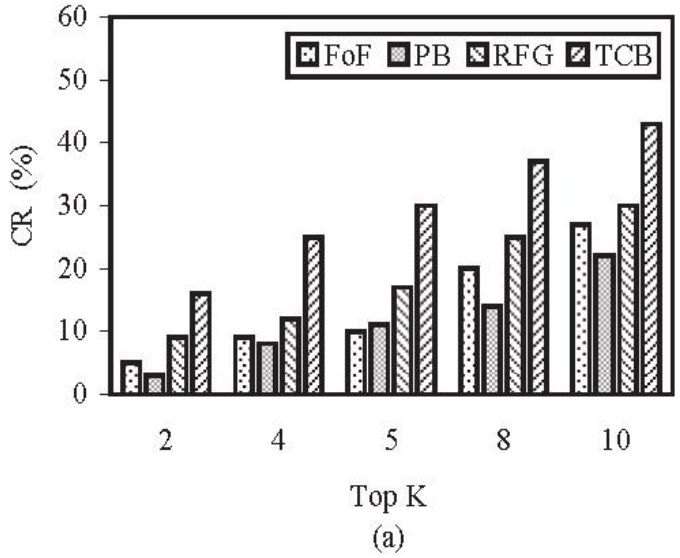
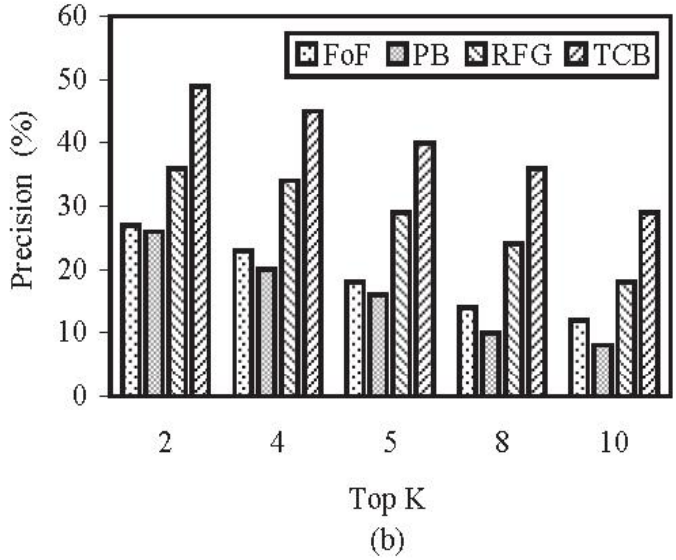


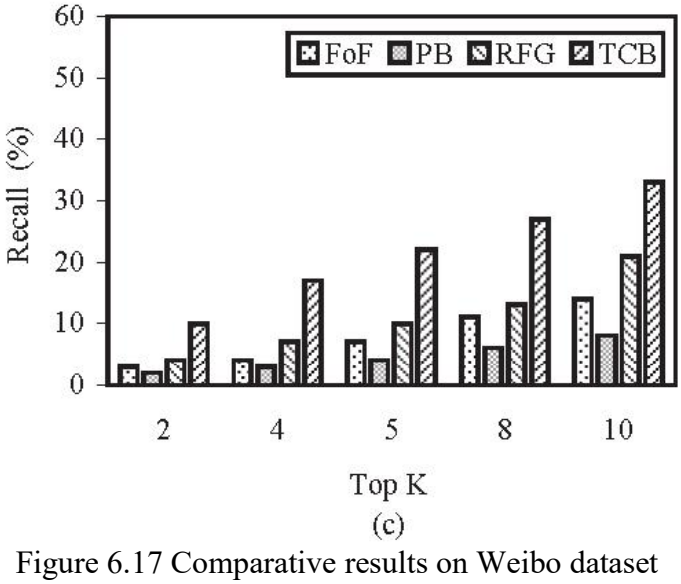

Figure 6.16 Comparative results on LinkedIn dataset.




(a)

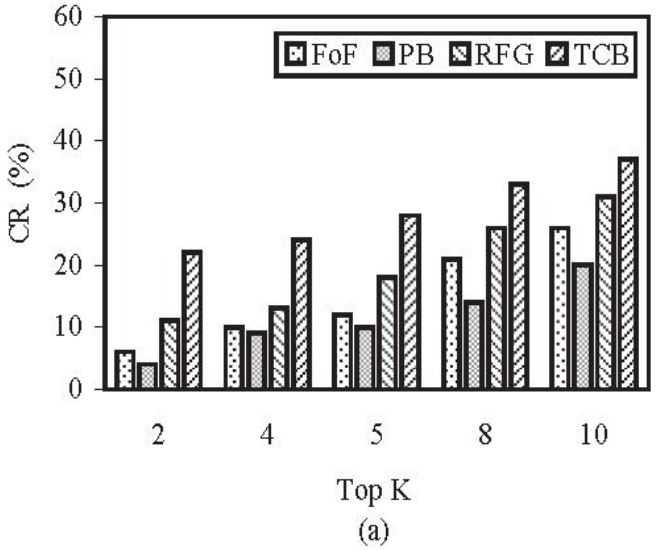



(b)

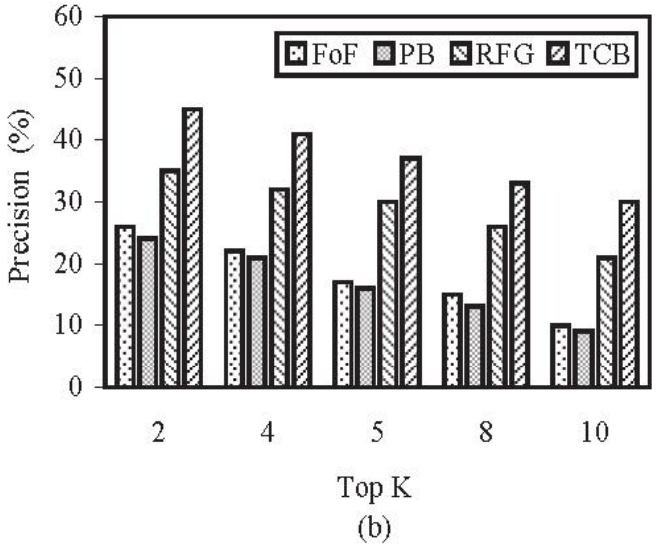



(c)

