Henley
Business School
UNIVERSITY OF READING

Node Similarity in Complex Networks and Its Applications in Online Recommendations

Lei Hou

Submitted in partial fulfilment of the requirements of

University of Reading for

the degree of Doctor of Philosophy

Informatics Research Centre

Henley Business School

November 2018

Declaration

I confirm that this is my own work and the use of all material from other sources has been properly and fully acknowledged.

Name: Lei Hou

Certificate of Readiness to Be Included in Library

I grant powers of discretion to the University Librarian to allow this thesis to be copied in whole or in part without further reference to me. This permission covers only single copies made for study purposes, subject to normal conditions of acknowledgement.

Acknowledgements

It has been a very pleasant process doing my Ph.D. here, through which I can literally feel my steps toward a mature researcher. For any of these steps forward, most credits should go to these who have been pushing me, leading me, or walking alongside with me.

Profound gratitude goes to my supervisor, Prof. Kecheng Liu, who has been so supportive to my research. He has been not only a supervisor making me a better researcher, but also a senior friend sharing his experiences of life, which has been making me a better person. I'm grateful for Dr. Lily Sun's generous care and support over the years. Discussions with her have always been encouraging and enlightening. I would also like to thank again my external supervisor, who was the supervisor of my masters' program, Prof. Jianguo Liu, for his generous help and rigorous comments. It was largely them who made my PhD research smoothly progressed.

I gratefully acknowledge my department, BISA (Business Informatics, Systems and Accounting), and Henley Business School, for the funding which made my work possible, as well as the environment it provided which has been encouraging. The annual events of BISA and also the confirmation viva have been very helpful in improving my research, in which I have received valuable and enlightening comments and questions from Prof. Keiichi Nakata, Prof. Yuksel Ekinci, Dr. Vickey Li, Dr. Yinleng Tan and so many more.

My lovely colleagues have made my personal and professional time at Reading enjoyable. It has been a pleasure to work closely with Xue Pan, Prince Kwame Senyo, Diego Fuentealba and Qi Li. And I also thank Xin Xiang, Biao Mi, Luqiao Zhang, Xiaodong Wang, Xing Huang for making my drinking time not alone. All of them have been a source of not only advice and collaboration, but also friendships for me.

I thank my parents for their unconditional love and putting up with me being so far away from home. Special acknowledgement goes to my loving and encouraging Fiancée, Xue Pan, who has been there supporting me through ups and downs in so many years.

There are so many people who have helped me throughout my Ph.D. that I can never make a complete list. Hereby, I thank them all for their generous helps and supports.

Related Publications

A significant part of the research in the present thesis has been summarised as articles which either have been published in journals/conferences, or are to be submitted. The following list indicates the related publications derived from the author's PhD research.

Journal Articles

Hou, L., Pan, X., & Liu, K. (2018). Balancing the popularity bias of object similarities for personalised recommendation. *European Physical Journal B*, 91, 47.

Pan, X., **Hou, L.**, Liu, K., & Niu, H. (2018). Do reviews from friends and the crowd affect online consumer posting behaviour differently? *Electronic Commerce Research and Applications*. (accepted, to appear)

Hou, L., & Liu, K. (2017). Common neighbour structure and similarity intensity in complex networks. *Physics Letters A*, 381(39), 3377-3383.

Pan, X., **Hou, L.**, & Liu, K. (2017). Social influence on selection behaviour: Distinguishing local-and global-driven preferential attachment. *PLOS ONE*, 12(4), e0175761.

Hou, L., Liu, K., Liu, J., & Zhang, R. (2017). Solving the stability–accuracy–diversity dilemma of recommender systems. *Physica A*, 468, 415-424.

Liu, J., **Hou, L.**, Pan, X., Guo, Q., & Zhou, T. (2016). Stability of similarity measurements for bipartite networks. *Scientific Reports*, 6, 18653.

Conference Proceedings

Hou, L., Liu, K., & Liu, J. (2017, November). Navigated Random Walks on Amazon Book Recommendation Network. In: *The 6th International Conference on Complex Networks and Their Applications*. pp. 935–945. Lyon, France.

Hou, L., Pan, X., & Liu, K. (2017). Identifying informative objects for mining similar users in recommender systems. In: *Proceedings of International Conference on Logistics, Informatics and Service Sciences*. Kyoto, Japan. **(Best paper award)**

Pan, X., **Hou, L.**, & Liu, K. (2017). On the prediction of future helpfulness for online reviews. In: *Proceedings of International Conference on Logistics, Informatics and Service Sciences*. Kyoto, Japan.

Working Papers

Hou, L., Pan, X., Liu, K., & Liu, J. Random walks on recommendation networks for information retrieval. (to be submitted)

Hou, L., Pan, X., Liu, K., & Liu, J. Traps in recommendation networks break down the navigability. (to be submitted)

Abstract

In the online world, especially the e-commerce websites, the interactions among various users and products can be naturally modelled and studied as complex networks. The network-based similarities, also known as the association rules, have thus found wide applications in examining the co-accessing patterns among products and providing recommendations for users accordingly. Focusing on two major forms of online recommendations, including personalised recommendations which are made for specific users, and recommendation networks which are hyperlinks connecting similar products as a networked system, this thesis explores the application of network-based similarity measures in recommendations, and examines the performances of them.

For the personalised recommendation, the recommendation list for a specific user is shown to be changing vastly when the system evolves, due to the unstable quantification of object similarities, which is defined as the recommendation stability problem. To improve the recommendation stability is thus crucial for the user experience enhancement and the better understanding of user interests. By ranking the similarities in terms of stability and considering only the most stable ones, this thesis presents a top-n-stability method based on the Heat Conduction algorithm for tackle the stability problem as well as guarantee the accuracy and diversity of recommendations. Experiments on four benchmark datasets indicate that the proposed algorithm can significantly improve the recommendation stability and accuracy simultaneously and still retain the high-diversity nature of the Heat Conduction algorithm. Furthermore, we show that the dilemma among stability, accuracy and diversity is caused by the popularity bias of network-based similarity measures, that the popular objects tend to have more common neighbours with others and thus are considered more similar to others. Such popularity bias of similarity quantification will result in the biased recommendations, with either poor accuracy or poor diversity. Based on the bipartite network modelling of the user-object interactions, this thesis calculates the expected number of common neighbours of two objects with given popularities in random networks. A Balanced Common Neighbour similarity measure is accordingly developed by removing the random-driven common neighbours from the total number. Recommendation experiments in three data sets show that balancing the popularity bias in a certain degree can significantly improve the recommendations' accuracy and diversity simultaneously.

Objects such as products, news, articles in most online systems are connected to similar others through hyperlinks as recommendations for users. Recommendation networks of objects are thus resulted enabling users to explore the massive relevant online information by surfing from one to another. While it connects overwhelming online objects as networks and seems to be a good way for users to navigate the haphazard content-browsing systems, two outstanding questions still exist that, 1) can the users locate their interests by surfing on the network, and 2) is every object accessible in the network? By mining the co-accessing pattern among objects, we construct recommendation networks according to the object similarity matrix, and thereby theoretically explore its topology and dynamics. Modelling the users' surfing behaviour as random walks, we examine how many history records of a target user can be retrieved during such process. Most measures are shown with limited accuracy and cannot help users to explore niche objects which may be not popular but fit some users' interests. In order to achieve a good accuracy quickly in a short-term random walk, we show that the recommendation list should be short, where each object is expected to have generally 2~6 recommended objects. In terms of accessibility, the recommendation networks are shown to be unnavigable due to the emergence of traps, which are dense communities with few or even no links connecting outside. Such vicious cycles trapping surfing users constantly make a handful of objects dominating most of the web traffic. According to the local structure of the network, a simple measure entitled the local return rate is developed, which can be used to accurately and efficiently identify the significant traps in large-scale recommendation networks.

To summarise, this thesis uncovers some fundamental challenges with network-based online recommendations including the stability problem and popularity bias for personalised recommendation and the information monopoly for recommendation networks. In addition to the proposed algorithm, measure and analytical methods, the results inform the needs of more careful system design in practice, and shed lights on the future studies on such challenges to develop better network-based similarity measures.

Contents

List of Figures

List of Table

List of Notions and Abbreviations

Symbol or Abbreviations	Meaning
$\mathbf{A}$	Adjacency matrix of a network, with elements $\boldsymbol{A} = \{a_{ij}\}_{N\times N}$, where $a_{ij} = 1$ if node i is connected with node j, and $a_{ij} = 0$ otherwise.
N	The size (number of nodes) of a unipartite network; or the number of objects in user-object bipartite network.
M	The number of users in user-object bipartite network.
L	In the context of online recommendation, it represents the length of the recommendation lists, i.e. number of recommendations per user or per object.
$\langle . \rangle$	The mean value calculator for the entities inside.
k_i	The degree (number of links) of a node i.
k_i^{out}	The out-degree (outgoing links) of a node i in the context of directed networks.
$\langle k \rangle$	Average degree of a network.
s_{xy}	Similarity between node x and node y.
Γ_i	The set of nodes who are the neighbours of a target node i
$\lvert . \rvert$	If the entity inside is a set, then it represents the number of elements in this set; If the entity inside is a value, it returns the corresponding absolute value.
CN	Common Neighbour similarity measure
SAL	Salton similarity measure
SOR	Sorensen similarity measure
JAC	Jaccard similarity measure
HP	Hub-Promoted similarity measure
HD	Hub-Depressed similarity measure
LHN	Leicht-Holmes-Newman similarity measure
AA	Adamic-Adar similarity measure
RA	Resource Allocation similarity measure
MD	Mass Diffusion similarity measure
HC	Heat Conduction similarity measure
$P(L)$	Precision of recommendations with list length of L
$R(L)$	Recall of recommendations with list length of L
RS	Ranking score of the recommendations, representing accuracy
$Nov(L)$	Novelty of recommendations with list length of L
$S(L)$	Personalisation of recommendations with list length of L
Δ	The stability of a recommended objects.
TNS-HC	Top-N-Stability method on the HC algorithm
BCN	Balanced Common Neighbour similarity measure.
LRR	Local Return Rate of random walks.

Chapter 1. Introduction

1.1 Background and Motivation

Thanks to the rapid development of the Internet technology, our society has witnessed us picking up an advanced lifestyle in the recent decade. We communicate online, buy goods online, read news online, watch movies online…… Basically, we live a significant part of our lives on the Internet. Especially with the technology of Web 2.0, we are not only the receiver but also the creator of online information. While there should be more than enough information fulfilling every aspect of our daily needs, the amount of online information is significantly more than any individual could ever possibly process which is known as the information overload problem (Bawden *et al.* 1999; Hwang, & Lin 1999; Eppler & Mengis 2004). In such overwhelming sea of information, though anything could be just a few clicks away (Albert *et al.* 1999), users are still struggling to find the most relevant information.

The explosion of online information makes the technique helping users to filter information urgently needed. The search engine is a basic and well-established such technique, and significantly helps users in filtering the irrelevant information by searching for specific keywords (Hanani et al. 2001). However, the search engine still does not fulfil all the needs of online users. On one hand, search engine returns the same results for different users searching the same keywords, even if they have totally different interests and are expecting different target information. One the other hand, users may not know exactly what they are looking for, such as a user looking for an interesting movie just to kill a boring night, or a researcher looking for some related papers. Regarding these limitations, many other techniques have been developed, among which, the recommender system (Resnick & Varian 1997; Ricci *et al.* 2011) is of great significance serving online users.

The recommender system analyses the patterns of users accessing online information and evaluates the correlation, normally referred as similarity, among these information, and thereby predicts appropriate information that might be of a user's interests automatically as recommendations. Accordingly, users do not have to search for information by themselves and may be able to explore information that are out of their knowledge. Generally, there are two widely-implemented forms of recommender systems, namely the personalised recommendation (Schafer *et al.* 2007; Lv *et al.* 2012) and the recommendation network (Oestreicher-Singer & Sundararajan 2012a; Lin *et al.* 2017).

The personalised recommendation deals with the question that what objects would a target user be interested in in the future. Accordingly, such system considers not only the similarity among objects, but also, more importantly, the target user's historical behaviours. For example, when a target user bought a book on network theory from Amazon, more books on network theory or graph theory will be recommended to him/her by Amazon displaying on the homepage in the following days. Briefly speaking, such system recommends either objects that are similar to the target user's historical selections (object-based algorithm), or objects that have been selected by users that have similar history records to the target user (user-based algorithm). Therefore, the personalised recommendations are make for a specific user, hence personalised, and are generally different for different users.

The recommendation network, on the other hand, is emerged through similar objects. Such recommendations are made for an object. All the users who browse a particular object get the same recommendations. Generally, for a specific object, such system recommends several other objects that are similar to this one on its webpage, to inspire users to keep browsing relevant objects. Accordingly, every object in a website would have a list of recommended objects connecting with hyperlinks. Such system thus would result in a massive network of objects, and this is the reason that it is normally referred as recommendation networks.

In practical systems, both personalised recommendations and recommendation network are normally available. In Amazon for example, when a user logged in to the system, there are lists of objects displayed, entitled such as "Related to items you've viewed", "Inspired by your browsing history" and "Recommendations for you" etc. which can be regarded as personalised recommendations. But while a user actually browsing an object, it is where the recommendation network presented entitled "Customers who bought this item also bought". In terms of the process of users accessing information, the personalised recommendations are to be taken before accessing, while the recommendation network is to be taken while accessing.

Despite the differences between personalised recommendation and recommendation networks, they share the same key technique, namely, how to quantify the similarity among objects. One of the most efficient and widely-applied such methods is the collaborative-based similarity (Goldberg *et al.* 1992), also known as the association rules (Sandvig *et al.* 2007). Normally, two objects would be considered similar if they share common neighbours, i.e. the same users who selected both. Consequently, such similarity describes to what extend would two objects be interested in, appreciated or purchased by the same user.

The study of complex networks (Albert & Barabási 2002, Newman *et al.* 2006) provides rich theoretical foundation for the collaborative similarities. The networks science originates from the graph theory (Erdös & Rényi 1959; 1960), and got significant boost in late 1990s (Watts & Strogatz 1998; Barabási & Albert 1999) since more and more data of real-world networks became available and computable. A network normally consists of a set of nodes (any kind of entities) and a set of links (any kind of interactions) connecting these nodes. Due to its elegant modelling, complex network has been applied to study a wide range of practical systems, from the biological network (Cline *et al.* 2007), to the Word-Wide Web (Albert *et al.* 1999), from the transportation network (Colizza *et al.* 2006) to the global social network (Guimera *et al.* 2003). While the online recommendations deal with the interactions between objects and users, complex networks provide an effective model, namely the user-object bipartite networks (Zhou *et al.* 2007; Daminelli *et al.* 2015), based on which the user behaviour and object (node) similarities can be explored.

1.2. Research Questions and Objectives

Given the significance of online recommendations, and the solid foundation that network science could provide, the present PhD thesis aims to *explore node similarity quantifications in complex networks, and examine and improve online recommendations accordingly*.

To achieve such objective, the research follows four steps, each focusing on one major research question described as follows:

There are dozens of similarity measures proposed for quantifying node similarities in networks and had been applied in making recommendations. However, the real networks, such as online user-product interactions, are always evolving and incomplete. With such dynamic changes of the data, we explore: **Research Question 1** (**RQ1) can the object similarities remain stable over the data change and how can we ensure the recommendation stability**? In exploration of such a question, the objectives include to

- establish a framework to evaluate the stability of similarity measures in bipartite networks;
- explore the influence of similarity stability on the recommendation stability;
- propose an algorithm which can ensure the stability, diversity and accuracy simultaneously.

Most existing network-based similarity measures have apparent degree bias, which is the close correlation (either positive or negative) between the node degree and the similarity. A severe problem it would cause is that popular objects always tend to be similar (in positive correlation) or dissimilar (in negative correlation) to others, leading the online recommendations to be either inaccurate or uniform. Accordingly, we study **Research Question 2** (**RQ2) how can we develop a balanced similarity measure by comparing the empirical network with random ones**? The objectives are to

- uncover the existence of degree bias of existing similarity measures and evaluate its severity;
- conclude the numerical correlation between the expected number of common neighbours and the degrees in random networks;
- develop a new similarity measure for both unipartite and bipartite networks by removing the random-based common neighbours.

The recommendation networks are, in nature, information networks. However, previous research only explored their commercial value and their influences on product demand, leaving many fundamental aspects of recommendation networks uncharted. Accordingly, this PhD thesis focuses on two major questions regarding recommendation networks, i.e. the navigation accuracy and information accessibility.

While the major function of recommendation networks is to aid users in their content exploration in the system, we ask **Research Question 3** (**RQ3) how accurately can recommendation networks navigate users to find what they interested in**? The objectives include to

- theorise the construction of recommendation networks by a top-L projection method on user-object interactions;
- report a general picture of the topological structure of recommendation networks;
- develop an evaluation method for the accuracy of the recommendation networks;
- conclude the optimised recommendation list length to achieve best accuracy.

Since the recommendation networks provide an easy way to explore various objects online, it is important to ensure that at least most objects are accessible. Therefore, we study **Research Question 4** (**RQ4) are the objects accessible and is the recommendation network navigable?** The objectives are to

- conclude the theoretical navigability of random recommendation networks;
- quantify the accessibility and navigability of empirical recommendation networks for further evaluation of the existing construction methods;
- discover and quantify the existence of traps which block the web traffic;
- develop a method for efficient identification of traps in large-scale networks.

1.3. Organisation of the Thesis

The present PhD thesis is organised as shown in Figure 1.1, where the main research is divided into two themes namely the personalised recommendation and the recommendation networks.

Chapter 1 is the introduction to this thesis which sets the context and defines the research questions and objective. The research design and methodologies adopted in the thesis are also introduced and their validity is justified via discussions.

Chapter 2 reviews the literature related to the study in this thesis, which is composed by four parts, including the complex networks and bipartite modelling, network-based similarity measures, personalised recommendation and recommendation networks.

Chapter 3 introduces the datasets that will be applied in latter studies, including the book recommendation network we collected from Amazon and several open datasets of user-object interactions.

The original research of this PhD thesis will be majorly reported in Chapter 4 to Chapter 7, consisting of two themes of online recommendations, namely the **personalised recommendation** (Chapter 4 and 5), and **recommendation networks** (Chapter 6 and 7).

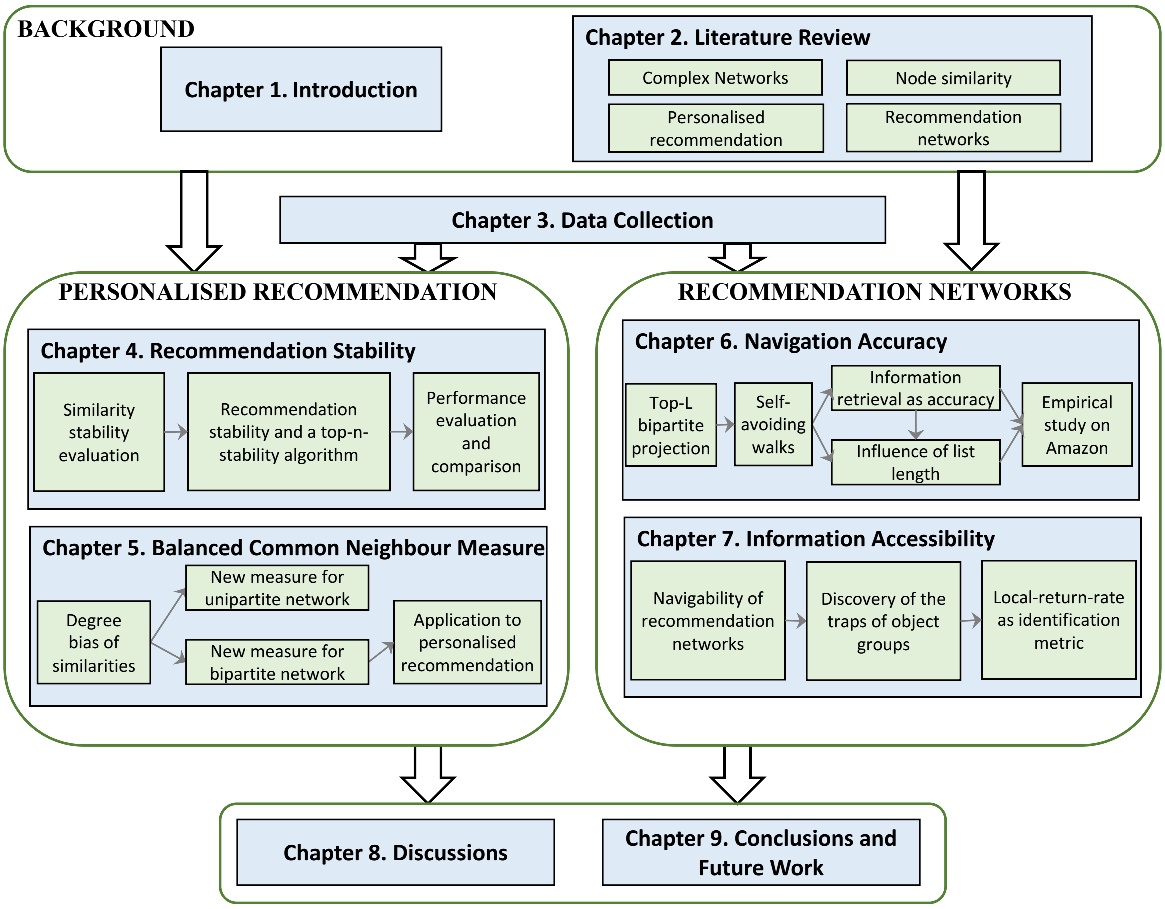

BACKGROUND
Chapter 1. Introduction
Chapter 2. Literature Review
Complex Networks
Node similarity
Personalised recommendation
Recommendation networks
Chapter 3. Data Collection
PERSONALISED RECOMMENDATION
Chapter 4. Recommendation Stability
Similarity stability evaluation
Recommendation stability and a top-n-stability algorithm
Performance evaluation and comparison
Chapter 5. Balanced Common Neighbour Measure
Degree bias of similarities
New measure for unipartite network
New measure for bipartite network
Application to personalised recommendation
RECOMMENDATION NETWORKS
Chapter 6. Navigation Accuracy
Top-L bipartite projection
Self-avoiding walks
Information retrieval as accuracy
Influence of list length
Empirical study on Amazon
Chapter 7. Information Accessibility
Navigability of recommendation networks
Discovery of the traps of object groups
Local-return-rate as identification metric
Chapter 8. Discussions
Chapter 9. Conclusions and Future Work

Figure 1.1 | Organisation of the present PhD Thesis.

Chapter 4 studies the stability problem of similarity quantification and personalised recommendation, tackling **RQ1**. After empirically analysing the similarity stability, we propose a top-n-stability algorithm to secure the good performance of personalised recommendations in terms of stability, accuracy and diversity. This chapter is largely based on our published papers including

- Hou, L., Liu, K., Liu, J., & Zhang, R. (2017). Solving the stability–accuracy–diversity dilemma of recommender systems. *Physica A*, 468, 415-424.
- Liu, J., Hou, L., Pan, X., Guo, Q., & Zhou, T. (2016). Stability of similarity measurements for bipartite networks. *Scientific Reports*, 6, 18653.

Chapter 5 focuses on the degree (popularity) bias of similarity measures in networks, tackling **RQ2**. We theoretically study both unipartite and bipartite random networks, to calculate the expected common neighbours between nodes. Accordingly, a balanced common neighbour similarity measure is developed. Furthermore, we apply the new measure to make personalised recommendation, and the performances are studied. The calculation and results are based on our published papers including

- Hou, L., Pan, X., & Liu, K. (2018). Balancing the popularity bias of object similarities for personalised recommendation. *European Physical Journal B*, 91, 47.
- Hou, L., & Liu, K. (2017). Common neighbour structure and similarity intensity in complex networks. *Physics Letters A*, 381(39), 3377-3383.

Chapter 6 theorises the construction of recommendation networks as a top-L projection, and explores the accuracy of such system helping users to find the right objects, which gives answer to **RQ3**. This chapter is based on a working paper, and a published conference paper which are

- Hou, L., Pan, X., Liu, K., & Liu, J. Random walks on recommendation networks for information retrieval. (working paper)
- Hou, L., Liu, K., & Liu, J. (2017, November). Navigated Random Walks on Amazon Book Recommendation Network. In: *The 6th International Conference on Complex Networks and Their Applications*. pp. 935–945. Lyon, France.

Chapter 7 explores the navigability of recommendation networks corresponding to **RQ4**. We theoretically and empirically study the navigability of recommendation networks which is the portion of objects that can be visited by users, and uncover the common existence of traps in such systems. We further develop a simple local method, which is efficient to identify traps in large-scale recommendation networks. This chapter is based on a working paper

- Hou, L., Pan, X., Liu, K., & Liu, J. Traps in recommendation networks break down the navigability. (working paper)

Chapter 8 discusses the results of the thesis in depth and their validation via comparative analysis. The limitation of the study in this thesis is discussed as well.

Chapter 9 summarises the major findings and contributions, and discusses the future work.

1.4. Research Methodology

The research in this thesis is, in nature, multi-disciplinary and being at the interface among management science, computer science, mathematics and statistical physics. As a consequence, our methodology is a combination of a series of both theoretical and empirical approaches.

In abstract, the studies in this thesis generally follow the following procedure: 1) develop recommendation algorithm, 2) implement the algorithm to empirical data, and 3) evaluate the performance. The development of recommendation algorithms is a theoretical step, where we, according to literature and observations of user behaviour, propose theories and assumptions based on which the algorithm is generated. The implementation of algorithms is a very practical step, where we make the algorithms applicable by programming them as a system, which takes empirical data in and gives recommendations accordingly. The evaluation largely depends on the partition of empirical data, i.e. we divide the user-object interactions into two parts, namely the training data and testing data. Training data will be fed to the developed recommendation system, and the outputs of the algorithms will be compared to the testing data to assess their performances. Metrics will be adopted from literature, as well as developed by ourselves, to evaluate a recommendation algorithm's performances in different dimensions such as accuracy, diversity and stability. A key step to conclude that our proposed algorithm is a valid contribution is to compare the performance of proposed algorithm to benchmark algorithms which have been widely recognised and applied in the field.

The following is the detailed methodologies applied in this thesis.

The theoretical part of the research includes the development of algorithms for similarity measurement and recommendation, design of the evaluation framework for recommendations, and the analysis of the expected consequences of certain behavioural patterns. The adopted methodologies are literature analysis, mathematical modelling, complex network theory and mathematical analysis. Through the review of relevant literature and theories, we extract and conclude the ontology of consumer behaviour or the relations among objects. Then abstract mathematical models are developed to describe and represent the complex system of user-object interactions. With the behaviours being regarded as dynamics on the complex networks, we apply mathematical analysis via probabilistic and statistical theory to theoretically calculate the consequence of certain behavioural patterns, which will serve as part of the evidence to inform the results and conclusion, or as the comparison for empirical studies.

The present thesis is also extensively data-driven, and thus adopts a large amount of empirical approaches. The adopted methodologies include web crawling, data analytics, Monte Carlo simulation, and algorithm implementation via programming etc. For the data collection, we retrieved six second-handed data sets on the user-object interactions from published papers, as well as developed a web crawler based on Python (with selenium package) and crawled the data from Amazon book recommendation. For all the analysis of these data sets, including their statistical features and in-depth patterns such as similarities, stabilities etc., we programmed from scratch based on Python environment with no developed software or tools been applied. The proposed personalised recommendation algorithms as well as the benchmark algorithms are all implemented via C-programming and Python, and tested through the training and testing data partition on the applied data sets. The Monte Carlo method is used to simulate the users surfing behaviour in Chapter 6 and 7, where the simulation programme is also developed by ourselves based on C-programming. To more clearly present the analytical results, many figures of plots are presented in this thesis, which are produced via Matlab and Python (with matplotlib package).

1.5. Novelty of the Thesis

This thesis explores the node similarity quantification in complex networks in order to improve the performances of recommender systems. As the major novelty of the thesis, we focus on the mismatch between the practical needs of recommender system and the current similarity measures.

An important need of the current online recommendation systems is the ability to uncover niche objects, which is the major focus of this thesis. The ‘niche object’ is a term for the differentiation from the concept of ‘popular object’. Unlike popular objects which are basically the common interests of mass population of users, the niche objects are normally not popular, but can represent the interests of small groups of users. Niche objects should be distinguished from the unpopular objects. The latter are simply unpopular, while the niche objects still fit some users’ interests. For most users, their interests may consist of both common interests which can be fitted by the popular information, and personalised interests which can only be fitted by the niche information. However, numerous valuable niche information is nowadays hidden in the dominance of popular information. There are many channels that are continuously enhancing the dominance of the popular information. With the development of the mass media, the popular information such as Oscar movies, best seller books or famous-branded products are almost impossible to be avoided by most users. In addition, when users initiatively accessing information, the search engines are already putting emphasises on the popular information, due to the bias of the indexing and ranking (Fortunato *et al.* 2006b; De Corniere & Taylor 2014). Therefore, what users expect from the recommender system are the alternatives that are out of their knowledge. For most users, recommending popular information is of little value, as they may have already well aware of these information (McNee *et al.* 2006; Vargas & Castells 2011). If a user wants these popular information, it's easy for him/her to access through search engine or the mass media. Instead, if this user is still looking for information, niche information would be normally expected. To summarise, though there are more than enough information online to fit users' interests, to find niche information still takes enormous efforts, and remains to be the challenge for online recommendations.

The widely applied collaborative-based recommendations fail to fulfil such need of finding niche information, due to the popularity bias of existing network-based similarity measures. Since the key technique in collaborative filtering is the quantification of object similarities, the choice of similarity measure basically determines the performance of the resulted recommendations. However, the existing network-based similarity measures are found with serious popularity bias (Chapter 5), which is the correlation between measured similarities and the object popularities. Popular objects tend to be considered by these measures to be either very similar (positive correlation) or dissimilar (negative correlation) to others, rather in a balanced manner. As a consequence, such bias may cause is that the recommendations are thus also biased. The recommendation lists for users are dominated by either very popular objects, or very unpopular objects, and thus the goal of finding niche objects cannot be achieved.

This thesis aims to quantify and try to overcome such popularity bias in Chapter 4 and Chapter 5, by looking at the stability of object similarities and developing a balanced similarity measure respectively. In addition, we also explore the other form of recommender system, i.e. the recommendation network. While previous related research only focusses on its marketing value, this thesis for the first time develops a systematic framework for the evaluation of its accuracy of navigation (Chapter 6) and accessibility of objects (Chapter 7).

Chapter 2. Literature Review

In this chapter, literature on some of the key aspects relating to this thesis is reviewed comprehensively. As suggested by the major objectives of this thesis, the review is organised as following: section 2.1 introduces the concepts of complex networks and bipartite networks, and the random walks in networks; section 2.2 reviews the network-based similarity measures which are applicable to collaborative online recommendation; section 2.3 and 2.4 summarises the progresses and key studies on the two major techniques of online recommendation, namely the personalised recommendation and recommendation networks respectively.

2.1. Complex Networks and Bipartite Modelling

The term 'network' has been used in various scenarios and disciplines. In this thesis, the network is a general term referring to a system in which a number of actors interact with each other. The reason we use the term 'complex networks' lies in the fact that the systems/networks we consider are mostly with nontrivial structures which make them distinguished from the random graphs. In most part of the present thesis, we study the complex networks of user-object interactions, i.e. the actors are users and objects (such as products, movies etc.) and the interaction normally represents the purchase, comment behaviours etc.

Mathematically, a network is a collection of nodes $\boldsymbol{V} = \{v_1, v_2, \cdots, v_N\}$ and links $\boldsymbol{E} = \{e_1, e_2, \cdots, e_K\}$. The connection of a network is normally represented by an adjacency matrix $\boldsymbol{A} = \{a_{ij}\}_{N\times N}$ where $a_{ij} = 1$ if node i is connected with node j, and $a_{ij} = 0$ otherwise. For example, a social network is generally the collection of individuals as nodes and relationships as links, while the World-Wide Web is the collection of webpages as nodes and hyperlinks as links. The large amount of nodes and links in a network would make it rather 'complicated' than 'complex'. What really makes a network 'complex' is the fact that in most networked-systems these massive amount of nodes are interacting with each other through links, and there are clear patterns with these interactions.

The field of complex networks is rather big and multidisciplinary. In terms of network types, there are unipartite networks, bipartite networks (Newman *et al.* 2002), temporal networks (Holme &

Saramäki 2012), inter-dependent networks (Buldyrev *et al*. 2010) etc. Many aspects of complex networks are also paid significant attentions, such as the modelling (Barabási & Albert 1999; Papadopoulos *et al*. 2012), robustness and resilience (Albert *et al.* 2000; Gao *et al.* 2016), spreading dynamics (Pastor-Satorras & Vespignani 2001; Moreno *et al.* 2004), and the synchronisation (Arenas *et al*. 2008). There are also many disciplines being interested and contributed to the study of complex networks including Physics (Boccaletti *et al.* 2006), Mathematics (Newman 2003), Biology (Thiery & Sleeman 2006), Computer Science (Silva & Zhao 2016), Social Science (Alvarez-Galvez 2016), Political Science (Conover *et al*. 2011) and so on. In terms of the applied context, there are even more categories since basically any system with entities interacting with each other can be modelled as complex networks.

Consequently, it is almost impossible to make a complete review on complex networks especially in a thesis. This section thus focuses on these that are most relevant to the present thesis. For more thorough reviews, there are quite many good papers available, such as Newman (2003), Boccaletti et al. (2006) and Barabási (2016) etc.

2.1.1. Basic structural properties

The real-world networks, though are complex, always exhibit regularities. Especially, many totally different networked systems have been found with very similar properties such as small average shortest path, high clustering feature, power-law degree distribution and community structures. Such properties attracted significant attentions in the past decades and made complex networks efficient in modelling real systems.

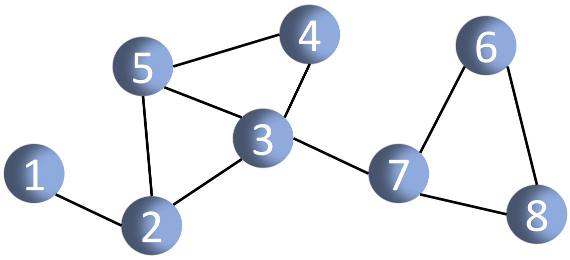


Figure 2.1 | A toy network consisting of 8 nodes and 10 links.

Shortest-path length

Normally, networks are connected, which means there would be at least one path that an arbitrary node can reach another one. The length of the paths connecting two nodes became an interest of the field from very early stage of graph theory. As shown in Figure 2.1, which is a connected network, every node can reach every other. Between node 1 and node 4, there are many paths such as {1,2,3,4}, {1,2,5,4} and {1,2,3,5,4}. The length of a path is the number of links involved in the path, and thus the aforementioned paths have length of 3, 3, and 4 respectively. Apparently, the length of the shortest path between node 1 and 4 is $d_{1,4} = 3$. To characterise such basic property of a network, one can use the average shortest-path length L, which is the mean value of the shortest-path length between every pair of nodes (Watts & Strogatz 1998), i.e.

$$L = \frac{\sum_{i,j \in \boldsymbol{V}; i \neq j} d_{ij}}{N(N-1)}, \tag{2.1}$$

where $\boldsymbol{V}$ is the set of nodes in the network and N is the size of the network, i.e. total number of nodes. An alternative method, when the network is not completely connected (not every pair of nodes has a path, $d_{ij} = \infty$), is to calculate the mean of the reciprocal of the shortest-path length, which is defined as efficiency E (Latora & Marchiori 2003) and reads,

$$E = \frac{\sum_{i,j \in \boldsymbol{V}; i \neq j} 1/d_{ij}}{N(N-1)}. \tag{2.2}$$

Such quantity has great implications for real-world networks, describing how well the network is connected and how efficient the network accommodates diffusion dynamics. In the 1960s, Thiery and Milgram carried out an experiment in which participants were required to send a parcel to a stranger. The only approach they can take is to send the parcel to their friends and let their friends to further pass the parcel on. A shocking finding of this experiments is that the average number of intermediate friends lies between five and six, which means for two random people, they can know each other through generally six intermediate friends. Such phenomena have been known as the '*six degrees of separation*' (Thiery & Milgram 1967). In recent years, evidences have shown that the average shortest-path length of social networks is normally 4~5, such as in university email network (Kossinets & Watts 2006), and in Facebook (Wilson *et al.* 2009; Backstrom *et al*. 2012). The World-Wide Web, though consists of numerous webpages, is also found with a short average shortest-path length. Albert *et al*. (1999) have investigated the complete network within the domain of nd.edu, and found it with an average shortest-path length of $L_{nd.edu} = 11.2$. They conclude a correlation between the Web size N and the corresponding L, which reads (Albert *et al*. 1999)

$$L = 0.35 + 2.06\log{(N)}. \tag{2.3}$$

Accordingly, while the size of the Web at that time was estimated to be 8×10^{8}, the average shortest path length is believed to be $L_{web} = 18.59$, which means one can surf to a random webpage from any starting page by 18.59 click on average.

Clustering

For several decades, networks were regarded random, until it was found that real networks have strong clustering features that a node's neighbours tend to connect to each other (Watts & Strogatz 1998). To quantify such feature, the clustering coefficient of a node i is defined as

$$C(i) = \frac{2t_i}{k_i(k_i - 1)}, \tag{2.4}$$

where t_i is the number of triangles involving node i, and k_i is the node's degree, i.e. how many connections it possesses. Taking the toy network shown in Figure 2.1 as an example, the node 7 has three neighbours ($k_7 = 3$), and there is only one triangle which is {7,6,8} involving node 7. As a result, the clustering coefficient of the node is $C(7) = 1/3$. In general, a node's clustering coefficient describes the closeness of its neighbours.

The clustering coefficient of a network can thus be calculated by averaging over every node, i.e. $C = \langle C(i) \rangle$, where $\langle . \rangle$ represents the mean of the entity. Accordingly, the clustering coefficient C has been used to characterise "the cliquishness of a typical neighbourhood". Applying such quantity, Watts and Strogatz (1998) reported that many real-world networks have high clustering coefficients, such as the collaboration network of film actors ($C = 0.79$) and the neural network of the worm Caenorhabditis elegans (C =0.28), together with small shortest-path length, which has been widely discussed in the following decades as the 'small-world' property.

Based on such standard clustering coefficient, other definitions have also been developed. Lind et al. (2005) develop a clustering coefficient by calculating the ratio of the observed squares, rather than triangles, over the possible squares. While the standard clustering coefficient can be regarded as the tendency of resulting cycles of length three, thus denoting with C_3, this quantity measures the density of cycles of length four in a network, hence C_4. Accordingly, they further define higher order clustering coefficient C_n describing the occurrence of cycles of generic length (Lind & Herrmann 2007). Considering that the standard clustering coefficient is normally correlated with the node's degree, Soffer and Vazque (2005) propose a new definition which removes such degree-correlation biases.

Degree distribution

Normally, the degree of a node i, k_i, is the number of links it possesses. For example, in Figure 2.1, the degrees of nodes 1, 2, 3 are $k_1 = 1$; $k_2 = 2$ and $k_3 = 4$ respectively. In a directed network, where the links are directed, every node possibly has both in-coming links and out-going links. Thus, the degree in such networks can be distinguished as out-degree and in-degree.

The degree is the most fundamental measure of a node in networks. The degree characterises, for example, in a social network, how many friends a person has; or in the World-Wide Web, how many hyperlinks directing to a webpage. As a consequence, degree has been a focal point in the study of networks such as the spreading dynamics (Kitsak *et al.* 2010; Brockmann & Helbing 2013), controllability (Liu *et al.* 2011) and so on. In real networked systems, the degree varies largely, that some nodes could have very large degrees while the isolated nodes have degrees of zero. Such fact makes the distribution of degrees vital for understanding a network's structure and the dynamics of/in it. The degree distribution $P(k)$ is the probability density function of degree, which is the percentage of nodes with degree of k. Taking the toy network shown in Figure 2.1 as an example, the degree sequence is {1,3,4,2,3,2,3,2}, and thus we have $P(1) = 1$, $P(2) = 3$, $P(3) = 3$ and $P(4) = 1$.

One of the earliest significant studies that boosted the development of network science, is the discovery of the power-law degree distribution (Barabási & Albert 1999). It is shown that, in most real-world networks, the probability that a node has a degree k, follows $P(k) \sim k^{-\gamma}$ for large k, which is a power-law distribution. Such distribution exhibits linear pattern in a log-log-scaled plot, since one has $\log\big(P(k)\big) = -\gamma \log{(k)} + \varepsilon$, where γ characterises the decay speed of the degree distribution. The power-law distribution is a very heterogeneous distribution which implies that the nodes are very different from each other in terms of degree. Accordingly, there may emerge some nodes with extremely high degree, which are normally referred as hubs, while most nodes have only a few connections. One can imagine the example of a social network, such as Twitter, where celebrities may have millions of followers (in-degree), while most users have only tens or even less followers. It turned out surprisingly that most other networks, ranging from social networks to biology networks, from infrastructure networks to information networks, follow the similar power-law degree distribution. Barabási & Albert (1999) report that as shown in Figure 2.2, the network of film actors, the network of World-Wide Web and the network of power grid follow power-law degree distributions with $\gamma = 2.3$, $\gamma = 2.1$ and $\gamma = 4.0$ respectively.

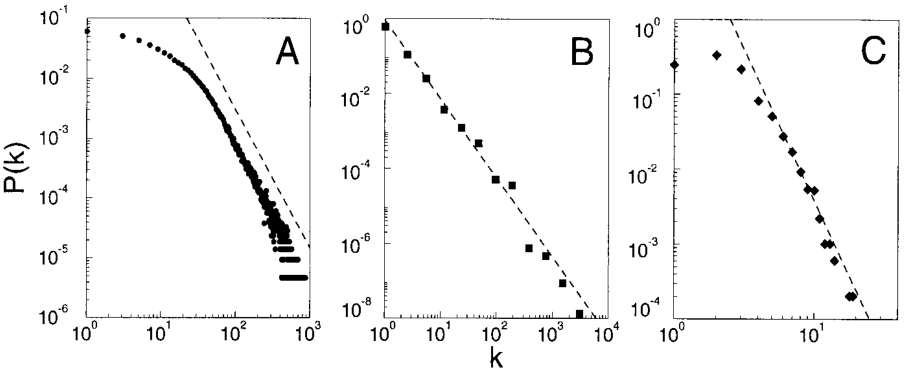


Figure 2.2 | Degree distributions of three typical real-world complex networks, which are (A) actor collaboration network, (B) WWW network and (C) the US power grid network. (From [Barabási, A. L. and Albert, R. (1999). Emergence of scaling in random networks. *Science* **286(5439)**, 509-512]. Reprinted with permission from AAAS.)

Community structure

It is commonly observed that, real-world networks generally have community structure, i.e. groups of nodes that have dense connections within the group, but sparse connections between groups (Girvan & Newman 2002). Figure 2.3 shows a network we collected from Twitter. Businesses develop official accounts in twitter to connect with their consumers, and normally quite a number of official accounts would be developed if the business is large enough, with each account being responsible for one area of its business. For example, @GooglePlay and @YouTube are two distinct accounts but both affiliated to the Google LTD. Following such way, the accounts of five IT companies, namely Google, IBM, Amazon, Microsoft and Dell, are collected along with the following relations among them. It can be clearly observed that the accounts that belong to the same company have much denser connections within them than with others. Thus, the network can be regarded as with five communities.

The determination of whether a network has community structure or not is rather arbitrary. Most commonly, the modularity can be applied to examine the goodness or the closeness of the community structure (Newman & Girvan 2004). For a given division of a network into n communities, an $n \times n$ matrix can be defined as $\boldsymbol{e}$ with e_{ij} as the ratio of links connecting community i and j, over the total links in the network. One can further calculate the row sum of the matrix as $a_i = \sum_j e_{ij}$ which is the fraction of the links that connecting to community i. For two arbitrary communities i and j, the expected links between them can thus be expected to be $a_i a_j$. Accordingly, the modularity is mathematically defined as

$$Q = \sum_i (e_{ij} - a_i a_i). \tag{2.5}$$

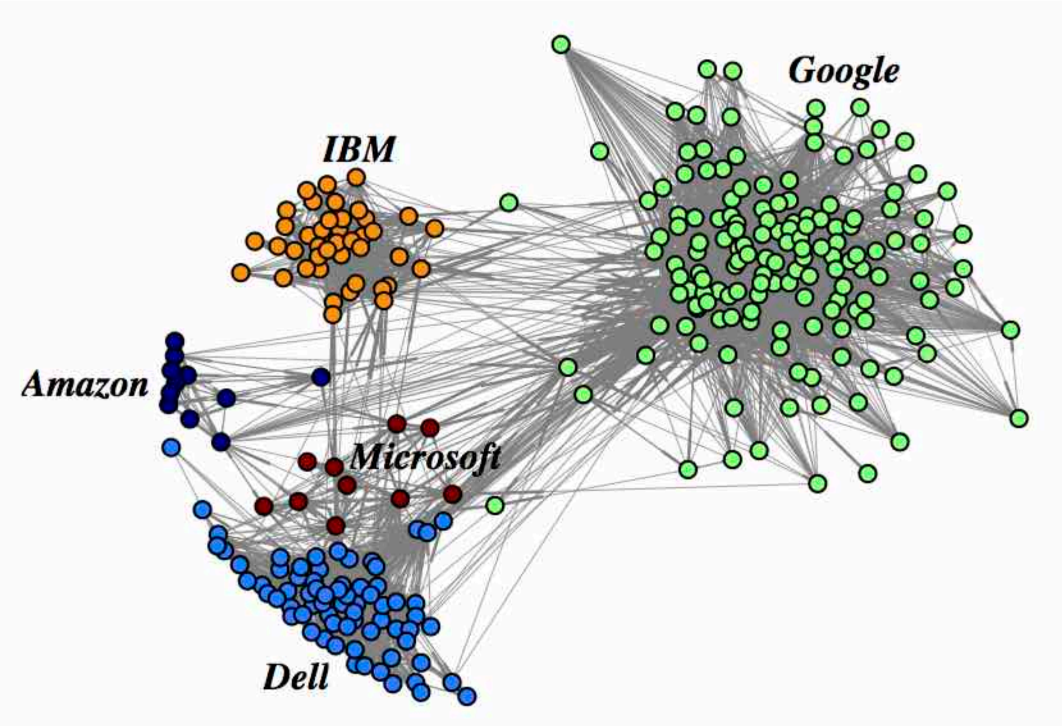


Figure 2.3 | A network from Twitter to show the community structure. In this network, each node is an official account of an IT company, and the directed links are the following relations among them. The network is colour-coded, with each colour represents one company.

According to its definition, the modularity describes how many of the links are connecting the nodes within the same community in comparison to the random cases. Basically, large value of modularity Q (approaching 1) suggests that the communities are well partitioned, while $Q = 0$ indicate that the ratio of inter- and intra-community links are similar to the random networks.

2.1.2. Network models

Inspired by the discovery of properties emerged from the empirical networked systems, models that generate artificial networks have been widely developed so that the networks can be studied in a controlled fashion. There are quite a number of models been developed, each specifically addressing some particular mechanism observed in empirical studies. Here we only introduce three widely-applied network models with very general mechanisms, which will be applied in the present thesis.

Erdös-Rényi random graph

Before the boost of network science in the late 1990s, networks were regarded totally random and the Erdös-Rényi (ER) networks (Erdös & Rényi 1959; 1960) was the most-investigated model.

In the ER model, an empty network with N nodes is considered at first. For each pair of the nodes, there is a fixed probability p for a link to be established to connect them. With $N(N-1)/2$ pairs of nodes, the total number of links is expected to be $pN(N-1)/2$. Accordingly, the average degree of the nodes is expected to be $\langle k \rangle = p(N-1)$. For an arbitrary node, the probability of its degree being k, which is also the degree distribution function of the ER network, is thus

$$P(k) = \binom{N-1}{k} p^k (1-p)^{N-1-k}, \tag{2.6}$$

which is a binomial distribution, where $\binom{b}{a} = \frac{b!}{a!(b-a)!}$. At the limit of large population $N \to \infty$, the exact solution of the degree distribution function for an ER network is

$$P(k) = \frac{(pN)^k e^{-pN}}{k!}, \tag{2.7}$$

which is a Poisson distribution. Consequently, the ER network is also normally referred as Poisson random graph (Newman 2003).

Imagine a node in an ER network, with degree of k. There could be at most $k(k-1)$ links among its neighbours, but the expected links are $pk(k-1)$. Consequently, the clustering coefficient of ER random graphs can be expected to be

$$C^{ER} = p. \tag{2.8}$$

Despite that it has been found that most real-world networks do not follow the Poisson degree distribution and have generally denser clustering property, ER model has been widely used as benchmark in various theoretical network studies, such as the cascading failure (Crucitti *et al.* 2004; Buldyrev *et al.* 2010), spreading dynamics (Nekovee *et al.* 2007; Manshour & Montakhab 2014) and controllability (Liu *et al.* 2011; Liu & Barabási 2016) in networks.

Barabási-Albert network

The discovery of the scale-free feature of networked systems (Barabási & Albert 1999) largely boosted the study of complex networks. As discussed earlier, the real-world networks are found mostly with power-law degree distributions, rather than the Poisson distribution predicted by the ER network.

