UNIVERSITY
OF TECHNOLOGY
SYDNEY

Enhanced Recommender Systems with Diffusion Dynamics and Machine Learning

by Ximeng Wang

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

under the supervision of
Associate Professor Guangquan Zhang
and Distinguished Professor Jie Lu

University of Technology Sydney
Faculty of Engineering and Information Technology
Australian Artificial Intelligence Institute

June 2022

Certificate of Original Authorship

I, *Ximeng Wang*, declare that this thesis is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the School of Computer Science at the Faculty of Engineering and Information Technology at the University of Technology Sydney. This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis. I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of the requirements for a degree at any other academic institution except as fully acknowledged within the text. This thesis is the result of a Collaborative Doctoral Research Degree program with Beijing Jiaotong University. This research is supported by the Australian Government Research Training Program.

SIGNATURE: Production Note:
Signature removed prior to publication.

DATE: 26, June, 2022

ACKNOWLEDGMENTS

It is a memorial and exciting journey at the University of Technology Sydney for pursuing my Ph.D. degree in the past years. I am sincerely grateful to the people who inspired and helped me in many ways.

I would like to express my thankfulness to my principal supervisor, Associate Professor Guangquan Zhang. Without his encouragement and guidance, I would not have been able to complete this Ph.D. program. His wisdom and critical comments leave a deep impression on me, which point out the directions of my research. Meanwhile, I am greatly indebted to my co-advisor, Distinguished Professor Jie Lu. Her confidence and enthusiasm inspired me to do the right thing even when the road got tough. She facilitated the collaborative doctoral research degree program between UTS and BJTU, and offered me an opportunity to join UTS. It is fair to say that I would not have been able to study at UTS without her contribution and help. In addition, I express sincere gratitude to my advisors at BJTU, Professor Yun Liu and Professor Fei Xiong. Their urge and instruction motivated and encouraged me to face difficulties and challenges positively.

I would like to thank every member of the DeSI Lab in the Australian Artificial Intelligence Institute. It was a wonderful experience to learn from these dedicated researchers. I especially thank Dr. Yi Zhang, Dr. Feng Liu, Dr. Hua Zuo, Dr. Junyun Xuan and Dr. Zheng Yan who provided insightful comments related to my research problems during my Ph.D. candidature, and thank Dr. Anjin Liu, Dr. Hang Yu, Dr. Ruiping Yin, Dr. Qian Liu, Dr. Bin Wang, Dr. Chenlian Hu, Dr. Yiliao Song, Dr. Shan Xue, Dr. Fan Dong, Dr. Fujin Zhu, Dr. Adi Lin, Dr. Qian Zhang, Dr. Guanjin Wang and Dr. Zhen Fang who have shared their opinions and comments with me. Furthermore, I want to thank for the kind help from Prof. Shirui Pan, Dr. Zhedong Zheng, Dr. Qingji Guan, Dr. Zhun Zhong and other friends, may our friendship last forever.

At last, I would like to express my heartfelt appreciation to my wife, parents, and families for their love and support.

ABSTRACT

Recommender systems are an effective tool for solving problems with information overload. As such, they have not only received much attention from academia, they have also been widely used in industry. However, although recommender systems have seen many significant research outcomes, they also face some challenges and problems.

This thesis focuses on enhancing recommender systems through the use of diffusion dynamics and machine learning, and solves four problems faced by existing recommendation methods: 1) Can diffusion-based recommendation methods get a better balance between accuracy and diversity; 2) How can trust diffusion processes be modeled in social networks and how can social information be introduced into diffusion-based recommendation methods; 3) Can opinion dynamics be integrated with machine learning to make better recommendations; and 4) How can the issue of preference conflicts in group recommender systems be alleviated such that the recommendations generated meet the requirements of most of the users in a group.

To address Problem 1), this thesis presents a mixed similarity diffusion process that integrates two kinds of similarity measures from both explicit and implicit feedback data. It also considers the degree of balance for different kinds of nodes in a bipartite network. This new diffusion process enhances both the accuracy and diversity of the recommendations.