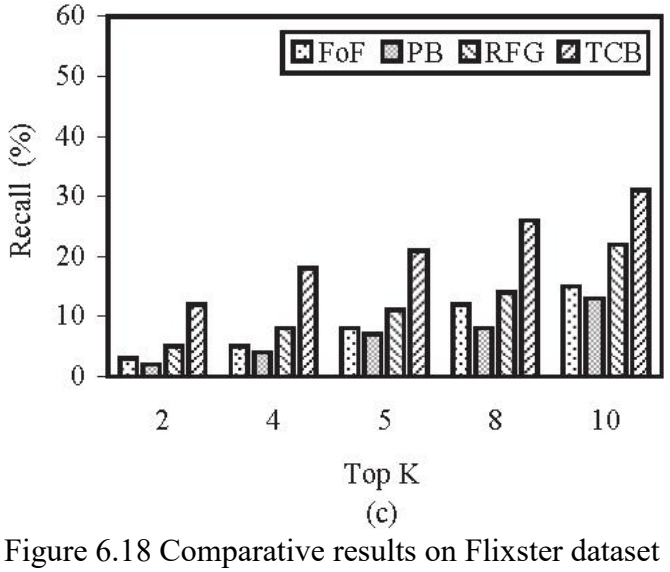
Figure 6.17 Comparative results on Weibo dataset




(a)




(b)




(c)

Figure 6.18 Comparative results on Flixster dataset

### 6.4.4 Demonstration

XSRecom can recommend potential scholars to the user with the use of the topic community-based method for friend recommendation when user logins in the SCHOLAT web site. Figure 6.19 displays the list of recommended scholars by XSRecom to the target user whoes ID is "Wei Wang". Every scholar has a similarity metric score value (we magnified the original value 100 times). The user can either click the scholar portrait to obtain more of their information or click the "See all" button to get more recommendations.

Figure 6.19 Scholars Recommendation Demo

## 6.5 Chapter Conclusion and Future Expectation

The first impression that Facebook gives us is of young, energetic and passion people who like to share their lifestyle moments to family, friends and sometimes strangers. Twietter is a

place full of people who like to follow up the most up-to-date topics and famous peoples in the world. LinkedIn provides graduates tremendous opportunities to find decent jobs and link potential industry partners together.

However one needs to know that social networking sites are far beyond general social networking sites like Facebook, Twitter and LinkedIn. Seeing the great potential profits of such market as well as people's perception of social networks are tend to be more specific. Newcomers like Instagram which dedicated itself for photo sharing; Pinterest which focus on 'little things" in life, succeeds with its innovative pin-board-style design. On the other hand, the good old sites like YouTube that revolves around video production, vlogging as well as music sharing also turned more focus on converting normal users into registered customers of Google+.

Yet, few are turning focus to the social networking services for research fields where massive requirements are still unsatisfied. Google Scholar was created by Google in 2004 and finally put onto the market in 2006; it is now one of the largest academic information search engine in the world, which mainly focus on providing searching service. Although they also provide a personal page for registration, however few social elements can be found where people are simply using it as a tool to manage citations [103]. ResearchGate4, founded in 2008 is the one that is most closely related work to SCHOLAT with similar basic functions such as personal homepage and cloud storage of research work. However, we provide more services such as search engine, and web-service interfaces. Although we share the same idea to connect and collaborate with potential colleagues and partners, ResearchGate has fewer collaboration elements such as institutional sub-platforms. Last, there is an undeniable improvement that we have comparing to ResearchGate in that we supports Chinese character and we have done tremendous works in developing methods and technologies that support the Chinese Language such as SOSN Ontology. Moreover, there is an aboundant market with fast growing rate awaits us to explore.

[4] Research Gate website http://www.researchgate.net/

# Chapter 7

## Conclusion and Future Work

*In this thesis, the ultimate objective is to build up a virtual environment for researchers to share knowledge, to make connections and to further establish possible collaborative relationships. Many steps and processes have been taken to achieve this goal from a different perspective. This chapter will summarise the work and result of this thesis as a whole and then points out the limitations of the current research. Finally, some future research directions have been put forward.*

## 7.1 Summary of Current Research Works

The objective of this thesis is to provide a more sensible way for people to understand the evolution of the academia online social network. With the models and algorithms developed we can solve issues in regards to the needs of the users in this network as finding valuable research partners. Ultimately, a virtual community can be established so as to allow future researchers to share knowledge and to carry on the work further. This thesis contains three significant steps to achieve the goal:

1) Understanding the network by analysing the user relationship. In particular, a new link prediction algorithm based on time varied weight has been created for better recommendation result in the co-authorship network.

2) Utilising user-generated content as the recourse features and social tagging system as the baseline to analyse the user-interest model. The UITGCF model has been built, and experiment result shows that it can successfully improve the accuracy of Top-N recommendation.

3) Academia social network is a domain where scarcely labelled data exists in terms of users attributes and features which is regarded as target domain. However many learning based methods requires sufficient labelled data to train precise ranking models. Thus some techniques are utilised to solve this issue such as cross-domain learning to rank. This solution provide a way that enables us to utilised the data from other domains where has sufficient labelled data which regarded as source domain. A sparse ranking model has been proposed to utilise the labelled data in the source domain to improve the ranking accuracy in the target domain.

Finally, in a live running website, some of my work has been implemented into practice. And the system has been introduced in chapter 6 where two major applications of my research work have been presented.

## 7.1 Limitation of the Current Research

There are some limitations of this thesis such as the size of the data tested for the algorithms, for example in chapter 3 the generality of the method may be tested on various kinds of networks and comparing the predictive accuracy. Also in chapter 4, if the work needs to experiment on another network with a larger size of data, it is necessary to adopt some parallel computing methods to assist the computation process.