To tackle the inability of existing network models, including the ER random graph and the small-world network (Watts & Strogatz 1998), and explain the origin of the commonly existed power-law degree distributions, Barabási and Albert (1999) argue two generic mechanisms. First, a growth mechanism should be incorporated to the network model, instead of setting a fixed population of nodes N. Second, the preferential attachment mechanism should be considered, which dominates the new nodes to have higher probability to connect to well-established (high-degree) nodes, rather than uniformed attachment. The Barabási-Albert (BA) model is thus developed as follows. Initially, consider a fully-connected network with a small number m_0 of nodes, i.e. every node is connected to every other node. At each of the following step $t = 1,2,3\ldots$, one new node is introduced along with m links to be connected to existing nodes. For each new-coming link, its probability to be connected to node i with degree k_i, is determined by

$$\Pi_i = \frac{k_i}{\sum_{j=1}^{m_0+t-1} k_j}, \tag{2.9}$$

i.e., the probability is proportional to the node's degree. Following such model, after sufficient steps, the degree distribution is

$$P(k) = 2m^2k^{-3}. \tag{2.10}$$

Accordingly, the BA networks have power-law degree distributions in the form $c \cdot k^{-\gamma}$ with an exponent $\gamma = 3$.

Such mechanisms have been validated in empirical networks (Pastor-Satorras et al. 2001; Jeong et al. 2003), and the BA network thus has been the most fundamental model in network studies.

Geometric popularity-similarity model

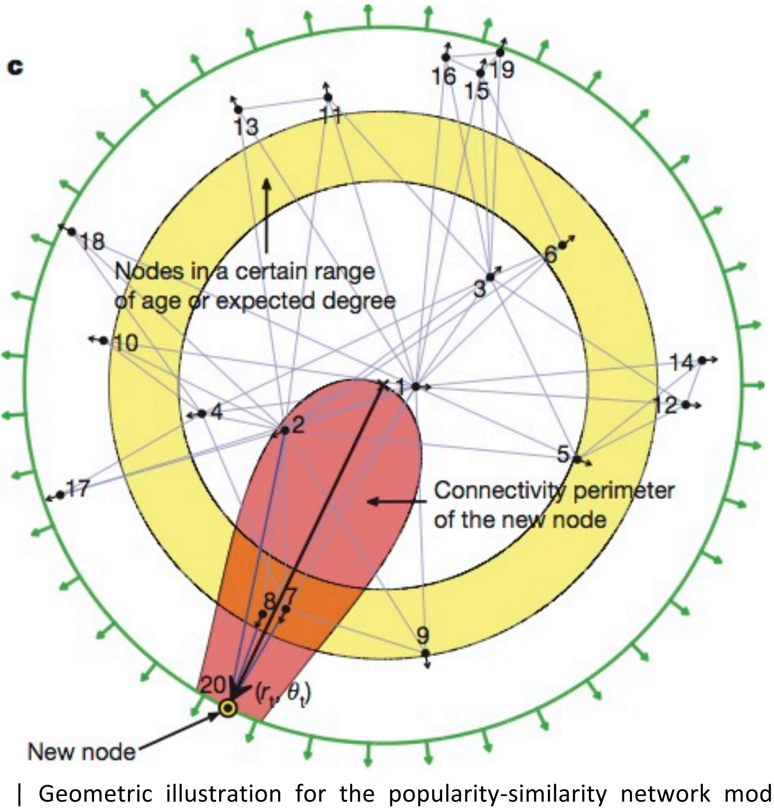


Figure 2.4 | Geometric illustration for the popularity-similarity network model. (From [Papadopoulos, F., Kitsak, M., Serrano, M. Á., Boguná, M., and Krioukov, D. (2012). Popularity versus similarity in growing networks. Nature 489(7417), 537]. Reprinted with permission from Springer Nature.)

Despite the success of the BA model in describing the power-law degree distribution, the clustering feature of the resulted networks is trivial. In line with this, Papadopoulos *et al.* (2012) argue that the popularity (the degree of nodes) is only one demission for the preferential attachment, and the trade-offs between popularity and similarity should be considered. In practical systems, nodes tend to connect to not only popular others, but also similar others. For example, in Twitter, a new user may connect to celebrities who have millions of followers, and at the same time, also his/her friends who may be not so popular.

Based on the geometric networks (Krioukov *et al.* 2010; Boguná *et al.* 2010), the popularity-similarity model is developed incorporating the preference of new links for both popularity (degree) and similarity (Papadopoulos *et al.* 2012). In the model, the nodes are assumed to be joining the network one by one and thus each node can be assigned with a birth time $t = 1, 2, 3, \dots$ Accordingly, the popularity is approximately regarded as the birth time, i.e. smaller index t represents higher popularity. For the similarity, a circle is assumed where each node t is assigned to a random position, described with an angular value θ_t. By doing so, the similarity between two nodes x and y can thus be represented as their angular distance θ_{xy}. With the popularity and similarity being modelled, the network can be generated as follows:

1) the network is initially empty.

2) at each step $t \geq 1$, a new node t comes into the network at a random angular position θ_t. The index (birth time) t can be mapped as the node's radial coordinate as $r_t = \ln(t)$. Thus, the new coming node locates at a position in a hyperbolic plane which can be represented by the polar coordinates (r_t, θ_t), as shown in Figure 2.4.

3) the new node connects to m existing nodes $s < t$ that are closest to t, in terms of the hyperbolic distance (Bonahon 2009). The hyperbolic distance between two nodes x at (r_x, θ_x) and y at (r_y, θ_y) is given by $d_{xy} = r_x + r_y + \ln(\theta_{xy}/2) = \ln(xy\,\theta_{xy}/2)$. Accordingly, the m new coming links will connects to nodes s minimising such distance d_{st}.

Following such procedure to reach a desired population, gives the network, where the average degree is approximately $\langle k \rangle = 2m$. Though totally different mechanisms are applied in comparison to the BA model, the popularity-similarity network can produce the similar power-law degree distribution with the exponent $\gamma = 2$. Most importantly, while BA networks fail to describe the clustering feature of real networks, the popularity-similarity networks have strongest possible clustering for the given degree distribution.

In the presented model, the degree distribution and clustering coefficient seems to be fixed as the popularity-similarity optimisation is exactly defined. Modifications can be made to the original model to achieve, as proved by Papadopoulos *et al.* (2012), power-law distribution with arbitrary exponent, and arbitrary clustering coefficient, such as adding the fading effect for the popularity, and considering the connections a stochastic process involving the hyperbolic distance.

2.1.3. Random walks on networks

It has been discussed that the complex network is an efficient model for many real-world systems and studying the structure of the networks gives extensive implications for the understanding, design, prediction and control of such systems. When it comes to the dynamics of such systems/networks, the random walk model is one of most powerful methods to be applied (Masuda *et al.* 2017).

A random walk on a network is basically to assume an agent who starts from a node and moves to other nodes following links randomly. Such simple process showed great power in a wide range of network studies, such as community structure detection (Rosvall & Bergstrom 2008; Fortunato & Hric 2016), node importance ranking (Newman 2005; Lv *et al.* 2016), opinion/epidemic spreading (Durrett 2010; Cataldi *et al.* 2010; Pastor-Satorras *et al.* 2015) and the user surfing behaviour on the Internet (Huberman *et al.* 1998; Nguyen *et al.* 2015) etc.

In general, some key metrics are of the most interest for the study involving random walks on networks, including the Mean First-Passage Time, Return Time and Navigability / Cover Time and so on.

Consider a network with a population of N nodes, and the links can be represented by the adjacency matrix $\boldsymbol{A} = \{a_{ij}\}_{N\times N}$. In a range of discrete time, a single walker randomly moves one step (from current node to one of its neighbours) at each time step. If starting at $t = 0$ from node i, a master equation can be laid out for the probability of the random walker being at an arbitrary node j at a given time t, as (Noh & Rieger 2004),

$$P_{ij}(t+1) = \sum_{v=1}^{N} \frac{a_{vj}}{k_v} P_{iv}(t), \qquad (2.11)$$

where k_v is the degree of the node v. The first-passage time from node i to j, denoting with τ_{ij}, basically answers the question that how long does it take for a random walker reach for the first time the target node j from the source i. When the target node is the same to the source node, i.e. $i = j$, such quantity τ_{ii} is also known as the return time, which represents the time that a random walker returns to the node he/she started for the first time. Similar to the transition probability shown in Eq. (2.11), the first-passage time / return time can also be described by a master equation which is written as (Masuda *et al.* 2017),

$$\tau_{ij} = 1 + \frac{1}{k_i} \sum_{v=1}^{N} a_{iv} m_{vj}. \qquad (2.12)$$

Of course, the above Eq. (2.11) and Eq. (2.12) may give different solutions for the mean first-passage time, depending on the detailed structure of the studied network. The mathematical analysis and computational studies on the mean first-passage time in different types of network models as well as empirical networks, has been a focus in the filed (Noh & Rieger 2004; Condamin *et al.* 2007; Hwang *et al.* 2012; Iacopini *et al.* 2018).

Random walks on networks can also help to reveal another crucial quantify for the understanding of the networked systems, namely the navigability – to what extend is the network navigable (Boguná *et al.* 2009; De Domenico *et al.* 2014). In brief, the navigability of a network is defined as the fraction of distinct nodes that can be visited by the random walk in a finite t steps. Normally, networks are with complex structures such as dense communities and highly clustered neighbourhood. As a result, a random walker will not necessarily find a new node in every step of his/her walk. Theoretically, a t-step random walk will at most visit $n(t) = t$ distinct nodes, and such maximum value is only likely to appear in tree-like networks. If for a large step t, only a limited number of distinct node $n(t)$ could be visited, the network can then be regarded unnavigable. Accordingly, the quantity $n(t)/N$ can be used to describe the navigability of the network. The time that yields $n(t) = N$, i.e. every of the possible nodes is visited for at least once, is known as the cover time (Chupeau *et al*. 2015; Maier & Brockmann 2017). For a connected network, where there is at least a path connecting every pair of nodes, the cover time will be a finite value, but normally a relatively large one.

To fit the purpose of investigating different networks in different scenarios, there are many variant random walk models (Lin and Zhang 2014; Bonaventura *et al.* 2014; Mondragon 2017). For example, a rather popular such model is the biased random walks (Fronczak & Fronczak 2009; Sinatra *et al.* 2011), in which the probability for a walker to move from current node to next is weighted by, for example, the degree of the target node. In other words, the walker is more likely to follow the link which connects to high-degree node. Another interesting model is the self-avoiding random walk (Herrero 2005; Kim *et al.* 2016), in which the random walker does not revisit any node, i.e. only keep visiting new nodes.

2.1.4. Bipartite modelling

Above we have briefly introduced the complex networks consisting of only one kind of nodes, i.e. all the nodes are of the same kind. For example, in a typical social network, there are only human beings as nodes. Such kind of networks can be referred as unipartite networks.

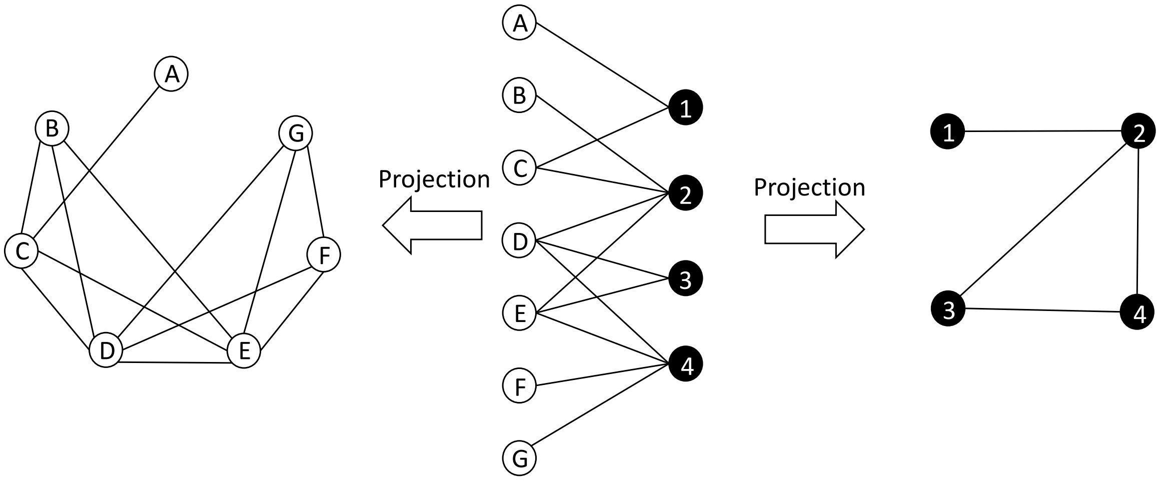


Figure 2.5 | A toy bipartite network with two kinds of nodes, namely the solid nodes and the empty nodes. The networks on the left and right are the projections of the bipartite network by connecting every pair of nodes sharing common neighbours.

However, there are, in the real world, sometimes that we need to model systems with two kinds of nodes. Accordingly, the bipartite network model can be applied (Newman et al. 2002), which conceptually appears as shown in Figure 2.5. Theoretically, links only exist between nodes of different kinds. Mathematically, a bipartite network is thus normally described as two sets of nodes $V = \{v_1, v_2, \cdots, v_N\}$ and $V' = \{v'_1, v'_2, \cdots, v'_M\}$ and the links are thus represented by the adjacency matrix $\boldsymbol{A} = \{a_{ij}\}_{N\times M}$, where $a_{ij} = 1$ indicates a link between nodes v_i and v'_j and $a_{ij} = 0$ otherwise.

The bipartite network model has found wide applications. For example, the collaborations of organisation on projects can be modelled as such network (Barber *et al.* 2006; Roediger-Schluga & Barber 2008). Organisations then compose one kind of nodes and the projects are the other kind. While one projects may have multiple participated organisations, every organisation would participate in multiple projects as well. Such participations can thus be modelled as the links. Another example is the scientific publication network (Guimera *et al.* 2005; Luong *et al.* 2015), where the authors and the publications are two sets of nodes respectively.

In particular, bipartite networks offer great potential in analysing user behaviour where users interact with entities such as music (Lambiotte & Ausloos 2005; Pongnumkul & Motohashi 2018), movies (Xu *et al.* 2016) etc. For generalisation, such networks are normally referred as the user-object bipartite networks, or user-item bipartite networks. In the present thesis, the term 'object' will be majorly used to represent all kinds of entities ranging from products, books, music, movie and so on. Since the online recommendations majorly deal with the interaction between users and objects, the user-object bipartite network model naturally provides an efficient theoretical representation for such systems. The user-object bipartite networks also exhibit strong structural regularities, similar to the unipartite networks. In such network, a user's degree k_u is the number of objects that s/he accessed (purchased, commented, browsed etc.), and thus can represents his/her activeness. On the other hand, an object's degree k_o is the number of users that accessed it, representing its popularity. Both the users' and objects' degree are found normally following the power-law distribution (Shang et al. 2010). Such distribution has huge implication for the study of the user-object systems. It is implied that most users are inactive, with only few interactions to objects, and as well, most objects are not popular at all. On the other hand, there always exist few hubs, i.e. users that are enormously active, and objects that are blockbusters. The heterogeneity in the system posts challenging problems especially for the recommender system. According to the square-based clustering coefficient (Lind & Herrmann 2007), the user-object bipartite networks are shown with strong clustering feature which can be used to describe individual user's interest diversity (Liu *et al.* 2013a).

Efforts have also been devoted to developing artificial bipartite networks reproduce the commonly-existed regularities, such as the catalogue model (Beguerisse Díaz *et al.* 2010) and the global-local preferential model (Pan *et al.* 2017).

There is a technique that can transfer bipartite networks to unipartite networks, which is known as the bipartite projection (Zhou *et al.* 2007; Zweig & Kaufmann 2011). Such technique is necessary due to the common needs of many systems. For example, while authors collaborating in articles can be represented by author-article bipartite networks, one may still be interested in the collaboration networks among authors (Newman 2001a; Abbasi *et al.* 2012). One simple method for the conversion is to establish links between nodes that share at least one common neighbour (Zhou *et al.* 2007). Normally, in networks, these nodes that connected with the target node are considered as its neighbours. The common neighbour of two nodes are thus the nodes that are connected to both of them. As shown in Figure 2.5, the bipartite network can be projected on basis of either kind of nodes. Taking the solid node {1,2,3,4} as an example, the node 1 shares one common neighbour with node 2, i.e. the empty node C, and thus a link should connect 1 and 2 in the projected network. On the other hand, node 1 does not have any common neighbour with the nodes 3 and 4, and thus no links existed between node 1 and nodes 3 and 4. Following such method, the projected unipartite network would result, in which the structure and interaction of one kind of nodes can be analysed. The bipartite projection has been widely applied (Buldú *et al.* 2007; Ribeiro *et al.* 2013; Del Vicario *et al.* 2017). With different considerations, there exist a number of other projection methods (Zweig & Kaufmann 2011; Wu *et al.* 2014; Banerjee *et al.* 2017) such as considering the temporal pattern and assign weights according to the co-occurrence of nodes.

2.2. Network-Based Similarity Measures

In networks, the most fundamental element is the links which describe the relation among nodes. However, networks are mostly sparse, that most pairs of nodes do not have actual connections. Two nodes having no direct connection does not necessarily mean that they have no relation. For example, two persons that don't know each other may still have common friends which can act as an indicator for their potential relation. Such relation can be generally described as 'similarity'. The similarity between anything of course can be considered relating to their internal feature, such as for two persons, their social status, background or education level etc. In terms of the networks, the similarity between two nodes can be measured by their common neighbours, which is normally referred as structural similarity. The underlying philosophy is basically the same to the widely-known association rules (Mobasher *et al.* 2001; Sandvig *et al.* 2007).

Therefore, the study of node similarity in networks tries to answer a simple question that, given the structure of a network, how to quantify the similarity between any pair of nodes. The most direct and widely-used such method is known as the Common Neighbour (CN) measure which regards the number of common neighbours as the similarity between two nodes. For the nodes x and y, the CN similarity is mathematically defined as

$$s_{xy}^{CN} = \left|\Gamma_x \cap \Gamma_y\right|, \tag{2.13}$$

where Γ_i is the set of the neighbours of node i, and $|.|$ gives the number of element of a set. Accordingly, the raw number of common neighbours is the similarity between two nodes. For example, in a social network, two people sharing 10 common friends would have a similarity of 10, and can be considered similar with large likelihood to establish connections in the future. However, imagine that these two people both have a very large number of friends, i.e. very large degrees, then the similarity of 10 seems to become less significant. In comparison, if another pair of two persons both have only 10 friends and all of these are their common friends, their similarity actually should be much higher despite the fact that CN still considers it to be 10. In brief, the CN similarity tend to overestimate the similarity between large-degree nodes, because they generally share more common neighbours with others even by chance. As a result, it has been argued that the CN measure can only well describe those strongly assortative networks where high-degree nodes tends to connect with high-degree nodes (Newman 2001b; Clauset *et al.* 2008).

Inspired by the CN measure, and its correlation with the degrees of the measured nodes, some variant measures have been trying to normalise the number of common neighbours. Three famous and fundamental normalisers are considered by the Salton (SAL) measure, Jaccard (JAC) measure and the Sorensen (SOR) measure, which are $\sqrt{k_x k_y}$, $(k_x + k_y - s_{xy}^{CN})$ and $(k_x + k_y)/2$ respectively. A Hub-Promoted (HP) measure (Ravasz *et al.* 2002) was developed by normalising the raw number of common neighbours by the smaller degree between the measured nodes, i.e. the normaliser $\min(k_x, k_y)$. Similarly, the Hub-Depressed (HD) measure uses the larger value for the normalisation i.e. $\max(k_x, k_y)$. The mathematical definitions of these measures can be simply achieved by dividing the CN similarity with the normaliser respectively, which read

$$s_{xy}^{SAL} = \frac{|\Gamma_x \cap \Gamma_y|}{\sqrt{k_x k_y}}, \tag{2.14}$$

$$s_{xy}^{SOR} = \frac{2|\Gamma_x \cap \Gamma_y|}{k_x + k_y}, \tag{2.15}$$

$$s_{xy}^{JAC} = \frac{|\Gamma_x \cap \Gamma_y|}{k_x + k_y - |\Gamma_x \cap \Gamma_y|}, \tag{2.16}$$

$$s_{xy}^{HP} = \frac{|\Gamma_x \cap \Gamma_y|}{\min(k_x, k_y)}, \tag{2.17}$$

$$s_{xy}^{HD} = \frac{|\Gamma_x \cap \Gamma_y|}{\max(k_x, k_y)}, \tag{2.18}$$

respectively.

By considering the number of paths connecting two measured nodes, Leicht *et al.* (2006) developed a global-based LHN measure which defines the similarity as

$$s_{xy}^{LHN-global} = \frac{2m\lambda}{k_x k_y}\left[\left(\boldsymbol{I} - \frac{\alpha}{\lambda}\boldsymbol{A}\right)^{-1}\right]_{xy}, \tag{2.19}$$

where m is the number of links in the network, λ is the largest eigenvalue of the adjacency matrix $\boldsymbol{A}$, and $\alpha < 1$ is a free parameter. This similarity measure considers not only the very local structure, i.e. the direct neighbours, but also the higher-order neighbours. There is also a local version of this measure which considers only the paths with length 2, which reads (Leicht *et al.* 2006)

$$s_{xy}^{LHN} = \frac{|\Gamma_x \cap \Gamma_y|}{k_x k_y}. \tag{2.20}$$

Besides the degree of the measured nodes, the degree of the intermediate nodes, i.e. common neighbours, is also affecting the similarity between two nodes. Imagine that two users commonly favouring the same popular music should be less similar than the case that two users commonly favouring the same niche music. Taking the degree of the intermediate nodes as weights to the CN measure, the Adamic-Adar (AA) measures (Adamic & Adar 2003) defines the similarity as

$$s_{xy}^{AA} = \sum_{v \in \Gamma_x \cap \Gamma_y} \frac{1}{\ln{(k_v)}}. \qquad (2.21)$$

Assuming that these intermediate nodes act as the transmitter that spreads the resources taking from node x evenly to its neighbours, the Resource Allocation (RA) measure (Zhou *et al.* 2009) regards the amount of resources that node y receives as the similarity, which writes as

$$s_{xy}^{RA} = \sum_{v \in \Gamma_x \cap \Gamma_y} \frac{1}{k_v}. \qquad (2.22)$$

So far, all of the above similarity measures were proposed for the unipartite networks. However, according to the definition of these measures, they can be naturally applied to bipartite networks without future modification.

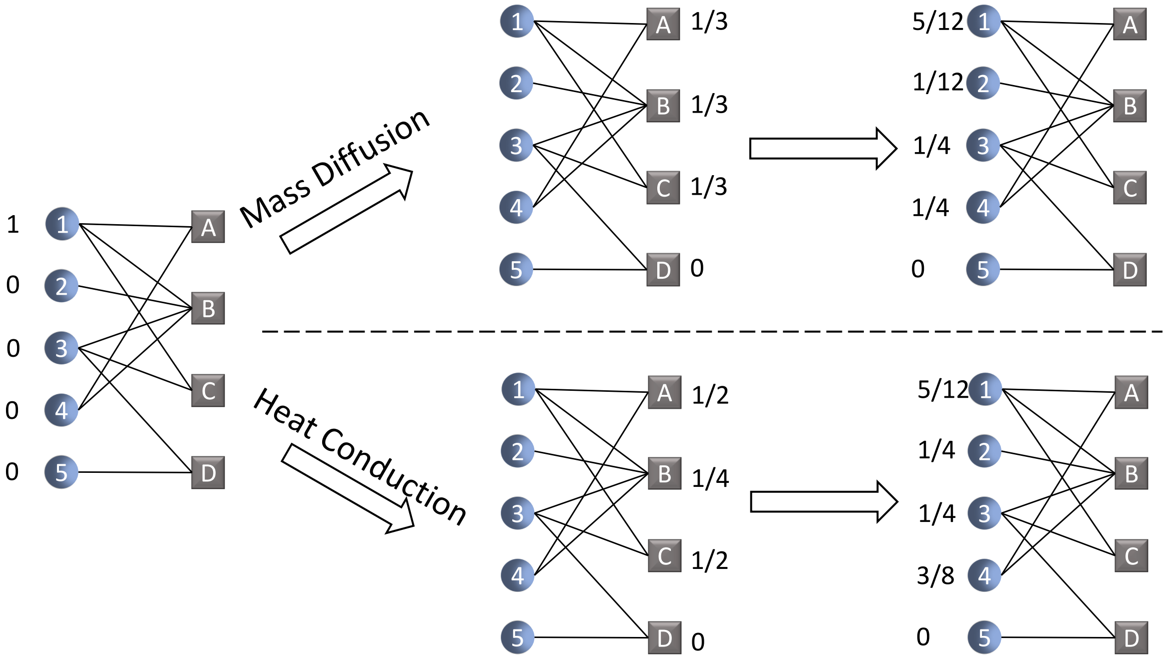


Figure 2.6 | Illustration for the Mass Diffusion (upper path) process and Heat Conduction (bottom path) process.

Particularly based on the bipartite network, two widely-discussed similarity measures, namely the Mass Diffusion (MD) measure (Zhou *et al.* 2007) and Heat Conduction (HC) measure (Zhang et al. 2007), have been developed considering the spreading process in the bipartite network. Both measures assume that a source node has a unit (1) of resource, and all the other nodes have nothing (0) initially as shown in Figure 2.6. They believe that the similarity between two nodes of the same kind (at the same side of the bipartite network) can be measured by the amount of resource spreads from the source node to the target node. The mechanisms of such spreading process are different for the two measures, leading to total different definition and resulted similarities.

The MD measure assumes that the resource should spread similarly to the diffusion process of mass, where each node evenly distributes its resource to its neighbours. In the example shown in

Figure 2.6, node 1 has a unit of resource initially. The first step of the spreading is from the circle nodes to square nodes, where each of node 1's neighbours receives 1/3 resources. In the second step, the resource spreads from square nodes back to circle nodes, where each of the nodes A, B and C, distributes the resource evenly back to circle nodes according to their degree. For example, node A gives node 1 and node 4 each with 1/6 resource; node B gives node 1, 2, 3, and 4 each with 1/12 resource. As a result, node 4 then has resource 1/6 + 1/12 = 1/4. Accordingly, MD measure considers the similarity from node 1 (source node) to node 4 (target node) as 1/4. In a general process, the MD similarity between a source node x and a target node y can be mathematically defined as

$$s_{xy}^{MD} = \frac{1}{k_x} \sum_{v \in \Gamma_x \cap \Gamma_y} \frac{1}{k_v}. \tag{2.23}$$

On the other hand, HC measure considers such process as the conduction of heat, where each node receives heat that should be the average of its neighbours'. For example, in Figure 2.6, node A has two neighbours namely node 1, 2 with heat of 1 and 0 respectively. Thus, in the first step, node A's heat is the average of node 1 and 2, resulting 1/2. Similarly, node B has the average heat among node 1, 2, 3, 4, which is 1/4. In the second step, where the heat conducts back to circle nodes, the same mechanism applies. The node 4 gets the average heat between its neighbours, node A and B, which is $(1/2 + 1/4)/2 = 3/8$. Mathematically, the HC similarity between a source node x and a target node y writes

$$s_{xy}^{HC} = \frac{1}{k_y} \sum_{v \in \Gamma_x \cap \Gamma_y} \frac{1}{k_v}. \tag{2.24}$$

According to the mathematical definition shown in Eq. (2.23) and (2.24), one may find MD and HC very similar to each other. Despite the spreading process, both MD and HC measure takes the form of the RA similarity, which is the weighted summation over common neighbours, but dividing with different degrees. MD divides RA with the degree of the source node, while HC divides RA with the degree of the target node. As a result, one has the relation $s_{xy}^{MD} = s_{yx}^{HC}$. However, due to the different dividers, the MD and HC measures have vastly different performances in the link prediction and personalised recommendations. Additionally, while all the other measures are mathematically symmetrical, i.e. one has $s_{xy} = s_{yx}$, the HC and MD similarities are not symmetrical, $s_{xy}^{MD} \neq s_{yx}^{MD}$ and $s_{xy}^{HC} \neq s_{yx}^{HC}$.

2.3. Personalised Recommendation

2.3.1. Recommendation techniques

Given the history records of a group of users interacting with objects, such as books, movies, products, what should be recommended to a target user which have the highest possibility fitting his/her interests? This is the basic task that personalised recommendation aims to accomplish (Park *et al.* 2012; Lu *et al.* 2015). To this end, many techniques have been developed. Given the advances in the artificial intelligence, the neural networks have been applied to develop recommender systems (Christakou *et al.* 2007; Covington *et al.* 2016; Katzman *et al.* 2018). The genetic algorithms have also found applications in making recommendations (Bobadilla *et al.* 2011; Hassan & Hamada *et al.* 2017). Embedding the social network analytics, recommendations can be made according to the trust of users (Massa & Avesani 2009; Wang *et al.* 2015), or social tags (Konstas *et al.* 2009; Zhang *et al.* 2011). Taking advantages of different techniques, hybrid methods have also been developed (Zhou et al. 2010; Tarus et al. 2017). Besides, the matrix factorisation is also found capable to generate accurate recommendations (Koren *et al.* 2009; Hernando *et al.* 2016; Xu 2018).

Among various techniques, two major methods, namely the content-based technique and the collaborative filtering, are most widely-applied.

Content-based technique

Among the techniques, the content-based method (Pazzani & Billsus 2007; Aggarwal 2016) is the most straightforward one. Basically, such system defines objects in terms of attributes which can be possibly attracted from the object description, category or user-generated tags etc. The attributes of objects are vastly different for different types of systems. For example, in a movie recommender, the attributes could be movie type, director, production company or actors. The system analyses the common attributes of a target user's historical selections, and accordingly recommends objects that have the same attributes to this user. If a user has watched several Sci-Fi movies, then more Sci-Fi movies would be recommended; if a user bought a book written by a particular author, more books by this author would be recommended. Such philosophy is simple, but the determination of the attributes is rather complicated. First of all, attributes are not always available, or there may be not enough dimensions of attribute to accurately define the objects. Secondly, every system would largely depend on in-depth investigation to determine appropriate combinations of attributes due to the lack of interoperability among different systems. Thirdly, such mechanism may produce redundant recommendations, such as a user bought a computer would not need another computer anymore.

Collaborative filtering

The collaborative filtering (Goldberg *et al.* 1992; Resnick *et al.* 1994; Nilashi *et al.* 2014) is a technique that makes recommendations based on the wisdom of the collective, which is also known as neighbourhood method or memory-based method (Breese *et al.* 1998; Ning *et al.* 2015). It examines the co-accessing patterns between objects and users, and defines the similarity according to how frequently two objects, or two users, have the same connections. The similarity measures introduced earlier in section 2.2 are basically the collaborative-based similarities, all of which can be applied in such recommender systems.

The collaborative filtering can be further categorised as user-based system (Zhao & Shang 2010) and object-based system (Sarwar et al. 2001). The user-based system firstly evaluates the similarity among users, and recommend a target user what his/her friends (similar users) likes. On the other hand, an object-based recommender system evaluates the object similarity and recommends a target user with objects that are similar to his/her historical selections. In practice, the object-based collaborative filtering has the widest applications, such as the Amazon (Schafer *et al.* 2001; Linden *et al.* 2003) and YouTube (Davidson *et al.* 2010). Accordingly, the present thesis will majorly focus on the object-based systems. Actually, the algorithms are basically applicable directly to the user-based systems with simple change of position for object nodes and user nodes.

The object-based collaborative filtering normally targets at an individual user, say u, to recommend objects according to his/her historical records. With the similarity among objects being calculated according to an arbitrary measure introduced in section 2.2, the system calculates a score for every object α for the target user u, as

$$w_{u\alpha} = \sum_{o \in \Gamma_u} s_{o\alpha}, \tag{2.25}$$

where Γ_u is the set of objects that user u has historically selected and $s_{o\alpha}$ is the similarity between the object o and α. Such score $w_{u\alpha}$ describes the likelihood that the object α being selected by user u in the future. Normally, a length of recommendation list should be determined as L, which largely varies for different systems. Thus, for the target user u, L objects that have the highest score would be recommended.

The recommendation technique of the collaborative filtering is easy, but the performance of the recommendations largely relies on the measure of quantifying the similarity. There are apparent advantages with this technique. Firstly, it needs only the structure of the user-object interactions as input. As a result, it does not have to handle the complexity caused by the consideration of internal attributes of objects and user profiles. Secondly, such method is applicable directly to a wide range of systems with different types of objects, because it does not require a unified attributes definition. Thirdly, due to the application of association rules (structure similarities), it is possible to uncover the hidden patterns of consumer behaviour, and thus make diverse and accurate recommendations.

2.3.2. Evaluation metrics

In order to test the recommendation algorithms, normally one may divide the user-object interaction data into two sets, *i.e.* a training set and a testing set according to a certain ratio. In the experiments, one uses the training set as the historical data to generate recommendation list with length L for each user and then evaluate the performance by comparing the recommended list and the actual selections in the testing set. There are many aspects of the recommendation performances have been addressed, among which the accuracy and diversity got the most attentions.

Accuracy metrics

For a target user u, in the testing set, there are a list of objects that the user actually selected, denoting with Γ_u^{test}. These objects in Γ_u^{test} are regarded as the user's future selections. On the other hand, the recommender system will calculate the score of each object for the target user and rank all the objects accordingly. The accuracy metrics are basically to examine whether the objects in Γ_u^{test} are ranked at the top of the list.

Since the objects ranked at the top-L positions will be recommended to the target user, displaying on the webpage, it is apparent that the most fundamental metrics should check the match between these L objects and the Γ_u^{test}. If an object that recommended to the user (ranked in the top-L position) also appears in the testing set Γ_u, this object can be called a hit. The number of hits for the target user u, denoting with $h_u(L)$, can thus represent the accuracy of the recommendations to the user. Depending on the ways to normalise the number of hits, two metrics namely Precision and Recall are normally used, which are defined as $p_u(L) = h_u(L)/L$ and $r_u(L) = h_u(L)/|\Gamma_u^{test}|$ respectively. One may notice that, the precision describes how many of the L recommendations are accurate, while the recall is the ratio of objects in Γ_u^{test} being retrieved. Averaging over all the users, the precision and recall can be used to describe the accuracy of the recommender system, which write

$$P(L) = \frac{1}{M}\sum_u \frac{h_u(L)}{L}, \tag{2.26}$$

and

$$R(L) = \frac{1}{M}\sum_u \frac{h_u(L)}{|\Gamma_u^{test}|}, \tag{2.27}$$

respectively, where M is the total number of users. The values of precision and recall locates in the range [0, 1], with lower-limit 0 representing totally inaccurate (no hits at all), and upper-limit 1 representing completely accurate.

According to the definition, one can find that the precision and recall both rely on the pre-defined length of the recommendation list, L. The ranking score RS is another widely-applied accuracy metric which is independent from the recommendation list length. The principle of ranking score is that the higher ranking for the objects in testing set represents better accuracy. If we denote the rank of an object o in user u's list with r_{uo}, the ranking score RS reads,

$$RS = \frac{1}{|\Gamma^{test}|}\sum_u \sum_{o\in\Gamma_u^{test}} \frac{r_{uo}}{N - |\Gamma_u^{train}|}, \tag{2.28}$$

where Γ^{test} is the full testing set, and $|\Gamma^{test}|$ is thus the number of records in the testing set regardless of the user, Γ_u^{train} is the user u's selections in the training set, and N is the total number of objects. Consequently, the accuracy *RS* ranges in (0,1) and the larger value of *RS* represents less accurate recommendation and the smaller value of *RS* stands for more accurate recommendation.

Diversity metrics

In practice, to ensure the accuracy of the recommendation, a system can simply recommend the most popular objects to every user. It has been shown that recommending according to the global ranking of the object popularity may lead to reasonable accuracy (Zhou *et al.* 2010), because after all, popular objects fit most users' interests. However, recommending only popular objects is of less meaning, since these popular objects such as Oscar movies or famous-branded products are well known by the users already. The users thus may expect more personalized recommendations. Accordingly, many studies have been arguing that the recommendations should be diverse and novel (McNee *et al.* 2006; Vargas & Castells 2011).

To address the ability of a recommender to recommend unpopular objects, the metric Novelty $Nov(L)$ is widely used, which is defined as the average popularity (degree) of all recommended objects, i.e. (Lv et al. 2012),

$$Nov(L) = \frac{1}{M \cdot L} \sum_{u} \sum_{o \in \Omega_u} k_o, \tag{2.29}$$

where Ω_u is the set of L objects being recommended to user u, and k_o is the popularity of an object o, i.e. how many users have selected this object in the training set. Accordingly, the smaller value of novelty would suggest better ability of the algorithm in recommending unpopular objects.

The recommender is also expected to be able to generate personalised recommendations, which requires the recommendation lists of different users to be as different as possible. For any two users i and j, assuming there are $Q_{ij}(L)$ same objects in their recommendation lists of length L, one can use the Hamming distance to calculate the degree to which two lists are different, as $H_{ij}(L) = 1 - Q_{ij}(L)/L$. Averaging over all pairs of users, the metric Personalisation $S(L)$ is defined as

$$S(L) = \langle H_{ij}(L) \rangle = \frac{1}{M \cdot (M-1)} \sum_{i,j \in \boldsymbol{U}, i \neq j} (1 - \frac{Q_{ij}(L)}{L}), \tag{2.30}$$

where $\boldsymbol{U}$ is the full set of users. Hence, the upper limit $S(L) = 1$ represents the totally diverse condition, under which every user's list is completely different from others', and the lower-limit $S(L) = 0$ means all the recommendation lists are identical.

2.4. Recommendation Networks

In many online systems, recommendations are not only made for target users, but also for objects. It is a common scenario that on the webpage of an object, there would be a list of recommendations consisting of other objects with hyperlinks that the system considers 'similar' to the current one. Such similar object recommendations can be widely found in online systems, ranging from retail website Amazon's "Customers who bought this item also bought" list, to scientific article database ScienceDirect as shown in Figure 2.7. While the objects in online systems are displayed in dedicated webpages, there are hyperlinks connecting the source object to the recommended objects in the similar-object list. Accordingly, one can imagine the massive volume of objects in a content-browsing system connected to each other via such recommendation hyperlinks, as a directed network as shown in Figure 2.8, which is normally refereed as the recommendation network, or product network. In such network, the node is the webpage of an object, and the link is the hyperlink representing the recommendation determined by the similarity between two objects. In particular, given the mechanism of constructing such network, the out-degree of each node is fixed, which is the length of the recommendation list.

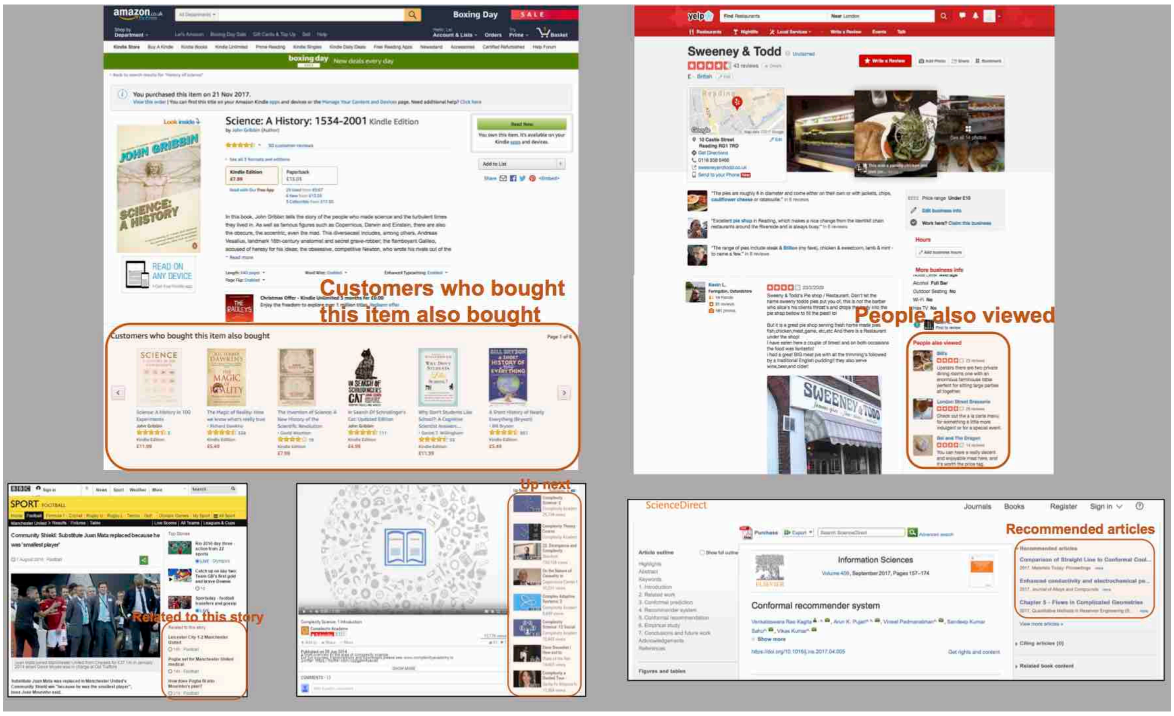


Figure 2.7 | Screenshots from amazon.com, yelp.com, bbc.co.uk/news, youtube.com and sciencedirect.com respectively to illustrate the common existence of similar-object recommendation in real online systems.

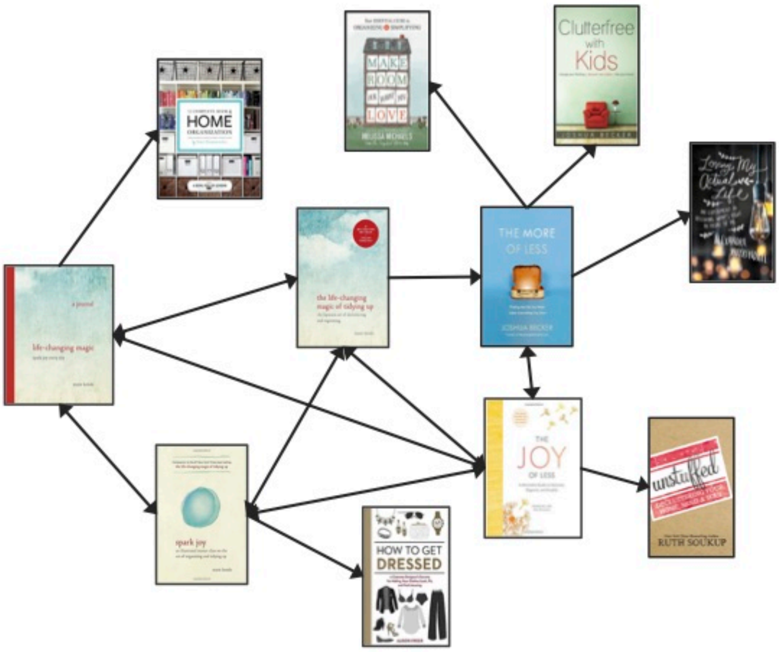


Figure 2.8 | An example of recommendation networks from amazon. Every book, as a node, is the webpage of the corresponding book, and the directed links are recommendation hyperlinks.

The recommendation networks are in nature different from the personalised recommendation which is introduced in section 2.3. Theorising the problem as link prediction in user-object bipartite networks, the personalised recommendation considers the historical behaviour of a target user and recommends objects specifically for this user. The recommendations are thus personalised, i.e. different for different users. In contrast, a recommendation network is emerged through recommendations for objects, i.e. similar objects. Accordingly, the recommendations for an object, are not made for any particular user, but for all the users browsing this object. Both kinds of recommendations may share the same technique of quantifying similarities. Though the personalised recommendations can be user-based, most such systems are based on object similarities. As a consequence, the first, also the most important step for both recommendations is to evaluate the object similarity $\boldsymbol{S} = \{s_{ij}\}_{N\times N}$. For similar object recommendation, i.e. the recommendation networks, the system directly connects the most similar objects with hyperlinks, and thus networks can be achieved. On the other hand, for personalised recommendations, the system still needs to consider a target user u's historical records Γ_u, which is a set of objects that historically been selected by the user. Accordingly, the system calculates the recommendation score of each object for the target user according to Eq. (2.25) and the objects with highest scores get recommended. In practical systems, both personalised recommendations and recommendation network are normally available. The personalised recommendation can be regarded as a mean of starting to access information, being an alternative to the search engine. While a user is accessing the objects, regardless of whether he/she started from clicking on recommendation or searching keywords, the recommendation network can be a good way to correct the initial bias, and leads the user to relevant objects.

In comparison to the personalised recommendation, the study on recommendation networks has a relatively short history. To the best of our knowledge, the earliest such study is on the music recommendation networks. Cano *et al.* (2006) collect the recommendation networks from four popular music websites, and analysed the basic structural topology of these networks. It is shown that, recommendation networks are similar to most other empirical networks, with short average shortest-path length and high clustering coefficient. Among the four music recommendation networks, there are two constructed by the collaborative efforts of users, i.e. according to the co-accessing pattern, and the other two are constructed by experts. They find that the collaborative-based recommendation networks have power-law in-degree distribution, while the expert-generated networks have exponential in-degree distribution. In addition to the empirical music recommendation networks, Buldú *et al.* (2007) project the playlists (can be regarded as bipartite networks between playlists and individual music) as a network of music, and explored the growing dynamic of the network. Besides, it is found that the music recommendation network is biased towards the popular artists (Celma & Cano 2008).

Besides the topology of the recommendation networks, significant amount of attentions have been paid to the interplay between recommendation networks and product performances, such as the demand and sales. Oestreicher-Singer and Sundararajan (2012a, 2012b) analyse the book recommendation network collected from Amazon, and find that the PageRank centrality, which is a measure for a node's position in a network, and some other quantities such as the in-degree, are closely associated with the books' demand measured by the sales rank of the book in Amazon. Leem and Chun (2014) further examine the other centrality measures, including degree centrality, closeness centrality, betweenness centrality, eigenvector centrality, and confirmed the argument of Oestreicher-Singer and Sundararajan (2012a, 2012b) that the demand of the books is vastly influenced by the position of the books in the recommendation networks. Lin *et al.* (2017) report that the network diversity and network stability have also significant influence over the product demands. Assuming the revenue of a product as the summation of its intrinsic value and incoming value, Oestreicher-Singer et al. (2013) develop a metric to estimate the network value of products, which consists the value generated by itself, and the value it contributes to other products. Together with the social network among users, product recommendation network composes the dual-network structure in content-browsing systems. Goldenberg *et al.* (2012) argue that such dual-network structure in the Youtube system largely facilitates the exploration for the contents.

The recommendation networks are also found able to spread the demand, and the word-of-mouth. While the external events, such as a book review in a TV show, may boost the sales of the corresponding product, Carmi *et al.* (2009, 2017) argue that such boosts may spread over the recommendation network. For a focal product that stimulated by an external event, its sales would increase significantly in the following days. Such growth of sales can also be observed in the focal product's, not only the direct, but up to 4th order neighbours. In other words, a product's sudden sale growth can spread as far as four clicks away on the recommendation networks. In addition, Lin and Wang (2018) show that products that are connected to each other in the recommendation networks tend to have similar ratings and product sales.

From the review of the related literature, one may notice that, the previous studies have been majorly focusing on the influence of the presence of recommendation networks on product performances. However, these studies are all based on the collected empirical networks which are constructed by the website according to unknown mechanisms. In other words, with the construction method uncontrolled, their findings may only limit to the particular system. Furthermore, all these studies only focused on products, leaving the influence on users accessing information still unknown. These limitations are in the major objectives of this thesis, which will be further discussed in the Chapter 6 and 7.

2.5. Summary

The present thesis aims to explore the recommender system including personalised recommendation and recommendation networks via studying the object similarity measures.

For the personalised recommendation study (Chapter 4 and 5), we adopt the Collaborative Filtering method as the foundation. Base on such system, we develop recommendation algorithms such as the Top-N-Stability algorithm (Chapter 4) and the Balanced Common Neighbour measure (Chapter 5). When evaluating the performance of the recommendation algorithms, we not only propose specific new metric, such as recommendation stability, but also adopt the standard metrics such as the precision, recall, diversity and novelty etc. As the validation for the proposed algorithm, we carry out comparative analysis to check whether the proposed algorithm can outperform benchmark algorithms, including these methods introduced in this Chapter such as CN, AA, RA etc.

For the recommendation network study (Chapter 6 and 7), we firstly theorise its construction according to the bipartite projection which is introduced in this Chapter. The similarity measures including the CN, SAL, SOR, HPI, LHN, AA, RA and HC, are then applied to construct recommendation networks based on empirical user-object interactions. We further adopt the random walk model to describe the users' surfing behaviour on these networks so that the systems' accuracy and accessibility can be evaluated.

Chapter 3. Data Collection and Scenarios

The present thesis is largely data-driven. Though there are theoretical studies consisting of only mathematical analysis based on random network models, most of these studies will come down to applications in empirical data.

As the thesis title suggested, the crucial data will be the networks relating to online recommendations, which are user-object bipartite networks. Accordingly, the nodes in typical such data are users and objects, which can be basically anything ranging from books, products, restaurants to movies etc. The meaning of links subjects to the detailed scenarios from which the data was collected. For example, in Amazon, where the objects are products, the links between users and objects then represent the purchase behaviour, i.e. a user bought a product. For systems such as Yelp, which is a review sharing website, the links will be commenting behaviour, such as a user commented a restaurant. In addition to the user-object interactions, empirical recommendation networks will also be collected and applied in the thesis in order to get fundamental understanding of the structure of such systems.

In this chapter, we introduce the data sets which will be used in the thesis. We develop a Python-based web crawler to collect the book recommendation networks from Amazon according to a width-first search, and the associated user-book interactions, which will be introduced in section 3.1. From open sources, we also collected several user-object bipartite networks which will be introduced in section 3.2.