To address Problem 2), a trust diffusion process is simulated via a trust network that introduces explicit trust into the diffusion process, while the similarity between users indicates implicit trust. Moreover, a special resource allocation process, designed for a tripartite network, combines both kinds of trust to model user preferences in a more exact manner.

To address Problem 3), a social recommendation model is used to integrate opinion dynamics and user influence into a matrix factorization framework. The model characterizes the impact of neighbors on user opinions through evolutionary game theory and uses a payoff matrix to improve the training process of the matrix factorization. In addition, user influence that originates from the trust network is added to the proposed recommendation model.

To address Problem 4), a virtual coordinator combined with group recommendation solves preference conflicts through a negotiation process. The virtual coordinator brings a global perspective to optimizing the evaluation processes of individual user preferences in a group in order to create a balanced set of group recommendations. Additionally, personal influence is inferred from the trust relations to define the impact of the virtual coordinator on each group member.

To conclude, this thesis proposes a set of recommendation methods for both personalized and group recommendation that go some way to solving current challenges in recommender systems.

List of Publications

1. **Wang, X.**, Liu, Y., Lu, J., Xiong, F. & Zhang, G., 2019, 'TruGRC: Trust-aware group recommendation with virtual coordinators', *Future Generation Computer Systems*, vol. 94, pp. 224-236.

2. **Wang, X.**, Liu, Y., Zhang, G., Xiong, F. & Lu, J., 2017, 'Diffusion-based recommendation with trust relations on tripartite graphs', *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2017, no. 8, p. 083405.

3. **Wang, X.**, Liu, Y., Zhang, G., Zhang, Y., Chen, H. & Lu, J., 2017, 'Mixed similarity diffusion for recommendation on bipartite networks', *IEEE Access*, vol. 5, pp. 21029-21038.

4. **Wang, X.**, Liu, Y. & Xiong, F., 2016, 'Improved personalized recommendation based on a similarity network', *Physica A: Statistical Mechanics and its Applications*, vol. 456, pp. 271-280.

5. Xiong, F.*, **Wang, X.***, Pan, S., Yang, H., Wang, H. & Zhang, C., 2020, 'Social recommendation with evolutionary opinion dynamics', *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 10, pp. 3804-3816. (*Co-first Authors)

TABLE OF CONTENTS

List of Figures

Figure **Page**

List of Tables

CHAPTER 1

INTRODUCTION

1.1 Background

As information technology develops, more and more people are communicating and sharing information over the Internet. Internet technologies have accelerated the integration of all fields of society. The digital economy has gradually formed and become an important force in promoting the economic growth of various countries (Zhang et al., 2021). Internet platforms, such as e-commerce sites, social networks and video websites, have developed rapidly and their powerful information processing capabilities are bringing individuals into the information era (Bakos and Katsamakas, 2008). Internet information services are now the most convenient way to gain knowledge and, as such, they have become an indispensable part of most people's daily lives. However, at the same time, these information services often struggle to filter information, which can lead to problems with information overload (Rutkowski and Saunders, 2010).

To solve these information overload problems, researchers have devised various tools, including search engines (Langville and Meyer, 2011) and recommender systems (Adomavicius and Tuzhilin, 2005). While being good at locating specific information, search engines do not consider a user's preferences, and they return the same results to all people no matter their different habits. The typical modus operandi of search engines is to let users provide suitable keywords and to passively return results in accordance with those keywords (Morita and Shinoda, 1994), with the results being the same every time - that is, the results will not be personalized. To the contrary, recommender systems are a personalized information filtering technology that predicts whether an active user will like a particular object based on his/her particular history of choices (Ghorab et al., 2013). Recommender systems integrate many kinds of diverse user information as history or profile of choices, such as transaction records, rating scores, social relations, location information, and so on - all of which are mined to generate personalized recommendations based on that user's unique preferences.