## 7.3 Future Work

Most of the current link prediction methods are solving the problem in a static sense which means the network where nodes in are presupposed as unchanged in the future. However real networks of any kind are changing over time and social networks in particular are very actively changeable. Although many mechanisms have been applied to solve this issue, however, there is still a gap between real-time link prediction reality and static link prediction methods. On the other hand, since most of the data we applied to conduct experiments comes from commercial websites, the data quality is questionable somehow since many malicious programs are developed to emulate real users who in fact are only machines, so in the future, how to detect these 'skeletons' users and distinguish them from real users so as to improve the validity of the results would be a very interesting direction.

From another perspective, most of the researchers on link prediction currently focus on predicting the links that may be created in the future. Few people are doing research on an inverse idea of predicting the links that will disappear in the future. It is a very interesting idea, however, predicting disappearing links is not simply an inverse process of predicting the new ones since the mechanism is different from behind. So simply applying current methods to it, such as ranking the low similarities of a pair of nodes to the top list, would not work for this new problem. It is quite interesting if there will be more sensible and applicable methods to be developed to solve this new issue.

# Reference


[1] 2017. *SCHOLAT* [Online]. Available: www.scholat.com.

[2] ADAMIC, L. A. & ADAR, E. 2003. Friends and neighbors on the Web. *Social Networks,* 25**,** 211-230.

[3] AGARWAL, V. & BHARADWAJ, K. K. 2012. A collaborative filtering framework for friends recommendation in social networks based on interaction intensity and adaptive user similarity. *Social Network Analysis and Mining,* 3**,** 359-379.

[4] AHMED, N. M. & CHEN, L. 2016. An efficient algorithm for link prediction in temporal uncertain social networks. *Information Sciences,* 331**,** 120-136.

[5] AKCORA, C. G., CARMINATI, B. & FERRARI, E. 2013. User similarities on social networks. *Social Network Analysis and Mining,* 3**,** 475-495.

[6] AL HASAN, M., CHAOJI, V., SALEM, S. & ZAKI, M. Link prediction using supervised learning. SDM06: workshop on link analysis, counter-terrorism and security, 2006.

[7] AL HASAN, M. & ZAKI, M. J. 2011. A survey of link prediction in social networks. *Social network data analytics.* Springer.

[8] ALE EBRAHIM, N. 2016. Academic social networking (ResearchGate & Academia) and the research impact. *Retrieved from Research Support Unit, Centre for Research Services, Institute of Research Management and Monitoring (IPPP)", University of Malaya: https://dx. doi. org/10.6084/m9. figshare,* 3464156**,** v1.

[9] AMARAL, L. A. N., SCALA, A., BARTHELEMY, M. & STANLEY, H. E. 2000. Classes of small-world networks. *Proceedings of the National Academy of Sciences,* 97**,** 11149-11152.

[10] ARMENTANO, M. G., GODOY, D. & AMANDI, A. A. 2013. Followee recommendation based on text analysis of micro-blogging activity. *Information Systems,* 38**,** 1116-1127.

[11] BAI, J., DIAZ, F., CHANG, Y., ZHENG, Z. & CHEN, K. Cross-market model adaptation with pairwise preference data for web search ranking. Proceedings of the 23rd International Conference on Computational Linguistics: Posters, 2010. Association for Computational Linguistics, 18-26.

[12] BALBY MARINHO, L., HOTHO, A., JÄSCHKE, R., NANOPOULOS, A., RENDLE, S., SCHMIDT-THIEME, L., STUMME, G. & SYMEONIDIS, P. 2012. Recommender Systems for Social Tagging Systems. Springer US.

[13] BARABÁSI, A. L., JEONG, H., NÉDA, Z., RAVASZ, E., SCHUBERT, A. & VICSEK, T. 2002. Evolution of the social network of scientific collaborations. *Physica A: Statistical Mechanics and its Applications,* 311**,** 590-614.

[14] BARTAL, A., SASSON, E. & RAVID, G. 2009. Predicting Links in Social Networks Using Text Mining and SNA. *2009 International Conference on Advances in Social Network Analysis and Mining.* IEEE.

[15] BASILICO, J. & HOFMANN, T. 2004. Unifying collaborative and content-based filtering. *Twenty-first international conference on Machine learning - ICML '04.* ACM Press.

[16] BASSETT, D. S. & BULLMORE, E. T. 2016. Small-world brain networks revisited. *The Neuroscientist***,** 1073858416667720.

[17] BILGIC, M., NAMATA, G. M. & GETOOR, L. 2007. Combining Collective Classification and Link Prediction. *Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007).* IEEE.

[18] BLEI, D. M., NG, A. Y. & JORDAN, M. I. 2003. Latent dirichlet allocation. *Journal of machine Learning research,* 3**,** 993-1022.

[19] BLONDEL, V. D., GUILLAUME, J.-L., LAMBIOTTE, R. & LEFEBVRE, E. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment,* 2008**,** P10008.

[20] BORWEIN, J. & LEWIS, A. 2006. Convex Analysis and Nonlinear Optimization. *CMS Books in Mathematics.* Springer New York.

[21] BRICKLEY, D. & MILLER, L. 2015. *FOAF (Friend of a Friend)* [Online]. Available: http://www.foaf-project.org/ [Accessed 10 Oct 2016].

[22] BRODER, A., KUMAR, R., MAGHOUL, F., RAGHAVAN, P., RAJAGOPALAN, S., STATA, R., TOMKINS, A. & WIENER, J. 2000. Graph structure in the Web. *Computer Networks,* 33**,** 309-320.

[23] BUI, D. T., TUAN, T. A., KLEMPE, H., PRADHAN, B. & REVHAUG, I. 2016. Spatial prediction models for shallow landslide hazards: a comparative assessment of the efficacy of support vector machines, artificial neural networks, kernel logistic regression, and logistic model tree. *Landslides,* 13**,** 361-378.