3.1. Collection of Amazon Recommendation Network

To study the recommendation networks, we collect data from the “Customers who bought this item also bought” list in the Amazon, which is a retail website where users can buy products and leave comments. In the system, each product has a dedicated webpage displaying its basic information, user comments, and most importantly a list of similar other products as recommendations. In such recommendation lists, there are normally 100 similar products recommended, but displayed in pages. Hyperlinks are available for users to click on and surf to the corresponding recommended products. To collect the recommendation network is basically to collect such hyperlinks.

The empirical recommendation networks of Amazon have already been widely used in previous studies. However, most of these studies applied a depth-first searching strategy (Oestreicher-Singer & Sundararajan 2012a, 2012b; Carmi et al. 2017). Therefore, the collected network does not have a unified out-degree, i.e. products have different number of recommended others. In Amazon system, and also most other similar systems, an apparent feature of recommendation networks is that the recommendation list length is fixed. Accordingly, previous strategies did not capture such feature. In the present thesis, we therefore adopt a width-first searching strategy to collect the hyperlinks so that every product would have the same number of recommendations.

The Amazon data was collected in two steps, namely the recommendation network collection and the user-object bipartite network collection respectively, over the January of 2016.

Recommendation network collection

In the Amazon system, each book has a unique ID, and the webpage of the book is composed as http://www.amazon.com/dp/ID, where the 'ID' should be replaced by a real ID. Consequently, to collect the book network is actually to collect the corresponding webpages and the recommendation hyperlinks connecting them.

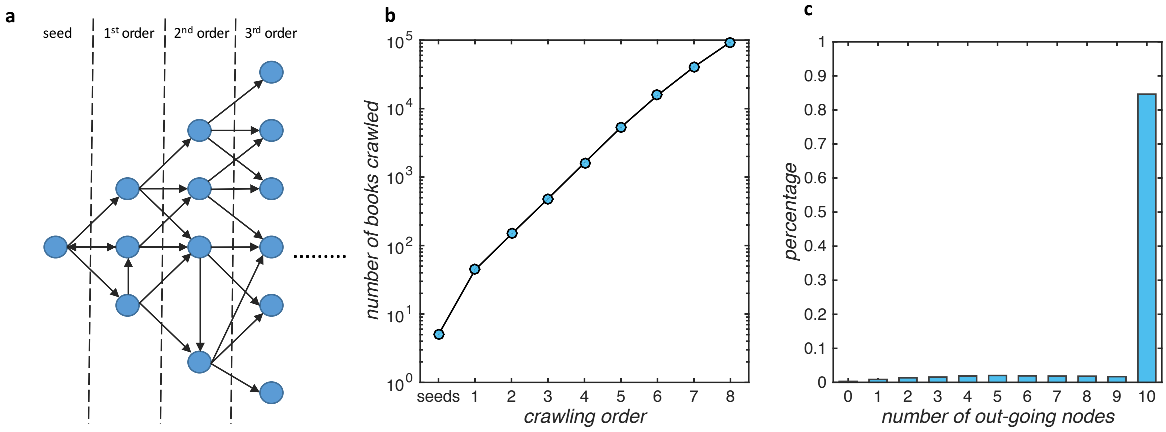


Figure 3.1 | Illustration for the collection of Amazon book recommendation network.

We firstly selected 5 books as the seeds of the crawling from the Amazon's bestseller list (www.amazon.com/gp/bestsellers/books). Note that, the list may change from time to time and in our collection, the seeds were collected on 1st January 2016. For each of the seed books, we collect books from its recommendation list known as '*Customers who bought this item also bought*' list. The books in the seeds' recommendation list are regarded as the 1st-order books (out-going nodes of the seeds) as shown in Figure 2.1a. While normally 100 similar objects are offered in each object's full recommendation list, there are generally 5 to 10 recommendations in the first page depending on the window size of the web browser. Assuming the recommendations displayed in the first page would get most attentions, we collected 10 books ranking at the top of the list as the current one's out-going nodes. As the crawling goes on and on, we then have the 2nd-order books, 3rd-order books and so on. The crawling continued for 8 steps. And for the 8th-order books, we collect the first 10 recommendations that have already been included in previous steps out from the list as their out-going books, so that there will be no 9th-order books. Figure 2.1b reports the number of books that are newly crawled at each order and in total of 157,856 books was collected. All of the collected books are reachable for the seeds within 8 steps. According to the crawling strategy, the out-degree of every node should be 10. However, some of the books may appear in others' recommendation list but somehow have no or just a few out-going books in the Amazon system. Additionally, those 8th-order books may have less than 10 out-going books that have been collected in the early steps. In summary, as shown in Figure 2.1c, 84.62% of all the crawled books have exactly 10 out-going books.

In this book recommendation network, the nodes are the Amazon webpages of the books, and the links are actual recommendation hyperlinks established by Amazon.

User-book bipartite network collection

In addition to the book recommendation network, we also collect all the reviews for each of the collected 157,856 books. Note that, in Amazon system, different versions of a same book such as Kindle edition, hardcover edition and paperback edition etc., share the same review webpage. Considering that version selection is also a reflection of the users' interest and the different versions of the same book have different recommendation lists, we only collected the comments devoted to the very specific version of the crawled book. In total of 4,520,102 reviews are collected (after cleaning the reviews to the other versions) which are posted by 2,540,369 users. Furthermore, the Amazon system marks the comments that posted by users who actually have bought this book as 'verified purchase'. Accordingly, every review between a user and a book represent a purchase behaviour, and our data is basically a sample of the full sale record of these books.

In this user-book bipartite network, the nodes are Amazon users and books, and the links are purchase relation between the corresponding users and books.

3.2. Open User-Object Interaction Datasets

Thanks to the recent studies on network science and ecommerce, many datasets of user-object interactions have been published, we retrieve five of the widely-investigated user-object bipartite networks, from online systems Yelp, Epinions, MovieLens, Netflix and Last.FM respectively.

The **Yelp** is a business review website where users can write comments on various businesses, mostly restaurants, but also including hotels, bars etc. Normally, users write comments to share their experience about the corresponding business. Being enthusiastic on scientific research, Yelp has published their data and been holding challenges for many years. The dataset used in this thesis was downloaded from Yelp challenge website www.yelp.co.uk/datase_challenge. While they constantly update the published dataset, the data we use was accessed in January 2016, which consists of 1,569,264 comments on 61,184 businesses posted by 366,715 users. On each business's webpage, there are basic information about the business, user comments, and most importantly a recommendation list entitled “People also viewed”, consisting $L = 3$ similar businesses.

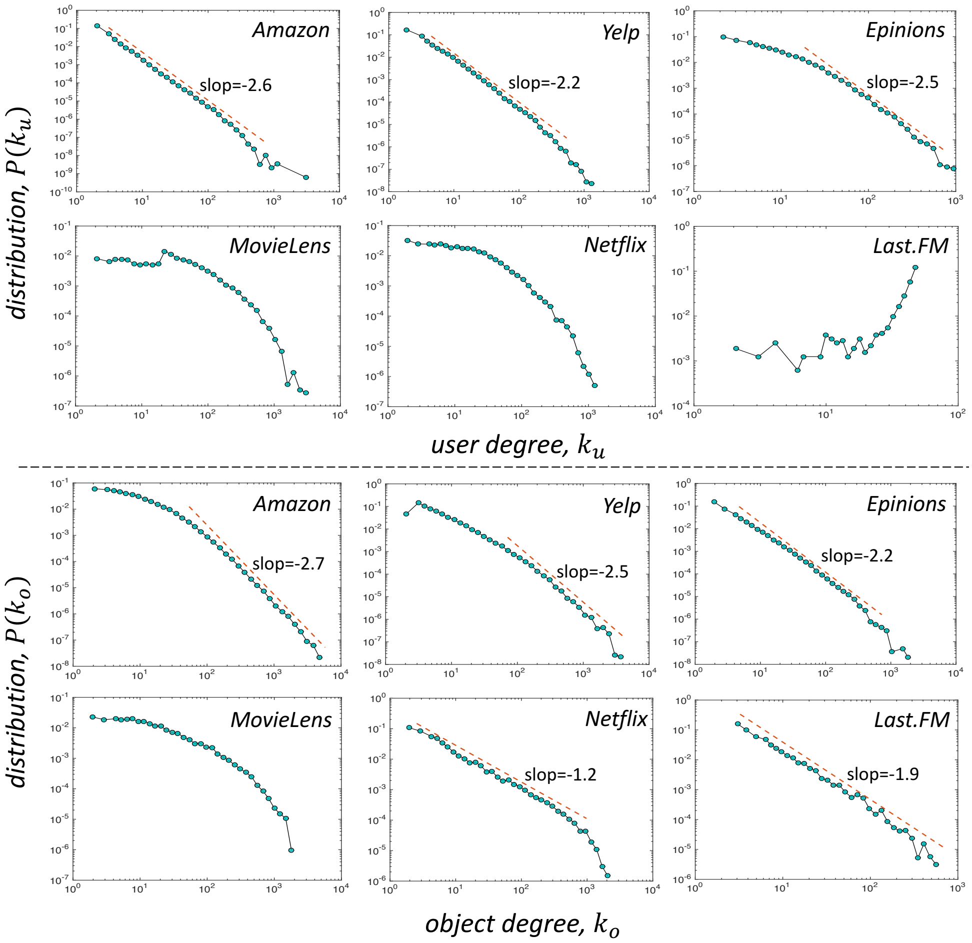


Figure 3.2 | User degree distributions (upper section) and object degree distributions (bottom section) of the applied six user-object bipartite networks. The distributions are plotted in log-log scales and a linear pattern indicts power-law distribution. The dashed red lines mark the linear pattern of the distributions with slope (power-law) exponent as shown in the plots.

The **Epinions** is a product review website, which is actually similar to Yelp. In Epinions, users normally write reviews on products such as cars, books, movies or electronics. Currently, we do not find the presence of recommendation network system in Epinions. When a user browsing a product’s reviews, the system does recommend other products, but the hyperlinks direct the user to other retail website where he/she can purchase the product, rather than the internal review webpage. Accordingly, such kind of recommendation does not connect the (webpages of) products in Epinions system as a network. An interesting point about the Epinions system is that users can decide to trust others, and accordingly it has been widely used to study the trust-based recommendations. The dataset applied in this thesis was collected by Massa and Avesani (2007), consisting of 664,824 reviews on 139,738 different products made by 40,163 users.

The **MovieLens** is a movie recommendation website running by a research lab entitled GroupLens. In the system, users can assign ratings and tags to various movies, and the system tries to predict users’ interests and recommend more movies. The MovieLens dataset used in this thesis is published by the GroupLens at https://grouplens.org/datasets/movielens/. After cleaning, the data contains 698,054 ratings on 5,850 movies posted by 5,547 users.

The **Netflix** is an online system where users can watch TV shows and movies. In 2009, Netflix held a recommendation competition with a prize of 1,000,000 dollars (https://www.netflixprize.com). The dataset in this thesis is a part of the competition dataset, consisting of 419,247 ratings on 5,081 movies posted by 8,608 users.

The **Last.FM** is a website that allow users to listen to songs and explore artists. The data applied in this thesis is entitled 'hetrec2011-lastfm-2k', which is released in Cantador *et al.* (2011). The full dataset contains social network among nearly 2000 users, the tags assigned to songs and artists and music artists listening information. In this thesis, we only focus on the 82,155 records of 1,885 users listening to 6,953 artists.

In summary, in total of six empirical user-object bipartite networks (including the Amazon data) are collected for the study in this thesis. To gain basic structural understanding on these networks, Figure 3.2 reports the degree distributions for both users and objects respectively. Though it has been found that most networks have power-law degree distributions, the distributions of these bipartite network data may largely depend on the collection and sampling strategies. The result shows that the user degree in the Amazon, Yelp and Epinions, and the object degree in Amazon, Yelp, Epinions, Netflix and Last.FM follow the power-law degree with different exponents, ranging from 1.2 to 2.7. In addition, we summarise the statistics of all these bipartite networks in Table 3.1.

Table 3.1 | Statistics of six user-object bipartite network datasets. In the table, M, N and T represent the number of users, objects and total links (data records) respectively. The sparsity is calculated as $T/(M \cdot N)$. While degree distributions of some of these datasets are in power-law form, i.e. $P(k_u) \sim k_u^{-\gamma^u}$ for users or $P(k_o) \sim k_u^{-\gamma^o}$ for objects, we report the exponents γ^u and γ^o for user distribution and object distribution respectively, as shown in Figure 3.2.

	objects type	M	N	T	Sparsity	$\boldsymbol{\gamma^u}$	$\boldsymbol{\gamma^o}$
Amazon	book	2,540,369	157,856	4,520,102	1.13×10^{-5}	2.6	2.7
Yelp	Business	366,715	61,184	1,569,264	6.99×10^{-5}	2.2	2.5
Epinions	Review	40,163	139,738	664,824	1.18×10^{-4}	2.5	2.2
MovieLens	Movie	5,547	5,850	698,054	2.15×10^{-2}	None	None
Netflix	Movie	8,608	5,081	419,247	9.59×10^{-3}	None	1.2
Last.FM	Artist	1,885	6,953	82,155	6.26×10^{-3}	None	1.9

Chapter 4. Recommendation Stability and A Top-n-Stability Algorithm

Networks, as well as user-object interaction data, are always evolving. Besides, the networks being investigated may also be mapped incompletely, and contain false positives and negatives (Ghoshal & Barabási 2011). Accordingly, can the similarities in networks be measured stably over the change of data, becomes a crucial question (RQ1 of the thesis), since the similarity measures have been widely applied in machine learning (Frey & Dueck 2007; Armano & Javarone 2013), socio-economic dynamics (Mäenpää & Jalovaara 2014), and other studies in network science (Clauset *et al.* 2008; Daminelli *et al.* 2015).

In addition, the recommendation list for a specific user has been found changing vastly when the system evolves, due to the unstable quantification of object similarities, which can be defined as the recommendation stability problem. To improve the similarity stability and recommendation stability is thus crucial for the user experience enhancement and the better understanding of user interests. While the stability as well as accuracy of the recommendation can be guaranteed by recommending only popular objects, studies have been addressing the necessity of diversity which requires the system to recommend unpopular objects (McNee *et al.* 2006; Vargas & Castells 2011). As a result, it becomes a difficulty to make recommendations which are stable, accurate and diverse at the same time. Section 4.2 presents a top-n-stability method based on the Heat Conduction algorithm (denoting with TNS-HC henceforth) for solving the stability–accuracy–diversity dilemma. In section 4.3, we show that the TNS-HC algorithm can significantly improve the recommendation stability and accuracy simultaneously and still retain the high-diversity nature of the Heat Conduction algorithm. Furthermore, we compare the performance of the TNS-HC algorithm with a number of benchmark recommendation algorithms. The result suggests that the TNS-HC algorithm is more efficient in solving the stability–accuracy–diversity triple dilemma of recommender systems.

Through the study in this Chapter, we confirm that the object similarities as well as the recommendations are not stable over the data change. The TNS-HC algorithm is developed by removing the unstable similarities, the stability of recommendations can be largely removed and at the meantime, better accuracy can be achieved. More importantly, besides the proposed algorithm, the unstable similarities are proved to be false information, which disturbs the recommender system.

4.1. Similarity Stability in Bipartite Networks

The study on the evaluation of similarity stability was carried out from the late stage of the author's master programme in collaboration with Prof. Jianguo Liu, to the early stage of this PhD programme. Considering the similarity stability study has been partially reported in the author's master thesis, here in this section, we only briefly introduce the main findings. For detailed experiments and results, please refer to the published article [Liu, J., Hou, L., Pan, X., Guo, Q., & Zhou, T. (2016). Stability of similarity measurements for bipartite networks. Scientific Reports, 6, 18653].

The similarity stability of fifteen similarity measures were investigated in six empirical user-object bipartite networks. Though the exact object similarities are unknown to be taken as benchmark to test the accuracy of the measures, we evenly and randomly divided each bipartite network into two sub-networks and compare the two similarity matrixes calculated according to these sub-networks. Three metrics were developed, namely the average bias, standard deviation of bias, and the Pearson coefficient, to measure the stability of a similarity measure.

The experiments suggested that, most existing similarity measures are unstable describing object similarities over data change. The relatively stable measures include the Common Neighbour (CN), Adamic-Adar (AA) and Resource Allocation (RA). In particular, these fifteen measures were found well clustered.

In order to show that the stability pattern of similarity measures is not caused by particular data set context, here we create an artificial network where the structure is randomly generated according to the preferential attachment, and examine the stability of similarity measures. The artificial network is generated according to the following steps:

1) Initially we assume an empty bipartite network (no links) with M users and N objects.
2) A link is added at each step to connect a user and an object, which are selected according to the probability

$$p(u) = \frac{(k_u^\gamma + 1)}{\sum_{i \in \boldsymbol{U}} (k_i^\gamma + 1)}, \tag{4.1}$$

and

$$p(o) = \frac{(k_o^\gamma + 1)}{\sum_{i \in \boldsymbol{O}} (k_i^\gamma + 1)}, \tag{4.2}$$

respectively, where $\boldsymbol{U}$ and $\boldsymbol{O}$ are the sets of users and objects respectively, and γ is a tuneable parameter which controls the intensity of the preferential attachment.
3) Repeat the step 2) for a given T steps.

In such way, a user-object bipartite network can be generated with power-law degree distributions on both user side and object side. Setting the parameters as $N = M = 5{,}000$, $T = 100{,}000$ and $\gamma = 1.4$, we divide the generated bipartite networks randomly and equally into two subnetworks. According to each of the subnetworks, a similarity matrix can be calculated by applying an arbitrary similarity measure. We use the average difference between such two matrixes, denoting with μ, and the deviation σ/μ of the differences as two dimensions of the evaluation. As shown in Figure 4.1, three clusters can be observed.

The first cluster, which is the relatively stable cluster, majorly consists of the CN, AA and RA measures. A common feature of these three measures is the fact that they only consider the information of common neighbours (users). While the CN measure directly counts the number of common neighbours, the AA and RA measures weight the common neighbours by $1/\ln(k_u)$ and $1/k_u$ respectively. The second cluster majorly contains the SAL, JAC, SOR, HP and HD measures. According to the mathematical definitions, as introduced in section 2.2, these measures can be regarded as the variations of the CN measure. They divide the CN similarity with the degree information of the two measured objects, such as $\sqrt{k_\alpha k_\beta}$ for the SAL measure, $k_\alpha + k_\beta$ for the SOR measure, $\max(k_\alpha, k_\beta)$ for the HD measure and so on. The third cluster consists of the MD and HC measures which consider the degree information of both the target objects and their common neighbours. Another common feature is that, both the MD and HC measures are designed based on the spreading process on bipartite networks. Although the basic considerations are different, mathematical definitions of the MD and HC measures are very similar to each other leading to $s_{xy}^{MD} = s_{yx}^{HC}$. Accordingly, the stabilities of the MD and HC measures are identical.

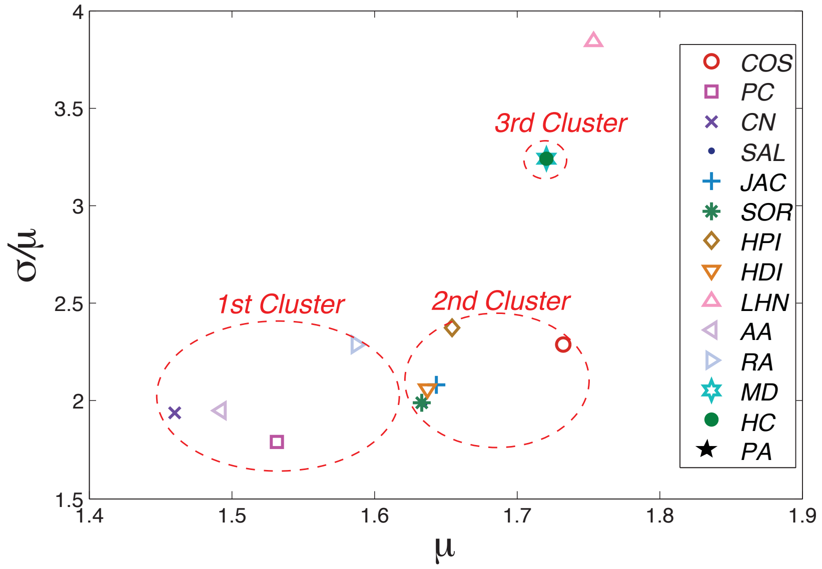


Figure 4.1 | Stability location map for fifteen similarity measures in the artificial bipartite network. Generally, the left-bottom represents more stable situation and right-top means unstable situation.

4.2. A Top-n-Stability Algorithm Based on Heat Conduction Measure

It has been suggested that most of the measures may generate totally different evaluations of the similarity when using different samples even from the same period of data. Therefore, a serious question rises that, if the similarity measures are unstable evaluating the object similarities, how could one be sure of that the measured similarity is the reflection of real similarity. Furthermore, given the recommender systems highly depending on the similarity quantifications, the unstable similarity measures offering inappropriate quantifications puts the system at risk, i.e. the recommendations will also be unstable.

Recommendations becoming unstable may cause risks such as 1) users finding recommendations unreasonable which leads to bad experiences, and 2) the uncontrollable performance of the recommendation algorithms in practical applications. The recommendation stability problem is of both theoretical and practical concerns. From the theoretical perspective, if the extracted similarity is unstable, it would be hard to evaluate whether a user is interested in an object or not. From the practical perspective, the stability problem would be a gap between laboratory investigation and real-time application because practical systems are always vastly evolving.

Note that, there are some related researches on the stability problem of recommender systems, such as Adomavicius and Zhang (2012, 2015). They define the stability as the consistency between the original recommendations and the recommendations using the combination of the historical data and some of the original recommendations (assuming some of the original recommendations have been adopted by the users). The stability we address here is defined differently from their study. While Adomavicius and Zhang (2012, 2015) used the output (recommendations) of the first recommendation experiment as the input (historical records) for the second recommendation experiment to examine the consistency of the prediction, which can be regarded as the recommendation algorithm's self-consistency, we explore the influence of the users' real behaviour growth on the similarity quantification and recommendation change. Other studies also have discussed the systems' ability to remain stable under malicious attached such as records faked for specific purpose (O'Mahony *et al*. 2004; Burke *et al*. 2015), which has also been referred as the robustness of the recommender systems.

Besides the stability problem of the recommendation, there is also the accuracy-diversity problem. If a recommender system wants to be accurate, the safest way to achieve this is to recommend users the most popular objects, because those popular objects will be interested by most of the users (otherwise, it won't be popular). However, users will hardly regard it as useful recommendations – they can find popular information by themselves easily. What the users looking for from the recommender system is more personalised recommendations according to their specific, unique interests. Therefore, being accurate is far not enough for a recommender system (McNee *et al.* 2006). However, most of the existing similarity measures are based on the Common Neighbour, which makes the similarity between two popular objects generally high such as the RA, and MD measure. Consequently, this kind of 'popular is similar' mechanism (the popularity bias) further makes the similarity-based recommender system tending to recommend popular objects rather than unpopular objects. Gradually, the importance of recommendations being diverse has got more and more attention. While some studies try to enhance the diversity by directly making inventions to the recommendation list (Ziegler *et al.* 2005; Hurley & Zhang 2011), most others tried to avoid the popular-preference from the definition of object similarity.

Similar to the accuracy problem, by only recommending popular objects, the system could have very high stability. However, the recommender system then falls again into the dilemma that whether should the recommender system recommend popular objects to achieve high stability and accuracy or recommends unpopular objects to achieve high diversity. Accordingly, there rises the triple dilemma of stability–accuracy–diversity.

Here in this section, we develop an algorithm based on the Heat Conduction measure (Zhang et al. 2007) to tackle such dilemma.

4.2.1. The algorithm

The HC measure is initially proposed based on the heat conduction process to generate highly diverse recommendations. However, the HC measure focusing too much on the unpopular objects recommendations, makes its accuracy relatively low. In order to solve the dilemma, some efforts have been devoted to further refine the HC measure (Liu *et al.* 2011), or to combine the HC method with others (Zhou *et al.* 2010).

Consider a user-object bipartite network, with a set of users $\boldsymbol{U} = \{u_1, u_2, \dots u_M\}$ and a set of object $\boldsymbol{O} = \{o_1, o_2, \dots o_N\}$. The links can be represented by an adjacency matrix $\boldsymbol{A} = \{a_{uo}\}_{M\times N}$, where $a_{uo} = 1$ if there is a link between user u and object o, and $a_{uo} = 0$ otherwise.

As has been introduced in section 2.2, the HC measure calculates the similarity between two objects o_i and o_j as

$$s_{o_i o_j} = \frac{1}{k_{o_j}} \sum_{u\in\boldsymbol{U}} \frac{a_{uo_i} a_{uo_j}}{k_u}, \tag{4.3}$$

where k_{o_j} and k_u are the degree of the object o_j and user u respectively. For a target user u, the scores of every unselected object o_j can thus be calculated as

$$w_{uo_j} = \sum_{o_i\in\Gamma_u} s_{o_i o_j}, \tag{4.4}$$

where Γ_u is the set of objects that has selected by user u historicaly.

Considering its advantage of uncovering diverse objects, here we adapt the HC algorithm to explore the question that how can we maintain the stability of recommendations. Based on this standard HC algorithm, we further consider the similarity stability, and only keep the most stable similarities. To measure the stability of similarity, we firstly divide the data (bipartite network) randomly into two subsets, i.e. each link of user-object has a probability of 50% to be assigned to one subset, or to the other subset otherwise. Hence, we can calculate the similarities for two subsets respectively according to the HC measure shown as Eq. 4.3. We denote the similarities between two objects o_i and o_j in two subsets of data as $s'_{o_i o_j}$ and $s''_{o_i o_j}$ respectively. A proper measure should result in the same similarity for two subsets, i.e. $s'_{o_i o_j} = s''_{o_i o_j}$. Therefore, the more difference between the two similarities $|s'_{o_i o_j} - s''_{o_i o_j}|$, the more unstable the similarity is. Considering the similarities may be of totally different scales, we further normalise the difference, leading to the definition of the stability of similarity from o_i to o_j as,

$$\delta_{o_i o_j} = \frac{|s'_{o_i o_j} - s''_{o_i o_j}|}{s'_{o_i o_j} + s''_{o_i o_j}}. \tag{4.5}$$

Consequently, the stability of similarity between any pair of objects would be in the range [0, 1], and the larger the value of $\delta_{o_i o_j}$ is, the more unstable the similarity should be considered.

We apply four empirical user-object bipartite networks, namely the MovieLens, Netflix, Last.fm and Epinions. The stability of the similarity between every pair of objects is calculated. According to the HC measure and Eq. 4.5. The similarity stability of HC measure is found with a strong correlation with the object popularity (degree of the objects), as shown in Figure 4.2. The results suggest that, the HC similarities directing to the unpopular objects o_j are generally less stable, i.e. larger $\delta_{\cdot o_j}$.

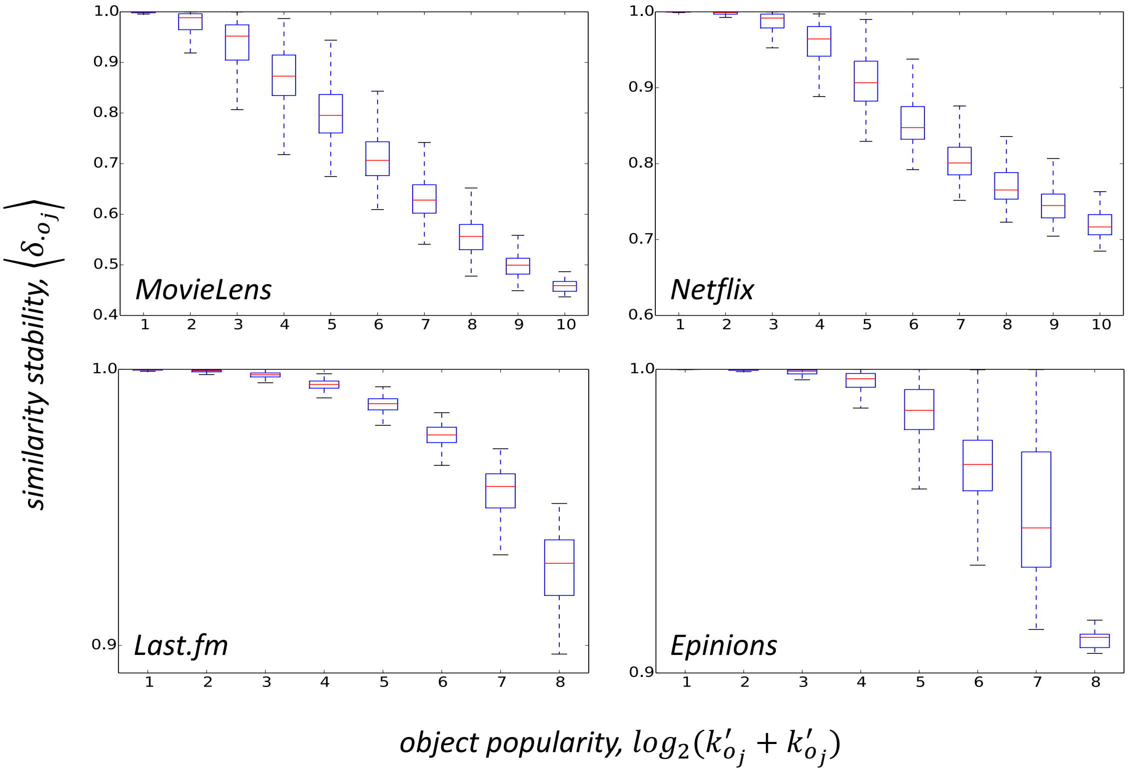


Figure 4.2 | Stability of HC similarity measure versus target object popularity. While the popularity of objects can be measured by its degree, we consider an object's degrees in both subsets of the data, i.e. $k'_{o_j} + k''_{o_j}$ as the horizontal axis. For every target object o_j, there are generally $N-1$ similarities directing to it. We accordingly use the average stability over all the objects pairs, i.e. $\langle \delta_{\cdot o_j} \rangle = \sum_{o_i \in O} \delta_{o_i o_j} / N$, as the vertical axis.

Some of the similarities are unstable when the network structure changes, and thus, such quantification cannot accurately reflect the real similarity at least in one of the subsets. Considering such situation, we only keep the most stable similarities when making recommendations. In the standard recommendation procedure, if an object o_i has been selected by the target user u in the training set, its similarity to any other object, $s_{o_i o_j}$ will be contributed to the score of the object $w_{u o_j}$ as shown in Eq. 4.4. The new algorithm assumes that the object o_i only contributes to n objects to which o_i has the most stable similarities, and hence can be referred as Top-n-Stability algorithm based on Heat Conduction measure, denoting with TNS-HC henceforce. Consequently, one can firstly rank the possible similarities starting from an object o_i in terms of stability, i.e. $\{\delta_{o_i 1}, \delta_{o_i 2}, \dots \delta_{o_i N}\}$ from low to high values. The n most stable ones of these similarities can then be considered while others can be ignored in the recommendation process. In other words, we update the similarities as,

$$s_{o_io_j} = \begin{cases} s_{o_io_j} & if\ Rank\left(\delta_{o_io_j}\right) \leq n \\ 0 & if\ Rank\left(\delta_{o_io_j}\right) > n \end{cases}. \tag{4.6}$$

Thus, we can use the updated similarity matrix to make recommendations according to Eq. 4.4. In the recommendation experiments, n is a tuneable parameter which can be gradually changed to explore the extent to which the unstable similarities should be removed.

4.2.2. Evaluation metric for recommendation stability

The evaluation of recommendation performances requires a training set and a testing set, as has been introduced in Section 2.3.2. In the development of the TNS-HC algorithm, the dataset is divided equally into two subsets so that the similarity stability can be calculated. For the purpose of evaluating recommendation performances, we use these two subsets of data as the training set and testing set by turns. Accordingly, we actually have two recommendation experiments for every data partition.

Since the proposed TNS-HC algorithm is to tackle the stability-accuracy-diversity dilemma, the recommendation performance is measured in three aspects. We take the Ranking Score RS as the metric for accuracy, and the personalisation S as the diversity metric. Please refer to the Section 2.3.2 for the definition of RS and S, in Eq. (2.28) and Eq. (2.30) respectively. As the stability problem is a newly-defined aspect for personalised recommendation, here we define the stability metric as follows.

As the two subsets are to be taken as the training set and the testing set by turns, there will be two recommendation lists for every user. For a specific target user u, if an object o_j ranked at the top L in one of the lists, it means the system predicts the object o_j to be potentially interested by the user u according to that subset. Therefore, we regard the object o_j's rank in another list as the recommendation stability of object o_j for user u denoted with Δ_{uo_j}, which can be described as

$$\Delta_{uo_j} = \begin{cases} r'_{uo_j}/(N - k''_u) & if\ r''_{uo_j} \leq L \\ r''_{uo_j}/(N - k'_u) & if\ r'_{uo_j} \leq L' \end{cases} \tag{4.7}$$

where r'_{uo_j} and r''_{uo_j} are is the ranks of object o_j in user u's two recommendation lists respectively, N is the total number of objects, and the k'_u and k''_u are the degree of the user in two subsets respectively, i.e. how many objects has the user selected in the corresponding set. Accordingly, a small value of Δ_{uo_j} suggests that the object o_j being recommended in one subset, is also ranked at a top position in another subset, and hence stable.

Following the results of the similarity stability versus object degree, we show the correlation between the recommendation stability of a recommended object $\Delta_{o_j}$and its popularity $k'_{o_j} + k''_{o_j}$ in Figure 4.3. Similar to the similarity stability, the recommendation stability of the objects also has strong correlation with object popularity. Although the recommendation stabilities empirically distribute in a wide range (sometimes almost from 0 to 1) for each popularity level, the unpopular objects with lower degree are shown generally less stable to be recommended.

To further quantify the stability of each recommendation experiment, we use the average value of all the recommended objects' stability to describe the algorithm's performance, which reads,

$$\langle\Delta\rangle = \frac{1}{|\boldsymbol{U}|}\sum_{u\in\boldsymbol{U}}\sum_{o\in\boldsymbol{\Omega_u}}\frac{\Delta_{uo}}{|\boldsymbol{\Omega_u}|}, \tag{4.8}$$

where $\boldsymbol{U}$ is the set of all the users, $\boldsymbol{\Omega_u}$ is the set of objects that being recommended (rank at the top L in the list) to user u according to either subset but have not been collected by the user in another subset. Note that, if an object o has been selected by a user in one of the subsets, it will be ranked at the bottom of the list. If in another list, the object o is in the top L positions (which is an accurate prediction), the stability would be approximately equals to 1 but cannot be regarded as unstable. Therefore, we don't include these objects that have already been historically selected by the user u in either subset to the set $\boldsymbol{\Omega_u}$.

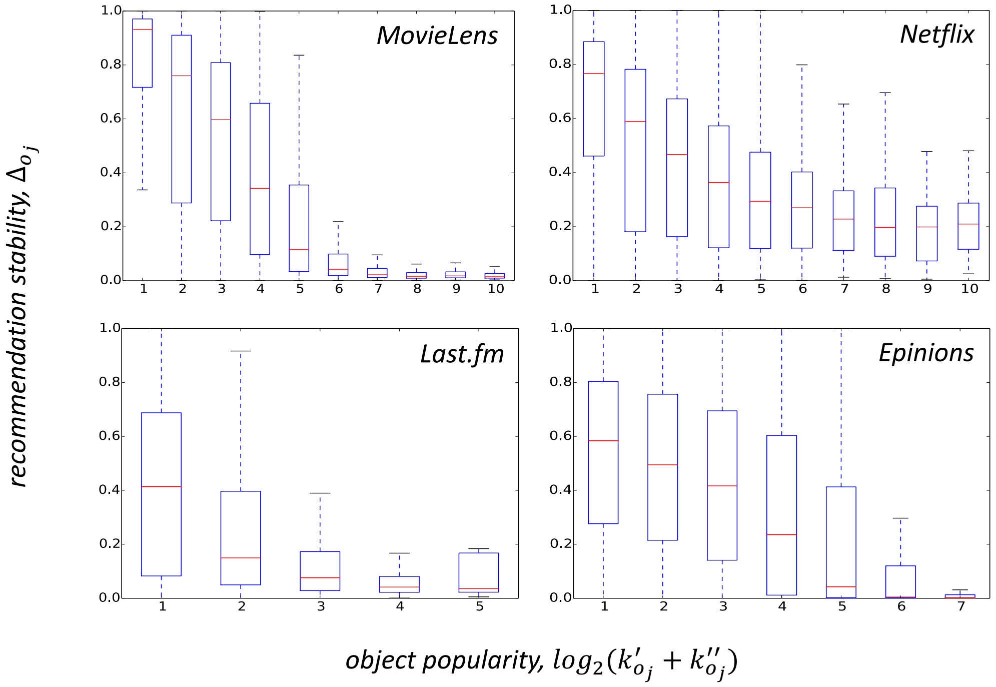


Figure 4.3 | Recommendation stability versus object popularity according to standard HC algorithm. The vertical axis is the average stability of an object o_j over all the times of it being recommended to a user.

According to this definition, a low value of recommendation stability $\langle\Delta\rangle$ means the system gives similar evaluation of the potentials of objects being selected using two subsets of data. On the other hand, a large value of recommendation stability $\langle\Delta\rangle$ would indicate that, for a general user, the potential of a particular object would be evaluated to be high in one subset but low in another. Accordingly, the lower value of stability $\langle\Delta\rangle$ means the recommendation is more stable.

4.2.3. Validation for TNS-HC algorithm in personalised recommendation

By considering only the most stable similarities, we perform the top-n-stability algorithm based on the HC similarity measure (TNS-HC) in four empirical user-object bipartite networks. We gradually change the number of stable similarities to be considered in the recommendation to explore that, to what extent should the unstable similarities be removed to gain high recommendation stability $\langle\Delta\rangle$, accuracy RS and diversity S simultaneously. The recommendation performances on four datasets, namely the MovieLens, Netflix, Last.fm and Epinions, are reported in Figure 4.4.

As the total number of objects N varies for each dataset, we take the normalised top-n, i.e. n/N to illustrate the horizontal axis. $n/N = 1$ means all the similarities are considered and thus reveals the original HC recommendation algorithm. Given the definitions of the three metrics, lower values of the stability $\langle\Delta\rangle$ and the accuracy RS represent better performances, while higher values of the diversity S indicate better performances. While each of the metrics has the optimised value of n/N, we take the n/N minimising the accuracy RS as the optimised value for the overall performance, as identified by the blue dashed lines in the figure. In practical cases, one can take value optimising any function of the three metrics as such optimised top-n.

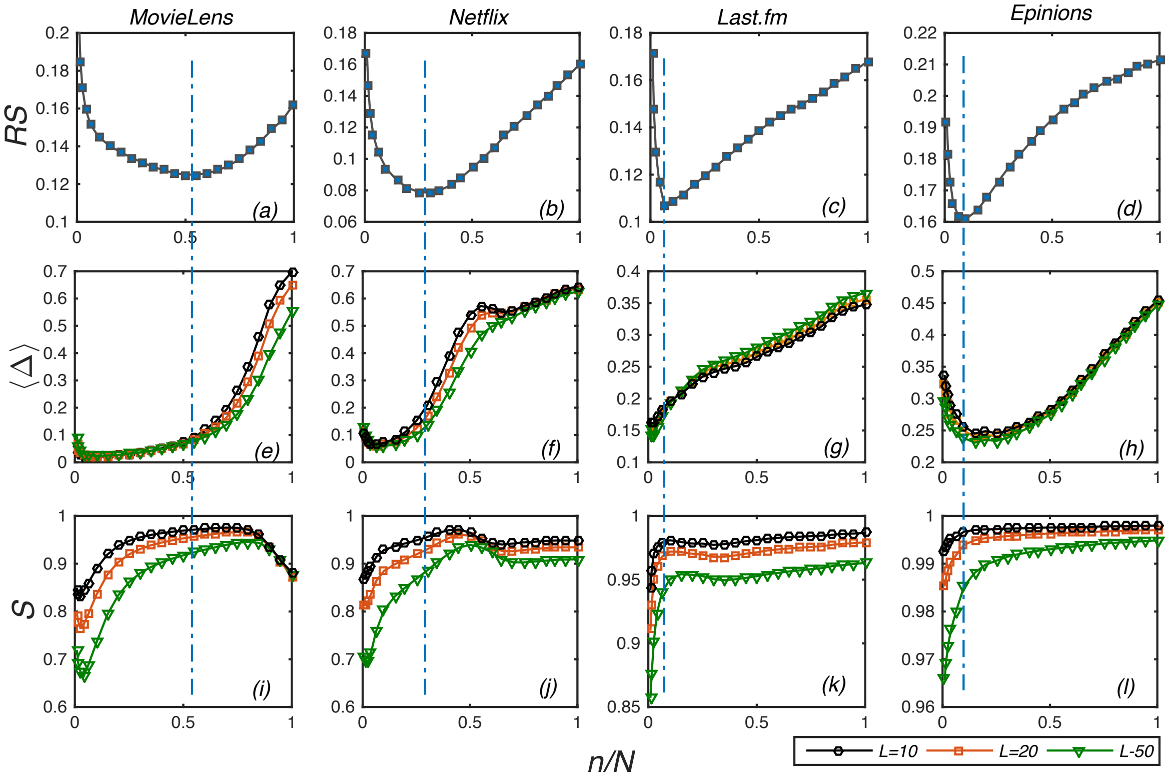


Figure 4.4 | Recommendation performances, namely the accuracy RS, stability $\langle\Delta\rangle$ and diversity S of the TNS-HC algorithm. The ranking score RS is a metric independent from the recommendation list length. For stability and diversity, we show the results based on list lengths $L = 10, 20$ and 50, as represented by the black circles, red squares and green triangles respectively. All the results are averaged over 20 independent experiments with different data partitions and recommendations.

While the standard HC ($n/N = 1$) is able to generate highly diverse recommendations with diversity S generally larger than 0.9, the recommendation lists are quite unstable (low stability $\langle\Delta\rangle$). In the MovieLens and Netflix datasets, the stability $\langle\Delta\rangle$ even goes beyond the value of random scenario. If recommending objects uniformly at random which means all the objects are randomly ranked into a list, the objects that recommended by one list would randomly distribute in another list, which leads to the random stability $\langle\Delta\rangle = 0.5$. According to the definition, the HC algorithm experts in recommending unpopular objects. However, the unpopular objects are generally less stable in the system as illustrated in Figure 4.3. Hence, the standard HC algorithm recommending unpopular objects has poor recommendation stability $\langle\Delta\rangle$.

When removing the unstable similarities (see the curves from right to left in Figure 4.4), a surprising finding is that the accuracy, i.e. ranking score RS, is getting better. The reason lies in the fact that, unstable similarities can be regarded as false positives in the evaluation, which cannot represent the real similarities among objects. Considering such false positives in the recommendation would confuse the algorithm resulting false recommendations. As a consequence, when the system removes these unstable similarities, the accuracy of recommendation can be largely improved. However, to make accurate recommendations, the system still need sufficient amount of data. The recommendation accuracy can thus be optimised with a certain amount of unstable similarities being removed. Empirically in the four datasets applied here, the optimised data volumes $(n/N)^o$ are 55%, 30%, 6% and 10% for MovieLens, Netflix, Last.fm and Epinions respectively. The optimised data volume for the Last.fm and Epinions datasets are surprisingly low, indicating that there are indeed large amounts of false evaluation of similarities in these systems.

The TNS-HC algorithm can also largely improve the recommendation stability $\langle\Delta\rangle$ by removing the unstable similarities with the optimised value of $(n/N)^o$. Taking $L = 20$ as an example, with only the stable information, the stability $\langle\Delta\rangle$ can be improved by 87%, 73%, 50% and 45% for MovieLens, Netflix, Last.fm and Epinions respectively, as shown in Table 4.1. On the other hand, the high-diversity nature of the HC algorithm is retained by the TNS-HC algorithm. For the MovieLens dataset, the diversity is also improved by 9.9% to a quite diverse level $S(20) = 0.96$. As to the other three datasets, the diversity of the standard HC algorithm has already reached a very high level ($S > 0.9$) due to the sparsity of the datasets (Table 3.1). Hence, the TNS-HC does not improve the diversity in these three datasets further.

Table 4.1 | Numerical results of standard HC and the optimised TNS-HC algorithm. In the table, $(n/N)^o$ represents the optimised value of n/N minimising the accuracy RS. For the ratios, the numerical value is calculated as $|m^{HC} - m^{TNS-HC}|/m^{HC}$ for every metric m. And if for a metric, the performance of the TNS-HC algorithm is better than the HC algorithm, the value would be marked as positive (+), and (-) otherwise.

		MovieLens			*Netflix*			*Last.fm*			*Epinions*		
L		10	20	50	10	20	50	10	20	50	10	20	50
HC	RS		0.162			0.16			0.131			0.211	
	$\langle\Delta\rangle$	0.697	0.651	0.552	0.643	0.637	0.625	0.347	0.356	0.366	0.453	0.451	0.446
	S	0.882	0.871	0.874	0.948	0.935	0.909	0.986	0.979	0.963	0.997	0.997	0.994
TNS-HC	$(n/N)^o$		0.55			0.3			0.06			0.1	
	RS		0.125			0.078			0.107			0.161	
	$\langle\Delta\rangle$	0.092	0.083	0.077	0.208	0.17	0.134	0.184	0.179	0.173	0.256	0.249	0.238
	S	0.971	0.957	0.923	0.955	0.93	0.885	0.979	0.968	0.94	0.996	0.994	0.985
Ratio %	RS		+22.83			+51.25			+18.32			+23.7	
	$\langle\Delta\rangle$	+86.8	+87.3	+86.1	+67.1	+73.3	+78.6	+46.9	+49.7	+52.7	+43.5	+44.8	+46.6
	S	+10.1	+9.9	+5.6	+0.73	-0.53	-2.64	-0.7	-1.12	-2.38	-0.1	-0.3	-0.9

Table 4.1 summarises the numerical results of the standard HC algorithm and the TNS -HC algorithm in the optimised situation, and compares the two algorithms. The TNS-HC algorithm simultaneously gaining high stability, high accuracy as well as high diversity, is an efficient method to overcome the stability-accuracy-diversity triple dilemma. Additionally, the stability $\langle\Delta\rangle$ and the diversity S for recommendation with different recommendation list length L have very similar behaviour against

the change of the parameter n. In other words, the length of recommendation list does not significantly influence the top-n-stability method on the HC algorithm.

We also report the accuracy of the recommendations in terms of the precision and recall in Figure 4.5, which has been introduced and defined in Section 2.3.2. Since they are all accuracy metrics, the precision and recall behave very similar to the ranking score. However, they have different optimised value for the data volume. To achieve highest value for precision and recall, the similarities to be considered should be generally less than that for the ranking score.

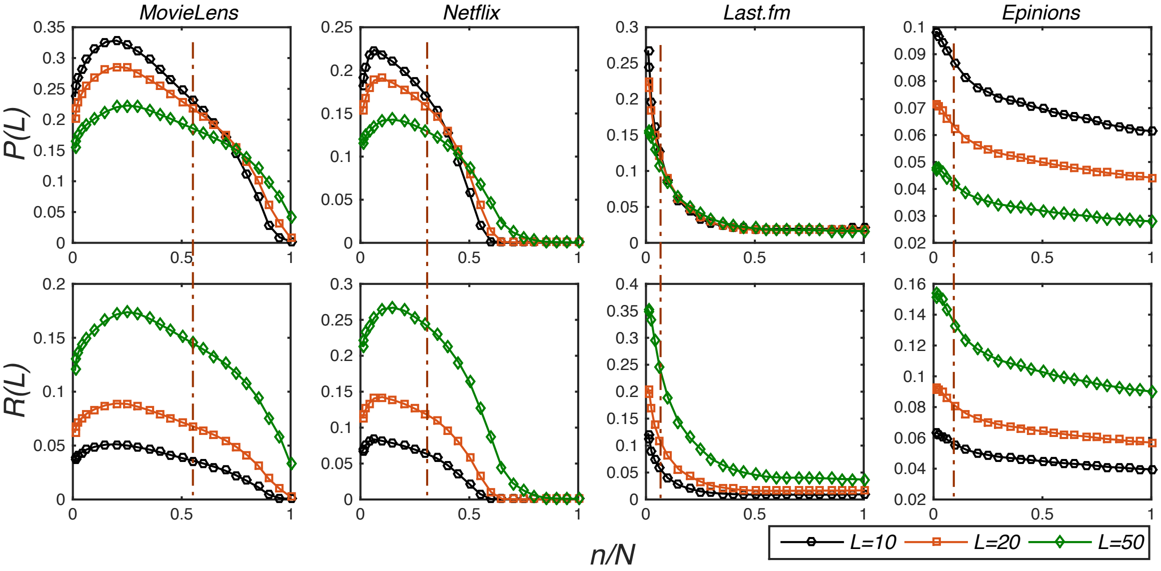


Figure 4.5 | Precision $P(L)$ and Recall $R(L)$ of the TNS-HC algorithm with different lengths of recommendation lists. The dashed line in each column is the optimised value of n/N minimising the ranking score RS which has been reported in Figure 4.4. As the length of recommendation list L does not influence the value of ranking score RS, these results with different lengths have the same optimised value of n/N. Note that, the higher values of precision and recall represent more accurate performances.

4.3. Comparison to benchmark algorithms

The TNS-HC algorithm gives a solution to the triple dilemma, which could generate accurate, diverse and most importantly, stable recommendations. To examine whether the TNS-HC algorithm can out-perform other algorithms in terms of the triple dilemma, we carry out comparative analysis to further compare its performances with some of the benchmark recommendation algorithms. Since the TNS-HC algorithm is based on the HC algorithm we firstly compare the proposed method with some of the other hybrid of the HC measure. We also compare with other single algorithms based on other similarity measures.