Recommender systems have been regarded as one of the most effective methods for solving information overload problems and have therefore been widely applied in the e-commerce field. For instance, some shopping websites including Amazon and JD.com have built their own product recommender systems (Smith and Linden, 2017; Liu et al., 2020). Additionally, YouTube and Netflix have developed video and movie recommender systems to offer better services to customers (Zhou et al., 2010a; Amatriain and Basilico, 2015).

At present, ways to effectively evaluate user preferences and make reasonable recommendations based on big data and a plethora of user features have become an important research topic. This is because online user behaviours are so complex and so diverse. For this reason, researchers are turning to the theory of complex networks and advanced machine learning techniques for answers. With these tools, it should be possible to improve the performance of recommender systems for better and more accurate results.

1.2 Motivation

Recommender systems have attracted much attention, not only from academia but also from industry. In past years, many multi-disciplinary research results and industrial applications have been developed to make recommender systems more accurate. However, user behaviors on the Internet is often complicated and is growing more so with the development of information technology and exploding volumes of data. Modeling user preferences and making recommendations can therefore be a highly problematic task, and one fraught with many challenges.

One of the greatest challenges to recommender systems today is data sparsity. Although the number of users and items on e-commerce sites might be growing rapidly, the amount of information available on the interactions between those users and items is limited. Thus, in proportional terms, data sparsity is actually increasing (Guo et al., 2014, 2017a). This is a serious problem that has no fundamental solution, as the rate that relationship information is growing is falling far behind the amount of entity information. The current workaround typically involves a data filling method or enhancing the anti-sparsity capacity of the algorithms. That said, data sparsity remains a long-term challenge.

The second challenge is the dilemma of accuracy versus diversity. In studies on recommendation algorithms, accuracy is generally considered to be the most important indicator. Most recognize that recommending popular items leads to higher accuracy because highly popular items are easier for users to accept (Lü and Liu, 2011). Diversity reflects the differences between recommendation results for different users. If there are significant variations between the recommendations given to each user, diversity values will be high. However, there is a contradiction between accuracy and diversity (Zhou et al., 2010b). The traditional idea of recommending popular items to users provides promising results when it comes to improving accuracy. However, it tends to lead to similar recommendation results for each user, which reduces the diversity of the recommendations. Therefore, finding a balance between accuracy and diversity is a challenge in recommender systems.

The third challenge is recommendation with auxiliary features. With the combination of e-commerce and social networks, user behaviors have become complicated, and the application scenarios of recommender systems have gradually diversified. The combination has given rise to many kinds of additional features related to users and objects (Hyun et al., 2021; Ni et al., 2022a,b), including social relations, geographical location information, group information, product reviews, and so on. These features are useful supplements to the interaction information between users and objects. Some studies have demonstrated that integrating auxiliary features into recommender systems not only improves the accuracy of recommendations but also alleviates the issue of data sparsity (Guo et al., 2017a; Yu et al., 2019; Sun et al., 2020). Hence, research on how to effectively use auxiliary features to make better recommendations is important.

This thesis presents some methods that contribute to solving the outlined challenges.

1.3 Research Questions and Objectives

This thesis aims to develop several methods of enhancing recommender systems using diffusion dynamics and machine learning. In the process, the following research questions will be answered.

Research Question 1 (RQ1): Can diffusion-based recommendation methods get a better balance between accuracy and diversity?

Research Question 2 (RQ2): How can trust diffusion processes be modeled in social networks and how can social information be introduced into diffusion-based recommendation methods?

Research Question 3 (RQ3): Can opinion dynamics be integrated with machine learning to make better recommendations?

Research Question 4 (RQ4): How can the issue of preference conflicts in group recommender systems be alleviated such that the recommendations generated meet the requirements of most of the users in a group?

Achieving the following objectives is expected to answer the above research questions.

Research Objective 1 (RO1): (in answer to RQ1) To discover the factors that impact the accuracy and diversity of recommendations and create an improved diffusion-based recommendation model that supports diffusion processes in bipartite networks so as to achieve a better balance between accuracy and diversity.