[24] BURGES, C., SHAKED, T., RENSHAW, E., LAZIER, A., DEEDS, M., HAMILTON, N. & HULLENDER, G. 2005. Learning to rank using gradient descent. *Proceedings of the 22nd international conference on Machine learning - ICML '05.* ACM Press.

[25] BURGES, C. J., RAGNO, R. & LE, Q. V. Learning to rank with nonsmooth cost functions. NIPS, 2006. 193-200.

[26] BÜTTCHER, S., CLARKE, C. L. & CORMACK, G. V. 2016. *Information retrieval: Implementing and evaluating search engines*, Mit Press.

[27] CAI, P., GAO, W., ZHOU, A. & WONG, K.-F. Query weighting for ranking model adaptation. Proceedings of the 49th Annual Meeting of the Association for

Computational Linguistics: Human Language Technologies-Volume 1, 2011. Association for Computational Linguistics, 112-122.

[28] CAO, Z., QIN, T., LIU, T.-Y., TSAI, M.-F. & LI, H. 2007. Learning to rank. *Proceedings of the 24th international conference on Machine learning - ICML '07.* ACM Press.

[29] CHAPELLE, O. & KEERTHI, S. S. 2009. Efficient algorithms for ranking with SVMs. *Information Retrieval,* 13**,** 201-215.

[30] CHAWLA, N. V., BOWYER, K. W., HALL, L. O. & KEGELMEYER, W. P. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research,* 16**,** 321-357.

[31] CHEN, D., XIONG, Y., YAN, J., XUE, G.-R., WANG, G. & CHEN, Z. 2009. Knowledge transfer for cross domain learning to rank. *Information Retrieval,* 13**,** 236-253.

[32] CHEN, D., YAN, J., WANG, G., XIONG, Y., FAN, W. & CHEN, Z. 2008. TransRank: A Novel Algorithm for Transfer of Rank Learning. *2008 IEEE International Conference on Data Mining Workshops.* IEEE.

[33] CHEN, H.-H., MILLER, D. J. & GILES, C. L. 2013. The predictive value of young and old links in a social network. *Proceedings of the ACM SIGMOD Workshop on Databases and Social Networks - DBSocial '13.* ACM Press.

[34] CHEN, X., XIA, M., CHENG, J., TANG, X. & ZHANG, J. Trend prediction of internet public opinion based on collaborative filtering. Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), 2016 12th International Conference on, 2016. IEEE, 583-588.

[35] CHOUDHURY, N. & UDDIN, S. 2016. Time-aware link prediction to explore network effects on temporal knowledge evolution. *Scientometrics,* 108**,** 745-776.

[36] CHUNG, F. & ZHAO, W. 2010. PageRank and Random Walks on Graphs. *Fete of Combinatorics and Computer Science.* Springer Berlin Heidelberg.

[37] CLAUSET, A. & EAGLE, N. 2012. Persistence and periodicity in a dynamic proximity network. *arXiv preprint arXiv:1211.7343*.

[38] CORBELLINI, A., MATEOS, C., GODOY, D., ZUNINO, A. & SCHIAFFINO, S. 2016. An Evaluation of Distributed Processing Models for Random Walk-Based Link Prediction Algorithms Over Social Big Data. *New Advances in Information Systems and Technologies.* Springer.

[39] CRAMMER, K. & SINGER, Y. 2005. Online Ranking by Projecting. *Neural Computation,* 17**,** 145-175.

[40] DABROWSKI, M., SYNAK, M. & KRUK, S. R. Bibliographic Ontology. *Semantic Digital Libraries.* Springer Berlin Heidelberg.

[41] DAVIS, D., LICHTENWALTER, R. & CHAWLA, N. V. 2012. Supervised methods for multi-relational link prediction. *Social Network Analysis and Mining,* 3**,** 127-141.

[42] DEAN, J. & GHEMAWAT, S. 2008. MapReduce. *Communications of the ACM,* 51**,** 107.

[43] DEAN, J. & GHEMAWAT, S. 2010. MapReduce. *Communications of the ACM,* 53**,** 72.

[44] DELICIOUS. 2016. *hetrec2011-delicious-2k* [Online]. Available: https://delicious.com/ [Accessed 4 June 2016].

[45] DOERFEL, M. L. & MOORE, P. J. 2016. Digitizing Strength of Weak Ties: Understanding Social Network Relationships through Online Discourse Analysis. *Annals of the International Communication Association,* 40**,** 127-148.

[46] DUNBAR, R. I. 1992. Neocortex size as a constraint on group size in primates. *Journal of human evolution,* 22**,** 469-493.

[47] DUNLAVY, D. M., KOLDA, T. G. & ACAR, E. 2011. Temporal Link Prediction Using Matrix and Tensor Factorizations. *ACM Transactions on Knowledge Discovery from Data,* 5**,** 1-27.

[48] FAN, X., WANG, J., PU, X., ZHOU, L. & LV, B. 2011. On Graph-Based Name Disambiguation. *Journal of Data and Information Quality,* 2**,** 1-23.

[49] FOUSS, F., PIROTTE, A., RENDERS, J.-M. & SAERENS, M. 2007. Random-Walk Computation of Similarities between Nodes of a Graph with Application to Collaborative Recommendation. *IEEE Transactions on Knowledge and Data Engineering,* 19**,** 355-369.

[50] FREUND, Y., IYER, R., SCHAPIRE, R. E. & SINGER, Y. 2003. An efficient boosting algorithm for combining preferences. *Journal of machine learning research,* 4**,** 933-969.

[51] FREUND, Y. & SCHAPIRE, R. E. 1995. A desicion-theoretic generalization of on-line learning and an application to boosting. *Lecture Notes in Computer Science.* Springer Berlin Heidelberg.

[52] FRIEZE, A., KANNAN, R. & VEMPALA, S. 2004. Fast monte-carlo algorithms for finding low-rank approximations. *Journal of the ACM,* 51**,** 1025-1041.