4.3.1. Hybrid algorithms based on HC

As the proposed TNS-HC algorithm is to rank the stability of similarities to keep only the stable similarities, it is necessary to consider another method of ranking, which is the top-n-popularity algorithm (TNP-HC). In addition, there are also many studies which integrate HC algorithm with other mechanisms leading to new hybrid methods, such as the Biased Heat Conduction (Biased-HC) (Liu *et al.* 2010), and Heat Conduction – Mass Diffusion (HC+MD) (Zhou et al. 2010) algorithms.

Similar to the proposed TNS-HC algorithm, all these hybrid algorithms evolve a free parameter to be optimised through experiments. Here these algorithms will be introduced in turn and the optimisation will be presented.

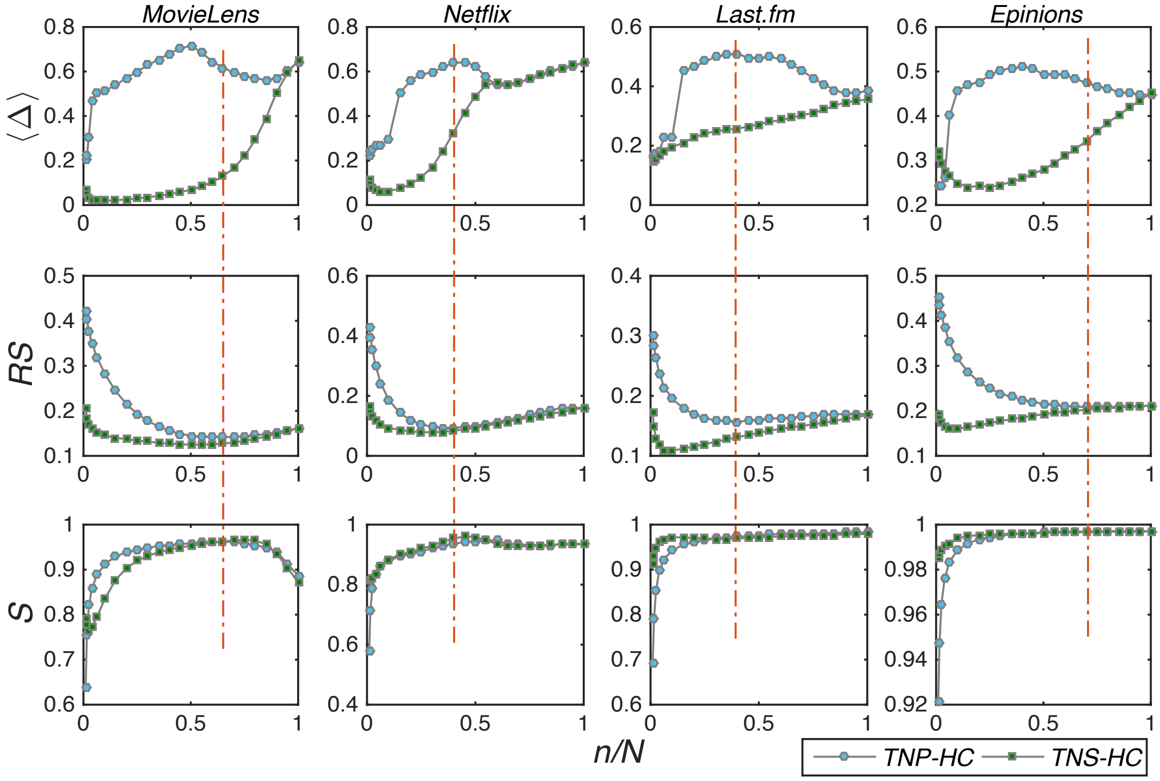


Figure 4.6 | Recommendation results of TNP-HC and TNS-HC on four datasets. The results are based on the recommendation list length of $L = 20$. The dashed line in each column is the optimised value of n/N minimising the ranking score RS for the TNP-HC algorithm.

Top-N-Popularity algorithm based on Heat Conduction (TNP-HC)

While the Top-N-Stability (TNS) method is to only consider these similarities with highest stabilities, the Top-N-Popularity (TNP) ranks the objects in terms of popularity (degree) and only consider these similarities to the most popular objects. Following the equation of TNS, i.e. Eq. (4.6), the TNP is to update the similarity matrix as,

$$s_{o_io_j} = \begin{cases} s_{o_io_j} & if\ Rank\left(k_{o_j}\right) \leq n \\ 0 & if\ Rank\left(k_{o_j}\right) > n \end{cases}, \tag{4.9}$$

where the $Rank\left(k_{o_j}\right)$ is the global ranking position of o_j in terms of popularity. Figure 4.6 reports the results of TNP-HC in comparison with TNS-HC. Note that, in both methods, $n/N = 1$ reveals the standard HC algorithm, and thus, TNP-HC and TNS-HC have exactly the same results for $n/N = 1$. While the TNP-HC has similar diversity comparing with the TNS-HC, it cannot improve the recommendations' accuracy and stability as significantly as TNS-HC. As the accuracy has been improved, though not significant, the optimised value, $(n/N)^o$ is obtained to be 0.65, 0.4, 0.4 and 0.7 for MovieLens, Netflix, Last.fm and Epinions respectively.

Biased Heat-Conduction algorithm (Biased-HC)

As has been introduced in Section 2.2, the HC algorithm considers the process of heat conduction in the bipartite network where a node receives the average heat of its neighbouring nodes. The Biased-HC (Liu *et al.* 2010) assumes that the heat is unevenly conducted. By introducing a free parameter ξ, the Biased-HC defines the similarity between two objects o_i and o_j as,

$$s_{o_i o_j}^{Biased-HC} = \frac{1}{k_{o_j}^{\xi}} \sum\nolimits_{v \in \Gamma_{o_i} \cap \Gamma_{o_j}} \frac{1}{k_v}. \tag{4.10}$$

As a consequence, $\xi = 1$ reveals the standard HC algorithm while $\xi = 0$ reveals the standard RA algorithm. The recommendation performances of the Biased-HC algorithm are reported in Figure 4.7. While the accuracy and stability can be largely improved by the Biased-HC algorithm, the diversity in Netflix and Last.fm dataset, is largely sacrificed for the such improvements. The optimisation of the parameter ξ is 0.85, 0.5, 0.4 and 0.35 for MovieLens, Netflix, Last.fm and Epinions dataset respectively.

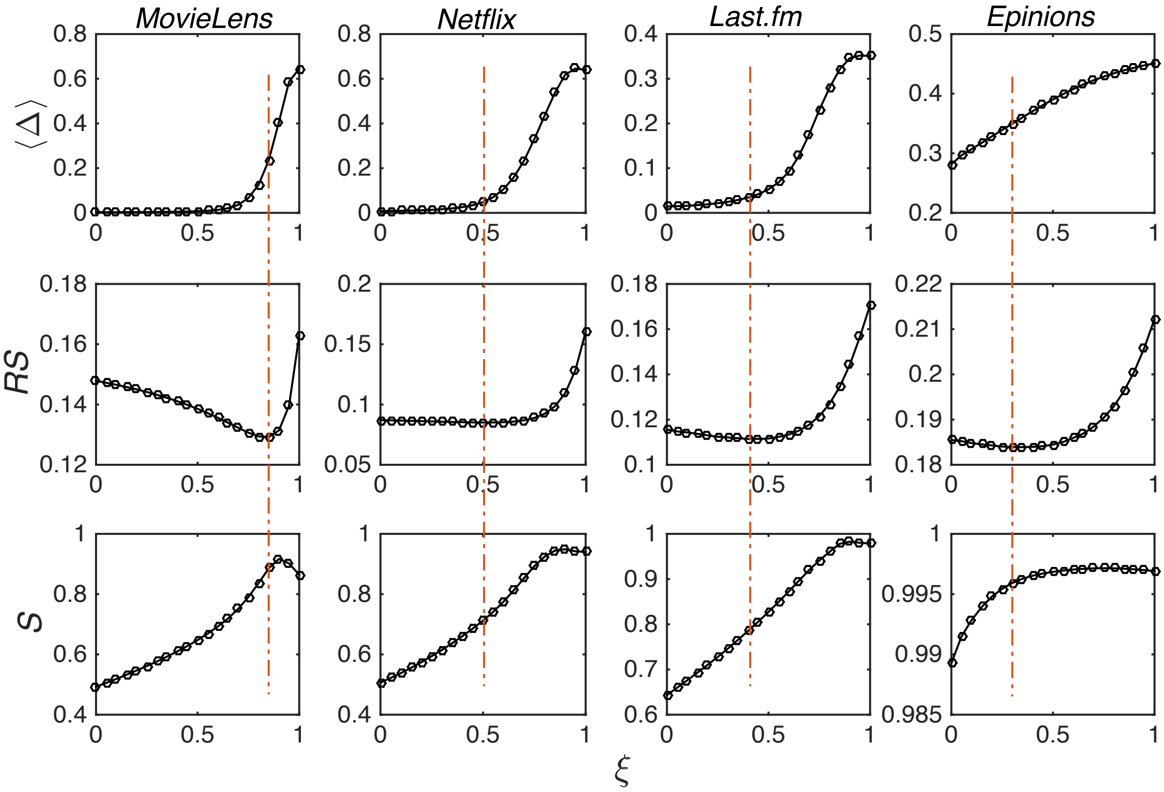


Figure 4.7 | Recommendation results of Biased-HC on four datasets. The red dashed line in each column marks the optimised parameter ξ for each dataset to minimise the accuracy RS. When calculating the stability $\langle\Delta\rangle$ and diversity S, the recommendation list length is set as L = 20.

Heat Conduction – Mass Diffusion (HC+MD)

Given the fact that the HC algorithm can generate diverse recommendations and the MD (Zhou *et al.* 2010) algorithm (introduced in section 2.2) can generate accurate recommendations, a hybrid method is proposed combining such two algorithms to achieve simultaneously accurate and diverse recommendations. Both the HC and MD algorithm define the object similarity according to the weighted common neighbours ($\sum_{v\in\Gamma_{o_i}\cap\Gamma_{o_j}}\frac{1}{k_v}$), but HC divides it by the target object's degree k_{o_j} while MD divides it by the source object's degree k_{o_i}. By introducing a parameter ξ, the HC+MD algorithm explores the trade-off between such two algorithms, and defines the object similarity as

$$s_{o_io_j}^{HC+MD} = \frac{1}{k_{o_i}^{1-\xi}k_{o_j}^{\xi}}\sum\nolimits_{v\in\Gamma_{o_i}\cap\Gamma_{o_j}}\frac{1}{k_v}. \tag{4.11}$$

Therefore, $\xi = 0$ gives the standard MD method, and $\xi = 1$ gives the standard HC method. By adjusting the parameter ξ, the hybrid algorithm HC+MD can find the optimised value to offer both accurate and diverse recommendations as shown in 4.8. While one can define any utility function to determine the optimised parameter ξ, here we also choose the parameter ξ to minimise the ranking score RS, to be same with the proposed TNS-HC. With MD and HC as lower-boundary ($\xi = 0$) and upper-boundary ($\xi = 1$), the HC+MD recommendations' stability $\langle\Delta\rangle$ become larger when tuning ξ from 0 to 1. Therefore, at the optimised ξ, the stability of recommendation $\langle\Delta\rangle$ is better than HC but worse than MD. The optimised value of ξ is 0.85, 0.5, 0.55 and 0.35 for MovieLens, Netflix, Last.fm and Epinions respectively.

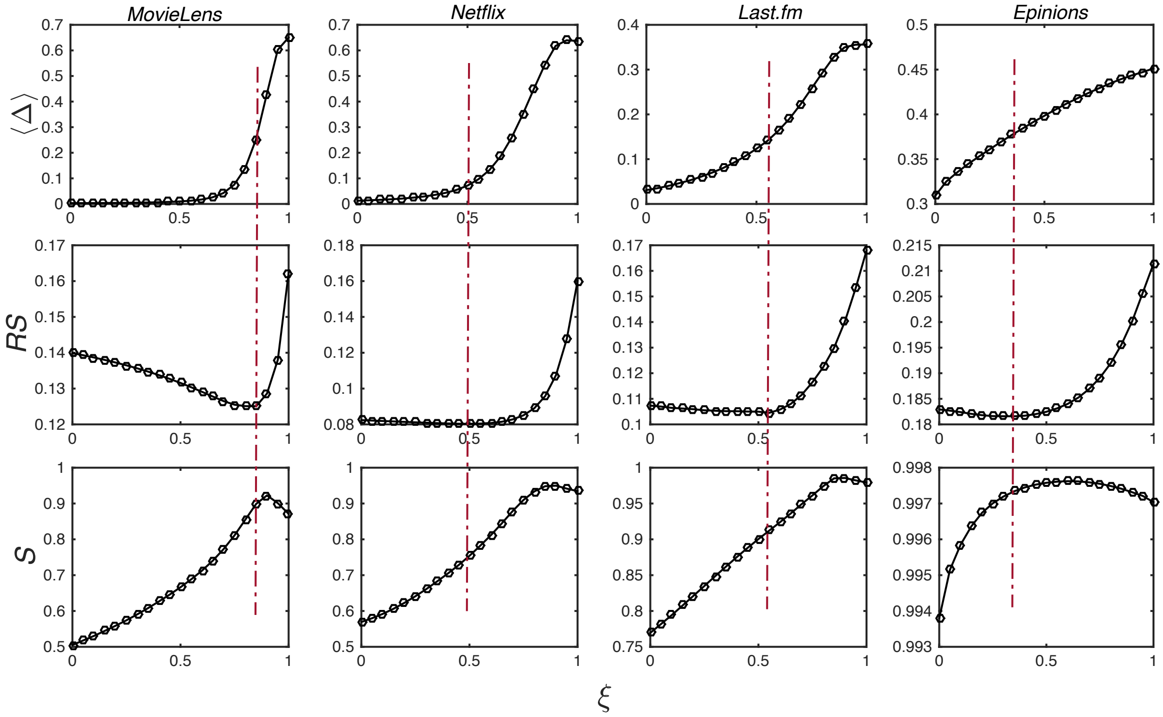


Figure 4.8 | Recommendation results of HC+MD on four datasets. The red dashed line in each column marks the optimised parameter ξ for each dataset to minimise the accuracy. With respect to the proposed TNS-HC, here we take $L = 20$ for the length of recommendation list for the calculation of the stability $\langle\Delta\rangle$ and diversity S.

4.3.2. Individual algorithms

In addition to the hybrid algorithms, we consider individual algorithms with different similarity measures, including HC (Eq. 2.24), Improved Heat Conduction (IHC), CN (Eq. 2.13), SAL (Eq. 2.14), JAC (Eq. 2.16), MD (Eq. 2.23). Since the definitions of the HC, CN, SAL, JAC and MD measures have been introduced in section 2.2, here we only introduce the mathematical definition of IHC, which defines the object similarity as

$$s_{o_i o_j}^{IHC} = \frac{1}{k_{o_j}^2} \sum\nolimits_{v \in \Gamma_{o_i} \cap \Gamma_{o_j}} \frac{1}{k_v}. \qquad (4.12)$$

While the CN-based methods consider the population of two objects' neighbourhood (users who selected both of them), it has been argued that, the density of the neighbourhood is also very important for the evaluation of the similarity (Cannistraci *et al.* 2013; Daminelli *et al.* 2015). Accordingly, an algorithm entitled Local-Community-Paradigm (LCP) has been proposed for the link prediction in bipartite networks by considering both the number of the common neighbours and the number of local links among those common neighbours (Daminelli *et al.* 2015).

Different with other similarity measures which calculate the similarity between two objects, the LCP directly calculates the similarity between a user and an object. Firstly, one needs to find the local community between a user u and an object o_i. Notably, LCP defines the local community differently from the classical CN method of bipartite network. Instead of the nodes (users) that connecting both of the target nodes, the LCP considers any nodes (both users and objects) that are on the paths with length three between the user u and the object o_i as their common neighbour. Thus, the LCP defines the similarity between the user u and any other object o_i as,

$$s_{uo_i}^{LCP} = CN_{uo_i}^{LCP} \cdot LCL_{uo_i}, \qquad (4.13)$$

where the $CN_{uo_i}^{LCP}$ is the number of nodes in the local community of u and o_i, and the LCL_{uo_i} represents the number of links in the local community, i.e. links among these common neighbours. This method was proposed for the link prediction of bipartite networks and thus they rank all the user-object pairs together in terms of the similarity and the top ones are considered as predictions. In order to introduce such method to the recommendation scenario, we rank the similarities for each user, and these objects with the highest similarity with the target user are considered as the recommendations.

4.3.3. Comparison

With all the classical recommendation algorithms introduced, here we compare the performances of different algorithms with the proposed TNS-HC algorithm. Note that, all the object-similarity-based algorithms are to firstly develop the similarity matrix for objects, and then make recommendations for each user following Eq. 2.25. For the LCP algorithm, since the similarities between users and objects are directly calculated, the recommendations are made accordingly. As the quantification for the stability metric requires the data to be partitioned according to 5:5 to training and testing set, all the recommendation experiments are based on such partition.

Based on a recommendation list length of $L = 20$, Table 4.2 reports the stability $\langle\Delta\rangle$, accuracy RS and diversity S of the recommendations resulted from each of these introduced algorithms. One can find from the numerical comparison within the HC-based algorithms that, the proposed TNS-HC algorithm has relatively good performances for all the three metrics. As to other HC-based algorithms, some still have low stability and somehow low accuracy such as HC itself, IHC algorithm and the TNP-HC algorithm, and some sacrifice the diversity when trying to gain better stability and accuracy such as the HC+MD and the Biased-HC algorithms especially in the Netflix and Last.fm datasets. The TNS-HC algorithm, on the other hand is able to improve the stability and accuracy without sacrificing the high diversity, and thereby has very balanced, yet good, performances in terms of the stability, accuracy and diversity. The CN-based algorithms have generally very good stabilities due to the apparent popularity correlation of the CN nature which is, popular objects are more likely to have more common neighbours with others. However, the accuracy and especially the diversity of CN-based algorithms are low in comparison with the proposed TNS- HC algorithm. The LCP algorithm behaves very similar to the CN-based algorithms with very good stability but low accuracy and diversity.

4.4. Summary

Due to the rapid data change in online systems, it becomes an urgent question that how could we guarantee the stability of the similarity quantification and recommendation? The stability problem will result in a great gap between the laboratory investigations and the practical applications of recommender systems. Furthermore, there arises the dilemma between diversity and stability. While high-diversity requires the system to recommend those unpopular objects, the local structures of those unpopular objects are generally unstable. Considering additionally the basic accuracy requirement of recommender systems, the challenge lies in how to overcome the triple dilemma of stability-accuracy-diversity.

Though it is found that when the data changes, the similarity between objects also change vastly, not every pair of objects has unstable similarities. Regarding the unstable similarities as false quantifications, the stable similarities are identified and we only consider a certain amount of stable similarities in the recommendation process, resulting the Top-N-Stability method based on the Heat Conduction (TNS-HC) algorithm. Applying the proposed TNS-HC algorithm, we explore whether the removal of unstable similarities can benefit the recommender system, to assess the role of unstable similarities. Gradually removing the unstable similarities is shown to be able to largely improve the recommendation's stability and accuracy simultaneously, and yet retain the high-diversity nature of the HC algorithm. In comparison to benchmark algorithms, the proposed TNS-HC algorithm is one of the best algorithms in terms of each metric, and arguably the best algorithm if considering all three metrics simultaneously. Therefore, the TNS-HC algorithm can be regarded as an efficient method overcoming the stability-accuracy-diversity dilemma of personalised recommendations.

This Chapter aims to tackle the RQ 1 of the thesis (can the object similarities remain stable over the data change and how can we ensure the recommendation stability?) For the first part of the question, it is confirmed in section 4.1 that the objects are not stable at all. On the other hand, the proposed TNS-HC algorithm is proved through recommendation experiments and comparative analysis to be an effective method to ensure the recommendation stability. As a consequence, the RQ1 has been fully addressed by this Chapter.

Table 4.2 | Comparisons of TNS-HC algorithm with benchmark algorithms. The results of TNS-HC, TNP-HC, HC+MD and Biased-HC are based on the optimised parameters respectively. The recommendations are all based on the list length $L = 20$. The random recommendation is to ranking the items randomly for each user as their recommendation list. All the results are averaged over 20 independent experiments.

		MovieLens			*Netflix*			*Last.fm*			*Epinions*		
		$\langle\Delta\rangle$	*RS*	*S*	$\langle\Delta\rangle$	*RS*	*S*	$\langle\Delta\rangle$	*RS*	*S*	$\langle\Delta\rangle$	*RS*	*S*
HC-based	TNS-HC	0.083	0.124	0.956	0.17	0.078	0.930	0.179	0.107	0.968	0.249	0.160	0.993
	HC	0.650	0.162	0.871	0.636	0.160	0.934	0.356	0.167	0.979	0.451	0.211	0.997
	IHC	0.622	0.216	0.848	0.584	0.227	0.930	0.419	0.171	0.955	0.430	0.226	0.997
	TNP-HC	0.616	0.141	0.961	0.637	0.093	0.936	0.508	0.157	0.973	0.474	0.207	0.996
	HC+MD	0.251	0.125	0.899	0.071	0.080	0.754	0.140	0.104	0.912	0.376	0.181	0.997
	Bised-HC	0.235	0.129	0.885	0.048	0.085	0.712	0.034	0.111	0.785	0.360	0.183	0.996
CN-based	CN	0.002	0.152	0.435	0.006	0.090	0.422	0.012	0.116	0.640	0.207	0.203	0.956
	SAL	0.102	0.150	0.76	0.174	0.096	0.800	0.249	0.107	0.902	0.398	0.225	0.982
	JAC	0.026	0.145	0.824	0.039	0.091	0.830	0.131	0.106	0.910	0.369	0.212	0.991
	MD	0.003	0.139	0.512	0.012	0.082	0.569	0.030	0.108	0.763	0.308	0.182	0.993
LCP		0.002	0.153	0.39	0.006	0.091	0.422	0.009	0.114	0.672	0.211	0.201	0.951
Random recommendation		0.499	0.519	0.996	0.500	0.508	0.996	0.503	0.503	0.997	0.500	0.504	0.999

Chapter 5. A New Similarity Measure for Personalised Recommendation

The degree preference has been considered as the key attractiveness driving the evolution of networks (Fortunato *et al.* 2006a; Ratkiewicz *et al.* 2010; Bagrow & Brockmann 2013) ever since the finding of the scaling phenomena (Barabási & Albert 1999). However, real networks are also found to be highly clustered (Watts & Strogatz 1998) and with dense community structure (Girvan & Newman 2002; Cui *et al.* 2015) which cannot be explained by the preferential attachment mechanism alone. Accordingly, node similarity is also argued to be a driver for networks (Papadopoulos *et al.* 2012) and has been applied to study the formation and evolution of different networks (Crandall *et al.* 2010; Chen *et al.* 2017; Zeng *et al.* 2014). As has been introduced in Section 2.2, a number of node similarity measures for complex networks have been proposed. Normally, the nodes that share the same neighbours are considered to be similar to each other. However, we show in Section 5.1 that the similarities quantified by these existing measures mostly have systematic bias regarding the node degree, i.e. hub nodes tend to have more common neighbours with others due to their rich consecutiveness. As a consequence, it is difficult to determine whether the common neighbours are due to the similarity between nodes or just random mechanism.

Such systematic bias, we refer as degree bias henceforth, mixes the degree preference and the similarity preferences of the network evolution mechanism and thus makes the quantified similarities incorrect and sometimes misleading. Therefore, the key question is that how many common neighbours two particular nodes are expected to share due to the degree preference in a given network, and how can we remove such number to get an unbiased similarity measure describing solely the similarity preference (RQ2 of the thesis)?

In this chapter, we theoretically study the expected number of common neighbours between two nodes with given degrees in random networks, and propose a new similarity measure, namely the Balanced Common Neighbour measure for unipartite network (Section 5.2) and bipartite network (Section 5.3) respectively, by removing the expected number from the empirical number of common neighbours.

This Chapter contributes a new similarity measure theoretically with no degree bias for both unipartite networks and bipartite networks. In addition, it is concluded that separating and controlling the preference for degree can largely improve the recommendation performance.

5.1. Degree Bias of Similarity Measures

To show the degree bias of the existing similarity measures, here we calculate the node similarities according to some of the widely-used measures in an empirical network. Considering that the present thesis majorly focuses on the online recommendation, here we apply the empirical user-book bipartite network collected from Amazon, which has been introduced in Section 3.1.

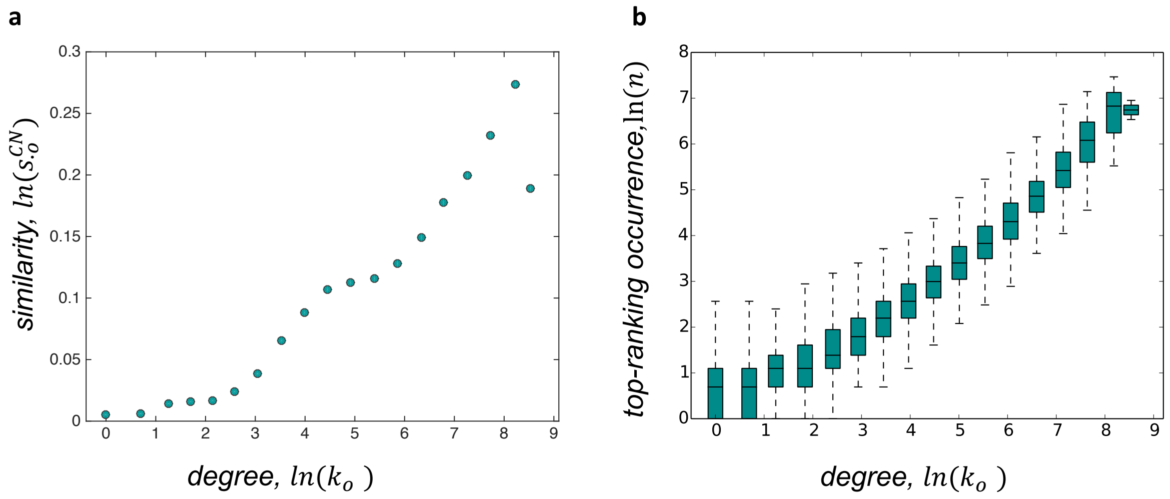


Figure 5.1 | Degree bias of CN similarity measure. Subplot (a) shows the correlation between the average similarity of an object $s_{\cdot o}^{CN}$, and its degree k_o. Subplot (b) is the occurrence of an object in others' top-10 most similar objects, versus its degree. We categorise the objects into groups in terms of their degree, and show the average value or the value distribution (box plot) of each group accordingly.

We start with examining the most fundamental measure, namely the Common Neighbour (CN) measure. For each object o in the network, we calculate the similarity from every other object to it, and then we have the similarities $\{s_{1o}^{CN}, s_{2o}^{CN}, \dots, s_{No}^{CN}\}$. For the non-zero similarities (at least one common neighbour between o and another object), we take average over them and denote with $s_{\cdot o}^{CN}$. Accordingly, the question then is transferred as: is the similarity $s_{\cdot o}^{CN}$ correlated with the degree of the object o? The similarities for every of the 157,856 objects are calculated and the relation between object similarity and object degree is shown in Figure 5.1a. Since there are too many data points (objects), we divide these objects into groups in terms of degree, and the similarity and degree in the figure are all the average value over each group of objects. Apparently, the CN similarity is positively correlated with the object degree, i.e. objects with higher degree tend to be more similar to others. To more closely show this, we measure how many times (occurrence, n) an object o is in the top-10 similar objects for others. Figure 5.1b shows the box plot, i.e. the distribution, of such occurrences for each group versus the average degree of the group. Being consistent with the average similarity (subplot a), the top-ranking occurrence is also positively correlated with the object degree for the CN measure. Note that, the top-ranking occurrence is directly linked to the possibility of an object being recommended. Thus, such results suggest that the popular (large-degree) objects have more chance to be recommended according to the CN measure. According to these results, we can conclude that the CN measure does have a strong, and positive degree bias.

The degree bias of CN measure is easy to be understood since it only considers the number of common neighbours between two nodes. It has been argued that CN measure can only well describe strongly assortative networks, where high-degree nodes tends to connect to high-degree nodes (Newman 2001b; Clauset et al. 2008). The other measures, which have been introduced in section 2.2, are basically normalisations of the CN measure. Here we further explore whether are these measures able to balance such degree bias of the CN measure.

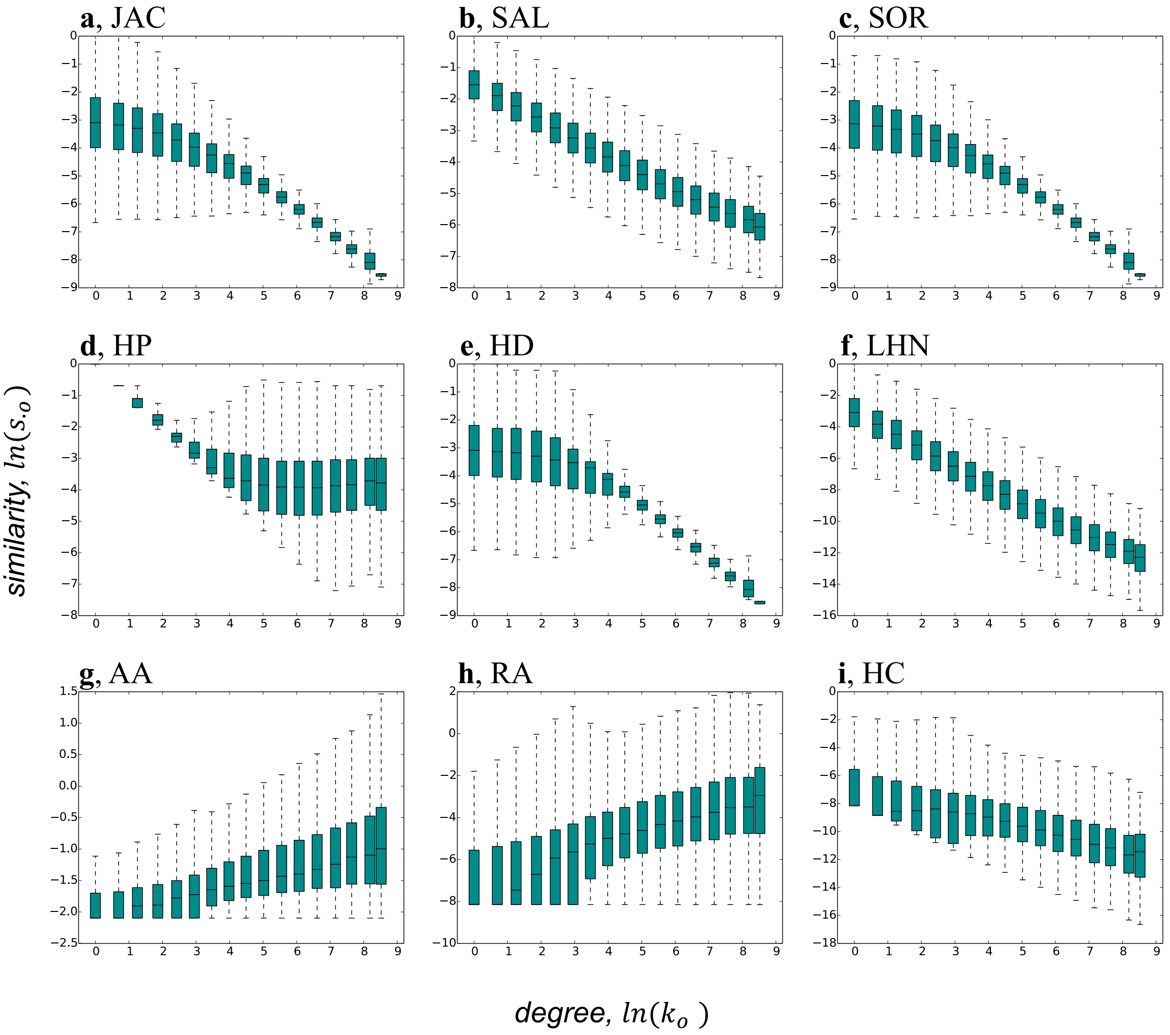


Figure 5.2 | Degree bias of similarity measures.

Following the same method of studying the CN measure, Figure 5.2 shows the relation of nine widely-used similarity measures versus object degree. The JAC (a), SAL (b), SOR (c), HPI (d), HDI (e) and LHN (f) measures can be regarded as variations of the CN measure, with additional consideration of the degree information of the two measured objects. After normalising the number of common neighbours between two measured objects with the degree information, the positive correlation between the CN similarity and the degree is revised to be negative correlations.

For all these measures, *i.e.* the JAC, SAL, SOR, HPI, HDI and LHN measures, unpopular objects tend to have higher similarity with others. When counting the number of common neighbours, the AA and RA measures weight each common neighbour (user) i with his/her degree, $\ln(k_i)$ and k_i respectively (Eq. 2.21; 2.22). Thus, the more active (higher degree) a user is, the lower the value that two objects being selected together by this user would be contributed to the similarity. However, such weightings for the CN measure does not remove the degree bias of the similarity. As suggested by the subplots (g) and (h), there are still positive correlations between the measured similarity and the degree. The HC measure (i) is based on the diffusion process on the bipartite network. While the HC measure has been found efficient in recommending unpopular objects, the reason partially lies in the fact that the unpopular (low-degree) objects tend to be more similar to others according to the HC measure. As shown in the subplot (i), the HC similarity has negative correlation with the popularity of the measured object.

To briefly conclude, the results show that most existing node similarity measures have apparent degree bias. Such bias may cause several problems for the study of the networked systems. In the context of personalised recommendation, there is a long-standing challenge which is the trade-off between the accuracy and diversity of recommendations (McNee *et al.* 2006; Vargas & Castells 2011). It has been shown that simply ranking the objects in terms of their popularities (degree) across the system, and recommending the most popular ones to every object would have quite accurate performance (Zhou et al. 2010). However, this is only due to the degree preference of such systems, rather than the similarity preference. This is also largely the reason for the fact that measures such as CN, AA and RA have good accuracy of recommendations that such similarities are positively correlated with the object degrees. With the degree preference and similarity preference mixed together, the trade-off between the accuracy and diversity would be very difficult to be studied, because such two mechanisms cannot be controlled efficiently and separately. Therefore, we believe to separate the similarity from degree preference can be an efficient approach to optimise the performance of personalised recommendations.

5.2. Balanced Common Neighbour Measure for Unipartite Network

Based on the CN measure, in this section we develop a new measure for the unipartite networks, which consists only one kind of nodes. Here we use the term ‘unipartite network’ to distinguish it from the bipartite network which will be studied in Section 5.3. But within this section (5.2), for simplicity, we use ‘network’ to refer to ‘unipartite network’.

5.2.1. Theoretical calculation

The CN measure believes that nodes sharing common neighbours are similar to each other. However, two nodes x and y that are not similar to each other at all, especially these with large degrees, could still have common neighbours by chance.

For example, in a network of 11 nodes, x and y with degrees $k_x = k_y = 6$, and not connecting to each other, should have at least 3 common neighbours. But having 3 common neighbours does not mean that they are similar, because they could have more and 3 is the lowest number possible. As a consequence, these two nodes should be considered not similar rather than similar to each other.

In other words, every pair of nodes x and y even with no similarity is expected to have a certain number of common neighbours, denoting with n_{xy}^{exp}, due to pure random mechanism or the degree preference. In a given network, if the observed number of common neighbours $n_{xy} = n_{xy}^{exp}$, we can consider these two objects x and y to be neutral to each other, i.e. not similar, nor dissimilar. Accordingly, the difference between the observed and expected number of common neighbours, which reads

$$s_{xy} = n_{xy} - n_{xy}^{exp}, \tag{5.1}$$

can be used to describe the tendency of x and y to connect the same nodes, which we argue is a more meaningful way to represent their similarity. Therefore, here we calculate the expected number of common neighbours between two nodes with given degrees in a given network. Such expected number can be regarded as the random-caused common neighbours which cannot reflect the real similarity between the measured objects. Accordingly, we can remove such random-caused common neighbours from the observed number to estimate their real similarity.

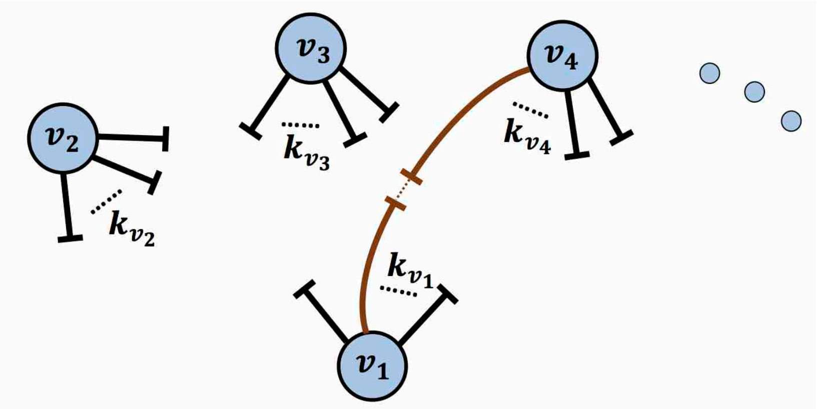


Figure 5.3 | Illustration of the random rewiring of networks. Each node v in the network has k_v half-links to be paired with others' and each pair of half-links has equal chance to be connected. Obviously, nodes with more half-edges are more likely to be connected to each other.

Consider a network of N nodes $V = \{v_1, v_2, \dots, v_N\}$ with a given degree sequence $\{k_1, k_2, \dots, k_N\}$. The expected number of common neighbours between two arbitrary nodes can be calculated by considering a random rewiring process of the given network. Assume all the links are broken into two half-links (stubs) and thus each node v has k_v half-links to be paired again with others as shown in Figure 5.3. This process is normally referred as the configuration model (Newman 2003; Kang & Seierstad 2007) which generates random networks with a given degree sequence. In the rewiring process, for each of a node i's half-links, the paired half-link is chosen randomly but from another node that has not been connected by i to avoid multi-links or self-loops. Therefore, the probability of the paired half-link coming from node j is thus $k_j / \sum_v k_v$. Considering all the k_i links that node i possessing, we have the probability of two random nodes i and j connecting with each other as (Chuang & Lu 2002a; 2002b),

$$p(i \leftrightarrow j) = \frac{k_i k_j}{\sum_v k_v}, \tag{5.2}$$

where $\sum_v k_v$ is the total number of half-links in the network. Accordingly, the probability of a node i being a common neighbour for nodes x and y, i.e. connecting to both x and y, can be written as,

$$p(i \leftrightarrow x, y) = p(i \leftrightarrow x) \cdot p(i \leftrightarrow y) = \frac{k_i(k_i - 1)}{(\sum_v k_v)^2} \cdot k_x k_y. \tag{5.3}$$

Following this logic, every node could possibly be the common neighbour for nodes x and y with different probabilities to be calculated according to Eq. (5.3). Considering all the possible common neighbours, we then have the expected number of common neighbours for x and y, which reads,

$$n_{xy}^{exp} = \sum_i p(i \leftrightarrow x, y) = \frac{\sum_v k_v(k_v - 1)}{(\sum_v k_v)^2} \cdot k_x k_y. \tag{5.4}$$

Therefore, as suggested by Eq. 5.4, the neighbourhood size for two arbitrary nodes x and y is expected to have a linear relation with the product of their degrees, i.e. $n_{xy}^{exp} \propto k_x k_y$. To validate whether such theoretical calculations are correct, we test the relation shown in Eq. (5.4) in the Barabási-Albert (BA) random network model (Barabási & Albert 1999), which has been introduced in section 2.1. The BA model is a random network model in which the links are attached randomly according to the degree preference without any predefined similarity. Accordingly, the nodes in a BA network are expected to be with no similarity and thus we should have $n_{xy}^{exp} = n_{xy}$. The simulated network starts from a complete network of $m_0 = 6$ nodes. At each of the following step, one node is added to the network to connect to $m = 5$ existing nodes. The probability of each node being connected is proportional to its current degree, i.e. $p(v) \propto k_v$. Nodes are added continuously until the network size reach $N = 10^4$. Considering most node pairs would have no common neighbour at all in a single realisation of network, we average n_{xy} over 10^4 realisations of the generated BA network. We rewire the generated BA network as follows:

1) select two from mN links uniformly at random;
2) chose one node from each link and switch if this will not result in multi-links or self-loops;
3) repeat 1) and 2) for $2mN$ times.

In such way, the degree of each node will not be changed while the network structure will be reshuffled for each realisation. We can average the number of common neighbours between two specific nodes n_{xy} over all realisations of the network accordingly.

Figure 5.4 shows the relation between the average number of common neighbours $\langle n_{xy} \rangle$ and the produc of the degrees of the measured nodes $k_x k_y$. Since both axes are in log scale, one should expect from Eq. (5.4) that $\log(n_{xy}^{exp}) = \log(k_x k_y) + c$, which indicates a slope of 1 in the log-scaled plot. As predicted by Eq. (5.4), the figure shows such relation, and thus we can conclude that the theoretical calculation for the expected number of common neighbours, i.e. random-caused common neighbours, is correct.

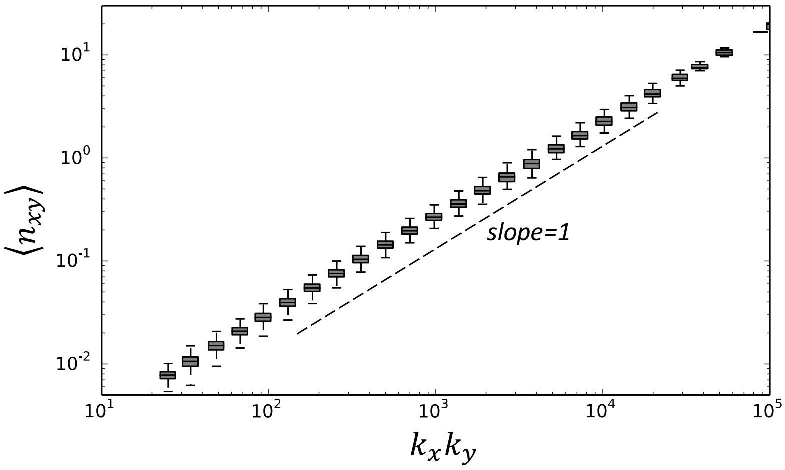


Figure 5.4 | Validation for the calculation of expected number of common neighbours. Since BA network is randomly generated, and the results are averaged over 10^4 random rewiring, the observed common neighbours can represent the expected number.

The parameter before the product $k_x k_y$ can be rewritten. Actually, in Eq. (5.4), $\sum_v k_v$ can be given by the product of the network size N and the average degree $\langle k \rangle$, i.e. $N\langle k \rangle$. Accordingly, we have also

$$\sum_v k_v(k_v - 1) = \sum_v (k_v^2 - k_v) = N(\langle k^2 \rangle - \langle k \rangle). \quad (5.5)$$

Therefore, we can rewrite the expression for the expected number of common neighbours as

$$n_{xy}^{exp} = \frac{\langle k^2 \rangle - \langle k \rangle}{N\langle k \rangle^2} k_x k_y. \quad (5.6)$$

The parameter for the product of the degrees basically describes the degree distribution feature of the whole network. The component $\langle k^2 \rangle / \langle k \rangle^2$ is usually used to described a network's degree heterogeneity H (Zhou *et al.* 2009; Vespignani 2012), which indicates how different the degrees are from node to node. With a unified degree for each node, a network has $\langle k^2 \rangle = \langle k \rangle^2$ and thus heterogeneity $H = 1$. The more heterogeneous the network's degree distribution is, the higher the value H would be. The BA network with the applied settings in this section has a degree heterogeneity $H = 2.79 \pm 0.08$. The parameter here is thus a function of the degree heterogeneity. Here we define it as a heterogeneity parameter denoting with $\mathcal{H}$, which consequently reads,

$$\mathcal{H} = \frac{\langle k^2 \rangle - \langle k \rangle}{N\langle k \rangle^2} = \frac{1}{N}\left(H - \frac{1}{\langle k \rangle}\right). \quad (5.7)$$

Introducing Eq. (5.7) into Eq. (5.6) gives us the final expression for the expected number of common neighbours for two randomly given nodes x and y as

$$n_{xy}^{exp} = \mathcal{H} \cdot k_x k_y. \quad (5.8)$$

Basically, the more heterogeneous the degrees are, the more common neighbours two nodes with given degrees would share, and on the other hand, nodes with higher degrees are likely to have more common neighbours with others. With the expected number of common neighbours n_{xy}^{exp} defined, we can then define the similarity between nodes x and y as

$$s_{xy} = n_{xy} - n_{xy}^{exp} = |\Gamma_x \cap \Gamma_y| - \mathcal{H} \cdot k_x k_y, \tag{5.9}$$

where Γ_v is the set of nodes that are connecting to node v and $|\Gamma|$ gives the number of nodes in the set.

The developed similarity s_{xy} indicates how many more (or less) common neighbours are nodes x and y sharing than expected (random). If the number of common neighbours is the same to the expected number, i.e. $s_{xy} = 0$, one can then consider x and y to be neutral to each other. On the other hand, if the nodes x and y share more (less) neighbours, i.e. $s_{xy} > 0$ ($s_{xy} < 0$), they are suggested to be similar (dissimilar) to each other. Hence, we call the proposed measure the Balanced Common Neighbour (BCN) measure.

Let's review again the example discussed earlier, that a network with 11 nodes and nodes x and y have degrees $k_x = k_y = 6$ and are not connecting to each other. Accordingly, x and y should randomly connect to 6 out of the remaining 9 nodes (aside from x and y). If ignore the degree distribution, every of these 9 nodes has a same probability of 6/9 to connect to x and y, and thus a probability of $\frac{6}{9} \cdot \frac{6}{9}$ to be the common neighbour for x and y. Counting all these 9 nodes, the expected number of common neighbours $n_{xy}^{exp} = 9 \cdot \frac{6}{9} \cdot \frac{6}{9} = 4$. As a consequence, having 3 common neighbours, though seems a lot, would indicate x and y dissimilar to each other, because they are expected to have 4. Only if they share 5 or 6 common neighbours, they can be regarded similar.

Comparison to LHN measure

There is a very similar measure entitled LHN measure (Leicht *et al.* 2006), in terms of the consideration and mathematical definition. The LHN considers the expected number of paths between two nodes with length of two, which is, in other words, the number of common neighbours. Although the same expression for the expected number of common neighbours n_{xy}^{exp} was derived, they defined the node similarity s_{xy}^{LHN} by dividing the observed number by the expected number, i.e. (Leicht *et al.* 2006),

$$s_{xy}^{LHN} = \frac{|\Gamma_x \cap \Gamma_y|}{k_x k_y}, \tag{5.10}$$

While such definition has shown good results in estimating the similarities in many networks, we believe our definition shown in Eq. (5.9) has advantages in following aspects. The real networks are usually extremely sparse, and thus most node pairs will share no common neighbours at all. For such nodes, the LHN measure considers the similarities uniformly to be zero. However, two hub nodes having no common neighbours has apparently different meaning from two low-degree nodes sharing no neighbours. The proposed BCN measure is able to estimate the similarity for node pairs even sharing no neighbours. Additionally, the BCN measure may yield negative values when the number of common neighbours is less than expected (random case), which can be regarded as the dissimilarity between the measured nodes. Especially, with the random case as the baseline, we can apply the defined similarity measure to explore that whether, or to what extent, is the similarity governing the complex networks (to be discussed in the Section 5.2.2).

Validation for BCN similarity

To test the accuracy of the proposed BCN similarity measure, we introduce the influence of similarities into the BA network model to generate networks with both degree preference and similarity preference.

Inspired by the popularity-similarity model (Papadopoulos *et al.* 2012), we randomly assign an angular position θ to each node. Nodes near to each other (with small angular distance), are considered to be similar to each other. Therefore, the predefined similarity between two nodes x and y can be written as

$$s_{xy}^{p} = 1 - \frac{2\Delta\theta_{xy}}{\pi}, \tag{5.11}$$

where $\Delta\theta_{xy}$ is the angular distance between the two nodes, i.e. $\Delta\theta_{xy} = \pi - |\pi - |\theta_x - \theta_y||$. Thus, the predefined similarity s_{xy}^{p} has a range of [-1, 1] and the larger the similarity is, the more similar the nodes are considered to be.

Instead of letting new node attach each of its m edges to an existing node i with probability proportional to only the degree, we define the probability of the new node j connecting i as

$$p(i) \propto \frac{k_i}{1 + e^{-\beta s_{ji}^{p}}}, \tag{5.12}$$

where β is a parameter controlling the influence of the predefined similarity. The case $\beta = 0$ gives the standard BA model where the links are attached according to only the degree preference with no enhancement from the similarity. For any positive β, the similar nodes are more likely to connect with each other and the larger the parameter β is, the stronger the influence of similarity would be governing the network evolution. Additionally, with such mechanism, positive predefined similarities $s_{xy}^{p} > 0$ will enhance the probability of attachment while negative values reduce such probability.

In the experiment, we randomly assign angular positions for $N = 10^4$ nodes, and accordingly use the same angular positions to generate 10^3 networks. Thus, in all the generated networks, a particular pair of nodes always has the same predefined similarity. We then calculate the similarity of every node pair according to the proposed BCN similarity measure, and average such similarity over these 10^3 networks, to check whether the BCN similarity can recover the predefined similarities. The results are shown in Figure 5.5. When setting the parameter $\beta = 0$, which indicates that the links does not emerge according to similarity, the estimated similarities do not differentiate from each other at all. As the parameter β increases, the node pairs with different predefined similarities become more and more distinguishable from each other. Especially the BCN measure can accurately detect whether two nodes are similar, neutral or dissimilar to each other.

According to the results shown in Figure 5.5, the fact that the BCN measure can recover the predefined similarities indicate the proposed BCN measure to be an efficient and accurate measure for similarity quantifications in networks.

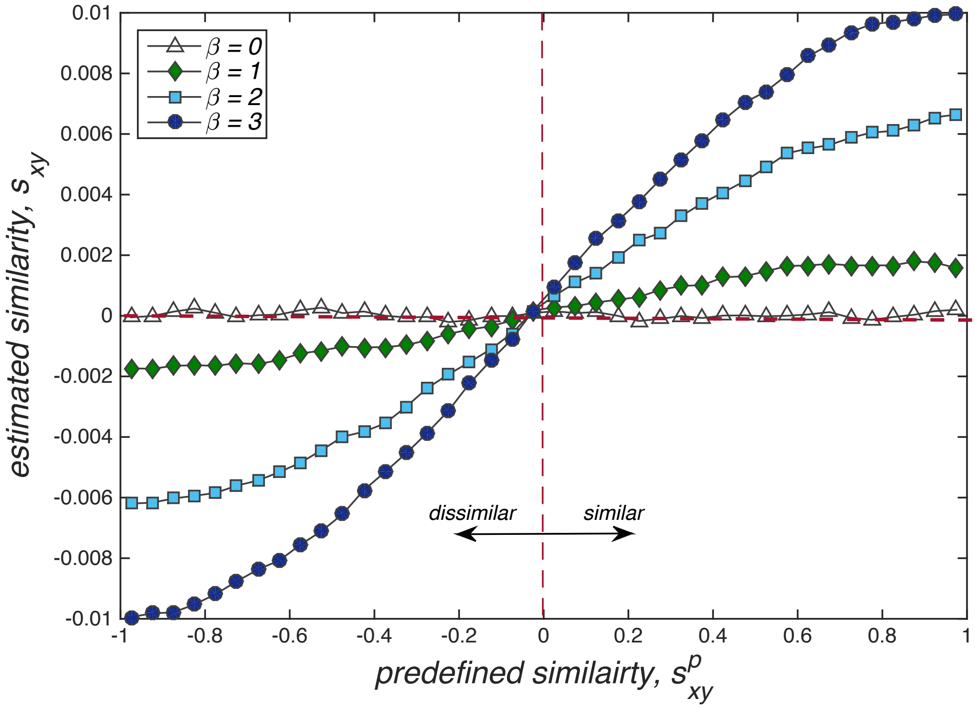


Figure 5.5 | Validation for the BCN similarity. The estimated similarity stands for the proposed BCN similarity, while the predefined similarity is given by the model according to Eq. (5.11). The vertical and horizontal dashed lines show the neutral case for the predefined and estimated similarity respectively.

5.2.2. Application to similarity intensity in networks

So far, the BCN measure has been proposed for the unipartite network, and thus cannot yet be applied to personalised recommendation. However, similarity measures may have significant applications in unipartite networks as well. Though it's a little bit off topic, here we explore one of the possible major applications of the proposed BCN measure in unipartite networks. Since we have defined a symmetrical similarity measure which can be used to detect whether two nodes are similar or dissimilar to each other in comparison to the random case, in this section we examine the connected nodes in a given network to explore whether and to what extent the links are established according to the similarity preference.

In a given network, for each link e with two nodes e_x and e_y on its ends, we examine the similarity between such two nodes. Note that, as e_x and e_y have already connected to each other, in the calculation of similarity, we exclude this link from the node degrees, leading the similarity to be

$$s_{e_x e_y} = \left|\Gamma_{e_x} \cap \Gamma_{e_y}\right| - \mathcal{H} \cdot \left(k_{e_x} - 1\right)\left(k_{e_y} - 1\right). \tag{5.13}$$

Accordingly, we define the *similarity intensity* of the network $\mathcal{S}$ as the average value of the similarity of every pair of connected nodes, which reads,

$$\mathcal{S} = \frac{1}{|\boldsymbol{E}|} \sum\nolimits_{e \in \boldsymbol{E}} s_{e_x e_y}, \tag{5.14}$$

where $\boldsymbol{E}$ is the full set of links in the network. Since the BCN similarity is symmetrical with positive values representing similar nodes and negative values representing dissimilar nodes, a positive value of $\mathcal{S}$ would suggest that the connected nodes share more common neighbours than expected which implies that the network is shaped by the similarity preference. On the other hand, a neutral value $\mathcal{S} = 0$ indicates that the formation of the network is irrelevant to the similarity. Additionally, larger values suggest stronger governance of similarity in the network evolution.

While the similarity intensity $\mathcal{S}$ can depict the similarity preference of a network, we also use the degree heterogeneity H to quantify the degree preference of the networks.

Similarity versus degree in artificial networks

We firstly analyse the similarity intensity and degree heterogeneity of artificial networks, including the ER network (Erdös & Rényi 1959; 1960), BA network (Barabási & Albert 1999), and Ring Lattice. In particular, we study the influence of link densities on such two features.

The ER random network (introduced in Section 2.1) takes a fixed probability for each node pair to establish a link. Since the links are established randomly, the ER networks have no similarity preference and thus one should expect a neutral similarity intensity $\mathcal{S} = 0$. As expected, the similarity intensity of ER networks is shown by Figure 5.6(a) to be neutral regardless of the network size and average degree. Additionally, following a Poisson degree distribution (Eq. 2.7), the ER random networks' degree heterogeneity is very close to the lower-limit $H = 1$ as shown in Figure 5.6 (b) especially for networks with dense links (higher average degree $\langle k \rangle$).

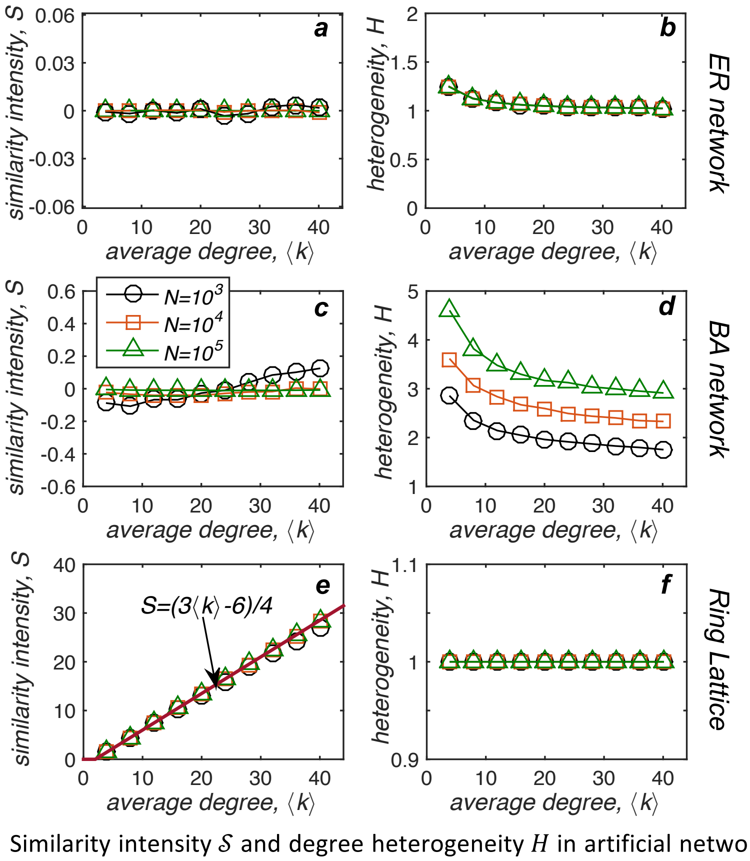


Figure 5.6 | Similarity intensity $\mathcal{S}$ and degree heterogeneity H in artificial networks. For ER and BA networks, the results for each size and average degree are averaged over 50 independent realisations. For the Ring Lattice, since it is a regular network, the results are based on one realisation for each size and average degree.

The BA network (introduced in Section 2.1) which introduces the degree preference to model the power-law degree distribution observed in real networks, has been considered as a standard heterogeneous network and thus has a relatively high degree heterogeneity. As shown in Fig. 5.6(d), the degree heterogeneity of BA networks is correlated with the network size and the average degree, but always takes value that is significantly larger than 1. On the other hand, since the links in BA networks are attached purely according to the degree preference, the similarity intensity $\mathcal{S}$ takes a neutral value (close to 0) similar to the ER random network as shown in Figure 5.6(c).

Different from the ER and BA networks, the Ring Lattice is a regular network which places N nodes evenly on a circle and lets each node connect to its $\langle k \rangle$ nearest neighbours. Therefore, every node has exactly the same degree and thus the degree heterogeneity is exactly $H = 1$ for ring lattice regardless of the size and average degree as shown in Figure 5.6(f). On the other hand, since the links are established according to the positions, the nodes near (similar) to each other will have many common neighbours leading to a high similarity intensity as shown in Figure 5.6(e). Assuming the nodes are numbered according to their positions, the neighbours of an arbitrary node i will be $\boldsymbol{V}_i = \{i - \frac{\langle k \rangle}{2}, \dots, i-1, i, i+1, \dots i + \frac{\langle k \rangle}{2}\}$. The number of common neighbours between i and j ($j \in \boldsymbol{V}_i$) can be given by $n_{ij} = \langle k \rangle - |j - i| - 1$. The average number of common neighbours for node pairs involving i is thus

$$\langle n_i \rangle = \frac{\sum_{j \in \boldsymbol{V}_i} (\langle k \rangle - |j - i| - 1)}{\langle k \rangle} = \frac{\langle k \rangle^2 - \langle k \rangle - 2 \sum_{m=1}^{\langle k \rangle/2} m}{\langle k \rangle} = \frac{3\langle k \rangle - 6}{4}. \quad (5.15)$$

Accordingly, we can theoretically have the similarity intensity of a ring lattice to be

$$\mathcal{S}^{ring} = \frac{3\langle k \rangle - 6}{4} - \mathcal{H} \cdot \langle k_x k_y \rangle. \quad (5.16)$$

Since in ring lattice, any $k_x = \langle k \rangle$, we have $\langle k_x k_y \rangle = \langle k \rangle^2$. We can thus rewrite Eq. (5.16) as

$$\mathcal{S}^{ring} = \frac{3\langle k \rangle - 6}{4} - \frac{\langle k \rangle^2}{N}(1 - \frac{1}{\langle k \rangle}). \quad (5.17)$$

For any ring lattice in which $N \gg \langle k \rangle^2$, we can approximately have $\mathcal{S}^{ring} = (3\langle k \rangle - 6)/4$. Therefore, as shown by Figure 5.6 (e), the similarity intensity of ring lattice is closely correlated with the average degree, but generally irrelevant to the network size.

Examination in empirical networks

We further examine the similarity intensity $\mathcal{S}$ and degree heterogeneity H of empirical unipartite networks. To explore the features of different networks, here we consider four classes of networks, including social networks, biological networks, infrastructure networks and animal networks, the statistics of which are reported in Table 5.1.

In the Coauthorship network (Leskovec *et al.* 2007), nodes are authors of academic papers and a link indicates that two authors have at least one joint publication. The Facebook (Viswanath *et al.* 2009), Yelp (data published by Yelp dataset challenge), Gowalla (Cho *et al.* 2011) and Flixster (Zafarani & Liu 2009) are social networking websites where users (nodes) can establish online friendships (links) with others. The Trust network (Papadopoulos *et al.* 2012) is based on an encryption program, entitled Pretty-Good-Privacy (PGP) where nodes are certificates and a link represents authorisation from the owner of a certificate to that of another. The Email network (Leskovec *et al.* 2009) describes the email exchanges (links) between employees (nodes) of the company entitled Enron. The Yeast (Stumpf *et al.* 2005) and PDZBase (Beuming *et al.* 2004) networks are the metabolic interactions (links) between proteins (nodes). For the Road networks of Pennsylvania (PA.) and California (CA.), a road is a link connecting intersections as nodes (Leskovec *et al.* 2009). For the power grid (Watts & Strogatz 1998), either a generator, a transformer or a substation is regarded as a node while the supply lines are regarded as links. The animal networks regard animals, i.e. dolphins (Lusseau *et al.* 2003), zebras (Sundaresan *et al.* 2007) and kangaroos (Grant 1973) respectively, as nodes and there will be a link connecting two individuals if they have at least one interaction during the observation. All these empirical networks are unipartite and considered as simple graphs, i.e. unweighted, undirected.

Table 5.1 | Statistics of networks applied in the similarity intensity study. In the table, N and M represent the number of nodes and links respectively; C is the clustering coefficient (Watts & Strogatz 1998) of the network; r is the degree assortativity coefficient (Newman 2002); H represents the degree heterogeneity, i.e. $H = \langle k^2 \rangle / \langle k \rangle^2$; and the $\mathcal{S}$ is the similarity intensity defined in this section.

Network Type	Network	N	M	$\langle k \rangle$	C	r	H	$\mathcal{S}$
Social	Coauthorship	18771	198050	21.1	0.63	0.45	3.09	19.65
	Facebook	63731	817035	25.64	0.22	0.42	3.43	12.36
	Trust (PGP)	10680	24316	4.55	0.26	0.42	4.14	6.58
	Email	36692	183831	10.02	0.49	0.13	13.97	7.1
	Yelp	174097	1288077	14.79	0.11	0.18	15.79	9.03
	Gowalla	196591	950327	9.66	0.23	0	31.71	3.41
	Flixster	2523386	7918801	6.27	0.08	0.11	35.07	2.73
Biological	Yeast	1846	2203	2.38	0.06	0.04	2.72	0.28
	PDZBase	212	242	2.28	0	0	2.33	-0.08
Infrastructure	Rooad (PA.)	1088092	1541898	2.83	0.04	0.26	1.12	0.13
	Power grid	4941	6594	2.66	0.08	0.18	1.45	0.29
	Road (CA.)	1965206	2776607	2.82	0.04	0.99	13.86	-2.35
Animal	Dolphin	61	159	5.16	0.26	0.24	1.32	1.17
	Zebra	27	111	8.22	0.87	0.81	1.33	3.51
	Kangaroo	17	91	10.7	0.82	0.11	1.13	1.88

For each of the empirical networks, we calculate its similarity intensity $\mathcal{S}$ and degree heterogeneity H and the results are reported in Table 5.1 and Figure 5.7. In order to explore whether the artificial network models (the ER, BA and Ring Lattice) can efficiently describe the features of empirical networks, we also simulate artificial networks according to the size and density of each empirical network and examine the similarity intensity $\mathcal{S}$ and degree heterogeneity H of the generated networks, which are shown in Table 5.2.

The biological networks are shown to have high degree heterogeneity and neutral similarity intensity which is very similar to the BA network. The infrastructure networks show different results on the degree heterogeneity that, while the road network in Pennsylvania as well as the power grid are less heterogeneous, the road network in California has extremely heterogeneous degrees. On the other hand, the similarity of infrastructure networks is neutrally or even negatively shaping the structure. Actually, the wiring patterns of such networks are constructed according to the geographical locations of the nodes (intersections in road network; generators, transformers or substations in power grid) which can be regard as location similarity. However, to achieve high efficiency, nodes in infrastructure networks, even geographically near to each other, would not share many common neighbours. Especially in road networks, the nodes (intersections) are mostly organised in squares resulting in second-order common neighbours rather than in triangles resulting in direct common neighbours. For the animal networks, the features are opposite to the biological networks and BA networks in terms of the degree heterogeneity and similarity intensity. Though with low degree heterogeneity, the similarity is shown to be playing a part in the interactions among animals. However, please be noted that the sizes of the studied animal networks are quite small which may cause uncertainties on their similarity intensities and degree heterogeneities.

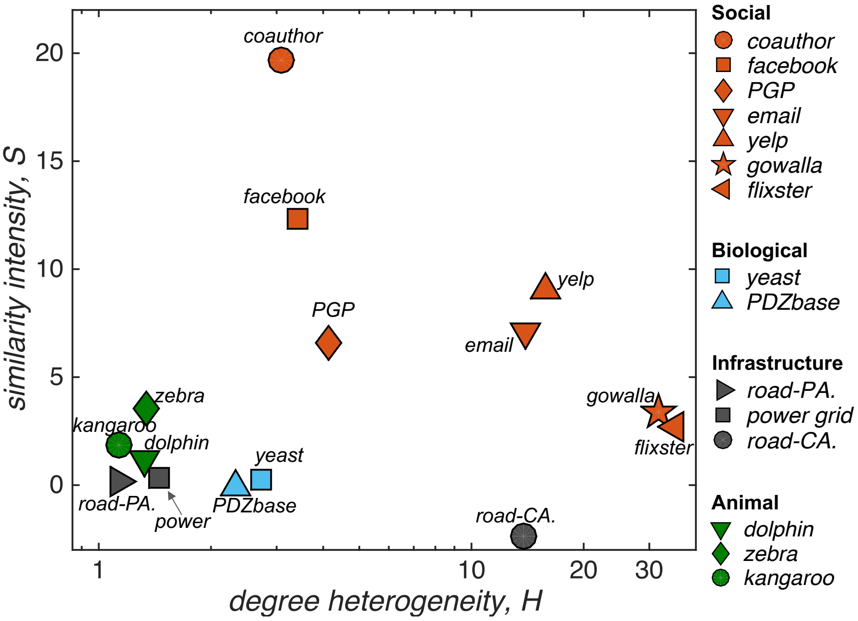


Figure 5.7 | The similarity intensity $\mathcal{S}$ versus degree heterogeneity H of fifteen empirical networks. While large H means the node degrees are very different (heterogeneous) from each other, the lower-limit $H = 1$ represents the case where each node has the same degree $k_v = \langle k \rangle, \forall v$. A large (positive) $\mathcal{S}$ indicates that the links tend to establish between similar nodes while small (negative) values suggest the links tend to connect dissimilar nodes.

Particularly, we address the social networks which are shown to be a special class of networks in terms of the degree heterogeneity and similarity intensity. The social networks have very heterogeneous degree distributions, normally more heterogeneous than the BA networks. While BA model can generate a power-law degree distribution with slope of 3, social networks in many cases may have much smaller slopes for the degree distribution leading to higher heterogeneities. A more interesting feature of social networks is the high similarity intensity. One can find that the similarity intensities of social networks are sometimes similar or even higher (Trust network) than the ring lattice. In other words, for social networks, each connected node pair shares much more common neighbours than the random case on average. Such result suggests the extremely strong governance of similarity in human interactions. Social networks with both high degree heterogeneity and high similarity intensity, stand alone as a special class of networks in comparison to others.

Table 5.2 | Comparison of degree heterogeneity and similarity intensity between empirical and artificial networks. For each empirical network, the artificial networks are generated according to its network size N and link density. For ER networks, the probability of each node pair connecting each other is set to be $p = 2M(N-1)/N$. For BA networks, we set $m_0 = 1 + 2M/N$ and $m = M/N$. As to the ring lattices, we let each node to connect $2M/N$ nearest neighbours.

		Degree Heterogeneity H				Similarity Intensity $\mathcal{S}$			
		Empirical	ER	BA	Ring	Empirical	ER	BA	Ring
Social	Coauthorship	3.09	1.02	2.46	1	19.65	0	0	29.91
	Facebook	3.43	1.02	2.71	1	12.36	0	0	35.96
	Trust (PGP)	4.14	1.12	2.96	1	6.58	0	-0.03	4.49
	Email	13.97	1.04	2.87	1	7.1	-0.01	-0.02	13.49
	Yelp	15.79	1.03	3.17	1	9.03	0	-0.01	19.49
	Gowalla	31.71	1.05	3.44	1	3.41	0	-0.01	11.99
	Flixster	35.07	1.11	4.93	1	2.73	0	-0.01	2.99
Biological	Yeast	2.72	1.08	3.03	1	0.28	0	-0.03	-0.01
	PDZBase	2.33	1.14	2.33	1	-0.08	0	-0.07	-0.01
Infrastructure	Rooad (PA.)	1.12	0.99	6.62	1	0.13	-0.01	-0.01	0
	Power grid	1.45	1.13	4.41	1	0.29	0	-0.03	-0.01
	Road (CA.)	13.86	1.13	8.19	1	-2.35	0	-0.01	0
Animal	Dolphin	1.32	1.05	2.24	1	1.17	0	-0.45	1.38
	Zebra	1.33	1.03	1.33	1	3.51	0	0.55	2.91
	Kangaroo	1.13	1.19	1.09	1	1.88	0	1.66	1.71

A number of artificial network models have been developed over the years, and many studies have been carried out based on these models to try to make implications for the understanding and control of the dynamics in real-world networks. But only if the network models can reveal the structural features of real networks, these theoretical studies could contribute to the knowledge of real-world systems. Here we show that the ER, BA and Ring Lattice cannot well describe the social networks in terms of the degree heterogeneity and similarity intensity simultaneously. Hence, more efforts shall be devoted to the development of network models with both high degree heterogeneity and similarity intensity. In addition, the examination of similarity intensity in this study provides a method to match the real networks with network models so that we can pick up the appropriate model according to the match to study with to make contributions to the target networks.

5.3. Balanced Common Neighbour Measure for Bipartite Network

So far, a BCN similarity measure for unipartite networks has been developed by comparing the empirical network to the random ones and removing the random-based common neighbours from the similarity. Considering that the present thesis focuses on the application in online recommendations, where the user-object bipartite network is normally the model, in this section we introduce the BCN measure to bipartite networks and apply it to the personalised recommendation.

5.3.1. Theoretical calculation

Similar to the calculation for the unipartite network, we argue the number of common neighbours for two objects α and β in a bipartite network consists of two components that one comes from random mechanism $n_{\alpha\beta}^{rand}$ and the other one comes from the similarity between them, $n_{\alpha\beta}^{sim}$. While the random component is completely popularity(degree)-correlated, the other one describes purely the similarity regardless of object popularities. To distinguish these two components is thus of significance for us to control the popularity tendency and the similarity tendency of the personalised recommendations to optimise the performances. To do so, we firstly assume the user-object bipartite network to be completely random and explore the expected number of common neighbours $n_{\alpha\beta}^{exp}$ for two arbitrary objects with given popularities as the estimation for the random component.

Considering a random user-object bipartite network with M users and N objects, we let T to be the total number of links between users and objects, i.e. $T = \sum_u k_u = \sum_o k_o$. The probability of a pair of randomly chosen user i and object α being connected with each other is therefore proportional to their degrees, which reads (Chuang & Lu 2002a; 2002b; Liu et al. 2013)

$$p(i \leftrightarrow \alpha) = \frac{k_i k_\alpha}{T}. \tag{5.18}$$

Accordingly, the probability of a user i being a common neighbour for objects α and β can be written as

$$p(i \leftrightarrow \alpha, \beta) = \frac{k_i(k_i - 1)}{T^2} k_\alpha k_\beta. \tag{5.19}$$

Therefore, every user has a probability to be a common neighbour for the objects α and β. Considering all the possible users can thus give us the expected number of common neighbours between α and β,

$$n_{\alpha\beta}^{exp} = \sum_u p(u \leftrightarrow \alpha, \beta) = \frac{\sum_u k_u(k_u - 1)}{T^2} \cdot k_\alpha k_\beta. \tag{5.20}$$

Actually, such derivation can also be obtained by considering this process as the hypergeometric distribution. Assuming an object β has randomly connected to k_β users, and we let T_β to be the total number of the remaining half-links originating from these k_β users. For each half-link of the object β, its probability to connect to a user u can be given by k_u/T, leading the expected degree of the user at the other end of β's half-link to be $\sum_u k_u^2/T$. Accordingly, we have

$$T_\beta = k_\beta \cdot (\sum_u \frac{k_u^2}{T} - 1). \tag{5.21}$$

The number of common neighbours between any other object α with β is thus determined by the process where α select k_α out of $T - k_\beta$ half-links. When one of the T_β half-links is selected, one common neighbour is generated for α and β. Therefore, the number of common neighbours between α and β can be described by a hypergeometric distribution $H(n_{\alpha\beta}; k_\alpha, T_\beta, T - k_\beta)$. For any bipartite network which is sparse enough, i.e. $T \gg k_o, \forall o$, we can approximately have

$H(n_{\alpha\beta};k_\alpha,T_\beta,T)$ to describe the distribution of $n_{\alpha\beta}$. The mean of such hypergeometric distribution is thus the expected number of the common neighbours between two random objects, which reads,

$$\begin{aligned} n_{\alpha\beta}^{exp} &= \langle H(n_{\alpha\beta};k_\alpha,T_\beta,T)\rangle = \frac{k_\alpha T_\beta}{T} = \frac{k_\alpha}{T}\cdot k_\beta\left(\sum_u \frac{k_u^2}{T} - 1\right) \\ &= k_\alpha k_\beta \cdot \frac{\sum_u k_u^2 - k_u}{T^2}. \end{aligned} \tag{5.22}$$

As a result, we have the exactly same solution for the expected common neighbours using the hypergeometric distribution with that using the network analysis (Eq. 5.20). In addition, since T is the total links between users and objects, we have $T = \sum_u k_u = M\langle k_u\rangle$. The expression for the expected number of common neighbours (Eq. 5.20 and Eq. 5.22) can thus be rewritten as

$$n_{\alpha\beta}^{exp} = \frac{\langle k_u^2\rangle - \langle k_u\rangle}{M\langle k_u\rangle^2}\cdot k_\alpha k_\beta. \tag{5.23}$$

Note that, such consideration does not exclude the case of multi-links, leading the calculated value slightly higher than the actual theoretical value for number of common neighbours, especially for these objects with very large degrees. However, the expression is valid for sparse limit or the limit of $M \to \infty;\ N \to \infty$.

As shown by Eq. 5.23, the expected number of common neighbours between two objects α and β is linearly correlated with the product of their popularities $k_\alpha k_\beta$, which is very similar to that for the unipartite networks as have been discussed in the Section 5.2. Similarly, we further use $\mathcal{H}$ to denote the parameter before the product of the popularities, which reads

$$\mathcal{H} = \frac{\langle k_u^2\rangle - \langle k_u\rangle}{M\langle k_u\rangle^2} = \frac{1}{M}\left(\frac{\langle k_u^2\rangle}{\langle k_u\rangle^2} - \frac{1}{\langle k_u\rangle}\right). \tag{5.24}$$

As a consequence, the parameter $\mathcal{H}$ describes the heterogeneity of the user degree distribution.

With the expected number of common neighbours as the estimation for the random component, one can compare the actual and expected number by taking either ratios or differences to get the similarity measure. Normally, real user-object systems are extremely sparse where most object pairs would have no common neighbours at all. In order to make these object pairs distinguishable from each other, we define the similarity between two objects α and β by taking the difference as $n_{\alpha\beta}^{sim} = n_{\alpha\beta}^{CN} - n_{\alpha\beta}^{exp} = |\Gamma_\alpha \cap \Gamma_\beta| - \mathcal{H}\cdot k_\alpha k_\beta$. This expression can be used as an object similarity measure and theoretically there would be no popularity bias (or degree-bias) for the quantified object similarities.

5.3.2. Application to personalised recommendation

Considering the popularity of objects may be an influential factor in recommender systems, here we explore to what extent should the popularity bias be balanced to achieve good performance. By introducing a free parameter λ, we define a new similarity measure for bipartite networks, namely the Balanced Common Neighbour (BCN) measure as

$$s_{\alpha\beta}^{BCN} = |\Gamma_\alpha \cap \Gamma_\beta| - \lambda \cdot \mathcal{H} k_\alpha k_\beta. \tag{5.25}$$

One may find from the expression that $\lambda = 0$ gives the standard CN measure. As the parameter λ increases, more and more popularity bias would be removed, until and $\lambda = 1$ gives the theoretical similarity with no popularity bias.

We carry out recommendation experiments based on three empirical datasets namely the MovieLens, Netflix and Last.FM (see Chapter 3 for descriptions of these datasets). For all the recommendation experiments here, we randomly divide 20% of the links into the testing set for each dataset, and take a recommendation list length $L = 20$. To Evaluate the performance of the recommendations, we use the precision $P(L)$ and recall $R(L)$ as the accuracy metrics, and the personalisation $S(L)$ and novelty $N(L)$ as the diversity metrics (see Chapter 2, Section 2.3.2 for the definition of these evaluation metrics). Furthermore, all of the results on the recommendation performance are averaged over 100 independent experiments.

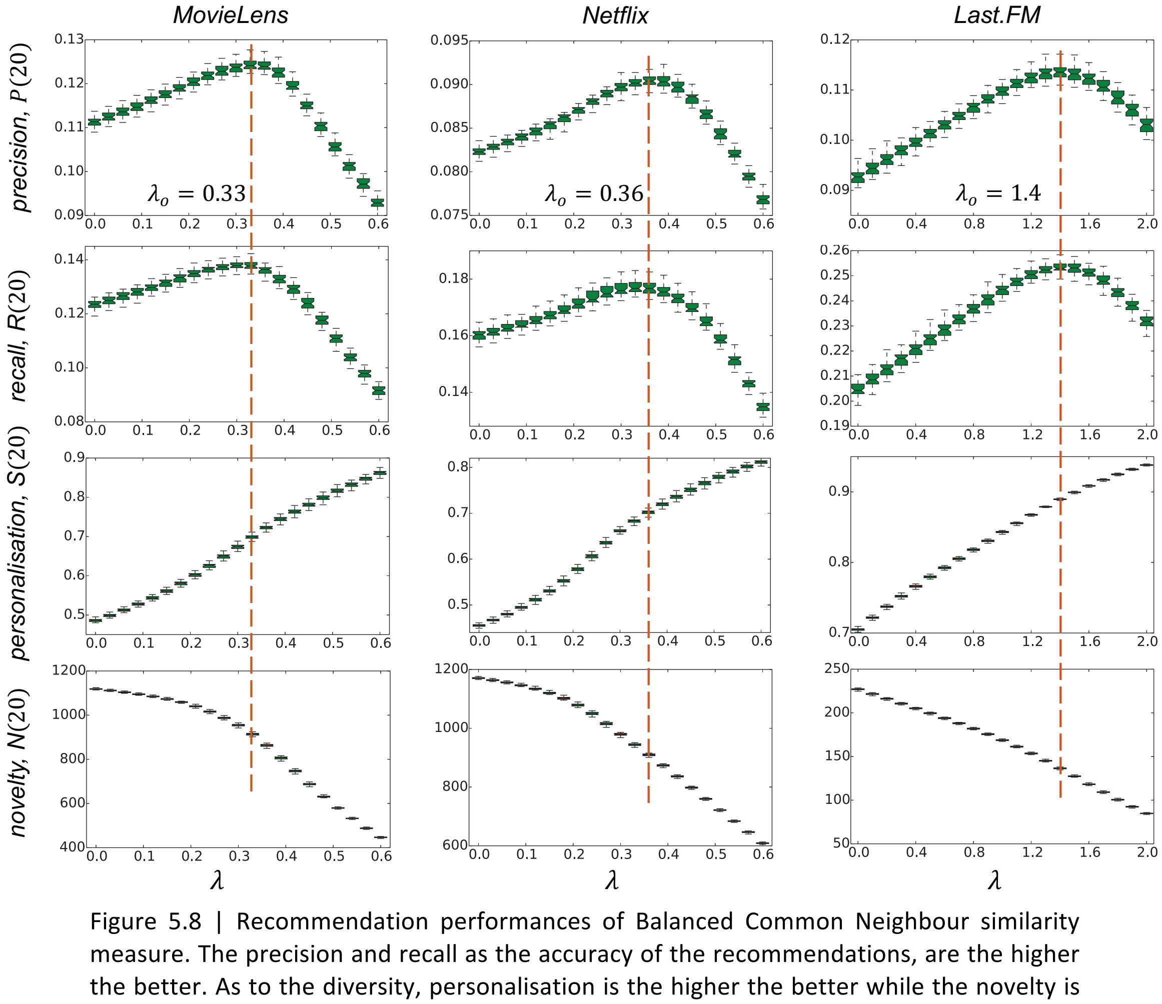


Figure 5.8 | Recommendation performances of Balanced Common Neighbour similarity measure. The precision and recall as the accuracy of the recommendations, are the higher the better. As to the diversity, personalisation is the higher the better while the novelty is the lower the better. Each column of subplots is the result for one dataset, where the red dashed line represents the optimised parameter λ_o maximising the precision. The results for each dataset are based on 100 independent recommendation experiments with random data partitions.

We start with exploring that to what extent should the popularity bias of object similarity should be balanced by tuning λ to achieve better recommendation performances in terms of accuracy and diversity. The results are shown in Figure 5.8.

As has been discussed earlier, $\lambda = 0$ gives us the standard CN measure, which can result in relatively good accuracy but poor diversity. The reason is that CN recommends generally the extremely popular objects, which have better chance to suite more users' common interests. However, every user's recommendation list would be dominated by the same popular objects, leading to poor personalisation and novelty. When gradually increasing the parameter λ, the BCN measure removes more and more randomly-generated common neighbours as suggested by Eq. 5.25. Accordingly, the average popularity of the recommended objects, i.e. the novelty $N(20)$, decreases with the increase of λ, since those not-so-popular objects are evaluated as more similar to others. The recommendations thus become more and more personalised. Therefore, Figure 5.8 suggests that to remove the random component of the common neighbours can remarkably improve the diversity of the recommendations. On the other hand, the recommendation accuracy will also be largely influenced by the balancing of the popularity bias. If slightly balance the popularity bias, the recommendations are shown to be more accurate. However, when applying a large λ, which means to totally remove the popularity bias ($\lambda = 1$) or even reverse the bias ($\lambda > 1$), the recommendation lists would be dominated by only unpopular objects, leading to poor accuracy. Here we take an optimised value λ_0 maximising the precision for each dataset. With the optimised value λ_o, both the accuracy and diversity of the the recommendations can be significantly improved in comparison to the algorithm applying the original CN measure. The optimised values λ_o are 0.33, 0.36 and 1.4 for MovieLens, Netflix and Last.FM datasets respectively.

Table 5.3 | Numerical results of recommendation with BCN measure.

	$P(20)$	$R(20)$	$S(20)$	$N(20)$
MovieLens				
$\lambda = 0$	0.111	0.123	0.486	1118.1
$\lambda_o = 0.33$	0.124	0.138	0.699	913.7
Improvement	11.64%	11.68%	43.72%	18.29%
Netflix				
$\lambda = 0$	0.082	0.160	0.455	1171.0
$\lambda_o = 0.36$	0.090	0.177	0.702	909.4
Improvement	9.95%	10.46%	54.37%	22.34%
Last.FM				
$\lambda = 0$	0.093	0.205	0.705	227.0
$\lambda_o = 1.4$	0.114	0.254	0.889	136.5
Improvement	22.52%	23.79%	26.12%	39.89%

As shown in Table 5.3, the accuracy metrics (precision $P(20)$ and recall $R(20)$) are improved about 10% for the MovieLens and Netflix datasets and more than 20% for the Last.FM dataset. The diversity performances are improved even more significantly, especially for the personalisation $S(20)$ which has been improved for 43.72%, 54.37% and 26.12% for MovieLens, Netflix and Last.FM datasets respectively.

We further compare the recommendation performances of the optimised BCN measure with that of classical similarity measures including the CN, LHN, HP, AA, RA, MD, and HC measures which have been introduced in Chapter 2. As shown in Table 5.4, the recommendation accuracies (precision $P(20)$ and recall $R(20)$) of the BCN measure are comparable to the MD measure which is normally considered as one of the most accurate algorithms. The BCN measure is more accurate than many of the accuracy-based measures such as the CN, AA, RA. In terms of the diversity, the BCN measure is comparable to the HC measure, which is designed to achieve good diversity. While some diversity-based measures such as LHN, HP, and HC sacrifice the accuracy a lot to focus on the extreme unpopular recommendations, the proposed BCN measure can achieve good accuracy and diversity simultaneously with reasonable preference on the popularity of recommended objects.

Table 5.4 | Comparison of recommendation performances of BCN measure to benchmarks. The results of BCN measure are based on the optimised value λ_o, i.e. 0.33, 0.36 and 1.4 for the MovieLens, Netflix and Last.FM respectively.

	MovieLens				**Netflix**				**Last.FM**			
	$P(20)$	$R(20)$	$S(20)$	$N(20)$	$P(20)$	$R(20)$	$S(20)$	$N(20)$	$P(20)$	$R(20)$	$S(20)$	$N(20)$
CN	0.111	0.123	0.486	1118.1	0.082	0.160	0.455	1171.0	0.093	0.205	0.705	227.0
LHN	0.001	0.001	0.469	1.1	0.001	0.001	0.922	1.4	0.017	0.044	0.994	2.1
HP	0.001	0.001	0.497	1.8	0.001	0.002	0.848	8.7	0.068	0.154	0.948	78.3
AA	0.114	0.130	0.500	1120.7	0.085	0.172	0.479	1175.6	0.093	0.203	0.707	227.2
RA	0.117	0.140	0.542	1114.2	0.089	0.199	0.542	1160.4	0.093	0.205	0.709	226.5
MD	0.121	0.145	0.560	1106.1	0.092	0.207	0.561	1144.9	0.115	0.255	0.794	196.8
HC	0.031	0.022	0.852	49.6	0.001	0.001	0.913	1.52	0.015	0.040	0.977	1.8
BCN	0.124	0.138	0.699	913.7	0.090	0.177	0.702	909.4	0.114	0.254	0.889	136.5

5.4. Summary

By comparing a given network with the random network, this Chapter develops a Balanced Common Neighbour similarity measure for the quantification of node similarities in networks especially for the object similarities in bipartite networks. Applying the BCN algorithm on recommendations, the diversity of recommendations can be largely improved. However, the accuracy will be sacrificed if removing all the random-driven common neighbours ($\lambda = 1$). To achieve good accuracy and diversity simultaneously, one should optimise the similarity quantification to remove the random-driven common neighbours to only a certain degree. The optimised value λ_o for the MovieLens and Netflix are 0.33 and 0.36 respectively, which are less than the theoretical value $\lambda = 1$. On the other hand, the optimised value for the Last.FM is 1.4 which is larger than the theoretical value. Such difference between the Last.FM dataset with others may be raised from the different object degree distributions, and the extremely even user degree distribution (see Figure 3.2). Without hub users, objects are less likely to share many common neighbours, leading to the possibility that we may need a relatively larger value of λ to balance the popularity bias. With the optimised value, the accuracy and diversity of the recommendations can be simultaneously improved and are comparable to accuracy-based algorithms (such as MD) and diversity-based algorithms (such as HC) respectively.

This Chapter explores RQ2 of the thesis (how can we develop a balanced similarity measure by comparing the empirical network with random ones?). The answer lies in the theoretical calculation for the expected number of common neighbours in random networks, which has been fully addressed in Section 5.2.1 and 5.3.1 for unipartite and bipartite networks respectively. In addition, the developed similarity measure has been applied to empirical scenarios to explore the similarity intensity of networks and the personalised recommendations.

Chapter 6. Navigation Accuracy of Recommendation Networks

In the previous chapters, the application of similarity measures in personalised recommendation has been discussed. In the online world, there is yet another commonly existed recommender system, which is normally referred as recommendation networks. From this chapter onwards, including Chapter 6 and Chapter 7, we will focus on the evaluation of recommendation networks and the application of similarity measures in such systems.

The world-wide web, as the major channel where people access information from, is naturally a complex network enabled by hyperlinks, and thereby its topology attracted a significant amount of scientific attentions (Albert *et al.* 1999; Barabási *et al.* 2000; Kleinberg & Lawrence 2001; Henzinger & Lawrence 2004; Fortunato *et al.* 2006b; Cheng *et al.* 2009). The connected webpages enable users to surf on such network to explore massive amount of information (Bilenko & White 2008). Particularly, in many online content-browsing systems, such as e-commerce websites and movie websites, where each object (product or movie etc.) is displayed in a dedicated webpage, the object similarities are evaluated and hyperlinks are established accordingly as recommendations. When browsing a specific object in such website, there is normally a recommendation list consisting of objects that the system considers to be similar with the current one. This kind of similar object list can be found in a wide range of websites, such as the "Customers who bought this item also bought" list in Amazon, and the "People also viewed" list in Yelp. As a consequence, these recommendation lists on various webpages enable the overwhelming online objects to connect to each other via hyperlinks as a massive recommendation network.

Regarding the topology, the recommendation networks are found with power-law distributed in-degrees and a small average shortest path length (Cano *et al.* 2006; Buldú *et al.* 2007). From the perspective of the commercial values (Oestreicher-Singer *et al.* 2013; Goldenberg *et al.* 2012), product sales and demands are argued to be closely related to the products' position in the network (Leem & Chun 2014; Oestreicher-Singer & Sundararajan 2012b) and a long-tail effect can be observed (Oestreicher-Singer & Sundararajan 2012a). In addition, the hyperlinks connecting different objects may be able to spread the web traffic, and as a result, the demands of objects are argued to be contagions in the recommendation networks (Carmi et al. 2009; 2017).

The major function of recommendation networks is, in nature, to serve users by navigating them to what they may be interested in. While the amount of online objects are too much for anyone to even skim over and the search engine normally provides only biased results (Fortunato *et al.* 2006b; De Corniere & Taylor 2014), the recommendation network can be a good way for users to access online information and correct the initial searching bias. By surfing on such network following the recommendation hyperlinks, users are thereby enabled to explore massive relevant objects to try to find interesting ones. However, it is still an open question that, how accurately can recommendation networks navigate users to find what they interested in (RQ3 of this thesis)? Additionally, while the structure of recommendation networks constructed by the website is totally determined by how the similarities of objects are evaluated, and there are quite a number of methods of quantifying the similarity according to the users' co-accessing pattern, how would different similarity measures influence the accuracy of recommendation networks is a question that at the central for the understanding and design of such systems.

By developing an evaluation framework based self-avoiding random walks, this Chapter shows that the navigation accuracy of recommendation networks constructed by similarity measures is very limited. Though CN, AA and RA measures are able to generate relatively high accuracies, about 7% retrieval accuracy, they fail to guarantee the retrieval for niche objects. In addition, we discover that the length of recommendation list is an important factor for the accuracy of recommendation networks. The optimal length should be relatively short, about 2-6 objects, in order to gain best retrieval accuracy.

The rest of this chapter is organised as follows: Section 6.1 theorises the construction of recommendation networks as the bipartite projection using object similarity measures and discusses the topology of the constructed networks; Section 6.2 models the user surfing behaviour as a self-avoiding random walk process and thereby explores the navigation accuracy of the recommendation networks; Section 6.3 discusses the influence of the recommendation list length on the accuracy of navigating users; Section 6.4 empirically studies the book recommendation network collected from Amazon; and Section 6.5 gives a summary for this Chapter.

6.1. Construction of Recommendation Networks

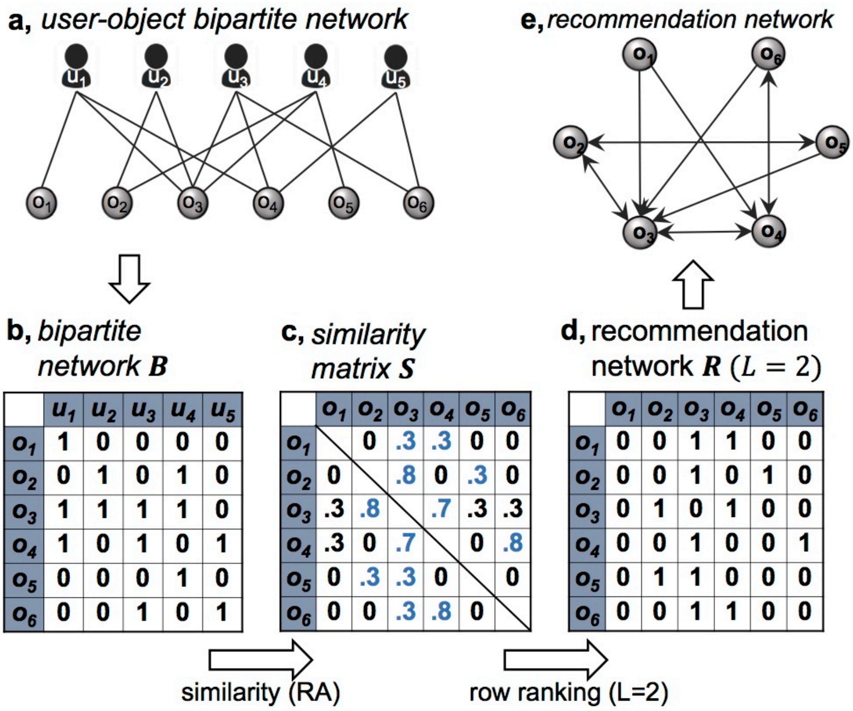
6.1.1. Top-L projection for the user-object bipartite networks

A recommendation network can be constructed by connecting similar objects with directed hyperlinks. The quantification of object similarities is thus at the central to the construction as well as the performances of recommendation networks.

There are a number of well-developed methods in network science to quantify the object similarities according to how are they connected by the same users, as has been discussed in

Chapter 2, section 2.2. Basically, the more common neighbours two objects share, the more similar they would be considered. In this Chapter, to explore the performance of recommendation networks constructed by different similarity measures, we apply eight such measures, including the CN, SAL, SOR, HPI, LHN, AA, RA and HC measure (for detailed definitions, please refer to section 2.2).

Consider a user-object bipartite network with a set of M users $\mathbf{U} = \{u_1, u_2, \dots, u_M\}$ and a set of N objects $\mathbf{O} = \{o_1, o_2, \dots, o_N\}$ (Figure 6.1a), and the links can thus be described by an adjacency matrix $\mathbf{B} = \{b_{uo}\}_{M \times N}$ where $b_{uo} = 1$ if there is a link between user u and object o and $b_{uo} = 0$ otherwise (Figure 6.1b). Applying an arbitrary similarity measure, the similarity between every pair of objects can be calculated and thus the similarity matrix can be obtained that $\mathbf{S} = \{s_{\alpha\beta}\}_{N \times N}$, as shown in Figure 6.1c.



	u_1	u_2	u_3	u_4	u_5
o_1	1	0	0	0	0
o_2	0	1	0	1	0
o_3	1	1	1	1	0
o_4	1	0	1	0	1
o_5	0	0	0	1	0
o_6	0	0	1	0	1

	o_1	o_2	o_3	o_4	o_5	o_6
o_1		0	.3	.3	0	0
o_2	0		.8	0	.3	0
o_3	.3	.8		.7	.3	.3
o_4	.3	0	.7		0	.8
o_5	0	.3	.3	0		0
o_6	0	0	.3	.8	0	

	o_1	o_2	o_3	o_4	o_5	o_6
o_1	0	0	1	1	0	0
o_2	0	0	1	0	1	0
o_3	0	1	0	1	0	0
o_4	0	0	1	0	0	1
o_5	0	1	1	0	0	0
o_6	0	0	1	1	0	0

Figure 6.1 | A toy example of constructing the recommendation network. (a) A user-object bipartite network with five users (top nodes) and six objects (bottom nodes), where the links represent the interactions. (b) The adjacency matrix for the bipartite network $\mathbf{B} = \{b_{uo}\}_{5\times6}$ where $b_{uo} = 1$ if there is a link between user u and object o and 0 otherwise. (c) The object similarity matrix $\mathbf{S} = \{s_{\alpha\beta}\}_{6\times6}$ which can be calculated by applying any similarity measure. In this example, we apply the Resource Allocation (RA) measure. (d) By ranking the similarities, and only keeping L highest values as 1 in each row of the matrix $\mathbf{S}$, the adjacency matrix for recommendation network $\mathbf{R}$ is obtained. (e) The recommendation network can thus be constructed according to the adjacency matrix $\mathbf{R}$, leading to a directed network with the same out-degree for each node, i.e. $k_o^{out} = L, \forall o$.

Though each object can be similar to quite a lot of others, most practical systems only recommend a fixed number of similar objects for each one considering the limited space on the webpages, rather than connecting every pair of objects sharing at common neighbours (Buldú *et al.* 2007). In a recommendation network, each object thus only connects to L others, to which it has the highest similarities. Therefore, the object similarity matrix $\mathbf{S}$ can be transferred to an adjacency matrix for the recommendation network $\mathbf{R}$ by ranking each row and keeping only the L highest values as 1 and others as 0, as shown by Figure 6.1d. In other words, in the recommendation network adjacency matrix $\mathbf{R} = \{r_{\alpha\beta}\}_{N\times N}$, we let

$$r_{\alpha\beta} = \begin{cases} 1, if\ Rank(\beta,\alpha) \leq L \\ 0, if\ Rank(\beta,\alpha) > L' \end{cases} \tag{6.1}$$

where $Rank(\beta,\alpha)$ is the rank of similarity $s_{\alpha\beta}$ in the row $\{s_{\alpha o_1}, s_{\alpha o_2}, \dots, s_{\alpha o_N}\}$ from high to low. Accordingly, the recommendation network (Figure 6.1e) can be constructed to be a directed network with a fixed out-degree for every object, i.e. $k_o^{out} = L, \forall o$.

Here we have described the construction of recommendation networks as the ranking process for the similarity matrix. Actually, it can also be regarded as the projection of bipartite networks (Zhou *et al.* 2007; Zweig & Kaufmann 2011), which connects every pair of objects who are accessed by at least one same user. In this way, every object would connect to a lot of others, resulting a very dense recommendation network, while the space of recommendation lists should be limited. Therefore, a key question is how to weight the connections so that one can rank them and keep only the top-L connections. This is where the similarity measures can be applied. By weighting each projected connection with an arbitrary similarity measure, one can keep only L outgoing links for each source object. As a consequence, the construction of recommendation networks can be regarded as a top-L projection of the user-object bipartite network.