[53] FU, W., SONG, L. & XING, E. P. 2009. Dynamic mixed membership blockmodel for evolving networks. *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09.* ACM Press.

[54] GENG, B., YANG, L., XU, C. & HUA, X.-S. 2012. Ranking Model Adaptation for Domain-Specific Search. *IEEE Transactions on Knowledge and Data Engineering,* 24**,** 745-758.

[55] GETOOR, L., FRIEDMAN, N., KOLLER, D. & PFEFFER, A. 2001. Learning Probabilistic Relational Models. *Relational Data Mining.* Springer Berlin Heidelberg.

[56] GUY, I., JACOVI, M., PERER, A., RONEN, I. & UZIEL, E. 2010. Same places, same things, same people? *Proceedings of the 2010 ACM conference on Computer supported cooperative work - CSCW '10.* ACM Press.

[57] HALPIN, H., ROBU, V. & SHEPHERD, H. 2007. The complex dynamics of collaborative tagging. *Proceedings of the 16th international conference on World Wide Web - WWW '07.* ACM Press.

[58] HAN-JIANG, L., YAN, P., YONG, T. & RONG, Y. 2013. FSMRank: Feature Selection Algorithm for Learning to Rank. *IEEE Transactions on Neural Networks and Learning Systems,* 24**,** 940-952.

[59] HAN, H., GILES, L., ZHA, H., LI, C. & TSIOUTSIOULIKLIS, K. 2004. Two supervised learning approaches for name disambiguation in author citations. *Proceedings of the 2004 joint ACM/IEEE conference on Digital libraries - JCDL '04.* ACM Press.

[60] HAN, H., XU, W., ZHA, H. & GILES, C. L. 2005. A hierarchical naive Bayes mixture model for name disambiguation in author citations. *Proceedings of the 2005 ACM symposium on Applied computing - SAC '05.* ACM Press.

[61] HAN, H., ZHA, H. & GILES, C. L. 2005. Name disambiguation in author citations using a K-way spectral clustering method. *Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries - JCDL '05.* ACM Press.

[62] HECKERMAN, D., MEEK, C. & KOLLER, D. 2004. Probabilistic models for relational data. Technical Report MSR-TR-2004-30, Microsoft Research.

[63] HERBRICH, R., GRAEPEL, T. & OBERMAYER, K. 2000. Large margin rank boundaries for ordinal regression.

[64] HUANG, Z., LI, X. & CHEN, H. 2005. Link prediction approach to collaborative filtering. *Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries - JCDL '05.* ACM Press.

[65] JAGADISHWARI, V. & UMADEVI, V. 2015. Empirical Analysis of Traditional Link Prediction Methods. *International Journal of Computer Applications,* 121.

[66] JÄRVELIN, K. & KEKÄLÄINEN, J. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems,* 20**,** 422-446.

[67] JEH, G. & WIDOM, J. 2002. SimRank. *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '02.* ACM Press.

[68] JELASSI, M. N., BEN YAHIA, S. & MEPHU NGUIFO, E. 2013. A personalized recommender system based on users' information in folksonomies. *Proceedings of the 22nd International Conference on World Wide Web - WWW '13 Companion.* ACM Press.

[69] JEONG, H. J., TAEYEON, K. & KIM, M. H. Link Prediction by Utilizing Correlations Between Link Types and Path Types in Heterogeneous Information Networks. International Conference on Data Mining and Big Data, 2016. Springer, 156-164.

[70] JIA, Y. & QU, L. Improve the Performance of Link Prediction Methods in Citation Network by Using H-Index. Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2016 International Conference on, 2016. IEEE, 220-223.

[71] JULIAN, K. & LU, W. 2016. Application of Machine Learning to Link Prediction.

[72] KABBUR, S., NING, X. & KARYPIS, G. 2013. FISM factored item similarity models for top-n recommender systems. *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '13.* ACM Press.

[73] KANG, I.-S., NA, S.-H., LEE, S., JUNG, H., KIM, P., SUNG, W.-K. & LEE, J.-H. 2009. On co-authorship for author disambiguation. *Information Processing & Management,* 45**,** 84-97.

[74] KASHIMA, H. & ABE, N. 2006. A Parameterized Probabilistic Model of Network Evolution for Supervised Link Prediction. *Sixth International Conference on Data Mining (ICDM'06).* IEEE.

[75] KASHIMA, H., OYAMA, S., YAMANISHI, Y. & TSUDA, K. 2009. On Pairwise Kernels: An Efficient Alternative and Generalization Analysis. *Advances in Knowledge Discovery and Data Mining.* Springer Berlin Heidelberg.

[76] KELLEY, S. 2016. FRBR, Before and after: A look at our bibliographic models. Taylor & Francis.

[77] KHADANGI, E., ZAREAN, A., BAGHERI, A. & BAGHERI JAFARABADI, A. 2013. Measuring relationship strength in online social networks based on users' activities and profile information. *ICCKE 2013.* IEEE.

[78] KOREN, Y. 2008. Factorization meets the neighborhood. *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD 08.* ACM Press.

[79] KUNEGIS, J. & LOMMATZSCH, A. 2009. Learning spectral graph transformations for link prediction. *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09.* ACM Press.

[80] LAI, H., PAN, Y., LIU, C., LIN, L. & WU, J. 2013. Sparse Learning-to-Rank via an Efficient Primal-Dual Algorithm. *IEEE Transactions on Computers,* 62**,** 1221-1233.

[81] LAI, H., PAN, Y., TANG, Y. & LIU, N. 2013. Efficient gradient descent algorithm for sparse models with application in learning-to-rank. *Knowledge-Based Systems,* 49**,** 190-198.

[82] LAST.FM. 2016. *hetrec2011-last.fm-2k* [Online]. Available: http://www.last.fm/ [Accessed 5 June 2016].

[83] LESKOVEC, J., KLEINBERG, J. & FALOUTSOS, C. 2005. Graphs over time. *Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining - KDD '05.* ACM Press.