6.1.2. Topological Structure of Recommendation Networks

Since it appears as networks, the recommendation network's degree distribution may be the most important feature determining its performances. Here in this section, we examine the degree distribution of the constructed recommendation networks.

Three datasets, including the user-book interactions from Amazon, the user-business interactions from Yelp and the user-product interactions from Epinions are applied. All three data sets can be modelled as user-object bipartite networks, and the descriptions can be found in Chapter 3. Note that, there are two Amazon datasets introduced in this thesis: one is the recommendation network collected directly from the website; the other one is the user-book bipartite network. In this section and also sections 6.2 and 6.3, we study the recommendation networks projected from the latter dataset, i.e. the bipartite network; and in section 6.4, we will empirically examine the recommendation network constructed by Amazon.

We apply eight similarity measures, including the CN, SAL, SOR, HPI, LHN, AA, RA and HC measure, to calculate the object similarity matrix for each of the three data sets, and therefore construct recommendation networks with recommendation list length L.

To set the baseline of the performances, we also construct random recommendation networks, where each object connects to L others chosen from the whole population uniformly at random.

While the out-degree is fixed for every object, i.e. $k_o^{out} = L, \forall o$ according to the mechanism of the construction method, the in-degree varies. In a recommendation network, an object's in-degree k^{in} represents its frequency of occurrences in others' recommendation list, and thus well links to its visibility for users. However, as shown by Figure 6.2, the in-degrees of the constructed recommendation networks have very heterogeneous distributions. For Amazon and Yelp system, all the measures except HC have power-law in-degree distributions with slopes of -2.7 approximately. The CN, AA and RA measures in Epinions system also show the power-law distributions. Such in-degree distributions suggest that there are a few objects frequently show up in others' recommendation list, while most others barely get recommended. The HC measure (also the SAL, SOR, HPI and LHN measures for Epinions dataset), on the other hand, can result in an exponential-like in-degree distribution which means the objects have relatively more equalised chances to be recommended and visited. As the baseline, the random recommendation networks connecting objects randomly, have normal (Gaussian) in-degree distributions, i.e. the in-degrees are extremely evenly distributed.

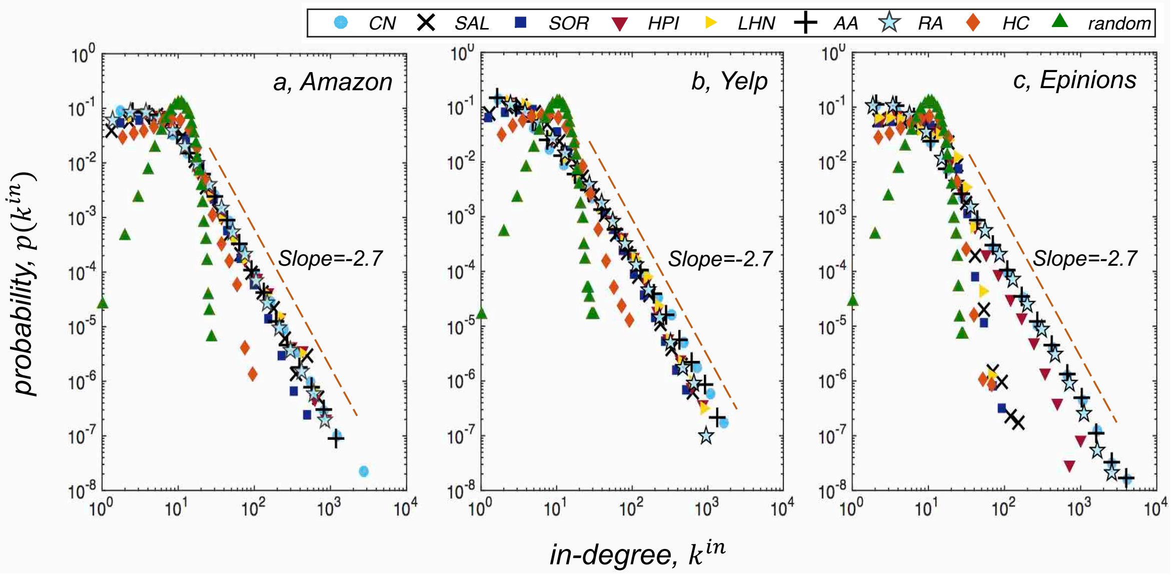


Figure 6.2 | In-degree distributions of recommendation networks. The plots are in log-log scale, and in each subplot the straight dashed line marks the slope of -2.7 indicating the corresponding distributions have an exponent $\gamma = 2.7$ for the power-law distribution. All the recommendation networks are constructed with a recommendation list length of $L = 10$.

The different behaviour of the similarity measures may be caused by the apparent popularity bias that the estimated similarities are highly correlated with the object popularity (as has been discussed in Chapter 5), either positively (the CN, AA, RA and HPI measures) or negatively (the SAL, SOR, LHN and HC measures). Such popularity bias will result in the fact that the links in recommendation networks will concentrate on those either every popular or very unpopular objects leading to the inequality of in-degree distribution.

6.2. Self-Avoiding Random Walks and Navigation Accuracy

6.2.1. Information retrieval experiments based on random walks

While the recommendation networks enable the numerous objects to connect to each other with hyperlinks, the user behaviour of browsing objects from one to another following the hyperlinks can thus be naturally modelled as the random walk (Masuda *et al.* 2017) on the recommendation networks. Considering that the users may not be interested in objects that have already been visited, we adopt the self-avoiding random walk model to describe the users' surfing behaviour.

Different from the classical random walk on network, users in self-avoiding random walk will not revisit any nodes. As shown in Figure 6.3, we assume a user to start walking from a random object at the initial time $t = 0$. At each following step, we let the user randomly move from the current node to one of its out-going nodes (another object that in the current one's recommendation list) which has not been visited in previous steps, following the hyperlink. Accordingly, each random walk will continue for a limited number of steps until the walker comes to a node whose out-going nodes have all been visited.

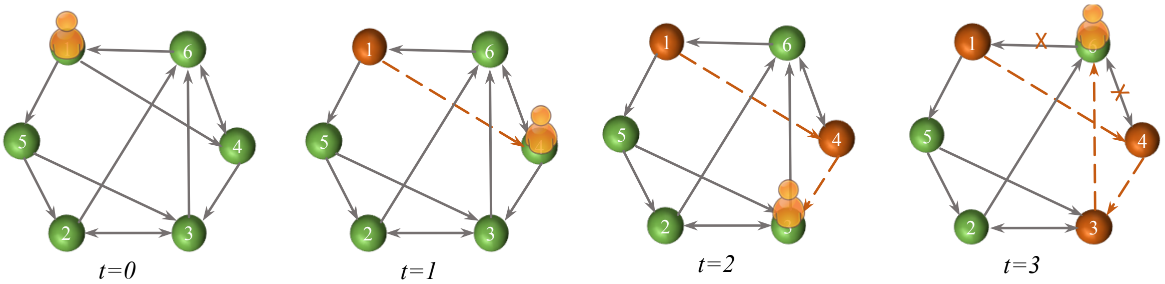


Figure 6.3 | An example of self-avoiding random walk on a toy recommendation network with six objects. As we assume the walker won't revisit any node, the random walk ends at $t = 3$ when all the out-going nodes have been visited in previous steps.

In the random walk process, an apparent question is that, are these objects visited by a random walker really of his/her interests? To examine the accuracy of the recommendation network navigation, we randomly divide the users into a training group and a testing group, for each of the three applied bipartite networks. The training group consisting of 90% of the users is used to construct the recommendation networks via the top-L projection. For each of the remaining 10% testing users, we assume that the objects they have selected in the data can well represent their interests, and examine how many of their historical selections can be retrieved by randomly walking on the constructed recommendation networks. For a target user u, who selected n_u objects in the data, we let him/her to perform self-avoiding random walks starting from one of his/her n_u selected objects. Suppose at step t of the random walk, $h_u(t)$ of the $n_u - 1$ remaining objects have been visited, we then have the retrieval percentage as

$$\gamma_u(t) = h_u(t)/(n_u - 1), \tag{6.2}$$

Accordingly, we use the retrieval percentage $\gamma(t)$ averaged over all testing users as the indicator for the accuracy of the recommendation networks. Apparently, higher values of retrieval percentage $\gamma(t)$ mean higher accuracy.

Note that, in the experiments, we carry out 10 independent experiments for each possible starting object for every user. And the recommendation list length of the constructed networks is $L = 10$.

As shown by Figure 6.4, limited amount of objects can be retrieved by letting users to walk on the recommendation networks. Generally, the retrieval percentages are less than 9% even for a random walk process of $t = 10^3$ steps. Especially, while eight similarity measures are studied, only the CN, AA and RA measures are shown to be able to construct recommendation networks with reasonable accuracies. The SAL, SOR, HPI, LHN, and HC measures generally have retrieval percentages less than 2%, being similar to that of the random recommendation networks. Having similar retrieval percentage with the random recommendation networks means these measures can mine the similarity between objects no better than random recommendations.

Actually, one should not expect users to keep walking on recommendation networks for too long time. Once realised that following the navigation cannot quickly locate interesting objects, the users may dropout or start from searching anew. Accordingly, a more reliable way to check the accuracy of such random walk, is the short-term accuracy. Taking $t = 10$ as an example for the short-term accuracy, the most accurate measure retrieves only 3% (CN), 3.2% (CN) and 1.9% (AA) objects for the Amazon, Yelp and Epinions dataset respectively if letting users to surf on the recommendation networks for 10 steps.

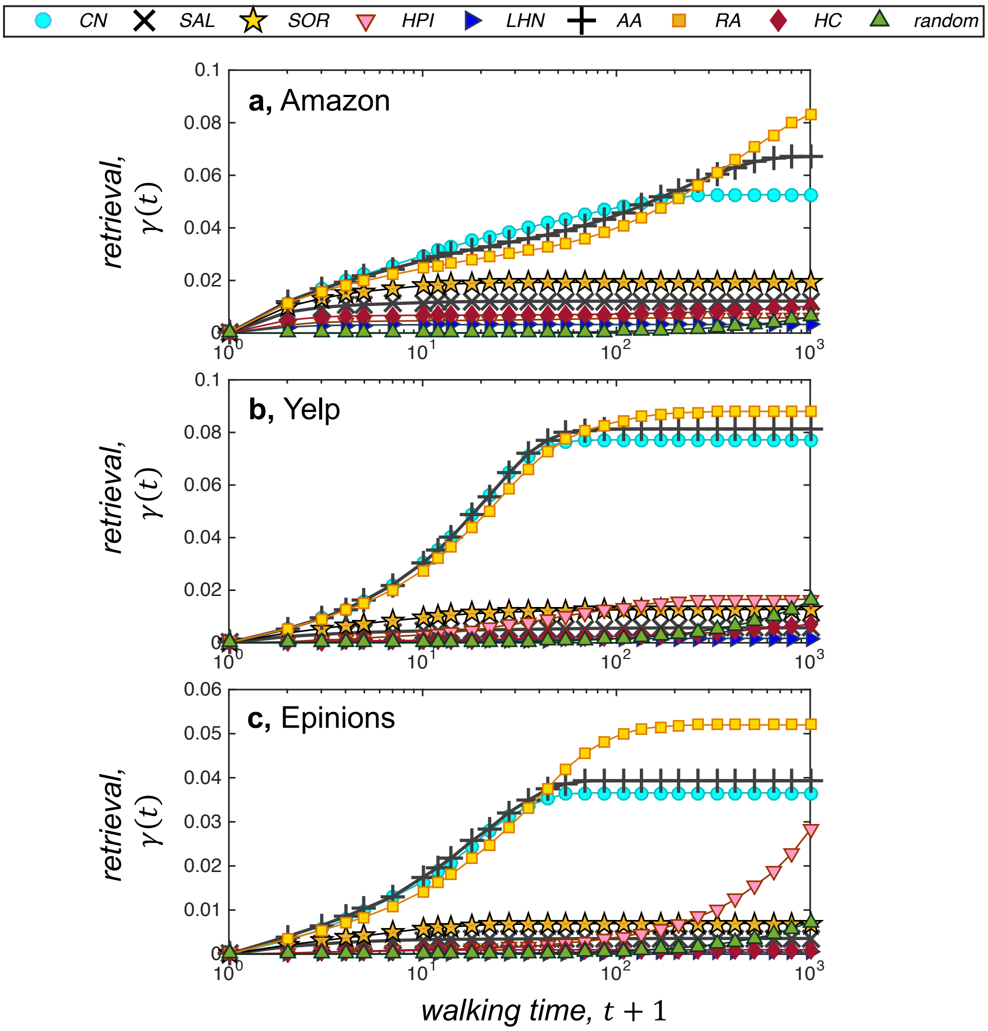


Figure 6.4 | The retrieval percentage $\gamma(t)$ of the random walk experiments versus walking steps for Amazon, Yelp and Epinions respectively.

6.2.2. Retrieval for niche versus popular objects

For the random walk on the recommendation networks, one may also be interested in the question that which objects are the users visiting. We measure how many users a specific object has been accessed by, i.e. the bipartite degree of the object k_o, as its popularity, and explore the popularity of the visited objects at each step t of the random walk, denoting with $k(t)$. Instead of letting testing users start random walk from their historical objects, we simulate $10N$ random walks starting from random objects for each recommendation network.

For the random recommendation networks, since there are no popularity preferences for the recommendation linkages, the popularity of visited objects at each step is expected to be the average popularity of the whole population of objects, i.e. $k(t) = \langle k \rangle$. As shown by Figure 6.5a-c, the random recommendation networks generate balanced popularity dynamics as expected. However, the popularity of the visited objects $k(t)$ for almost all the similarity measures (except for the HC and HPI measure in Amazon data set) deviates from the average level, which means the random walkers on such recommendation networks are navigated to either popular or unpopular objects.

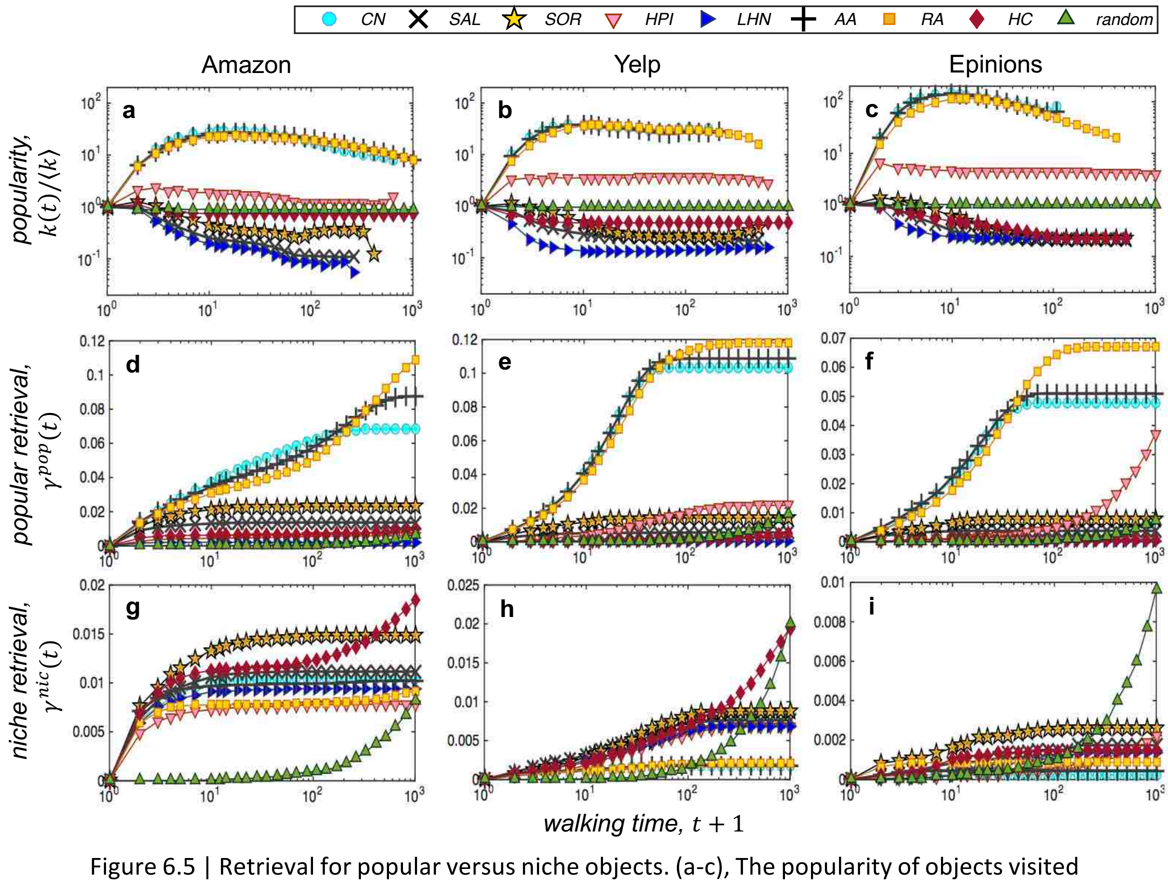


Figure 6.5 | Retrieval for popular versus niche objects. (a-c), The popularity of objects visited at each walking step. These results are based on random walks that start from random objects, not limited to these that are historically selected by the random walker. Note that, for some recommendation networks, the self-avoiding random walk may not last for 10^3 steps. Consequently, there may be missing data points at the tails of the results, such as for the LHN recommendation networks. (d-f), The retrieval percentage for popular (more than average) objects $\gamma^{pop}(t)$. (g-i), The retrieval percentage for niche (less than average) objects $\gamma^{nic}(t)$.

The CN, AA and RA measures, which have relatively high accuracy of retrieving users' history records, are shown to be rapidly navigating users to extreme popular objects in a short 10-steps random walk, which can be about 30 (for Amazon and Yelp data sets) or even more than 100 (for

Epinions data set) times more popular than the average. On the other hand, the recommendation networks constructed by the SAL, SOR and LHN measure (also the HC measure in Epinions data set) gradually navigate users to objects with quite low level of popularity. Such biased accessibility may be caused by the systemic bias of the similarity measures as discussed in Chapter 5. Measures such as the CN, AA and RA, do not take any normalisation for the weighted sum of the common neighbours, and thus the popular objects tend to be similar to most others since they have higher chances of getting common neighbours even randomly. On the other hand, measures such as SAL, SOR, LHN and HC normalise the common neighbours by the object degree, and result negative correlations between the object degree (popularity) and the similarity.

Despite the moderate accuracy regarding retrieving users' historical records, the recommendation networks are shown to be navigating users to either popular or unpopular objects rather than in balanced manners. Inspired by the study of personalised recommendation, which has been addressing the importance of diverse and novel recommendations, here we further separately examine the systems' accuracy of retrieving popular and niche objects.

Setting a baseline as the average popularity $\langle k \rangle$ of the whole population of objects, we regard any object o with popularity higher than average, i.e. $k_o > \langle k \rangle$, as a popular object, and these with popularity less than or equal to the average as niche objects. Accordingly, for a testing user u, his/her n_u historical records consist of n_u^{pop} popular objects and n_u^{nic} niche objects, with $n_u = n_u^{pop} + n_u^{nic}$. Similarly, we let each testing user to start from one of his/her historical object and examine how many popular and niche objects can be retrieved at a given step t, denoting with $h_u^{pop}(t)$ and $h_u^{nic}(t)$ respectively. We then have the retrieval percentage for popular objects $\gamma^{pop}(t)$, and niche objects $\gamma^{nic}(t)$ respectively, averaged over all the random walk experiments.

As shown by Figure 6.5d-f, the retrieval percentages for popular objects have similar patterns in comparison to the overall retrieval (Figure 6.4). The most accurate measures, i.e. the CN, AA and RA, generally have higher retrieval percentages for popular objects than that for all objects ($\gamma^{pop}(t) > \gamma(t)$). For other measures, there are no apparent differences between the retrieval for popular objects and for all objects. On the other hand, the retrieval for the niche objects $\gamma^{nic}(t)$ show different patterns in Figure 6.5g-i. Despite the high accuracies of CN, AA and RA measures for the popular objects, their ability of helping users to find niche objects is very limited. The relatively accurate measures for retrieving niche objects are SOR, HC and SAL, but have also quite low retrieval percentages ($\gamma^{nic}(t) < 2\%$), which can be even lower than that of the random recommendation networks in the long-term random walks.

Such results suggest that the recommendation networks constructed by the eight applied similarity measures can only navigate users to find some (generally about 10% or less) popular objects, while fail to help them find diverse and novel ones.

Similar to the results for the retrieval for all objects, its more reliable to examine the short-term retrieval percentages for niche and popular objects. We still take $t = 10$ as an example, the short-term accuracies for popular objects $\gamma^{pop}(t = 10)$ and niche objects $\gamma^{nic}(t = 10)$ are shown in Figure 6.6. One can regard these similarity measures as two clusters. The cluster consisting of the

CN, AA and RA measures, generally have better accuracies retrieving popular objects but is not good at retrieving niche objects. On the other hand, the second cluster consisting of SAL, SOR, HPI, LHN, and HC measures has poor retrieval percentages for both popular and niche objects.

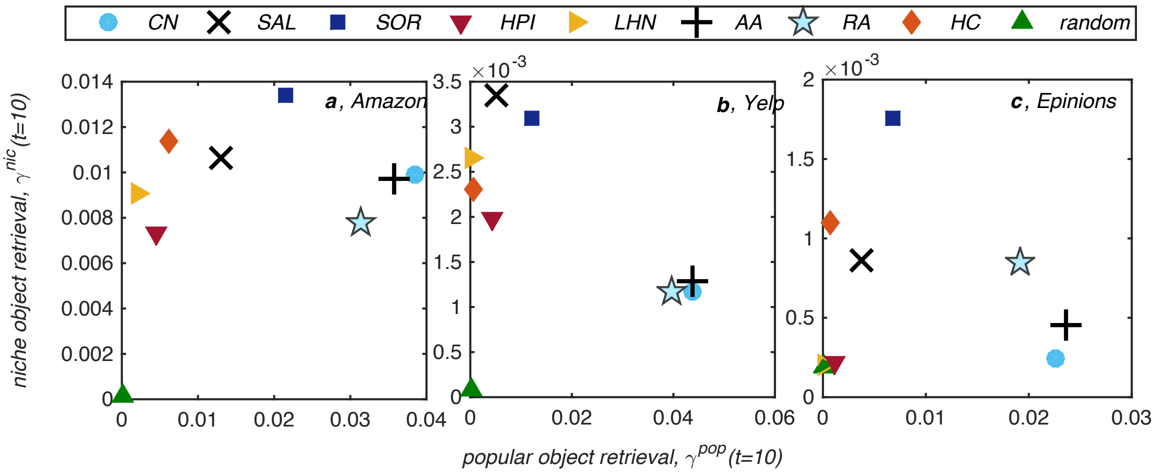


Figure 6.6 | Short-term retrieval percentages for popular and nice objects.

6.3. Influence of the Length of the Recommendation List

So far, all the analysis is based on the recommendation list length of $L = 10$. Actually, practical websites have different lengths from each other, such as $L = 3 \sim 10$ in Amazon depending on the window size of the Internet browser and $L = 3$ in Yelp. Here we explore the influence of the recommendation list length L on the network structure and the navigation accuracy.

Apparently, the length of recommendation list directly decides how many other objects can a specific object connect to with hyperlinks. Thus, the length L will largely influence the structure of the recommendation networks. Here we investigate two aspects of the network structure, which are network efficiency and the reachable objects.

As has been introduced in Chapter 2, Section 2.1.1, the shortest path length of a network is an important indicator for its efficiency. The distance from one object i to another j is the smallest number of links that can direct a random walker from the source to target, denoting with d_{ij}. Since in the recommendation network, not every pair of objects has a route, here we use the efficiency metric to describe such feature, which is defined as

$$E = \langle \frac{1}{d_{ij}} \rangle = \frac{\sum_{i,j \in O; i \neq j} 1/d_{ij}}{N(N-1)}, \tag{6.3}$$

which is basically the average value of the reciprocals of the distance, with the distance between these unconnected object pairs to be infinite, i.e. $d_{ij} = \infty$ if there are no paths connecting from i to j. Accordingly, a larger value for E means the network is more efficient.

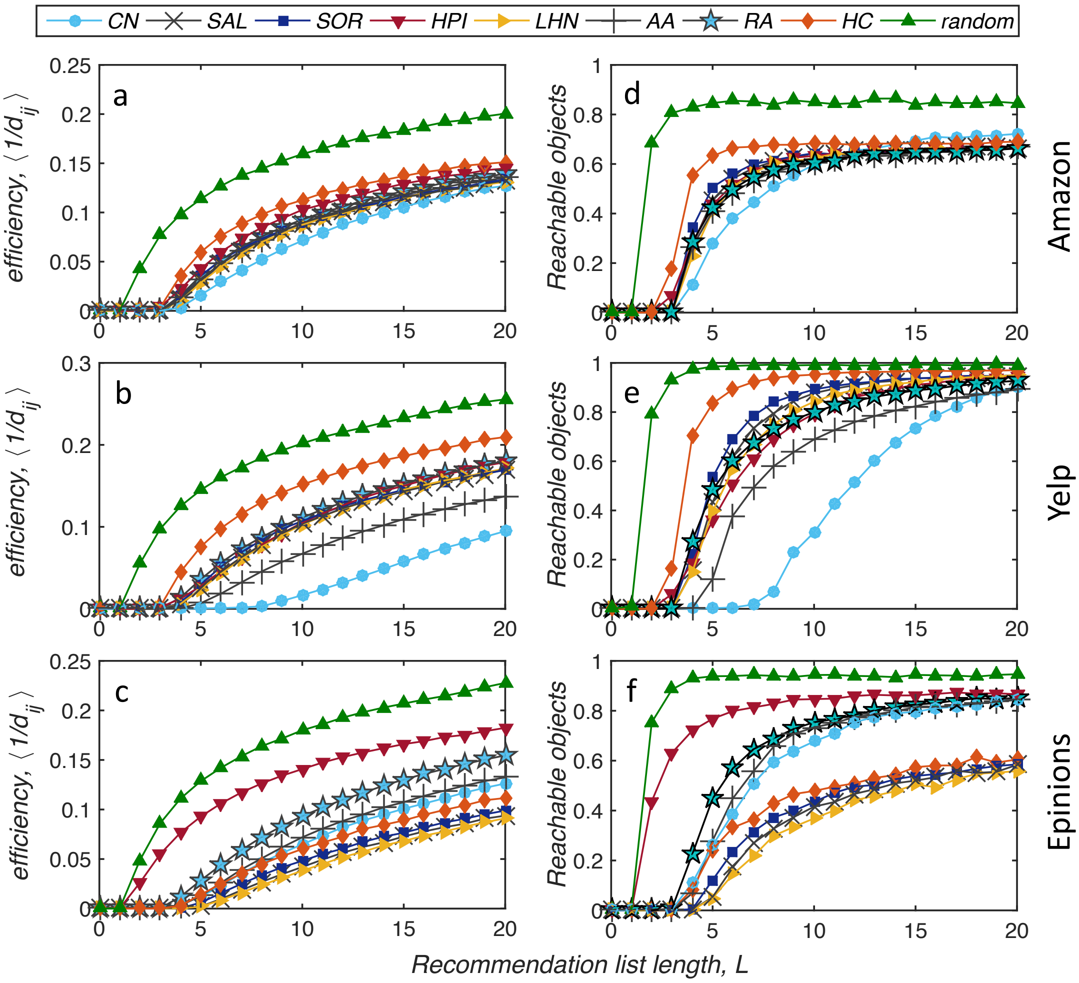


Figure 6.7 | Impact of recommendation list length on the network structure. The number of reachable objects is normalised by the population of the objects N.

Since larger list length L means that more links will be introduced into the recommendation networks, the shortest path length between two objects, i.e. the network efficiency, will be largely improved, as shown in Figure 6.7a-c. But one may notice that, the recommendation networks projected by all the applied measures are much less efficient than the randomly constructed ones in terms of the network efficiency. The reason lies in the fact that, random recommendation networks have more equalised in-degree distributions.

There are also critical points in the increase of the network efficiency versus recommendation list length. When the length $L < 4$ for Amazon and $L < 5$ for Yelp and Epinions datasets, the network efficiency is almost $E = 0$, regardless of the increase of the length. As soon as the length increases to values larger than the critical points ($L = 4$ for Amazon, $L = 5$ for Yelp and Epinions), the network efficiency dramatically increases. Such phenomenon is normally known as the second-order phase transition in Physics and has also been widely observed in networks (Buldyrev *et al.* 2010; Gómez-Gardenes *et al.* 2011; Gao *et al.* 2011).

Another apparent change that the increase of length L brings to the network structure is the number of reachable objects. For a specific source object, if there is at least one path connecting to any other objects o, we call o reachable for the source object. Therefore, the number of reachable objects is actually the average number of object that can be reached starting from the source object. As shown by Figure 6.7d-f, as the length L increases, more and more objects become reachable for random walkers. Apparently, the random recommendation networks have the highest ratio of reachable objects, which are generally more than 80% for Amazon and more than 95% for Yelp and Epinions systems when the length $L > 2$. Notably, the increase of reachable objects has also critical points which are the same to that of network efficiency, i.e. $L = 4$ for Amazon and $L = 5$ for Yelp and Epinions.

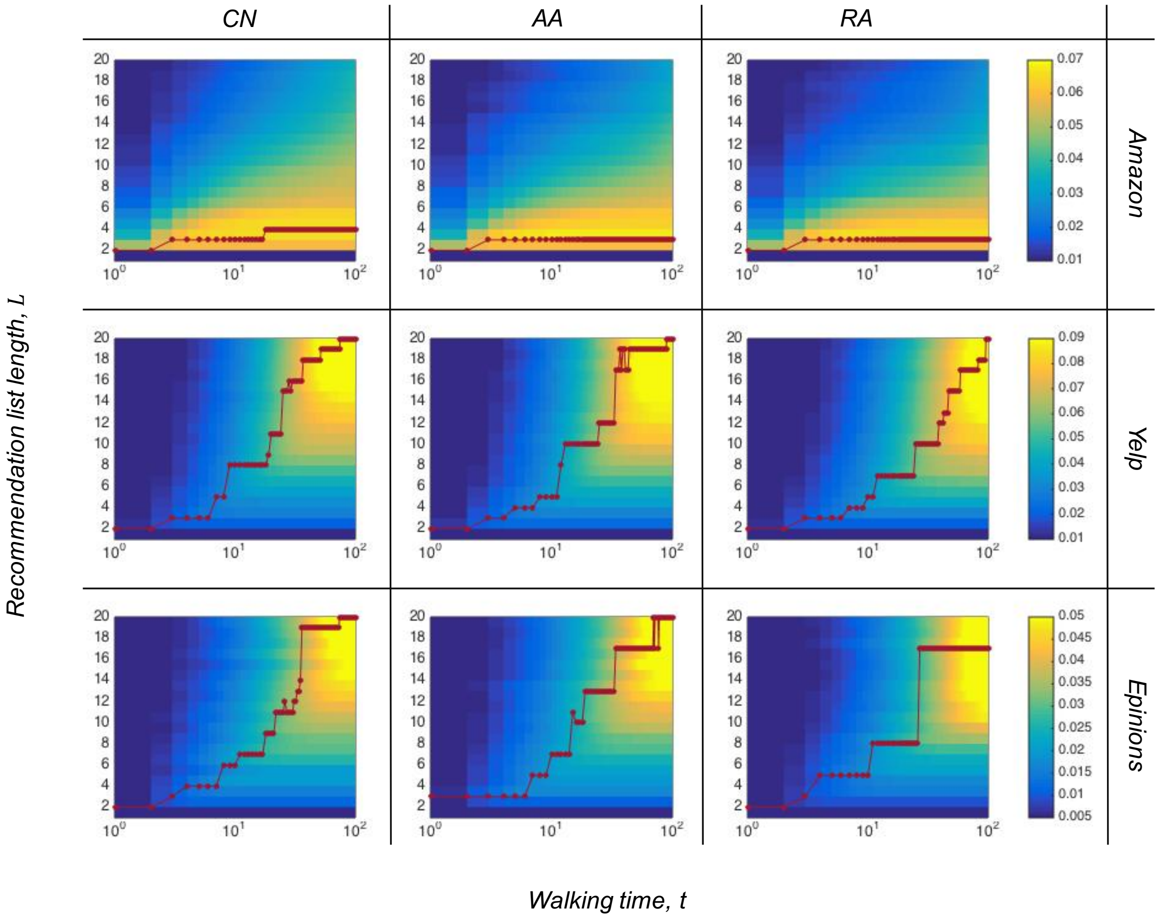


Figure 6.8 | Heatmaps of retrieval percentage for CN, AA and RA measures versus the recommendation list length and walking time. Each row (dataset) shares the same colour scale, and thus the results are comparable to each other within the same dataset.

While the recommendation list length has significant influence over the network structure, a key question is that how would it influence the accuracy of recommendation networks navigating users. With different length of recommendation list ($1 \leq L \leq 20$), we carry out again the information retrieval experiments and the results are shown in Figure 6.8 (for CN, AA and RA measures) and Figure 6.9 (for SAL, SOR, HPI, LHN and HC measures). In these heatmaps, cooler colour (blue) represents low retrieval percentages while warmer colour (yellow) represents higher accuracies. One may notice that the warm colour generally distributes on the right side (larger walking time t), which indicates that the longer the walkers surf on the recommendation network, the more history records can be retrieved. However, the warm colour does not necessarily distribute on the top of the heatmap (longer list length L), which means that the length L is not simply the longer the better.

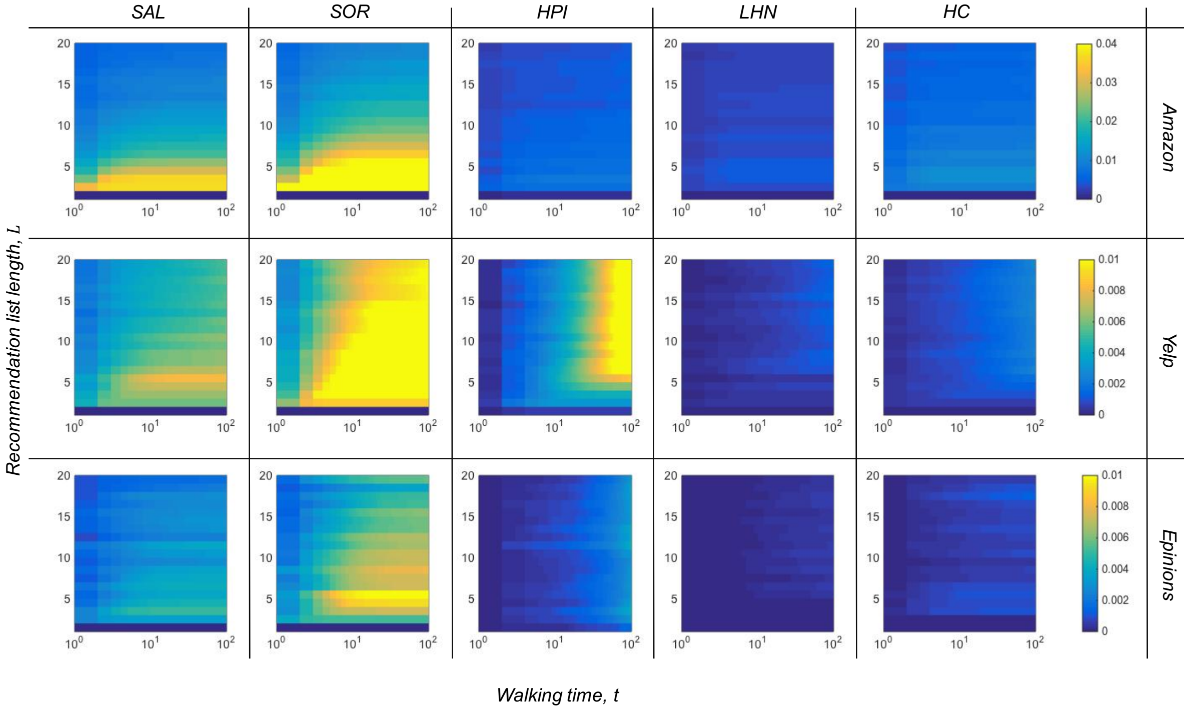


Figure 6.9 | Heatmaps of retrieval percentage for SAL, SOR, HPI, LHN and HC measures versus the recommendation list length and walking time. Each row (dataset) shares the same colour scale, and thus the results are comparable to each other within the same dataset.

Taking $t = 10$ as an example, we take the intersection from each heatmap and put it into two-dimensional plots as shown in Figure 6.10. The short-term retrieval percentage $\gamma(t = 10)$ generally increases at first when the length L increases, but will drop for large L. It means that there are optimal values of length L^{opt} to maximise the accuracy. For the short-term retrieval ($t = 10$), the optimal list length should be $L^{opt} = 2{\sim}4$ for Amazon dataset, and $L^{opt} = 4{\sim}6$ for Yelp and Epinions datasets as shown in Figure 6.10.

Actually, for every given time step, an optimal value for the length can be achieved. Considering the SAL, SOR, HPI, LHN and HC measures are not accurate, here we only show the optimal lengths for CN, AA and RA measures in Figure 6.8, where each red solid dot represents the length L^{opt} maximising the accuracy for the corresponding time step. The results show that if one wants to achieve high accuracy at a given time step t_o, the optimal length L^{opt} would normally be taken as a relatively small value. Only when one wants to optimise the accuracy for a very long-term random walk, for example $t_o > 100$, long recommendation lists can be considered (only for Yelp and Epinions dataset). For short-term retrieval, a long recommendation list may reduce the accuracy for users finding interesting objects. The reason may lie in the fact that, offering a lot of recommendations in the list would distract users from visiting the most similar ones.

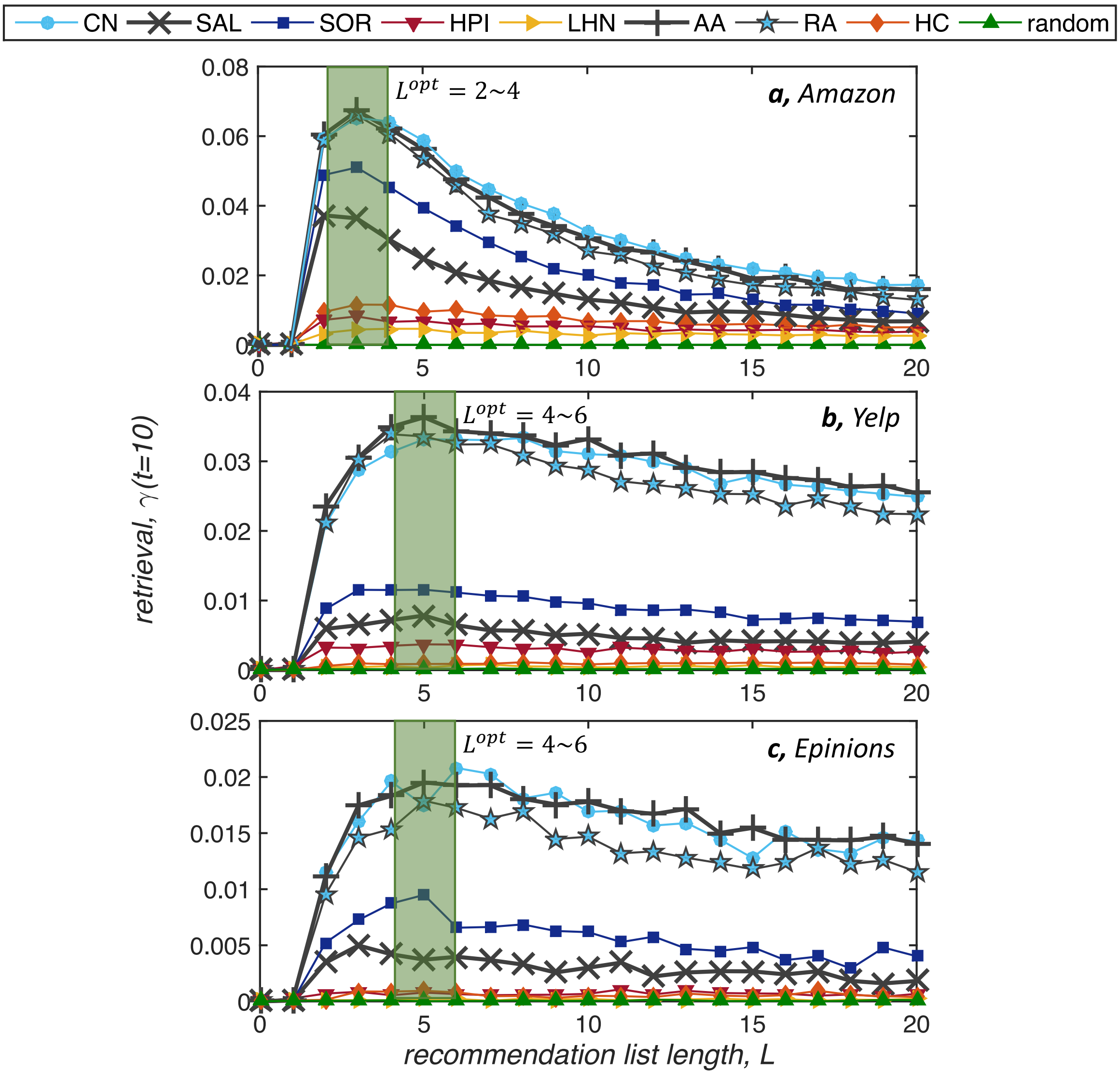


Figure 6.10 | Short-term retrieval percentage versus recommendation list length.

6.4. Empirical Study on Amazon's Book Recommendation Network

In the previous sections, we have focused on the performance of recommendation networks that are constructed by projecting the user-object bipartite networks. Actually, in many practical online systems, the mechanisms through which the recommendation networks are constructed are not exactly known, and they may include some promotions and advertisements in some way. Since we have collected the empirical book recommendation network from Amazon using a web crawler, here in this section, we examine the Amazon's recommendation network.

Please be noted that in Section 6.1 – 6.3, there are also results for dataset 'Amazon', which are referring to the networks projected from the Amazon user-book bipartite network, rather than the actual amazon recommendation network. Here we focus on the actual recommendation network of Amazon which is collected in January 2016 as has been introduced in Chapter 3, Section 3.1.

Previous research has also considered the empirical book recommendation network from Amazon. However, the networks were collected with a depth-first searching strategy (Carmi et al. 2017; Oestreicher-Singer & Sundararajan 2012a; 2012b). While in most actual system, the number of recommendations for each object is uniform and fixed, the depth-first search may fail to achieve this, resulting heterogeneous out-degrees for different objects. The difference in such fundamental structure may lead to biases in the observations and investigations. In our collection, we adopted a width-first searching strategy, which collects exactly 10 hyperlinks for every object.

While the out-degree is fixed for each book in our data collection, i.e. $k_o^{out} = 10, \forall o$, the in-degree generally follows a power-law distribution as shown in Figure 6.11a with an exponent of -3.3. In the Amazon recommendation network, a book's in-degree k_o^{in} represents its frequency of other books recommending it. Such in-degree distribution suggests that there are a few books frequently show up in others' recommendation list, while most others barely get recommended. Note that, there may be a significant amount of books that are not recommended in any lists at all. But due to the data collection mechanism which is based on a tree search, we can only collect those reachable books, i.e. those at least get recommended once. In addition, the in-degree of books shows a moderate correlation with the popularity, as shown in Figure 6.11b, with a Pearson Correlation Coefficient of 0.396. It suggests that the popular books tend to be more frequently recommended in others' lists. Such phenomena may be caused by the mechanism of Amazon constructing the recommendation networks, which is based on a collaborative filtering method with a certain form of network-based object similarity measure. To be more specific, it is the measure that Amazon adopted to evaluate the co-accessing pattern caused such correlation between in-degree and popularity of books, because most measures have been found to be biased by the popularity (as studied in Chapter 5), that popular objects tend to be evaluated as more similar to others. Such bias will lead to the heterogeneous visibility for the books, which will be discussed in the following.

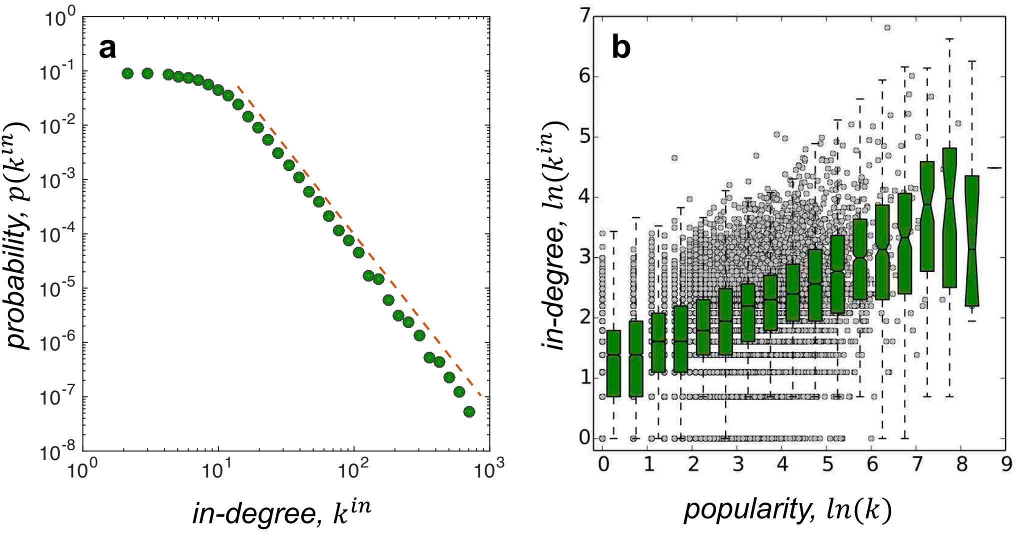


Figure 6.11 | In-degree distribution of books in Amazon recommendation networks.

To study the performance of the empirical book recommendation network, we also apply the self-avoiding random walks to examine its traffic distribution and accuracy.

In the random walk experiment, the traffic of each book may be of the first interest. Here we examine the frequency of each book being visited in random walks f to show the traffic on the recommendation network. As the users do not revisit any books in the defined random walk, each book can be visited at maximum for one time in an individual experiment. Accordingly, we define the visiting frequency of a book o as the times that it is visited over the $N^{experiments}$ experiments, i.e. $f_o = N_o^{visited}/N^{experiments}$. Therefore, a higher value of visiting frequency f_o means more chance for the book o to be visited by the random walkers. The upper-limit $f_o = 1$ would suggest that, the book o is visited by every user starting from any initial book. Here we perform 10^6 independent self-avoiding random walks.

Due to the heterogeneous in-degree distribution, the visiting frequency also approximately follows the power-law distribution as shown in Figure 6.12a, which has an exponent of 2.22 for the tail. Some books are dominating the attention of users who are surfing on the recommendation network. The visiting frequency f of some books could be as high as 0.2, which means these books would be visited by one of every five random walkers regardless of where they started the walk. Such dominance on network traffic is resulted by the heterogeneous in-degree distribution, as books with high in-degrees are more likely to be visited during the random walks. As shown by the red line in Figure 6.12b, the correlation between the visiting frequency and book in-degree is fitted to be $\ln(f_o) = 1.3\ln(k_o^{in}) - 10.52$ ($R^2 = 0.54$).

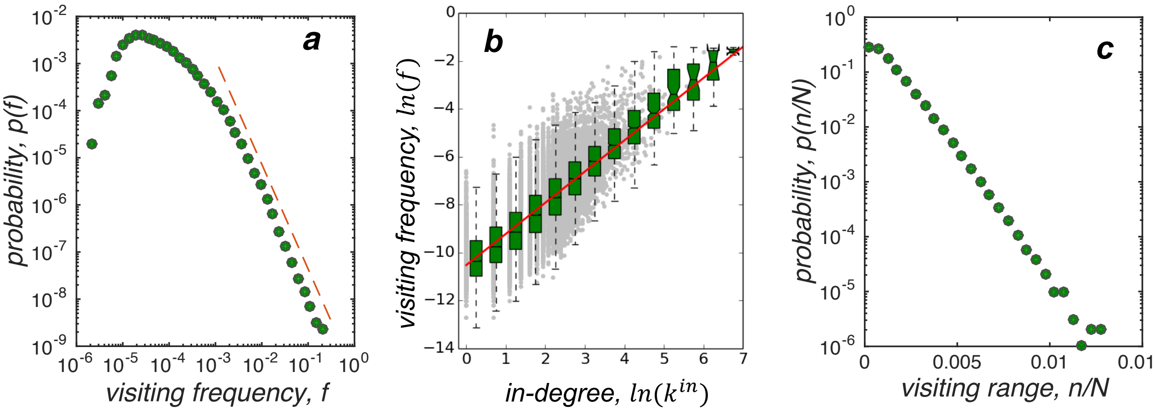


Figure 6.12 | Results of self-avoiding random walks on Amazon book recommendation network.

We also examine the visiting ranges of the random walks, which is defined as the maximum number of books n visited by a random walker without returning to any visited book. The visiting range can well describe for how long could the system keep users surfing on the recommendation networks, and thus is expected to be a large value. However, as shown in Figure 6.12c, following an exponential distribution (the plot is log-normal scaled), the visiting range is very limited. The users can only visit 1.3% of the books at most by randomly walking on the recommendation network, and for most cases, the visiting range is less than 0.1%. The limited visiting range may be caused by the frequent occurrence of reciprocal links. Normally, if a book i is similar to j, the book j will also be similar to i. Therefore, many pairs of books appear in each other's recommendation list. In the collected Amazon book recommendation network, 58% of the links are reciprocal.