[84] LI, B., CHAUDHURI, S. & TEWARI, A. Handling class imbalance in link prediction using learning to rank techniques. Thirtieth AAAI Conference on Artificial Intelligence, 2016.

[85] LI, J., ZHAO, G., RONG, C. & TANG, Y. 2011. Semantic description of scholar-oriented social network cloud. *The Journal of Supercomputing,* 65**,** 410-425.

[86] LI, P., BURGES, C. J., WU, Q., PLATT, J., KOLLER, D., SINGER, Y. & ROWEIS, S. McRank: Learning to Rank Using Multiple Classification and Gradient Boosting. NIPS, 2007. 845-852.

[87] LI, X. & CHEN, H. 2013. Recommendation as link prediction in bipartite graphs: A graph kernel-based machine learning approach. *Decision Support Systems,* 54**,** 880-890.

[88] LI, X., GUO, L. & ZHAO, Y. E. 2008. Tag-based social interest discovery. *Proceeding of the 17th international conference on World Wide Web - WWW '08.* ACM Press.

[89] LIBEN-NOWELL, D. & KLEINBERG, J. 2007. The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology,* 58**,** 1019-1031.

[90] LIPCZAK, M. 2008. Tag recommendation for folksonomies oriented towards individual users. *ECML PKDD discovery challenge,* 84**,** 2008.

[91] LIU, C., YEUNG, C. H. & ZHANG, Z.-K. 2011. Self-organization in social tagging systems. *Physical Review E,* 83.

[92] LIU, H., HU, Z., HADDADI, H. & TIAN, H. 2013. Hidden link prediction based on node centrality and weak ties. *EPL (Europhysics Letters),* 101**,** 18004.

[93] LIU, Y. & KOU, Z. 2007. Predicting who rated what in large-scale datasets. *ACM SIGKDD Explorations Newsletter,* 9**,** 62.

[94] LÜ, L., JIN, C.-H. & ZHOU, T. 2009. Similarity index based on local paths for link prediction of complex networks. *Physical Review E,* 80.

[95] LU, Z., SAVAS, B., TANG, W. & DHILLON, I. S. 2010. Supervised Link Prediction Using Multiple Sources. *2010 IEEE International Conference on Data Mining.* IEEE.

[96] MA, H., ZHOU, T. C., LYU, M. R. & KING, I. 2011. Improving Recommender Systems by Incorporating Social Contextual Information. *ACM Transactions on Information Systems,* 29**,** 1-23.

[97] MALIN, B., AIROLDI, E. & CARLEY, K. M. 2005. A Network Analysis Model for Disambiguation of Names in Lists. *Computational and Mathematical Organization Theory,* 11**,** 119-139.

[98] MANCA, M., BORATTO, L. & CARTA, S. 2015. Using Behavioral Data Mining to Produce Friend Recommendations in a Social Bookmarking System. *Communications in Computer and Information Science.* Springer International Publishing.

[99] MARTÍNEZ, V., BERZAL, F. & CUBERO, J.-C. 2016. A Survey of Link Prediction in Complex Networks. *ACM Computing Surveys (CSUR),* 49**,** 69.

[100] MENON, A. K. & ELKAN, C. 2011. Link Prediction via Matrix Factorization. *Machine Learning and Knowledge Discovery in Databases.* Springer Berlin Heidelberg.

[101] MILGRAM, S. 1967. The small-world problem. *In:* TODAY, P. (ed.) *PsycEXTRA Dataset.* Psychology Today: American Psychological Association (APA).

[102] MORAES, D., WAINER, J. & ROCHA, A. 2016. Low false positive learning with support vector machines. *Journal of Visual Communication and Image Representation,* 38**,** 340-350.

[103] NALLAPATI, R. M., AHMED, A., XING, E. P. & COHEN, W. W. 2008. Joint latent topic models for text and citations. *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD 08.* ACM Press.

[104] NEWMAN, M. E. J. 2001. Clustering and preferential attachment in growing networks. *Physical Review E,* 64.

[105] NEWMAN, M. E. J. 2001. From the Cover: The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences,* 98**,** 404-409.

[106] NEWMAN, M. E. J. 2004. Who Is the Best Connected Scientist?A Study of Scientific Coauthorship Networks. *Complex Networks.* Springer Berlin Heidelberg.

[107] NICKEL, M., MURPHY, K., TRESP, V. & GABRILOVICH, E. 2016. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE,* 104**,** 11-33.

[108] NICULESCU-MIZIL, A. & CARUANA, R. 2005. Predicting good probabilities with supervised learning. *Proceedings of the 22nd international conference on Machine learning - ICML '05.* ACM Press.

[109] NZEKO'O, A. J. N., TCHUENTE, M. & LATAPY, M. Time Weight Content-Based Extensions of Temporal Graphs for Personalized Recommendation. WEBIST 2017-13th International Conference on Web Information Systems and Technologies, 2017.

[110] OYAMA, S. & MANNING, C. D. 2004. Using Feature Conjunctions Across Examples for Learning Pairwise Classifiers. *Machine Learning: ECML 2004.* Springer Berlin Heidelberg.

[111] PAN, L., ZHOU, T., LÜ, L. & HU, C.-K. 2016. Predicting missing links and identifying spurious links via likelihood analysis. *Scientific reports,* 6.

[112] PAN, S. J. & YANG, Q. 2010. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering,* 22**,** 1345-1359.

[113] PAN, Y., LUO, H.-X., TANG, Y. & HUANG, C.-Q. 2011. Learning to rank with document ranks and scores. *Knowledge-Based Systems,* 24**,** 478-483.

[114] PAVLOV, D., MANNILA, H. & SMYTH, P. 2003. Beyond independence: probabilistic models for query approximation on binary transaction data. *IEEE Transactions on Knowledge and Data Engineering,* 15**,** 1409-1421.

[115] PAVLOV, M. & ICHISE, R. Finding experts by link prediction in co-authorship networks. Proceedings of the 2nd International Conference on Finding Experts on the Web with Semantics-Volume 290, 2007. CEUR-WS. org, 42-55.

[116] POPESCUL, A. & UNGAR, L. H. 2004. Cluster-based concept invention for statistical relational learning. *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '04.* ACM Press.

[117] POTTS, B. B. 1994. Book Reviews: Networks : John Scott: Network Analysis: A Handbook. London, England, and New bury Park, CA: Sage Publications, 1992. Stanley Wasserman and Katherine Faust: Social Network Analysis: Methods and Applications.

Cambridge, England and New York: Cambridge University Press, 1994. *Acta Sociologica,* 37**,** 419-423.

[118] QIN, T., LIU, T.-Y., XU, J. & LI, H. 2010. LETOR: A benchmark collection for research on learning to rank for information retrieval. *Information Retrieval,* 13**,** 346-374.

[119] SACHAN, M. & ICHISE, R. 2010. Using Abstract Information and Community Alignment Information for Link Prediction. *2010 Second International Conference on Machine Learning and Computing.* IEEE.

[120] SALTON, G., WONG, A. & YANG, C. S. 1975. A vector space model for automatic indexing. *Communications of the ACM,* 18**,** 613-620.

[121] SARUKKAI, R. R. 2000. Link prediction and path analysis using Markov chains. *Computer Networks,* 33**,** 377-386.

[122] SETT, N., SINGH, S. R. & NANDI, S. 2016. Influence of edge weight on node proximity based link prediction methods: an empirical analysis. *Neurocomputing,* 172**,** 71-83.

[123] SHALEV-SHWARTZ, S. & SINGER, Y. 2010. On the equivalence of weak learnability and linear separability: new relaxations and efficient boosting algorithms. *Machine Learning,* 80**,** 141-163.

[124] SHANG, M.-S., ZHANG, Z.-K., ZHOU, T. & ZHANG, Y.-C. 2010. Collaborative filtering with diffusion-based similarity on tripartite graphs. *Physica A: Statistical Mechanics and its Applications,* 389**,** 1259-1264.

[125] SHARMA, D. & SHARMA, U. 2014. Link prediction algorithm for co-authorship networks using Neural Network. *Proceedings of 3rd International Conference on Reliability, Infocom Technologies and Optimization.* IEEE.

[126] SHEN, Y. & JIN, R. 2012. Learning personal + social latent factor model for social recommendation. *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '12.* ACM Press.

[127] SHEPPARD, C. L., LAPLANT, B. & NEVILE, L. 2004. Dublin Core and the alternative interface access protocol. National Institute of Standards and Technology.

[128] SHU, J., SHEN, X., LIU, H., YI, B. & ZHANG, Z. 2017. A content-based recommendation algorithm for learning resources. *Multimedia Systems***,** 1-11.

[129] SIOUTAS, S., MYLONAS, P., PANARETOS, A., GEROLYMATOS, P., VOGIATZIS, D., KARAVARAS, E., SPITIERIS, T. & KANAVOS, A. Survey of machine learning algorithms on Spark over DHT-based Structures. International Workshop of Algorithmic Aspects of Cloud Computing, 2016. Springer, 146-156.

[130] SRILATHA, P. & MANJULA, R. 2016. Similarity Index based Link Prediction Algorithms in Social Networks: A Survey. *Journal of Telecommunications and Information Technology***,** 87.

[131] SRILATHA, P. & MANJULA, R. User behavior based link prediction in online social networks. Inventive Computation Technologies (ICICT), International Conference on, 2016. IEEE, 1-3.

[132] STAMOU, S. & NTOULAS, A. 2008. Search personalization through query and page topical analysis. *User Modeling and User-Adapted Interaction,* 19**,** 5-33.

[133] SYSTEMS, H. A. F. I. R. 2011. *hetRec-2011* [Online]. Available: http://grouplens.org/datasets/hetrec2011/ [Accessed 7 June 2016].

[134] SZWABE, A., CIESIELCZYK, M. & JANASIEWICZ, T. 2011. Semantically Enhanced Collaborative Filtering Based on RSVD. *Computational Collective Intelligence. Technologies and Applications.* Springer Berlin Heidelberg.

[135] TANG, A., HE, J., TANG, Y., PENG, Z. & TENG, L. 2014. Sparse ranking model adaptation for cross domain learning to rank. *網際網路技術學刊,* 15**,** 949-962.

[136] TANG, F., MAO, C., YU, J. & CHEN, J. The implementation of information service based on social network systems. Information Science and Service Science (NISS), 2011 5th International Conference on New Trends in, 2011. 46 - 49.

[137] TANG, F., ZHU, J., CAO, Y., MA, S., CHEN, Y., HE, J., HUANG, C., ZHAO, G. & TANG, Y. PARecommender: a pattern-based system for route recommendation. Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, 2016. AAAI Press, 4272-4273.

[138] TANG, F., ZHU, J., HE, C., FU, C., HE, J. & TANG, Y. SCHOLAT: An Innovative Academic Information Service Platform. Australasian Database Conference, 2016. Springer, 453-456.

[139] TASKAR, B., ABBEEL, P., WONG, M.-F. & KOLLER, D. 2007. Relational markov networks. *Introduction to statistical relational learning***,** 175-200.

[140] TORVIK, V. I. & SMALHEISER, N. R. 2009. Author name disambiguation in MEDLINE. *ACM Transactions on Knowledge Discovery from Data,* 3**,** 1-29.

[141] TRAVERS, J. & MILGRAM, S. 1969. An Experimental Study of the Small World Problem. *Sociometry,* 32**,** 425.

[142] TRAVERSO-RIBÓN, I., PALMA, G., FLORES, A. & VIDAL, M.-E. Considering Semantics on the Discovery of Relations in Knowledge Graphs. Knowledge Engineering and Knowledge Management: 20th International Conference, EKAW 2016, Bologna, Italy, November 19-23, 2016, Proceedings 20, 2016. Springer, 666-680.