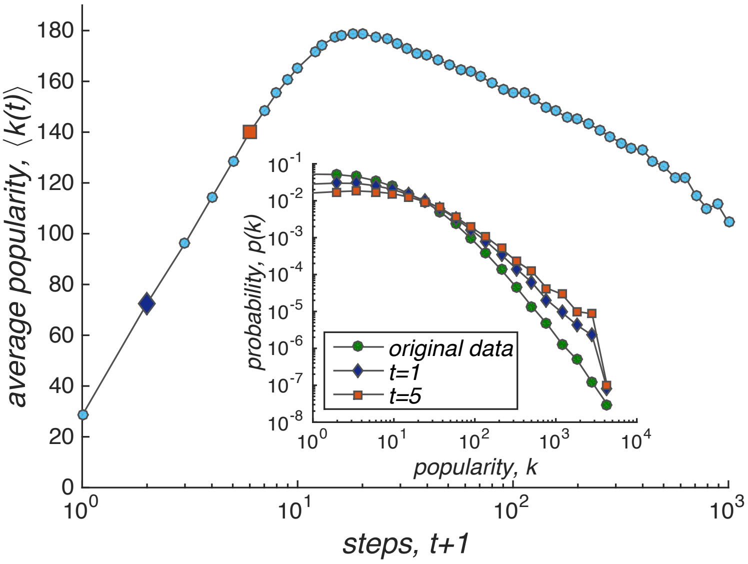


Figure 6.13 | The average popularity of books that are visited at each walking step. In the inset, the distributions of popularities are shown. The green circles are the distribution for all the books. The blue diamonds and red squares are the popularity distributions for the books that are visited at the step of $t = 1$ and $t = 5$, i.e. $p(k(t = 1))$ and $p(k(t = 5))$ respectively.

During the self-avoiding random walk, we are also interested in the popularity of books that the walker visits at each step. In a single random walk experiment, if denote the book that is visited at step t with $o(t)$, we examine the popularity of this book $k(t) = k_{o(t)}$. As the random walks are assumed to start from a randomly chosen book, the average popularity of the initial book should be $\langle k(t = 0)\rangle = \langle k\rangle = 28.6$. However, as shown by Figure 6.13, as soon as the random walk starts, the users are navigated rapidly to popular books. At the step of $t = 10$, the average popularity of the visited books is about 6 times higher than the average level of all the books. To pay closer attention, we show in the inset of Figure 6.13 the distribution of popularity of books that are visited at $t = 1$ and $t = 5$, in comparison to the overall distribution of book popularity. One can find that, the tails of the distributions $p(k(t = 1))$ and $p(k(t = 5))$ are significantly higher than that of the original distribution $p(k)$. This means the popular books have much higher possibility to be visited by random walkers at the steps $t = 1$ and $t = 5$ than expected. Although it has been found that the presence of recommendation network in Amazon redistributed the demands and thereby increased the sale of the 20% least popular products by 50% (Oestreicher-Singer & Sundararajan 2012a), the visibility and reachability of books with different popularity levels are extremely heterogeneous after all. While the search engine is already providing biased results with preference on popular objects, the recommendation network of Amazon is further enhancing the dominance of them.

A more straightforward method to examine the recommendation network is to look at its accuracy in navigating users to their potential interests. Such accuracy can be well reflected by how many of the recommended products are of users' interests leading to purchasing behaviours. We follow the information retrieval methods introduced in Section 6.2.1 here to study the accuracy of Amazon’s book recommendation network.

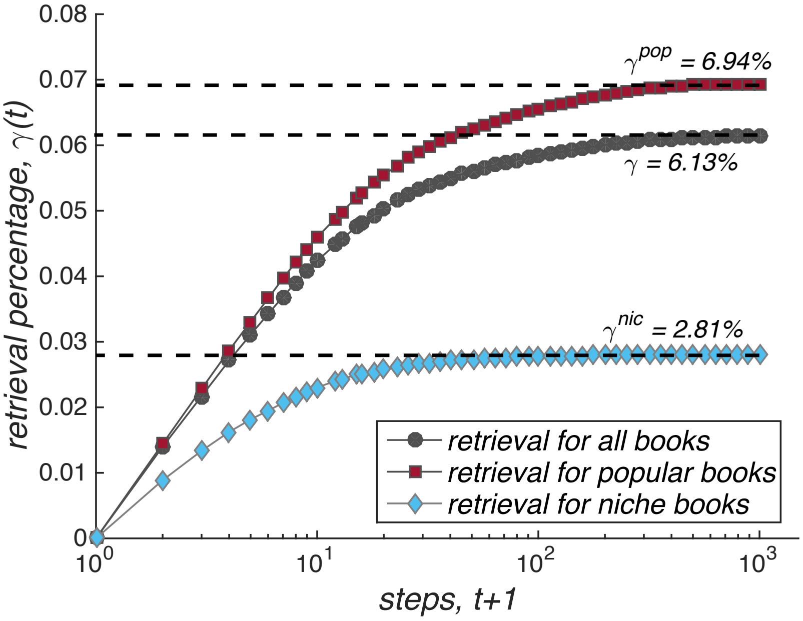


Figure 6.14 | Retrieval percentages for empirical Amazon book recommendation network. We show here only the results within 1000 steps of random walks considering the limited visiting range of the experiment. The popular and niche books are defined as the ones whose popularities are larger and less than the average respectively.

As shown in Figure 6.14, as the random walk starts, more and more historical purchases of the users can be retrieved, but only 6.13% books can be retrieved even for 1000 steps of random walk. One may find that the increase of the retrieval percentage slows down for the late stage of the random walk. There are possibly two reasons: 1) the books can be possibly retrieved are already found in the early stage; and 2) the random walk only lasts for limited steps as shown in the Figure 6.12c. Particularly, one should not expect the users to surf on the recommendation network for too many steps. Once a user finds the recommendation network not efficient in helping them finding interesting objects, s/he may drop out from the surfing. A more reliable way is thus to look at the short-term retrieval percentage. Taking t=5 as an example, only 3.42% of users' history records can be retrieved by randomly walking on the recommendation network.

To examine whether are the retrieved books are popular or unpopular ones, one can further divide the books into groups. We take the average popularity $\langle k \rangle = 28.6$ as a benchmark and regard the books whose popularity is larger than average as popular books and others as niche books which are not popular but can fit some users' interests. Accordingly, we can have also the retrieval percentage for popular books $\gamma^{pop}(t)$ and for niche books $\gamma^{nic}(t)$ respectively. As shown in Figure 6.14, though limited number of books can be retrieved, most of the retrieved books are popular ones. For 1000 steps of random walk, only 2.81% niche books can be retrieved. For a short-term retrieval, 3.66% of popular books and 1.95% of niche books can be retrieved within 5 steps.

6.5. Summary

By applying eight widely-used similarity measures to project the user-object interactions as recommendation networks, this Chapter explores whether such network can accurately navigate users to what they may be interested of.

Though it is expected that recommendation networks should be able to help users to find what they want, the accuracy of current methods are shown to be limited and only focused on extreme popular information. The relatively accurate measures, such as CN, AA and RA, rapidly navigating users to popular objects, fail to help users to find niche ones. The navigation for popular information, though accurate, is of less meaning than that for the niche information.

We also applied the developed evaluation methods to examine the empirical book recommendation network of Amazon. It is shown that the web traffic on the recommendation network is monopolised by the blockbusters, i.e. very popular books. Similar situation of the accuracy is found with the Amazon book recommendation network, that only popular books can be retrieved while very limited number of niche books can be found by the random walkers. Though it is unknown to us that what detailed algorithm is being used by Amazon to construct this book recommendation network, the results suggest it to be similar to the similarity measure group of CN, AA and RA.

Additionally, while in practical systems the recommendation list lengths are different from each other, we show that there are optimal lengths in terms of achieving best random walk accuracy for a given time step. Recommending more objects may provide diverse choices for users, but on the other hand may also distract users from visiting the most relevant information. Generally, though long recommendation list length could result in good accuracy for longer random walks, the length should be short if one wants the recommendation network to quickly locate the interesting objects.

This Chapter is to explore the RQ3 (how accurately can recommendation networks navigate users to find what they interested in?). Though the self-avoiding random walk model, such question is addressed by the developed evaluation framework as the information retrieval percentage. Experiments show that such recommendation networks have limited accuracy, and especially fail to help uncover the niche objects.

Chapter 7. Information Accessibility and Traps in Recommendation Networks

In Chapter 6, the accuracy of recommendation networks to navigate users to find interesting objects has been studied. Generally, recommendation networks are in nature information networks. Yet, the information accessibility, which is central to the understanding of such systems' function and efficiency, is still unknown. Since users normally surf on such network to explore online content, a fundamental question raises that are the recommendation networks navigable (RQ4 of this thesis)?

In this Chapter, we explore such simple question by examining how many distinct objects can a random walker visit in the recommendation networks (navigability). This Chapter concludes that the recommendation networks are not navigable, which means users cannot efficiently discover different objects by surfing on it. We also uncover the existence of traps in such networks which is the reason of the low navigability. As an additional contribution, an efficient local metric is developed for the identification of traps in recommendation networks.

The present Chapter is organised as following. Since the construction of recommendation networks has been introduced in Chapter 6, Section 6.1, here in this Chapter we do not repeat it, and directly apply the measures of CN, LHN, RA, AA and HC to project the user-object bipartite networks accordingly. Therefore, this Chapter starts by introducing a toy bipartite network model in Section 7.1, which will be used as a benchmark dataset alongside with two empirical datasets, including the Amazon user-book bipartite network and the Yelp user-restaurant bipartite network. Section 7.2 gives the calculations and experimental results for the accessibility and navigability of constructed recommendation networks. Section 7.3 uncovers the existence of traps in recommendation networks which are small collections of objects which have dense internal linkages but few or even no links connecting out. We further develop a local-structure-based metric to help identify the trap objects in large scale networks in Section 7.4. At last, Section 7.5 summarises this Chapter.

7.1. A Toy Bipartite Network Model

Apart from two empirical bipartite networks, we also aim to analyse the recommendation networks constructed according to a toy bipartite network, so that the results could be more generalised rather than limited to the applied datasets.

Considering the fact that most empirical user-object bipartite networks have power-law degree distributions for both users and objects (Figure 3.2), we aim to generate toy networks with similar properties. Inspired by the Barabási-Albert model (Barabási & Albert 1999), we also adopt a dynamical process for generating networks.

At the initial stage, we consider a complete bipartite network with $m_0^u = 10$ users and $m_0^o = 10$ objects, where every user-object pair is connected. For every following step:

1) an object node is added to the network with m^o links, each of which connects to an existing user node with a probability proportional to the user node's degree, i.e. $p \propto k_u^{bip}$;
2) a user node is added to the network with m^u links, each of which connects to an existing object node with a probability proportional to the object node's degree, i.e. $p \propto k_o^{bip}$.

Since m^o and m^u will determine the minimal degree for objects and users, and in order to make the bipartite degrees in a wider range, we set m^o and m^u to be a number selected from $\{1, 2, \ldots, 9\}$ uniformly at random. Such process is continued until the population of objects and users reach $N = 10^5$ and $M = 10^5$.

In such toy bipartite network, since every object and user brings on average 5 new links, the average degree for both users and objects are expected to be 10. In particular, the degree distribution for both users and objects will follow a same power-law form, as shown by Figure 7.1, with exponential of 2.9.

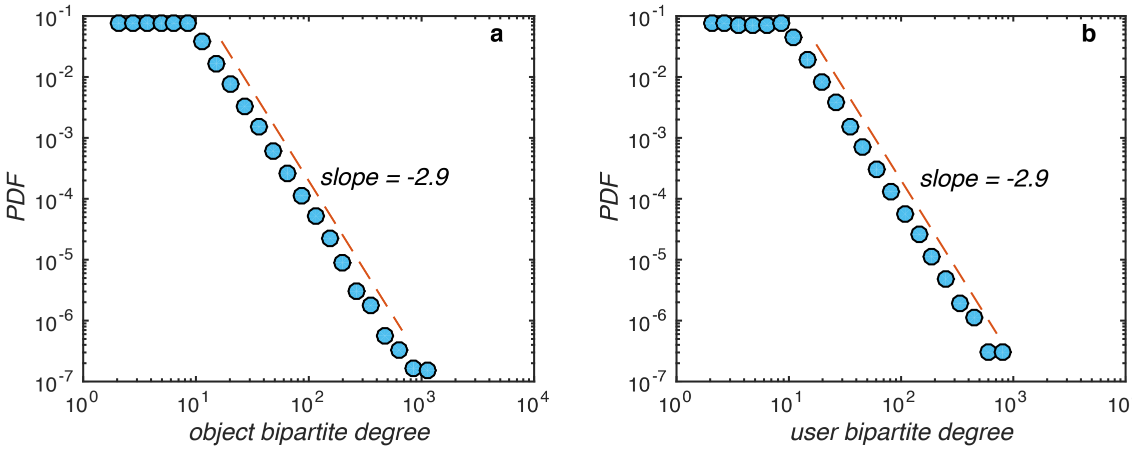


Figure 7.1 | Degree distributions of the toy bipartite network.

The toy bipartite network generated here will be used as one of the three bipartite network datasets in the following studies. The advantage of adding this toy dataset is that it has no user preference in the linkages. Therefore, any significant results uncovered from this dataset will not be caused by the system design of the website nor the user behaviour. Instead, the results will be accounted to the general mechanisms from normal bipartite network.

Hence, in the following studies, three datasets will be applied, including the ToyData, Amazon and Yelp. For the description of bipartite network datasets of Amazon and Yelp, please refer to Chapter 3, Section 3.2.

7.2. Accessibility and Navigability of Recommendation Networks

Following the method of top-L projection introduced in Chapter 6, here we apply similarity measures of CN, LHN, RA, AA and HC to calculate the object similarity and construct recommendation networks based on each of the three applied bipartite networks. In particular, we also consider a random recommendation network, which is to let each object connect to L random objects.

To study the accessibility and navigability of recommendation networks, we consider the random walk process in it. Different from the self-avoiding walks introduced in Chapter 6, here we use traditional (repeatable) random walk which is more suitable for the exploration of network navigability. To be more specific, a random walker starts from a random object and walks through the network following the hyperlinks regardless of whether an object has been visited or not. Therefore, such random walk is endless and a specific object could be visited for multiple times.

The first-passage time, characterising the time that a random walker reaches an object for the first time (Condamin *et al.* 2007; Perkins *et al.* 2014) as introduced in Chapter 2, Section2.1.3, is normally the first interest when studying how navigable is each entity or the whole network (Boguná *et al.* 2009). However, in directed networks especially recommendation networks where the links are sparse, many nodes may be not reachable at all, leading to difficulties for exploring the first-passage time. We thus firstly look at the number of distinct objects that can be visited in a t-steps random walk process, which has been regarded as a network's navigability (De Domenico *et al.* 2014), denoting with $n(t)$. Apparently, a larger value of navigability $n(t)$ represents a better connectivity for the network where random walkers can easily explore more objects.

Another important quantity to directly describing object accessibilities is that how frequently is every object being visited during the random walk. With N as the population of the objects, every object is expected to be visited once in a N-steps random walk. For simplicity, we thus define the visiting frequency of object v as the times that it being visited in a N-steps random walk, denoting with f_v.

7.2.1. Theoretical calculations for random networks

Considering that the random recommendation networks are constructed randomly, exact solutions for its navigability and accessibility can be theoretically calculated. Here we show the theoretical results for random networks so that the experimental results for other projected recommendation networks can be compared with the random case.

Navigability $n(t)$. Consider a random recommendation network with N objects and each object connects L others, i.e. $k_o^{out} = L, \forall o$. Suppose at step t, the random walker is visiting object o and $n(t)$ distinct objects in the recommendation network has been visited. For the next step, the walker will visit one of o's L outgoing objects. Since in the random recommendation networks, the links are randomly wired, these L outgoing objects should be an unbiased sample of the whole population N. Thus, in these L objects, we can expect $L \cdot n(t)/N$ objects to be visited in the previous steps. Accordingly, at step $t + 1$, the probability of the object to be visited being a new one (have not been visited before) can be written as,

$$prob(t+1) = 1 - \frac{1}{L} \cdot \frac{L \cdot n(t)}{N} = 1 - \frac{n(t)}{N}. \tag{7.1}$$

In other words, for step $t + 1$, the random walker has a probability to visit one more distinct object. Therefore, one can have the master equation for the navigability as

$$n(t+1) = n(t) + prob(t+1) = n(t) + 1 - \frac{n(t)}{N}. \tag{7.2}$$

Accordingly, a simple first order differential equation for the system can be written as

$$\frac{\mathrm{d}}{\mathrm{d}t} n(t) = n(t+1) - n(t) = 1 - \frac{1}{N} n(t), \tag{7.3}$$

with an initial condition of $n(t = 0) = 0$. Solving such equation gives us the expression for the navigability of random recommendation networks which reads,

$$\frac{n(t)}{N} = 1 - e^{-t/N}. \tag{7.4}$$

Expectedly, for a random walk of N steps, the ratio of objects can be visited is thus $n(t = N)/N = 1 - 1/e \approx 0.632$. Thus, 63.2% distinct objects can be expected to be visited in an N-steps random walk.

Since we have here the theoretical navigability for random recommendation networks, we should expect the constructed recommendation networks have navigability that as close to Eq. (7.4) as possible.

Visiting frequency f_v. Consider a target object v, with in-degree k_v in a random recommendation network. For any other object o, with L outgoing links, the probability of each link to connect to the target object is proportional to v' in-degree, i.e. $prob \propto k_v$. Considering that the summation of in-degrees over all the objects can be written as NL, we have such probability as $prob = k_v/(NL)$. Therefore, at every step, the probability of the random walker moving from the current object to the target object v is $k_v/(NL)$. In other words, the visiting frequency of v for every step is

$$\frac{\mathrm{d}}{\mathrm{d}t} f_v(t) = \frac{k_v}{NL}. \tag{7.5}$$

Accordingly, for a N-steps random walk in the random recommendation network, the visiting frequency of the target object v is

$$f_v = \frac{k_v}{L}. \tag{7.6}$$

7.2.2. Empirical results for recommendation networks

In this section, we carry out random walk experiments on recommendation networks constructed by five similarity measures (CN, LHN, RA, AA and HC) and the random mechanism. Here we construct recommendation networks with recommendation list length $L = 10$, and we simulate N independent random walk experiments, each of which lasts for N steps.

As suggested by Eq. (7.4), the number of distinct objects $n(t)$ increases with the walking steps t, and 63.2% objects can be expected to be visited for N steps of random walk. The results of the random walk experiments are shown in Figure 7.2, where in each subplot, the solid line represents the theoretical navigability predicted by Eq. (7.4). For the random recommendation networks, the navigability $n(t)$ is consistent as predicted. However, recommendation networks constructed by most similarity measures have much lower navigability. Taking the ToyData which has a population of $N = 10^5$ as an example, N steps of random walk can only cover $\sim 10^1$ objects for the CN, LHN and AA measure. For the empirical data Amazon and Yelp, N steps of random walk generally covers less than 1% objects which is much less than the expected 63.2%. Such result suggests that, the recommendation networks constructed by collaborative-based methods (top-L projections based on existing similarity measures) are generally unnavigable. Users surfing on such recommendation network cannot efficiently and massively explore different objects.

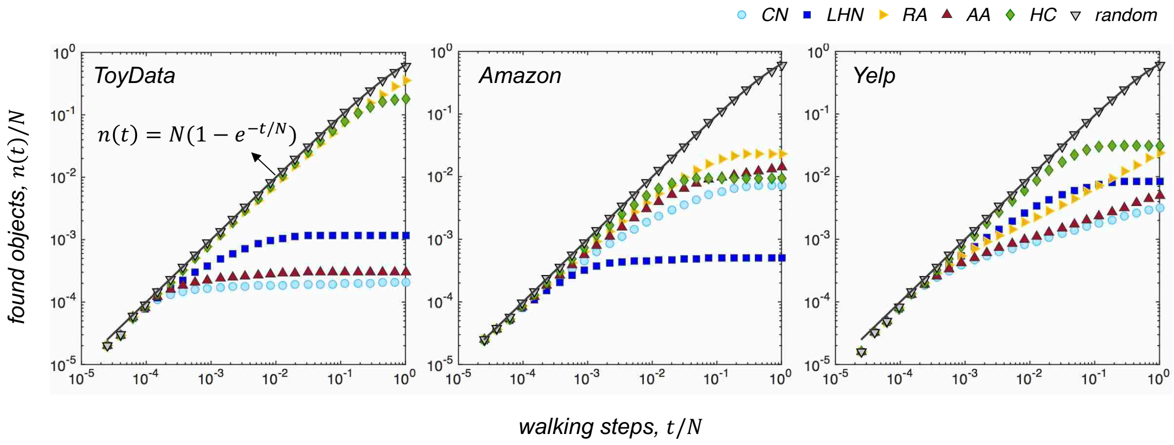


Figure 7.2 | The navigability of recommendation networks with $L = 10$.

For the visiting frequency, as shown in Figure 7.3, the random recommendation networks well follow the prediction of Eq. (7.6) which is marked by the solid line in each subplot. However, all the others largely deviate from it. Though the general pattern is still that high-degree objects tend to have high visiting frequency, some objects (not necessarily those with highest in-degree) have extreme values of visiting frequency which can be $\sim 10^5$ times higher than others. Note that, in a N-steps random walk, every object can be expected to be visited by once, leading the mean visiting frequency $\langle f \rangle = 1$. However, in the recommendation networks constructed by similarity measures, some objects are shown to have visiting frequency as high as $f_v = 10^2 \sim 10^4$.

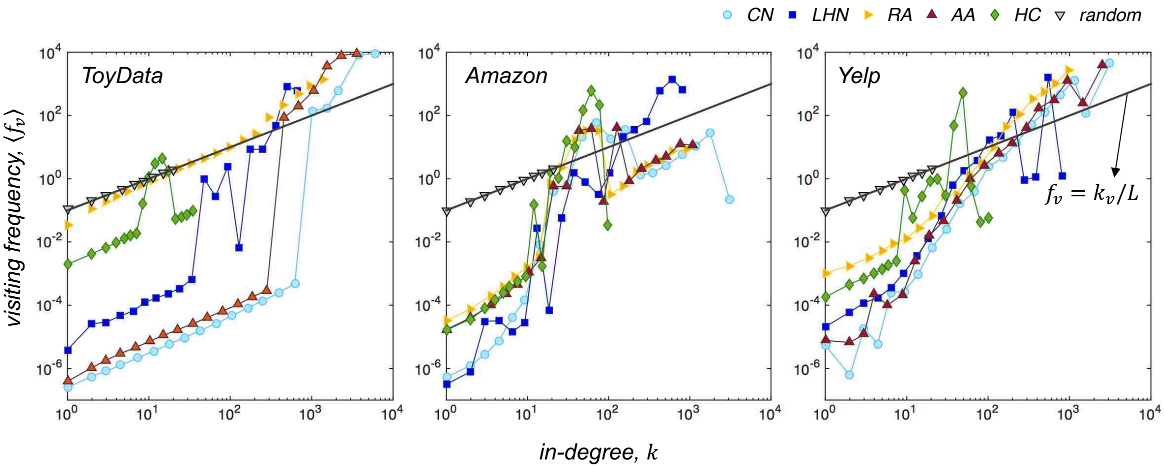


Figure 7.3 | Visiting frequency of objects versus in-degree in recommendation networks with $L = 10$.

The distributions of visiting frequency for different recommendation networks are shown in Figure 7.4. While most distributions follow power-law forms (for CN, LHN, RA and AA measures), there are intermittents in the distributions. Taking the AA measure in Yelp data set as an example, there are no objects with frequency in the range of $f_v = 10^{-3} \sim 10^1$, while some objects have frequencies of $f_v > 10^1$. Such extreme inequality of visiting frequency implies that, most of the traffic on the recommendation networks are dominated by some tens of objects, while most objects have very slim chance to be visited. This is also a direct clue that something is wrong with the random walk in recommendation networks.

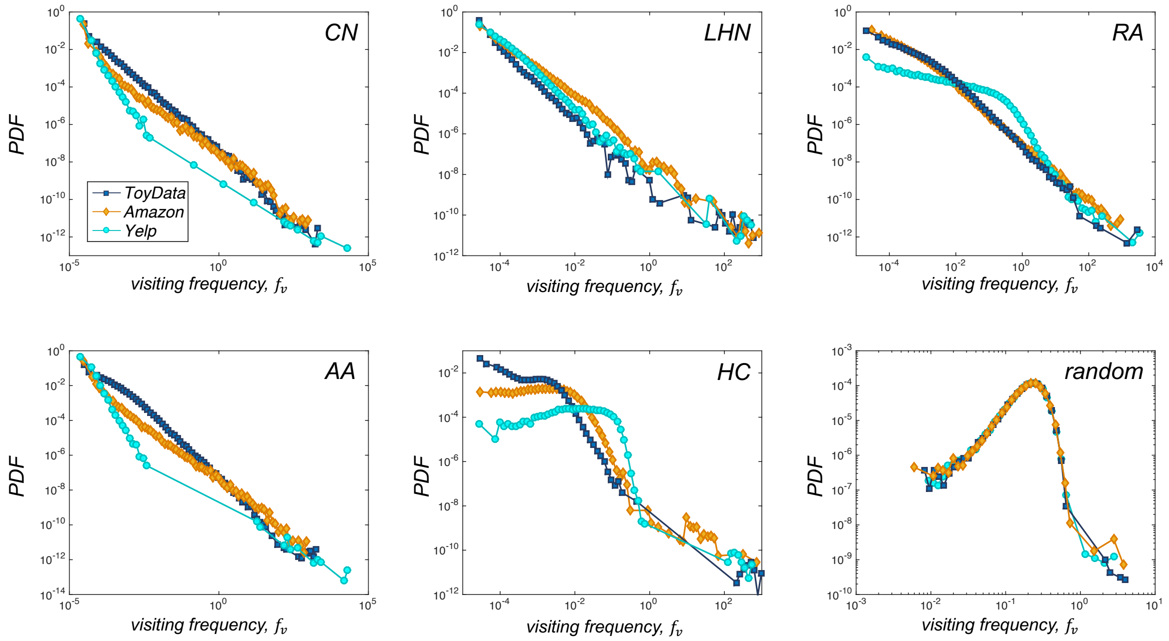


Figure 7.4 | Distribution of visiting frequency in recommendation networks with $L = 10$.

Actually, not only the distributions of visiting frequency are extremely heterogeneous, but also a significant number of objects are not accessible at all, i.e. have visiting frequencies of 0. As shown in Table 7.1, there are significant ratios of objects that are not accessible in recommendation networks except for the random ones. Generally, such objects are these with no in-degree, i.e. $k_v = 0$ and thus have no chance to be visited during random walks.

Table 7.1 | Ratio of objects that have visiting frequency of $f_v = 0$ in each recommendation networks.

	CN	LHN	RA	AA	HC	random
ToyData	52.88%	27.96%	11.82%	38.02%	1.65%	0%
Amazon	51.42%	41.53%	8.50%	14.29%	3.77%	0%
Yelp	44.53%	13.63%	11.36%	37.90%	1.25%	0%

7.3. Traps in Recommendation Networks

The unexpected results of the random walk experiment naturally make one wonder that what makes the recommendation networks constructed by similarity measures so unnavigable and why would the extreme objects exist dominating most traffic. Though the in-degrees of objects are quite heterogeneous, evidences have been found that networks with power-law degree distributions are still navigable (Boguná *et al.* 2009). Accordingly, a seemingly explanation is that there are 'traps' in the constructed recommendation networks.

We define a trap as a small collection of objects who have dense internal linkages, but few links connecting out to others as shown in Figure 7.5. In the toy recommendation network shown in the figure, each of the node has an out-degree of $L = 2$. The solid black nodes are the members of the illustrated trap, most out-going links of which are connected to each other. There is only one link from the trap nodes connecting to the outside (green dash-dot line) defined as an escaping link. Normally, for a significant trap which may account for a large share of traffic, there would be a lot of links connecting to the trap (trapping links as shown by the dotted lines). Accordingly, a random walker in such network will have a high probability getting into the trap while a very low probability getting out.

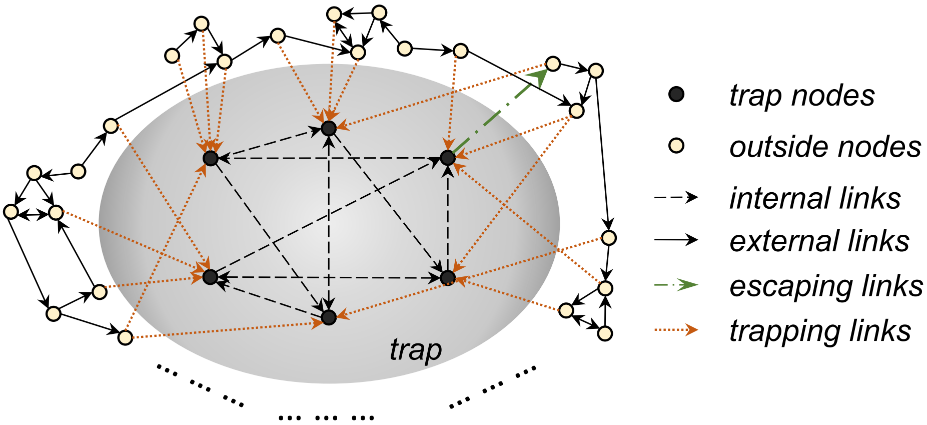


Figure 7.5 | Illustrative example of a trap in recommendation networks.

Theoretically consider a trap with size of s, i.e. there are s objects in it, there would be sL links originating from these trap nodes. An important quantity is that how many of these links are connecting to outside the trap which is defined as escaping links as shown by the green dash-dot line in the Figure 7.5. If there are N^{esc} escaping links, we thus can define the escaping probability pe for a random walker who is in the trap as,

$$pe = \frac{N^{esc}}{sL}. \tag{7.7}$$

The determination of whether a collection of nodes is a trap or not, is rather arbitrary. Generally, a low escaping probability pe is required. Furthermore, if the probability $pe > 0$, the trap can be regarded as an open trap, random walkers in which still have chance of getting out. On the other hand, a trap with $pe = 0$ can thus be defined as a closed trap, which means a random walker can never get out once trapped.

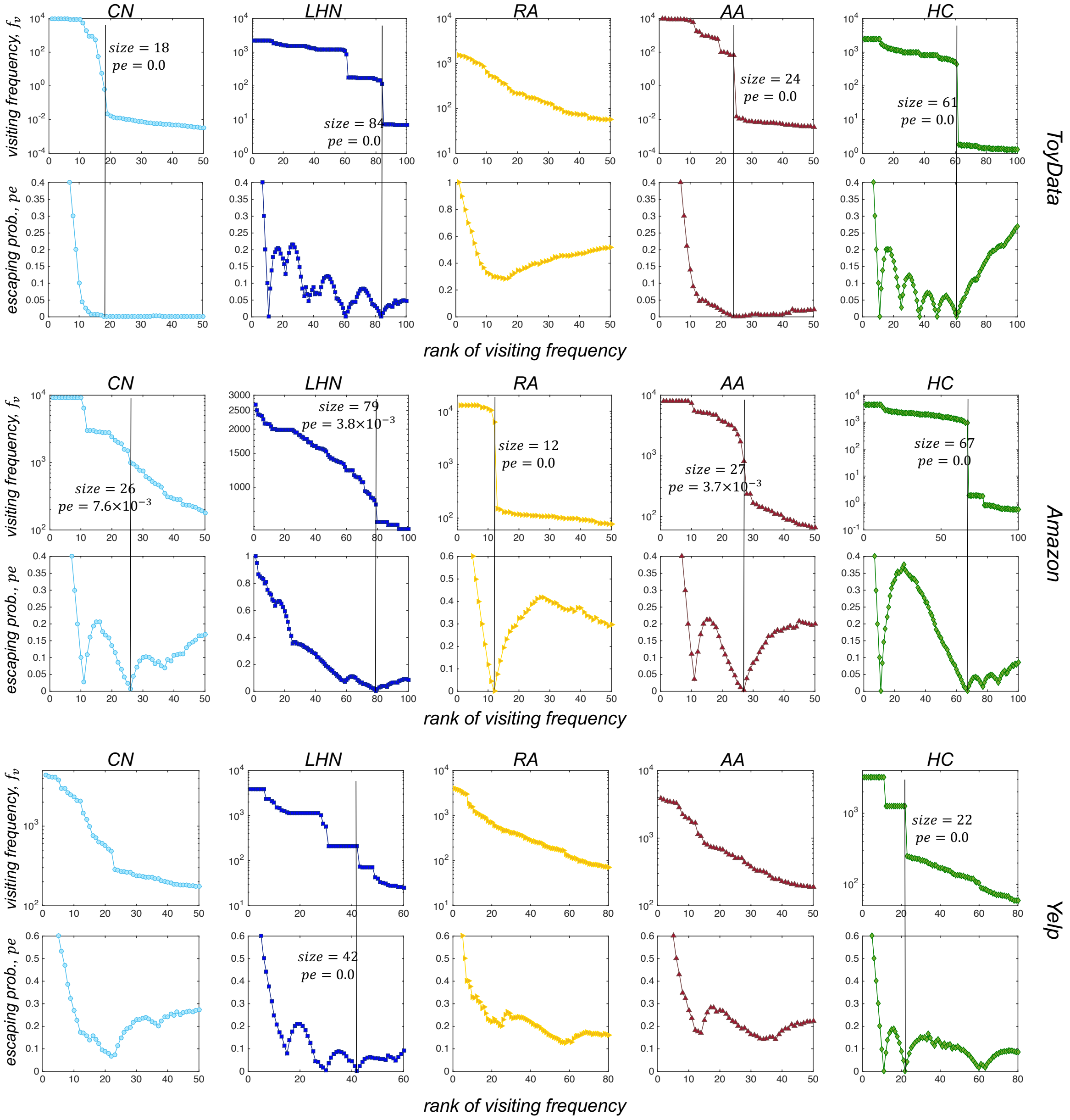


Figure 7.6 | Identifying traps among highest-visiting-frequency objects in recommendation networkswith $L = 10$.

To explore whether the extreme visiting frequencies of a few objects are due to the existence of traps, we rank the objects in terms of the visiting frequency, as shown in Figure 7.6 in the top row of each dataset. Significant drops can be observed in many of the ranking. For example, in the HC recommendation network based on the ToyData, the 61st object has a visiting frequency of 449.0, while that of the 62nd object is about 2.9. To check whether are these objects a trap, we examine the escaping probability pe for the n highest-visiting-frequency objects. In the bottom row of each data set, the corresponding escaping probabilities are shown. The escaping probability shown here is calculated by assuming the top-n objects with highest visiting frequency as a single trap. For some recommendation networks, the probability pe could reach a quite low level (or 0), and seems to be periodic. Such phenomenon suggests that there may exist smaller traps in a bigger trap. Taking the HC recommendation network in ToyData as an example, the 11 highest-visiting-frequency objects have a $pe(11) = 0$, while the top-61 objects also have a $pe(61) = 0$. It indicates that the trap of 11 objects is a closed trap within a bigger closed trap of size 61. For the determination of the traps, we follow the rules:

1) if there are values of $pe(s) = 0$, we regard the last such value s within the top-100 as the trap size;
2) if there are no such values of $pe(s) = 0$, we consider the last value s which has $pe(s) < 0.01$ as the trap size;
3) if the escaping probability pe is always higher than 0.01, we consider there is no traps.

According to such rules, we identified the traps among the top-ranking objects as shown in the Figure 7.6 where marked by a solid line. We summarise the statistics of the identified traps in Table 7.2. Generally, the traps are of very small size ($\sim 10^1$) in comparison to the whole populations, i.e. 10^5 for the ToyData, 157,856 for Amazon, and 61,184 for Yelp. Though with small size, the traps are accounting for most of the web traffic. Taking the CN measure in ToyData as an example, the 18 objects with highest visiting frequency are a closed trap with $pe = 0$ which accounts for 99.99% of the random walk traffic. In particular, whether an object belongs to a trap or not significantly separates its visiting frequency, leading to dramatic drops in its ranking as shown in Figure 7.6.

Table 7.2 | Traps among top-visiting-frequency objects in recommendation networks with $L = 10$. The values '-' mean that there is no trap among the top-visiting-frequency objects.

		CN	LHN	RA	AA	HC
ToyData	Size	18	84	-	24	61
	Traffic	99.99%	99.53%	-	99.99%	73.49%
Amazon	Size	26	79	12	27	67
	Traffic	85.09%	81.30%	90.07%	92.25%	98.89%
Yelp	Size	-	42	-	-	22
	Traffic	-	93.34%	-	-	81.09%

Focusing on only closed traps, we further uncover all the traps regardless of their traffic shares in recommendation networks with different out-degree L, and examine the severity of them.

For the identification of the closed traps, we adopt a greedy search following directed links in the recommendation network. Starting from an arbitrary node, we check how many other nodes can a greedy search find. For example, in Figure 7.7, starting from node 5, the first step of the search will find node 4 and 1, and at the second step node 2 and 3 will be found. After that, the search will not discover any new nodes. Accordingly, such greedy search can be ended and these nodes are regarded as a closed trap, i.e. {5, 4, 1, 2, 3}.

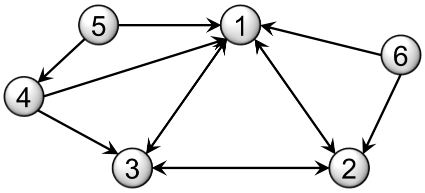


Figure 7.7 | Illustration for the closed trap identification where the out-degree L = 2.

An apparent flaw can be observed from such method of discovering closed traps, that different traps may have overlaps. Actually, one may find from Figure 7.7 that the smallest closed trap in this toy network is formed by nodes {1, 2, 3}, connecting to each other. However, due to the identification method, which is a greedy search starting from an arbitrary node, multiple closed traps are possible to be considered including: {5, 4, 1, 2, 3}, {4, 1, 2, 3} and {6, 1, 2, 3}. It is apparent that once a walker is trapped in any of these traps, he/she can only keep visiting nodes 1, 2 and 3. The marginal nodes such as 5 and 6 have no in-degree, and thus though been included in a trap, they will not be visited anymore. Several disadvantages may be resulted that, 1) the number of discovered traps would be significantly and falsely increased; 2) many non-trap nodes would be regarded as trap members.

To best avoid such situation, we adopt following strategies. First of all, we only start greedy searches from nodes with in-degree larger than 0 so that such nodes would at least have chance to be visited during a random walk. In this way, a significant amount of nodes would be prevented from being regarded as trap members since there are quite a lot nodes with in-degree of 0 as shown in Table 7.1. Secondly, when doing the greedy search iteratively for different nodes, we do not consider the nodes that have been included in any of the traps discovered in the early steps. For example, in Figure 7.7, if we firstly do the search starting from node 4, the trap {4, 1, 2, 3} will be discovered. Consequently, we do not carry out searches starting from nodes 1, 2, 3 any more. With such strategies, we believe that the inclusion of marginal objects into the traps will be largely avoided.

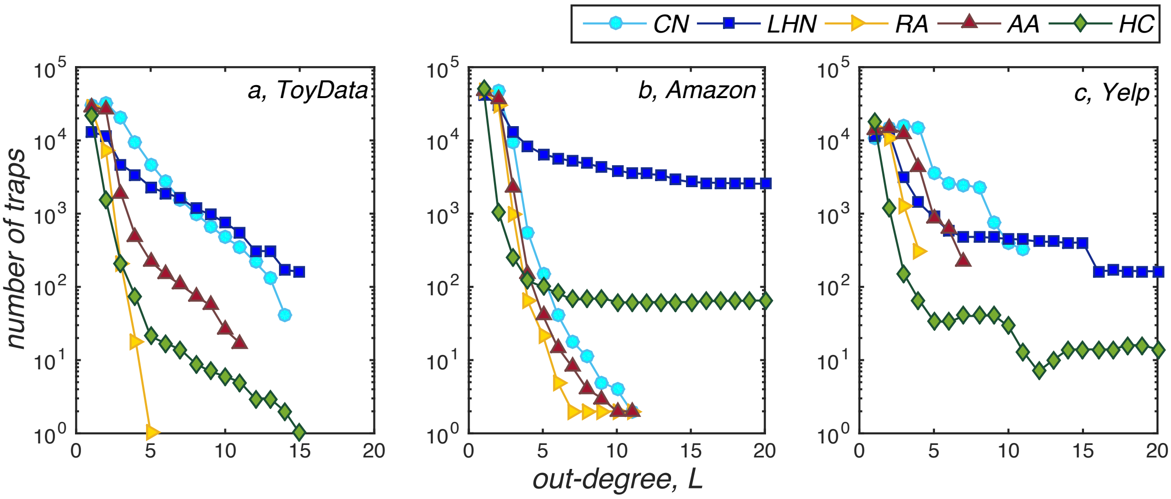


Figure 7.8 | Number of closed traps in recommendation networks with different out-degree.

In more general words, for each object in a network with an in-degree of $k > 0$, we explore how many others it can reach. If an object can only reach a very finite number $s - 1$ of objects, it and its reachable objects are thus a closed trap with size of s. Note that, here we consider only these with size $s < 1000$ as closed traps since the traffic would also be quite evenly distributed if the trap has a considerably large size. After all, the whole population can also be considered as a trap with escaping probability of 0. Figure 7.8 reports the number of closed traps uncovered following such strategy in recommendation networks with out-degrees $1 \leq L \leq 20$. As has been discussed earlier, the traps may have overlaps on the core set of objects, and thus the number of closed traps shown here may be larger than the actual number. Anyway, when a recommendation network takes a small size of recommendation list L, there would emerge a lot of closed traps. As the length increases, there would be generally less and less traps in the network. Actually, one can imagine that when the length $L = 1$, there will be a lot of pairs of objects that are connecting to each other and no links connecting to other objects. Such pairs of objects are typical closed traps. Similarly, $L = 2$ may results in many local structures as triangles, where three objects connecting to each other. Same logic could apply to any L, that $L + 1$ objects connects to each other as a closed trap. When the length L takes a small value, it's easier to result such structure while large L is harder. This explains the dynamics of the number of closed traps over the length.

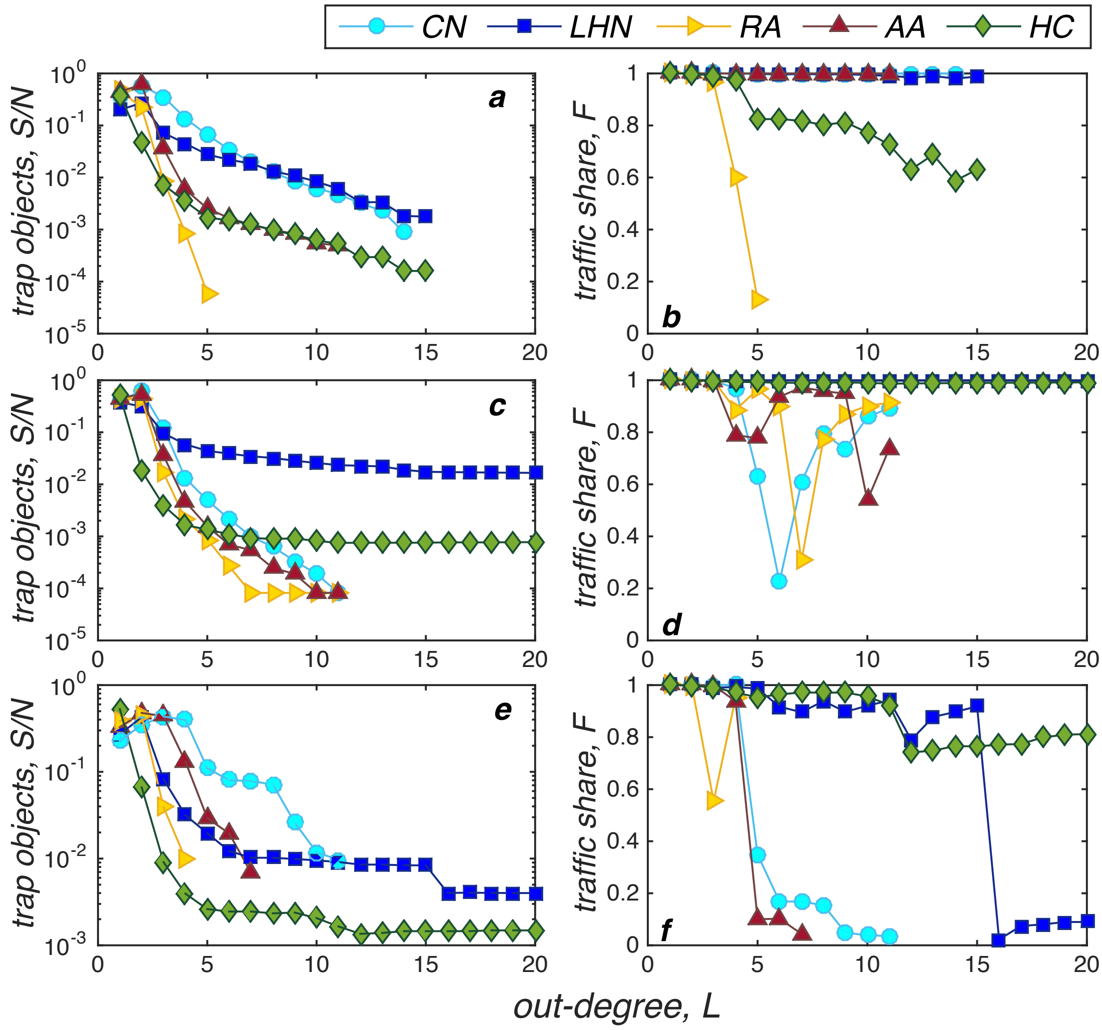


Figure 7.9 | Size and traffic share of closed traps in recommendation networks.

In particular, we examine how many distinct objects are involved in any closed trap, denoting with S, and how much of the random walk traffic are they accounting for, denoting with F. As the mean visiting frequency is expected to be $\langle f_v \rangle = 1$ according to the definition, the traffic share is calculated as $F = \sum_{v \in \Gamma} f_v / N$, where Γ is the set of trap objects with population of S. As suggested by Figure 7.9, not every recommendation network has closed traps. Generally, recommending more objects (larger out-degree L) significantly reduces the number of trap objects. With a large enough value of out-degree L, for example $L > 5$ for RA and AA measures in ToyData and $L > 11$ for CN, RA, AA measures in Amazon, no closed traps would emerge. Though a very small proportion of objects involving in traps when the recommendation list length is relatively short, these trap objects account for significant shares of random walk traffic because the walkers cannot get out once trapped.

7.4. Local Return Rate Metric for the Identification of Significant Traps

It is easy to uncover all the traps by examining every object in laboratory. However, in real online systems where there may be billions of objects, it is impossible to check every of them. On the other hand, not every closed trap harms the system significantly. Only these accounting for a large amount of traffic are breaking down the navigability of the system. As a consequence, an apparent challenge is how to identify significant trap members in very large networks.

We believe that an object with a very short return time (Condamin *et al.* 2007; Perkins *et al.* 2014) in random walks, would be significant trap members. Accordingly, a simple measure, namely the local return rate, can be developed as

$$R_v = (\gamma_v^{(2)} + \gamma_v^{(3)})/2. \tag{7.8}$$

where $\gamma_v^{(n)}$ is the fraction between the number of returning paths $r_v^{(n)}$ with length n, and all the paths with length n originating from object v, i.e.

$$\gamma_v^{(n)} = r_v^{(n)}/L^n. \tag{7.9}$$

A returning path is the path that starts from an object and also ends at it. For example, $i \to j \to i$ is a returning path for object i with length 2.

Actually, the metric can be extended to include more rates for higher-order paths (larger n), hence a more precise form to write the expression is

$$R_v = \sum_{n=1}^{\infty} \gamma_v^{(n)}. \tag{7.10}$$

However, the purpose of developing this metric is to more efficiently deal with very large recommendation networks so that the most significant traps can be quickly identified without costing too much computational power and time. In any networks, to find path with larger length is significantly more time-demanding than finding shorter paths. Therefore, here when applying the local return rate, we still stick with the Eq. (7.8), i.e. only consider paths with length of 2 and 3.

To identify the significant traps, originally one need to check every object as introduced in Section 7.3. With the developed local return rate, one can firstly identify objects with the highest value of R_v, and then further examine how many others can each reach to uncover significant closed traps. To show the power of the local return rate, here we only check 0.1% objects with highest R_v, so that the computational costs would theoretically be 1/1000 of the original methods. The traps identified by checking only the 0.1% highest local return rate are summarised in Table 7.3.

Table 7.3 | Traps identified involving 0.1% objects with highest local return rate in recommendation networks with out-degree $L = 10$.