[143] TYLENDA, T., ANGELOVA, R. & BEDATHUR, S. 2009. Towards time-aware link prediction in evolving social networks. *Proceedings of the 3rd Workshop on Social Network Mining and Analysis - SNA-KDD '09.* ACM Press.

[144] WANG, C., SATULURI, V. & PARTHASARATHY, S. 2007. Local Probabilistic Models for Link Prediction. *Seventh IEEE International Conference on Data Mining (ICDM 2007).* IEEE.

[145] WANG, F., LI, J., TANG, J., ZHANG, J. & WANG, K. 2008. Name Disambiguation Using Atomic Clusters. *2008 The Ninth International Conference on Web-Age Information Management.* IEEE.

[146] WANG, Z., YU, N. & WANG, J. Collaborative Filtering Recommendation Algorithm Based on Matrix Factorization and User Nearest Neighbors. Asian Simulation Conference, 2016. Springer, 199-207.

[147] WASSERMAN, S. & FAUST, K. 1994. *Social network analysis: Methods and applications*, Cambridge university press.

[148] WEI, C., HUANG, C. & TAN, H. 2009. A Personalized Model for Ontology-driven User Profiles Mining. *2009 International Symposium on Intelligent Ubiquitous Computing and Education.* IEEE.

[149] WOHLFARTH, T. & ICHISE, R. 2008. Semantic and Event-Based Approach for Link Prediction. *Practical Aspects of Knowledge Management.* Springer Berlin Heidelberg.

[150] XIAO, H., HUANG, M. & ZHU, X. 2016. SSP: Semantic Space Projection for Knowledge Graph Embedding with Text Descriptions. *arXiv preprint arXiv:1604.04835*.

[151] XIE, F., LIU, J., TANG, M., ZHOU, D., CAO, B. & SHI, M. Multi-relation Based Manifold Ranking Algorithm for API Recommendation. Advances in Services

Computing: 10th Asia-Pacific Services Computing Conference, APSCC 2016, Zhangjiajie, China, November 16-18, 2016, Proceedings 10, 2016. Springer, 15-32.

[152] XLIN, X., SHANG, T. & LIU, J. 2014. An Estimation Method for Relationship Strength in Weighted Social Network Graphs. *Journal of Computer and Communications,* 02**,** 82-89.

[153] XU, J. & LI, H. 2007. AdaRank. *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '07.* ACM Press.

[154] YANG, C.-Y., YANG, J.-S. & WANG, J.-J. 2009. Margin calibration in SVM class-imbalanced learning. *Neurocomputing,* 73**,** 397-411.

[155] YU, L. 2014. FOAF: Friend of a Friend. *A Developer's Guide to the Semantic Web.* Springer Berlin Heidelberg.

[156] YU, Y., CHEN, H. & YU, H. 2016. A social networks user relationship strength model based on Hawkes process. *Acta Electronica Sinica,* 44**,** 1362-1368.

[157] ZENG, R., DING, Y.-X. & XIA, X.-L. Link prediction based on dynamic weighted social attribute network. Machine Learning and Cybernetics (ICMLC), 2016 International Conference on, 2016. IEEE, 183-188.

[158] ZENG, S. 2016. Link prediction based on local information considering preferential attachment. *Physica A: Statistical Mechanics and its Applications,* 443**,** 537-542.

[159] ZHANG, C.-X., ZHANG, Z.-K., YU, L., LIU, C., LIU, H. & YAN, X.-Y. 2014. Information filtering via collaborative user clustering modeling. *Physica A: Statistical Mechanics and its Applications,* 396**,** 195-203.

[160] ZHANG, C., WANG, K., YU, H., SUN, J. & LIM, E.-P. 2014. Latent Factor Transition for Dynamic Collaborative Filtering. *Proceedings of the 2014 SIAM International Conference on Data Mining.* Society for Industrial and Applied Mathematics.

[161] ZHANG, C., ZHANG, H., YUAN, D. & ZHANG, M. Deep Learning Based Link Prediction with Social Pattern and External Attribute Knowledge in Bibliographic Networks. Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2016 IEEE International Conference on, 2016. IEEE, 815-821.

[162] ZHANG, D., TANG, J., LI, J. & WANG, K. 2007. A constraint-based probabilistic framework for name disambiguation. *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management - CIKM '07.* ACM Press.

[163] ZHANG, L., CHEN, G., TANG, Y. & CAI, Z. 2013. Disambiguating Authors in Academic Search Engines. *Lecture Notes in Computer Science.* Springer Berlin Heidelberg.

[164] ZHANG, Z., ZHENG, X. & ZENG, D. D. 2016. A framework for diversifying recommendation lists by user interest expansion. *Knowledge-Based Systems,* 105**,** 83-95.

[165] ZHAO, G., LEE, M. L., HSU, W., CHEN, W. & HU, H. 2013. Community-based user recommendation in uni-directional social networks. *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management - CIKM '13.* ACM Press.

[166] ZHAO, X., YUAN, J., LI, G., CHEN, X. & LI, Z. 2012. Relationship strength estimation for online social networks with the study on Facebook. *Neurocomputing,* 95**,** 89-97.

[167] ZHAO, Y., LI, L. & WU, X. Link Prediction-Based Multi-label Classification on Networked Data. Data Science in Cyberspace (DSC), IEEE International Conference on, 2016. IEEE, 61-68.

[168] ZHOU, T., LÜ, L. & ZHANG, Y.-C. 2009. Predicting missing links via local information. *The European Physical Journal B,* 71**,** 623-630.

[169] ZHOU, T. C., MA, H., KING, I. & LYU, M. R. 2009. TagRec: Leveraging Tagging Wisdom for Recommendation. *2009 International Conference on Computational Science and Engineering.* IEEE.

[170] ZHOU, T. C., MA, H., LYU, M. R. & KING, I. UserRec: A User Recommendation Framework in Social Tagging Systems. AAAI, 2010.