Dataset	Measure	size	matched	traffic share	accuracy
ToyData	CN	18	18	99.99%	18(18)
	LHN	12	12	1.93%	69(84)
		11	11	17.09%	
		11	11	24.63%	
		11	11	1.82%	
		11	11	13.16%	
		13	13	15.68%	
	RA	No trap			
	AA	24	24	99.99%	24(24)
	HC	11	11	26.39%	61(61)
		11	11	11.02%	
		11	11	8.97%	
		15	15	19.25%	
		13	13	7.84%	
Amazon	CN	16	14	20.97%	26(26)
		12	12	64.12%	
		19	0	0.05%	
	LHN	11	0	1.87%	24(79)
		11	11	13.8%	
		11	0	0.22%	
		11	0	2.68%	
		14	0	4.46%	
		11	0	4.22%	
		14	0	4.59%	
		13	13	8.53%	
	RA	12	12	90.06%	12(12)
	AA	12	12	56.2%	12(27)
	HC	11	11	30.29%	11(67)
Yelp	CN	58	0	3.69%	0(0)
	LHN	20	20	68.82%	42(42)
		11	11	3.82%	
		11	11	20.69%	
	RA	No trap			
	AA	No trap			
	HC	11	11	58.46%	22(22)
		11	11	22.61%	

For a given recommendation network, multiple closed traps can be possibly identified, each corresponding to a record (line) in the table. The size and traffic represent the population and ratio of random walk traffic of the identified trap. Regarding the traps identified in Figure 7.6 and Table 7.2 as the benchmark for the ground truth of significant traps, we also examine how many of the traps identified by local return rate can match with ground-truth taps. Take the CN measure in ToyData as an example, the 18 highest-visiting-frequency objects form a closed trap as shown in Figure 7.6. Thus, we regard these 18 objects as a ground-truth trap, and we examine whether our local return rate method can also uncover such 18 objects. As the results in Table 7.3 suggested, all these 18 objects are found by our method, and thus the number of matched objects is 18 as well. The column of accuracy shows the total number of matched objects and, in bracket, the full population of the top-visiting-frequency trap. The aim, of course, is to match all of the ground-truth trap objects, which is that the value is expected to be as close to the number in bracket as possible.

As suggested by Table 7.3, for most identified closed traps, the members well match with the top-visiting-frequency objects. Though some traps do not match with the top-visiting-frequency trap, their traffic shares are still significantly higher than average. Taking the CN measure in Yelp as an example, the trap with size of 58 is expected to account for 0.09% (58/61184) of the traffic, while actually account for 3.70%, which is about 41 times higher than expected. In addition, one may observe that 11 is the most commonly-appeared size for the closed traps. As the out-degree of every object is fixed at $L = 10$, a population of 11 objects could be the smallest, densest as well as simplest closed trap, where each object connects to every other. As discussed earlier, for any out-degree L especially these smaller values, the majority of traps are with seize $s = L + 1$.

We also examine the efficiency of the proposed local return rate in recommendation networks with different recommendation list length (out-degree) L. We don't aim to uncover every trap in a recommendation network. There are too many trivial closed traps in recommendation networks that are not accounting for significant traffic at all. Accordingly, instead of the number of trap objects, we compare the traffic share to explore whether the local return rate can uncover the significant traps. As has been discussed in the Figure 7.9, all the traps account for a significant amount of traffic, denoting with $F = \sum_{v \in \Gamma} f_v / N$, where Γ is the set of trap objects, identified by traversing all the objects. We also check the traffic share of the traps identified by these 0.1% top-local-return-rate objects, i.e. $F^{lrr} = \sum_{v \in \Gamma'} f_v / N$, where Γ' is the new set of trap objects. Therefore, a large value of F^{lrr}/F would suggest that, the local return rate is efficient in detecting significant traps, and vice versa.

The comparison between the traffic share F^{lrr} and F is shown in Figure 7.10. The recommendation networks with very small out-degree L, are normally segmented into thousands of traps. As a result, there may be less differences between the traffic shares of different traps, and F^{lrr}/F thus generally take small values since much less number of traps can be identified. When the out-degree L is relatively large, for example $L > 3$, the local return rate is shown to be efficient except for the LHN and HC measure in Amazon datasets.

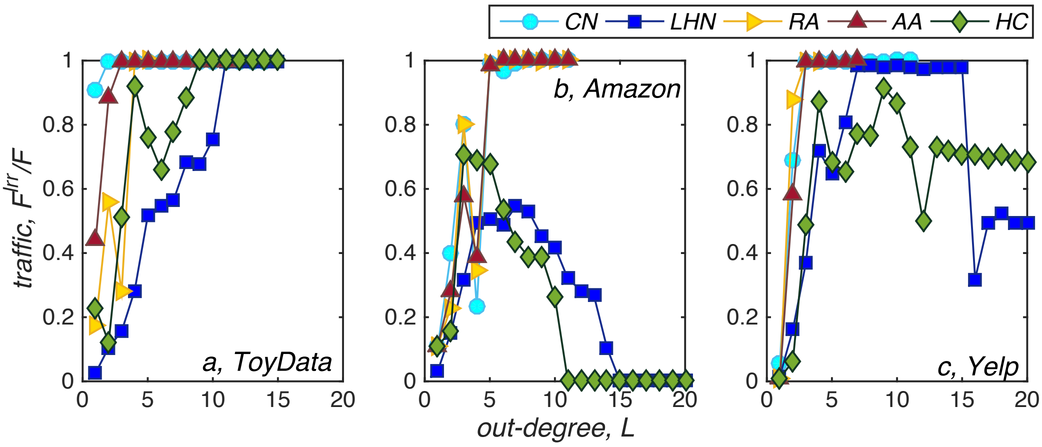


Figure 7.10 | Traffic share of closed traps identified by 0.1% top-local-return-rate objects in comparison to that of all the traps.

7.5. Summary

The recommendation network enables the numerous online objects such as products, movies or restaurants, to connect with each other by hyperlinks. Users can thus surf on the recommendation network to access massive information and find interesting ones. However, the recommendation networks constructed according to the user co-accessing patterns, are found unnavigable with a very limited number of objects to be reached during random walks. The existence of traps is shown to be the underlying reason for the poor navigability, which means users surfing on recommendation networks can easily fall into a small group of objects and stuck to them with little or even no chance of visiting other objects.

The emergence of traps in recommendation networks may be caused by the evaluation of the co-accessing patterns, i.e. the similarity measure. While the user interests, represented by the selection behaviour, is highly clustered (Liu et al. 2013), many similarity measures make the object co-accessing relations even more clustered. The similarity measures studied in this thesis are all collaborative-filtering-based (network-based) and only focus on the local structure of the network. Improvements are thus possible via adopting other approaches such as content-based similarities (Pera & Ng 2003; Lops et al. 2011) or global-based methods (Leicht et al. 2006; Katz 1953). However, the trade-offs between either internal similarity and user co-accessing patterns, or accuracy and computational efficiency, would become challenges.

This Chapter addressed RQ4 of the thesis (are the objects accessible and is the recommendation network navigable?). The short answer to this according to the results in this Chapter is: the objects are not easily accessible and the recommendation networks are totally unnavigable. For an N-steps random walk, only 1% objects in general can be visited, while the expected ratio is about 63%. To deepen the answer to the question, the reason of the networks being unnavigable is the emergence of traps, which has dense internal linkages that block the web traffic.

Chapter 8. Discussions

8.1 Validity Discussions

The validity of the results in this thesis comes from the proper scientific methods we adopted, including the Mathematical analysis, empirical experiments on large data sets, and the comparative analysis. The following is the discussion of the validity of the results in each of the Chapters 4 to 7.

Chapter 4 studies the stability problem of similarity measures and its influence on the recommendations. The major output is the TNS-HC algorithm which is proposed for tackling the stability-accuracy-diversity triple dilemma of the personalised recommendation. Through empirical recommendation experiments on four user-object interaction data sets, it is confirmed that the proposed TNS-HC algorithm can largely improve the recommendation performance from the original HC algorithm, by more than 18% and 45% for the accuracy and stability ($L = 20$). In order to further illustrate the effectiveness of the proposed method, we extensively compare it with other benchmark algorithms. The comparison shows that, the TNS-HC algorithm is among the best ones in terms of either accuracy, stability or diversity of the recommendation. Thus, according to the empirical experiments and comparison with other algorithms, we can conclude that the proposed method is effective and valid for overcoming the challenge of stability-accuracy-diversity triple dilemma.

In Chapter 5, the expected number of common neighbours is firstly studied via adopting the mathematical analysis with the assumption that the network is purely random, i.e. without any influence of user interests. By doing so, the resulted equation for the expected number of common neighbour (Eq. 5.4 and Eq. 5.22) should be valid for real networks without similarity-driven links. However, in real networks there always will be links governed by similarities, and hence the difference between the expected number of common neighbours and observed number of common neighbours is theoretically a representation for the similarity between any nodes (objects). This is the theoretical validity for the proposed BCN similarity measures (Eq. 5.9 and Eq. 5.25). Later on, by adopting the empirical experiment approach, the validation on the performance of BCN in improving personalised recommendations are carried out which proves that the proposed method is efficient to generate accurate and diverse recommendations. Similar to the Chapter 4, at the end of Chapter 5, the proposed method is also validated by comparing with other benchmark algorithms.

Chapter 6 and 7 are more exploratory, looking at the evaluation of recommendation networks. Since there are no previous work on evaluating the performance, we developed the evaluation framework based on mathematical modelling, which is further tested via empirical experiments. In order to validate the framework as well as to assess the results of the evaluation, we compared the empirical results with the theoretical results conducted via mathematical analysis. Through the comparison, the conclusion is made that the recommendation networks are not as efficient as expected in terms of navigating users to find interesting objects or to find diverse objects.

8.2 Limitations

Regarding limitations, this thesis is rather to uncover the challenges for online recommendation and explore its influences, than to deliver comprehensive solutions for the challenges. Though there are some possible solutions, including the top-n-stability algorithm, the balanced common neighbour similarity measure, the optimisation of the recommendation list length, and the local return rate metric, we focused more on the popularity bias of similarity measures and its consequence of failing to discover niche information. For the stability problem, the top-n-stability algorithm though shown good performance, an apparent limitation is its computational complexity. To apply such algorithm in practical systems, one need to calculate the stability for the similarity of every object pair, which is very time-consuming. The balanced common neighbour similarity measure has also similar limitation that it needs optimisation to determine the parameter in the algorithm. In the recommendation networks, optimisation of the recommendation list length and the metric are rather analytical methods to explore the problem, which do not fundamentally overcome the challenges. Hence, comprehensive solutions for the niche object finding should be developed in the future. Possible directions include hybrid similarity measures which combine network-based measure and content-based measure, and new mechanisms to generate recommendations or construct recommendation networks. Since the trade-offs between popularity preference and similarity preference is an important influencer for the online recommendations, to further and more closely study such systems' evolution and the consumer behaviours would also benefit the study of online recommendations.

Chapter 9 Summary

9.1. Concluding Remarks

The scale of the Internet is nowadays increasing vastly in both the amount of information and the number of users. Billions of new records being created every day brings abundant information for online users to access which basically fit every aspect of their daily needs. However, the overwhelming information and the fast evolution not only call for efficient online recommendation systems, but also leave us challenges for making proper recommendations. Focusing on the inability of current similarity measures fulfilling the practical needs of uncovering niche objects due to the apparent degree bias problem, this thesis explores the similarity quantification in complex networks and applies it to online recommendations.

The first challenge brought by the overwhelming information is that how can we accurately recommend users with what they interested in. Accordingly, a large body of research has been addressing such problem with a lot of similarity measures and recommendation algorithms proposed. Another challenge, as well as the urgent need of the users, is that how can we help users to discover niche information. For most users, their interests may consist of both common interests which can be represented by the popular information, and personalised interests which can only be fitted by the niche information. Thus, the accuracy and diversity of the recommendations are equally important. However, the existing similarity measures fail to generate both accurate and diverse recommendations due to the degree bias problem. In this thesis, we address such problem in both scenarios of personalised recommendation and recommendation networks.

We systematically outlined the inability of current network-based similarity measures to fulfil the need of online recommendation to discover niche information. Existing network-based measures were shown unable to stably quantify the similarity among objects, and through experiments, the unstable similarities were proved to be false quantifications, which severely affect the performance of online recommendations in terms of diversity, stability and even accuracy. While the commonly existed popularity bias of these similarity measures mixed the similarity preference with popularity preference, this thesis presented evidences for the fact that controlling the similarity preference and popularity preference separately can largely enhance the performance of personalised recommendations. A theoretical similarity measure without such popularity bias was developed in this thesis for both unipartite and bipartite networks, which may enrich the understanding of the quantification of node similarities in networks. Despite the increasing attention on the value of recommendation networks in recent years, the valuation of the fundamental performances of such system serving users is still a huge gap. For the first time, we comprehensively explored and evaluated the accuracy and navigability of recommendation networks. The popularity bias of similarity measures, again, was shown breaking down the accuracy of such system in terms of navigating surfing users to find interesting niche objects. On the other hand, the traps in recommendation networks were uncovered which explained the poor navigability. These findings may contribute not only to the knowledge of online recommendations, but also to the study of general networked systems such as social systems, and the biological networks etc.

The results in this thesis have also fruitful practical implications and applications. For example, the fact that the unstable similarities are false quantifications, should be considered by the system developers, and in practice, these unstable similarities should be removed to gain better performances. The developed algorithms, namely the TNS-HC and the BCN, are directly applicable to a wide range of practical systems, or can be easily tailored by combining with other methods as hybrid algorithms. This thesis also provided systematic frameworks to check the performance of any given recommendation networks, such as the self-avoiding random walks can be used to evaluate the accuracy of a practical recommendation network, and the local return rate metric can be applied to uncover traps in very-large-scale systems. The exploration of the recommendation list length also informs relevant system designers about how to determine an optimal length for their system. In every study of this thesis, multiple widely-used similarity measures were evaluated and compared. Therefore, the results can inform practitioners with the features of different measures and thereby help them make choice according to their needs.

9.2. Future Work

As has been discussed in section 8.2, the research in this thesis has many limitations. Some possible future directions can be followed to make such line of research more complete and comprehensive.

The first problem is the computational complexity of the proposed personalised recommendation algorithms including the TNS-HC and BCN, since both of them have a free parameter to be trained to achieve best performance. A possible solution of this is to intensively experiment with different kind of systems to conclude the pattern of optimised parameter as the reference for practice with a certain kind of system. For example, in the BCN experiments, the system Last.FM takes a totally different optimised value of $\lambda = 1.4$ in comparison to that of the MovieLens and Netflix which have the value of $\lambda = 0.33$ and $\lambda = 0.36$ respectively. A possible pattern is that the movie systems (MovieLens and Netflix) have smaller such optimised parameters while the music systems (Last.FM) normally have larger parameters. But to confirm this pattern, we need to carry out more experiments in more different systems. By doing so, we may be able to answer questions such as can we infer the optimised parameter by examining the observable feature of a certain system? Such results will largely benefit the practical application via reducing the computational cost of adopting the proposed algorithms.

While the algorithms and recommendation experiments are all carried out based on solely the collaborative filtering, hybrid recommendation systems should be considered in the future. Collaborative filtering is good at quantifying the similarity among objects via collective user behaviours, and hence uncovering hidden association patterns. However, as part of the conclusion of this thesis, such method is shown unable to stably and accurately measure the similarities among unpopular objects. Other methods, such as the content-based systems, will not have such problem.

As a consequence, the integration of collaborative filtering and content-based techniques has the potential to take the advantages from each method and achieve better overall performances.

The recommendation network study in this thesis is exploratory. The developed evaluation framework shows the inability of recommendation networks to guarantee the navigation accuracy and equal information accessibility. How to develop new methods to construct recommendation networks with accurate navigation and equal information accessibility should be the focus of the future research. A simple solution is to explore appropriate similarity measures to apply to the projection of bipartite networks as recommendation networks. But such method may sill result in systemic patterns of recommendation networks leading to information monopoly. Thus, the future research should focus attentions on new construction methods other than the direct projection of bipartite networks. For example, the recommendation networks can be made personalised, i.e. instead of all users share one unified network, each user should have a personalised recommendation network. Another direction may be to empower the memory of recommendation networks so that the network can be dynamically changing. To be more specific, the system should remember what a user has just browsed and remove the recent history from the recommendation network as the user keep surfing. Doing so will prevent users from being trapped, and provide users with more diverse navigations.

References

Abbasi, A., Hossain, L., and Leydesdorff, L. (2012). Betweenness centrality as a driver of preferential attachment in the evolution of research collaboration networks. *Journal of Informetrics* **6(3)**, 403-412.

Adamic, L. A., and Adar, E. (2003). Friends and neighbors on the web. *Social Networks* **25(3)**, 211-230.

Adomavicius, G., and Zhang, J. (2012). Stability of recommendation algorithms. *ACM Transactions on Information Systems (TOIS)* **30(4)**, 23.

Adomavicius, G., and Zhang, J. (2015). Improving stability of recommender systems: A meta-algorithmic approach. *IEEE Transactions on Knowledge and Data Engineering* **27(6)**, 1573-1587.

Aggarwal, C. C. (2016). Content-based recommender systems. In: *Recommender Systems* (pp. 139-166). Springer, Cham.

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

Albert, R., Jeong, H., and Barabási, A. L. (1999). Internet: Diameter of the world-wide web. *Nature* **401(6749)**, 130.

Albert, R., Jeong, H., and Barabási, A. L. (2000). Error and attack tolerance of complex networks. *Nature* **406(6794)**, 378.

Alvarez-Galvez, J. (2016). Discovering complex interrelationships between socioeconomic status and health in Europe: A case study applying Bayesian Networks. *Social Science Research* **56**, 133-143.

Aral, S., and Nicolaides, C. (2017). Exercise contagion in a global social network. *Nature Communication* **8**, 14753.

Arenas, A., Díaz-Guilera, A., Kurths, J., Moreno, Y., and Zhou, C. (2008). Synchronization in complex networks. *Physics Reports* **469(3)**, 93-153.

Armano, G., and Javarone, M. A. (2013). Clustering datasets by complex networks analysis. *Complex Adaptive Systems Modeling* 1(1), 5.

Backstrom, L., Boldi, P., Rosa, M., Ugander, J., and Vigna, S. (2012, June). Four degrees of separation. In: *Proceedings of the 4th Annual ACM Web Science Conference* (pp. 33-42). Evanston, Illinois, USA.

Bagrow, J. P., and Brockmann, D. (2013). Natural emergence of clusters and bursts in network evolution. *Physical Review X* **3(2)**, 021016.

Banerjee, S., Jenamani, M., and Pratihar, D. K. (2017, August). Algorithms for projecting a bipartite network. In: *2017 Tenth International Conference on Contemporary Computing* (pp. 1-3). IEEE.

Barabási, A. L. (2016). Network science. Cambridge university press.

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

Barabási, A. L., Albert, R., and Jeong, H. (2000). Scale-free characteristics of random networks: the topology of the world-wide web. *Physica A* **281(1-4)**, 69-77.

Barber, M. J., Krueger, A., Krueger, T., and Roediger-Schluga, T. (2006). Network of European Union–funded collaborative research and development projects. *Physical Review E* **73(3)**, 036132.

Bawden, D., Holtham, C. and Courtney, N. (1999). Perspectives on information overload. *Aslib Proceedings* **51(8)**, 249-255.

Beguerisse Díaz, M., Porter, M. A., and Onnela, J. P. (2010). Competition for popularity in bipartite networks. *Chaos* **20(4)**, 043101.

Beuming, T., Skrabanek, L., Niv, M. Y., Mukherjee, P., and Weinstein, H. (2004). PDZBase: a protein–protein interaction database for PDZ-domains. *Bioinformatics* **21(6)**, 827-828.

Bilenko, M., and White, R. W. (2008, April). Mining the search trails of surfing crowds: identifying relevant websites from user activity. In: *Proceedings of the 17th international conference on World Wide Web* (pp. 51-60). ACM.

Bobadilla, J., Ortega, F., Hernando, A., and Alcalá, J. (2011). Improving collaborative filtering recommender system results and performance using genetic algorithms. *Knowledge-Based Systems* **24(8)**, 1310-1316.

Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., and Hwang, D. U. (2006). Complex networks: Structure and dynamics. *Physics Reports* **424(4-5)**, 175-308.

Boguná, M., Krioukov, D., and Claffy, K. C. (2009). Navigability of complex networks. *Nature Physics* **5(1)**, 74.

Boguná, M., Papadopoulos, F., and Krioukov, D. (2010). Sustaining the internet with hyperbolic mapping. *Nature Communications* **1**, 62.

Bonahon, F. (2009). Low-dimensional geometry: From Euclidean surfaces to hyperbolic knots. American Mathematical Society.

Bonaventura, M., Nicosia, V., and Latora, V. (2014). Characteristic times of biased random walks on complex networks. *Physical Review E* **89(1)**, 012803.

Breese, J. S., Heckerman, D., and Kadie, C. (1998, July). Empirical analysis of predictive algorithms for collaborative filtering. In: *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence* (pp. 43-52). Morgan Kaufmann Publishers Inc.

Brockmann, D., and Helbing, D. (2013). The hidden geometry of complex, network-driven contagion phenomena. *Science* **342(6164)**, 1337-1342.

Buldú, J. M., Cano, P., Koppenberger, M., Almendral, J. A., and Boccaletti, S. (2007). The complex network of musical tastes. *New Journal of Physics* **9(6)**, 172.

Buldyrev, S. V., Parshani, R., Paul, G., Stanley, H. E., and Havlin, S. (2010). Catastrophic cascade of failures in interdependent networks. *Nature* **464(7291)**, 1025.

Burke, R., O'Mahony, M. P., and Hurley, N. J. (2015). Robust collaborative recommendation. In: *Recommender systems handbook* (pp. 961-995). Springer, Boston, MA.

Cannistraci, C. V., Alanis-Lobato, G., and Ravasi, T. (2013). From link-prediction in brain connectomes and protein interactomes to the local-community-paradigm in complex networks. *Scientific Reports* **3**, 1613.

Cano, P., Celma, O., Koppenberger, M., and Buldu, J. M. (2006). Topology of music recommendation networks. *Chaos* **16(1)**, 013107.

Cantador, I., Brusilovsky, P. L., and Kuflik, T. (2011). Second workshop on information heterogeneity and fusion in recommender systems (HetRec2011). ACM.

Carmi, E., Oestreicher-Singer, G., and Sundararajan, A. (2009). Spreading the Oprah effect: The diffusion of demand shocks in a recommendation network. In: *Thirtieth International Conference on Information Systems* (p. 78).

Carmi, E., Oestreicher-Singer, G., Stettner, U., and Sundararajan, A. (2017). Is Oprah Contagious? The Depth of Diffusion of Demand Shocks in a Product Network. *MIS Quarterly* **41(1)**, 207-221.

Cataldi, M., Di Caro, L., and Schifanella, C. (2010, July). Emerging topic detection on twitter based on temporal and social terms evaluation. In: *Proceedings of the tenth international workshop on multimedia data mining* (p. 4). ACM.

Celma, Ò., and Cano, P. (2008, August). From hits to niches?: or how popular artists can bias music recommendation and discovery. In: *Proceedings of the 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition* (p. 5). ACM.

Chen, L. J., Zhang, Z. K., Liu, J. H., Gao, J., and Zhou, T. (2017). A vertex similarity index for better personalized recommendation. *Physica A* **466**, 607-615.

Cheng, X. Q., Ren, F. X., Zhou, S., and Hu, M. B. (2009). Triangular clustering in document networks. *New Journal of Physics* **11(3)**, 033019.

Cho, E., Myers, S. A., and Leskovec, J. (2011, August). Friendship and mobility: user movement in location-based social networks. In: *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 1082-1090). ACM.

Christakou, C., Vrettos, S., and Stafylopatis, A. (2007). A hybrid movie recommender system based on neural networks. *International Journal on Artificial Intelligence Tools* **16(05)**, 771-792.

Chung, F., and Lu, L. (2002a). Connected components in random graphs with given expected degree sequences. *Annals of Combinatorics* **6(2)**, 125-145.

Chung, F., and Lu, L. (2002b). The average distances in random graphs with given expected degrees. *Proceedings of the National Academy of Sciences USA* **99(25)**, 15879-15882.

Chupeau, M., Bénichou, O., and Voituriez, R. (2015). Cover times of random searches. *Nature Physics* **11(10)**, 844.

Clauset, A., Moore, C., and Newman, M. E. (2008). Hierarchical structure and the prediction of missing links in networks. *Nature* **453(7191)**, 98.

Cline, M. S., Smoot, M., Cerami, E., Kuchinsky, A., Landys, N., Workman, C. *et al.* (2007). Integration of biological networks and gene expression data using Cytoscape. *Nature Protocols* **2(10)**, 2366.

Colizza, V., Barrat, A., Barthélemy, M., and Vespignani, A. (2006). The role of the airline transportation network in the prediction and predictability of global epidemics. *Proceedings of the National Academy of Sciences USA* **103(7)**, 2015-2020.

Condamin, S., Bénichou, O., Tejedor, V., Voituriez, R., and Klafter, J. (2007). First-passage times in complex scale-invariant media. *Nature* **450(7166)**, 77.

Conover, M., Ratkiewicz, J., Francisco, M. R., Gonçalves, B., Menczer, F., and Flammini, A. (2011). Political polarization on twitter. *ICWSM* **133**, 89-96.

Covington, P., Adams, J., and Sargin, E. (2016, September). Deep neural networks for youtube recommendations. In: *Proceedings of the 10th ACM Conference on Recommender Systems* (pp. 191-198). ACM.

Crandall, D. J., Backstrom, L., Cosley, D., Suri, S., Huttenlocher, D., and Kleinberg, J. (2010). Inferring social ties from geographic coincidences. *Proceedings of the National Academy of Sciences USA* **107(52)**, 22436-22441.

Crucitti, P., Latora, V., and Marchiori, M. (2004). Model for cascading failures in complex networks. *Physical Review E* **69(4)**, 045104.

Cui, X. M., Kim, W. S., Hwang, D. U., and Han, S. K. (2015). Estimation of inter-modular connectivity from the local field potentials in a hierarchical modular network. *EPL (Europhysics Letters)* **110(3)**, 38001.

Daminelli, S., Thomas, J. M., Durán, C., and Cannistraci, C. V. (2015). Common neighbours and the local-community-paradigm for topological link prediction in bipartite networks. *New Journal of Physics* **17(11)**, 113037.

Davidson, J., Liebald, B., Liu, J., Nandy, P., Van Vleet, T., Gargi, U. *et al.* (2010, September). The YouTube video recommendation system. In: *Proceedings of the fourth ACM conference on recommender systems* (pp. 293-296). ACM.

De Corniere, A., and Taylor, G. (2014). Integration and search engine bias. *The RAND Journal of Economics* **45(3)**, 576-597.

De Domenico, M., Solé-Ribalta, A., Gómez, S., and Arenas, A. (2014). Navigability of interconnected networks under random failures. *Proceedings of the National Academy of Sciences USA* **111(23)**, 8351-8356.

Del Vicario, M., Zollo, F., Caldarelli, G., Scala, A., and Quattrociocchi, W. (2017). Mapping social dynamics on Facebook: The Brexit debate. *Social Networks* **50**, 6-16.

Durrett, R. (2010). Some features of the spread of epidemics and information on a random graph. *Proceedings of the National Academy of Sciences USA* **107(10)**, 4491-4498.

Eppler, M. J. and Mengis, J. (2004). The concept of information overload: A review of literature from organization science, accounting, marketing, MIS, and related disciplines. *The Information Society* **20(5)**, 325-344.

Erdös, P. and Rényi, A. (1959). On random graphs, I. *Publicationes Mathematicae (Debrecen)* **6**, 290-297.

Erdös, P. and Rényi, A. (1960). On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci.* **5(1)**, 17-60.

Fortunato, S., Flammini, A., and Menczer, F. (2006a). Scale-free network growth by ranking. *Physical Review Letters* **96(21)**, 218701.

Fortunato, S., Flammini, A., Menczer, F., and Vespignani, A. (2006b). Topical interests and the mitigation of search engine bias. *Proceedings of the National Academy of Sciences USA* **103(34)**, 12684-12689.

Fortunato, S., and Hric, D. (2016). Community detection in networks: A user guide. *Physics Reports* **659**, 1-44.

Frey, B. J., and Dueck, D. (2007). Clustering by passing messages between data points. *Science* **315(5814)**, 972-976.

Fronczak, A., and Fronczak, P. (2009). Biased random walks in complex networks: The role of local navigation rules. *Physical Review E* 80(1), 016107.

Gao, J., Barzel, B., and Barabási, A. L. (2016). Universal resilience patterns in complex networks. *Nature* **530(7590)**, 307.

Gao, J., Buldyrev, S. V., Havlin, S., and Stanley, H. E. (2011). Robustness of a network of networks. *Physical Review Letters 107(19)*, 195701.

Ghoshal, G., and Barabási, A. L. (2011). Ranking stability and super-stable nodes in complex networks. *Nature Communications* **2**, 394.

Girvan, M., and Newman, M. E. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences USA* **199(12)**, 7821-7826.

Goldberg, D., Nichols, D., Oki, B. M. and Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM* **35(12)**, 61-70.

Goldenberg, J., Oestreicher-Singer, G., and Reichman, S. (2012). The quest for content: How user-generated links can facilitate online exploration. *Journal of Marketing Research* **49(4)**, 452-468.

Gómez-Gardenes, J., Gómez, S., Arenas, A., and Moreno, Y. (2011). Explosive synchronization transitions in scale-free networks. *Physical Review Letters* **106(12)**, 128701.

Grant, T. R. (1973). Dominance and association among members of a captive and a free-ranging group of grey kangaroos (Macropus giganteus). *Animal Behaviour* **21(3)**, 449-456.

Guimera, R., Danon, L., Diaz-Guilera, A., Giralt, F., and Arenas, A. (2003). Self-similar community structure in a network of human interactions. *Physical Review E* **68(6)**, 065103.

Guimera, R., Uzzi, B., Spiro, J., and Amaral, L. A. N. (2005). Team assembly mechanisms determine collaboration network structure and team performance. *Science* **308(5722)**, 697-702.

Hanani, U., Shapira, B. and Shoval, P. (2001). Information filtering: Overview of issues, research and systems. *User Modeling and User-Adapted Interaction* **11(3)**, pp.203-259.

Hassan, M., and Hamada, M. (2017, September). Improving prediction accuracy of multi-criteria recommender systems using adaptive genetic algorithms. In: *Intelligent Systems Conference* (pp. 326-330). IEEE.

Henzinger, M., and Lawrence, S. (2004). Extracting knowledge from the world wide web. *Proceedings of the National Academy of Sciences USA* **101(suppl 1)**, 5186-5191.

Hernando, A., Bobadilla, J., and Ortega, F. (2016). A non negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model. *Knowledge-Based Systems* **97**, 188-202.

Herrero, C. P. (2005). Self-avoiding walks on scale-free networks. *Physical Review E* **71(1)**, 016103.

Holme, P., and Saramäki, J. (2012). Temporal networks. *Physics Reports* **519(3)**, 97-125.

Huberman, B. A., Pirolli, P. L., Pitkow, J. E., and Lukose, R. M. (1998). Strong regularities in world wide web surfing. *Science* **280(5360)**, 95-97.

Hurley, N. and Zhang, M. (2011). Novelty and diversity in top-n recommendation--analysis and evaluation. In: *ACM Transactions on Internet Technology (TOIT)* **10(4)**, 14.

Hwang, S., Lee, D. S., and Kahng, B. (2012). First passage time for random walks in heterogeneous networks. *Physical Review Letters* **109(8)**, 088701.

Hwang, M. I. and Lin, J. W. (1999). Information dimension, information overload and decision quality. *Journal of Information Science* **25(3)**, 213-218.

Iacopini, I., Milojević, S., and Latora, V. (2018). Network dynamics of innovation processes. *Physical Review Letters* **120(4)**, 048301.

Jeong, H., Néda, Z., and Barabási, A. L. (2003). Measuring preferential attachment in evolving networks. *EPL (Europhysics Letters)* **61(4)**, 567.

Kang, M., and Seierstad, T. G. (2007). Phase transition of the minimum degree random multigraph process. *Random Structures & Algorithms* **31(3)**, 330-353.

Katz, L. (1953). A new status index derived from sociometric analysis. *Psychometrika* **18(1)**, 39-43.

Katzman, J. L., Shaham, U., Cloninger, A., Bates, J., Jiang, T., and Kluger, Y. (2018). DeepSurv: Personalized treatment recommender system using A Cox proportional hazards deep neural network. *BMC Medical Research Methodology* **18(1)**, 24.

Kim, Y., Park, S., and Yook, S. H. (2016). Network exploration using true self-avoiding walks. *Physical Review E* **94(4)**, 042309.

Kitsak, M., Gallos, L. K., Havlin, S., Liljeros, F., Muchnik, L., Stanley, H. E., and Makse, H. A. (2010). Identification of influential spreaders in complex networks. *Nature Physics* **6(11)**, 888.

Kleinberg, J., and Lawrence, S. (2001). The structure of the Web. *Science* **294(5548)**, 1849-1850.

Konstas, I., Stathopoulos, V., and Jose, J. M. (2009, July). On social networks and collaborative recommendation. In: *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval* (pp. 195-202). ACM.

Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer* **42(8)**, 42-49.

Kossinets, G., and Watts, D. J. (2006). Empirical analysis of an evolving social network. *Science* **311(5757)**, 88-90.

Krioukov, D., Papadopoulos, F., Kitsak, M., Vahdat, A., and Boguná, M. (2010). Hyperbolic geometry of complex networks*. Physical Review E* **82(3)**, 036106.

Lambiotte, R., and Ausloos, M. (2005). Uncovering collective listening habits and music genres in bipartite networks. *Physical Review E* **72(6)**, 066107.

Latora, V., and Marchiori, M. (2003). Economic small-world behavior in weighted networks. *The European Physical Journal B* **32(2)**, 249-263.

Leem, B., and Chun, H. (2014). An impact of online recommendation network on demand. *Expert Systems with Applications* **41(4)**, 1723-1729.

Leicht, E. A., Holme, P., and Newman, M. E. (2006). Vertex similarity in networks. *Physical Review E* **73(2)**, 026120.

Leicht, E. A., and Newman, M. E. (2008). Community structure in directed networks. *Physical Review Letters* **100(11)**, 118703.

Leskovec, J., Kleinberg, J., and Faloutsos, C. (2007). Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data (TKDD)* **1(1)**, 2.

Leskovec, J., Lang, K. J., Dasgupta, A., and Mahoney, M. W. (2009). Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics* **6(1)**, 29-123.

Lin, Z., Goh, K. Y., and Heng, C. S. (2017). The demand effects of product recommendation networks: an empirical analysis of network diversity and stability. *MIS Quarterly* **41(2)**, 397-426.

Lin, Z., and Wang, Q. (2018). E-commerce Product Networks, Word-of-mouth Convergence, and Product Sales. *Journal of the Association for Information Systems* **19(1)**, 23-39.

Lin, Y., and Zhang, Z. (2014). Mean first-passage time for maximal-entropy random walks in complex networks. *Scientific Reports* **4**, 5365.

Lind, P. G., and Herrmann, H. J. (2007). New approaches to model and study social networks. *New Journal of Physics* **9(7)**, 228.

Lind, P. G., Gonzalez, M. C., and Herrmann, H. J. (2005). Cycles and clustering in bipartite networks. *Physical Review E* **72(5)**, 056127.

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

Liu, J., Hou, L., Zhang, Y. L., Song, W. J., and Pan, X. (2013a). Empirical analysis of the clustering coefficient in the user-object bipartite networks. *International Journal of Modern Physics C* **24(08)**, 1350055.

Liu, J., Hu, Z., and Guo, Q. (2013b). Effect of the social influence on topological properties of user-object bipartite networks. *The European Physical Journal B* **86(11)**, 478.

Liu, J., Zhou, T., and Guo, Q. (2011). Information filtering via biased heat conduction. *Physical Review E*, 84(3), 037101.

Liu, Y. Y., and Barabási, A. L. (2016). Control principles of complex systems. *Reviews of Modern Physics* **88(3)**, 035006.

Liu, Y. Y., Slotine, J. J., and Barabási, A. L. (2011). Controllability of complex networks. *Nature* **473(7346)**, 167.

Lops, P., De Gemmis, M., and Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In: *Recommender Systems Handbook* (pp. 73-105). Springer, Boston, MA.

Lu, J., Wu, D., Mao, M., Wang, W., and Zhang, G. (2015). Recommender system application developments: a survey. *Decision Support Systems* **74**, 12-32.

Luong, N. T., Nguyen, T. T., Hwang, D., Lee, C. H., and Jung, J. J. (2015). Similarity-based Complex Publication Network Analytics for Recommending Potential Collaborations. *J. UCS* **21(6)**, 871-889.

Lusseau, D., Schneider, K., Boisseau, O. J., Haase, P., Slooten, E., and Dawson, S. M. (2003). The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology* **54(4)**, 396-405.

Lv, L., Medo, M., Yeung, C. H., Zhang, Y. C., Zhang, Z. K. and Zhou, T. (2012). Recommender systems. *Physics Reports* **519(1)**, 1-49.

Lv, L., Chen, D., Ren, X. L., Zhang, Q. M., Zhang, Y. C., and Zhou, T. (2016). Vital nodes identification in complex networks. *Physics Reports* **650**, 1-63.

Mäenpää, E., and Jalovaara, M. (2014). Homogamy in socio-economic background and education, and the dissolution of cohabiting unions. *Demographic Research* **30**, 1769-1792.

Maier, B. F., and Brockmann, D. (2017). Cover time for random walks on arbitrary complex networks. *Physical Review E* **96(4)**, 042307.

Manshour, P., and Montakhab, A. (2014). Contagion spreading on complex networks with local deterministic dynamics. *Communications in Nonlinear Science and Numerical Simulation* **19(7)**, 2414-2422.

Massa, P., and Avesani, P. (2007, October). Trust-aware recommender systems. In: *Proceedings of the 2007 ACM conference on Recommender systems* (pp. 17-24). ACM.

Massa, P., and Avesani, P. (2009). Trust metrics in recommender systems. In: *Computing with social trust* (pp. 259-285). Springer, London.

Masuda, N., Porter, M. A., and Lambiotte, R. (2017). Random walks and diffusion on networks. *Physics Reports* **716-717**, 1-58.

McNee, S. M., Riedl, J., and Konstan, J. A. (2006, April). Being accurate is not enough: how accuracy metrics have hurt recommender systems. In: *CHI'06 extended abstracts on Human factors in computing systems* (pp. 1097-1101). ACM.

Mobasher, B., Dai, H., Luo, T., and Nakagawa, M. (2001, November). Effective personalization based on association rule discovery from web usage data. In: *Proceedings of the 3rd international workshop on Web information and data management* (pp. 9-15). ACM.

Mondragon, R. J. (2017). Core-biased random walks in complex networks. arXiv preprint arXiv:1709.07715.

Moreno, Y., Nekovee, M., and Pacheco, A. F. (2004). Dynamics of rumor spreading in complex networks. *Physical Review E* **69(6)**, 066130.

Nekovee, M., Moreno, Y., Bianconi, G., and Marsili, M. (2007). Theory of rumour spreading in complex social networks. *Physica A* **374(1)**, 457-470.

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

Newman, M. E. (2001b). Clustering and preferential attachment in growing networks. *Physical Review E* **64(2)**, 025102.

Newman, M. E. (2002). Assortative mixing in networks. *Physical Review Letters* **89(20)**, 208701.

Newman, M. E. (2003). The structure and function of complex networks. *SIAM Review* **45(2)**, 167-256.

Newman, M. E. (2005). A measure of betweenness centrality based on random walks. *Social networks* **27(1)**, 39-54.

Newman, M. E., Barabási, A. L., and Watts, D. J. (2006). The structure and dynamics of networks. Princeton University Press.

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

Newman, M. E., Watts, D. J., and Strogatz, S. H. (2002). Random graph models of social networks. *Proceedings of the National Academy of Sciences USA* **99**, 2566-2572.

Nguyen, T. N., Kanhabua, N., Niederée, C., and Zhu, X. (2015, August). A time-aware random walk model for finding important documents in web archives. In: *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 915-918). ACM.

Nilashi, M., bin Ibrahim, O., and Ithnin, N. (2014). Multi-criteria collaborative filtering with high accuracy using higher order singular value decomposition and Neuro-Fuzzy system. *Knowledge-Based Systems* **60**, 82-101.

Ning, X., Desrosiers, C., and Karypis, G. (2015). A comprehensive survey of neighborhood-based recommendation methods. In: *Recommender systems handbook* (pp. 37-76). Springer, Boston, MA.

Noh, J. D., and Rieger, H. (2004). Random walks on complex networks. *Physical Review Letters* **92(11)**, 118701.

Oestreicher-Singer, G., Libai, B., Sivan, L., Carmi, E., and Yassin, O. (2013). The network value of products. *Journal of Marketing* **77(3)**, 1-14.

Oestreicher-Singer, G., and Sundararajan, A. (2012a). Recommendation networks and the long tail of electronic commerce. *MIS Quarterly* **36(1)**, 65-83.

Oestreicher-Singer, G., and Sundararajan, A. (2012b). The visible hand? Demand effects of recommendation networks in electronic markets. *Management Science* **58(11)**, 1963-1981.

O'Mahony, M., Hurley, N., Kushmerick, N., and Silvestre, G. (2004). Collaborative recommendation: A robustness analysis. *ACM Transactions on Internet Technology (TOIT)* **4(4)**, 344-377.

Pan, X., Hou, L., and Liu, K. (2017). Social influence on selection behaviour: Distinguishing local-and global-driven preferential attachment. *PLoS ONE* **12(4)**, e0175761.

Papadopoulos, F., Kitsak, M., Serrano, M. Á., Boguná, M., and Krioukov, D. (2012). Popularity versus similarity in growing networks. *Nature* **489(7417)**, 537.

Park, D. H., Kim, H. K., Choi, I. Y., and Kim, J. K. (2012). A literature review and classification of recommender systems research. *Expert Systems with Applications* **39(11)**, 10059-10072.

Pastor-Satorras, R., Castellano, C., Van Mieghem, P., and Vespignani, A. (2015). Epidemic processes in complex networks. *Reviews of Modern Physics* **87(3)**, 925.

Pastor-Satorras, R., Vázquez, A., and Vespignani, A. (2001). Dynamical and correlation properties of the Internet. *Physical Review Letters* **87(25)**, 258701.

Pastor-Satorras, R., and Vespignani, A. (2001). Epidemic dynamics and endemic states in complex networks. *Physical Review E* **63(6)**, 066117.

Pazzani, M. J., and Billsus, D. (2007). Content-based recommendation systems. In: *The adaptive web* (pp. 325-341). Springer, Berlin, Heidelberg.

Pera, M. S., and Ng, Y. K. (2013). A group recommender for movies based on content similarity and popularity. *Information Processing and Management* **49(3)**, 673-687.

Perkins, T. J., Foxall, E., Glass, L., and Edwards, R. (2014). A scaling law for random walks on networks. *Nature Communications* **5**, 5121.

Pongnumkul, S., and Motohashi, K. (2018). A bipartite fitness model for online music streaming services. *Physica A* **490**, 1125-1137.

Ratkiewicz, J., Fortunato, S., Flammini, A., Menczer, F., and Vespignani, A. (2010). Characterizing and modeling the dynamics of online popularity. *Physical Review Letters* **105(15)**, 158701.

Ravasz, E., Somera, A. L., Mongru, D. A., Oltvai, Z. N., and Barabási, A. L. (2002). Hierarchical organization of modularity in metabolic networks. *Science* **297(5586)**, 1551-1555.

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. (1994, October). GroupLens: an open architecture for collaborative filtering of netnews. In: *Proceedings of the 1994 ACM conference on Computer supported cooperative work* (pp. 175-186). ACM.

Resnick, P., and Varian, H. R. (1997). Recommender systems. *Communications of the ACM* ***40*****(3)**, 56-58.

Ribeiro, B., Perra, N., and Baronchelli, A. (2013). Quantifying the effect of temporal resolution on time-varying networks. *Scientific Reports* **3**, 3006.

Ricci, F., Rokach, L., and Shapira, B. (2011). Introduction to recommender systems handbook. In: *Recommender Systems Handbook* (pp. 1-35). Springer US.

Roediger-Schluga, T., and Barber, M. J. (2008). R&D collaboration networks in the European Framework Programmes: Data processing, network construction and selected results. *International Journal of Foresight and Innovation Policy* **4(3-4)**, 321-347.

Rosvall, M., and Bergstrom, C. T. (2008). Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences USA* **105(4)**, 1118-1123.

Sandvig, J. J., Mobasher, B. and Burke, R. (2007). Robustness of collaborative recommendation based on association rule mining. In: *Proceedings of the 2007 ACM conference on Recommender systems* (pp. 105-112) New York, NY, USA.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001, April). Item-based collaborative filtering recommendation algorithms. In: *Proceedings of the 10th international conference on World Wide Web* (pp. 285-295). ACM.

Schafer, J. B., Frankowski, D., Herlocker, J., and Sen, S. (2007). Collaborative filtering recommender systems. In: *The Adaptive Web* (pp. 291-324). Springer, Berlin, Heidelberg.

Schafer, J. B., Konstan, J. A., and Riedl, J. (2001). E-commerce recommendation applications. In: *Applications of Data Mining to Electronic Commerce* (pp.115-153). Springer US.

Shang, M. S., Lv, L., Zhang, Y. C., and Zhou, T. (2010). Empirical analysis of web-based user-object bipartite networks. *EPL (Europhysics Letters)* **90(4)**, 48006.

Silva, T. C., and Zhao, L. (2016). Machine learning in complex networks (Vol. 2016). Switzerland: Springer.

Sinatra, R., Gómez-Gardenes, J., Lambiotte, R., Nicosia, V., and Latora, V. (2011). Maximal-entropy random walks in complex networks with limited information. *Physical Review E* **83(3)**, 030103.

Soffer, S. N., and Vazquez, A. (2005). Network clustering coefficient without degree-correlation biases. *Physical Review E* **71(5)**, 057101.

Stumpf, M. P., Wiuf, C., and May, R. M. (2005). Subnets of scale-free networks are not scale-free: sampling properties of networks. *Proceedings of the National Academy of Sciences USA* **102(12)**, 4221-4224.

Sundaresan, S. R., Fischhoff, I. R., Dushoff, J., and Rubenstein, D. I. (2007). Network metrics reveal differences in social organization between two fission–fusion species, Grevy's zebra and onager. *Oecologia* **151(1)**, 140-149.

Tarus, J. K., Niu, Z., and Yousif, A. (2017). A hybrid knowledge-based recommender system for e-learning based on ontology and sequential pattern mining. *Future Generation Computer Systems* **72**, 37-48.

Thiery, J. P., and Sleeman, J. P. (2006). Complex networks orchestrate epithelial–mesenchymal transitions. *Nature Reviews Molecular Cell Biology* **7(2)**, 131.

Travers, J., and Milgram, S. (1967). The small world problem. *Phychology Today* **1(1)**, 61-67.

Vargas, S., and Castells, P. (2011, October). Rank and relevance in novelty and diversity metrics for recommender systems. In: *Proceedings of the fifth ACM conference on Recommender systems* (pp. 109-116). ACM.

Vespignani, A. (2012). Modelling dynamical processes in complex socio-technical systems. *Nature Physics* **8(1)**, 32.

Viswanath, B., Mislove, A., Cha, M., and Gummadi, K. P. (2009, August). On the evolution of user interaction in facebook. In: *Proceedings of the 2nd ACM workshop on Online social networks* (pp. 37-42). ACM.

Wang, Y., Yin, G., Cai, Z., Dong, Y., and Dong, H. (2015). A trust-based probabilistic recommendation model for social networks. *Journal of Network and Computer Applications* **55**, 59-67.

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

Wilson, C., Boe, B., Sala, A., Puttaswamy, K. P., and Zhao, B. Y. (2009, April). User interactions in social networks and their implications. In: *Proceedings of the 4th ACM European conference on Computer systems* (pp. 205-218) Nuremberg, Germany.

Wu, T., Yu, S. H., Liao, W., and Chang, C. S. (2014, October). Temporal bipartite projection and link prediction for online social networks. In: *2014 IEEE International Conference on* Big Data (pp. 52-59). IEEE.

Xu, C. (2018). A novel recommendation method based on social network using matrix factorization technique. *Information Processing & Management* **54(3)**, 463-474.

Xu, J., Wang, S., Su, S., Kumar, S. A., and Wu, C. (2016, June). Latent interest and topic mining on user-item bipartite networks. In: 2016 IEEE International Conference on Services Computing (SCC) (pp. 778-781). IEEE.

Zafarani, R., and Liu, H. (2009). Social Computing Data Repository at ASU [http://socialcomputing. asu. edu]. Tempe, AZ: Arizona State University, School of Computing. Informatics and Decision Systems Engineering.

Zeng, A., Vidmer, A., Medo, M., and Zhang, Y. C. (2014). Information filtering by similarity-preferential diffusion processes. *EPL (Europhysics Letters)* **105(5)**, 58002.

Zhang, Y. C., Blattner, M., and Yu, Y. K. (2007). Heat conduction process on community networks as a recommendation model. *Physical Review Letters* **99(15)**, 154301.

Zhang, Z. K., Zhou, T., and Zhang, Y. C. (2011). Tag-aware recommender systems: a state-of-the-art survey. *Journal of Computer Science and Technology* **26(5)**, 767.

Zhao, Z. D., and Shang, M. S. (2010, January). User-based collaborative-filtering recommendation algorithms on hadoop. In: *Third International Conference on Knowledge Discovery and Data Mining* (pp. 478-481). IEEE.

Zhou, T., Kuscsik, Z., Liu, J. G., Medo, M., Wakeling, J. R., and Zhang, Y. C. (2010). Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings of the National Academy of Sciences USA* **107(10)**, 4511-4515.

Zhou, T., Lv, L., and Zhang, Y. C. (2009). Predicting missing links via local information. *The European Physical Journal B* **71(4)**, 623-630.

Zhou, T., Ren, J., Medo, M., and Zhang, Y. C. (2007). Bipartite network projection and personal recommendation. *Physical Review E* **76(4)**, 046115.

Ziegler, C. N., McNee, S. M., Konstan, J. A., and Lausen, G. (2005, May). Improving recommendation lists through topic diversification. In: *Proceedings of the 14th international conference on World Wide Web* (pp. 22-32). ACM.

Zweig, K. A., and Kaufmann, M. (2011). A systematic approach to the one-mode projection of bipartite graphs. Social Network Analysis and Mining, 1(3), 187-218.