Research Objective 2 (RO2): (in answer to RQ2) To develop a trust-aware diffusion model for making recommendations within social networks.

Research Objective 3 (RO3): (in answer to RQ3) To develop a social recommendation method that integrates the advantages of both opinion dynamics and machine learning.

Research Objective 4 (RO4): (in answer to RQ4) To find a solution that alleviates preference conflicts in groups and to develop a group recommendation method to can make compromises in the recommendations generated such that most group members are satisfied.

1.4 Research Innovation and Contributions

The work of the thesis advances recommender systems through diffusion dynamics and machine learning. The main innovations are summarised as follows.

Innovation 1: This study is the first to consider both explicit and implicit feedback in diffusion processes and to integrate the similarity measures from these two kinds of feedback in a diffusion-based recommendation model. Compared to existing diffusion-based methods, the proposed diffusion process models user preferences more accurately.

Innovation 2: This study is the first to introduce trust relations into diffusion-based recommendation models. The trust diffusion process is designed to capture the impacts of explicit trust between users in tripartite networks. This process can not only be used for combining trust relations but also for combining other additional features.

Innovation 3: This study is the first to introduce opinion dynamics into a matrix factorization model so as to bring interpretability and physical meaning to machine learning models. As a result of this combination, social information can be organized well for recommendation tasks.

Innovation 4: This study is the first to put forward the concept of virtual coordinators as negotiators for clearing up user preference conflicts in group situations. The approach takes advantage of both the result and the profile aggregation strategies to improve group recommendations.

These innovations lead to the following contributions.

Contribution 1 is a mixed similarity diffusion model called MSD that makes recommendations balanced between accuracy and diversity through a bipartite network. MSD introduces both cosine similarity with explicit feedback and a resource allocation index with implicit feedback into the diffusion process. The impacts of node degrees are also considered to make the diffusion process more reasonable. Experiments on real-world datasets demonstrate the MSD model concurrently enhances both the accuracy and the diversity of the recommendations.

Contribution 2 is a trust-aware diffusion-based recommendation model called DBRT that captures explicit trust relations and the implicit trust inferred by user similarity to improve resource-allocation processes. Experiments indicate that considering the effects of trust makes the diffusion process more reasonable and enhances the performance of the recommender system.

Contribution 3 is a social recommendation model called REOD that applies opinion dynamics theory to improving recommendation models.

REOD imparts a new perspective on integrating opinion dynamics with a matrix factorization model. The framework updates user opinions according to a payoff matrix built on game theory during the training process. Experiments show that REOD outperforms several existing social recommendation models given both normal users and cold-start users.

Contribution 4 is a trust-aware group recommendation model called TruGRC. This model introduces a virtual coordinator into group recommendation, which brings a global perspective to optimizing the evaluation processes of individual user preferences. Moreover, TruGRC considers the interactions between group users and the virtual coordinator to represent a negotiation process. Experiments indicate that the virtual coordinator improves group recommendation performance at a range of group sizes.

From the methodological perspective, contributions 1 and 2 apply diffusion dynamics to improve recommender systems, and contributions 3 and 4 enhance recommender systems by machine learning. From the data perspective, contribution 1 only uses rating data and contributions 2, 3 and 4 introduce additional features into recommender systems.

1.5 Research Significance

With the rapid popularization of e-commerce and social networks, user features are growing more and more abundant. These data resources are a good foundation for research in recommender systems as they have both theoretical and practical significance.

From the perspective of research, recommender systems are a research hotspot in the field of information technology. For example, in recent years, the number of submissions to the annual ACM conference on recommender systems has dramatically increased, which indicates how much attention researchers are paying to developing and improving recommender systems. This thesis proposes several methods for using diffusion dynamics and machine learning to enhance the performance of recommendation algorithms. Specifically, this is the first research to integrate both explicit and implicit similarity into a diffusion process with a bipartite network. This is a new idea for modeling trust relations in diffusion processes. The results show that considering multiple similarity measures can make diffusion processes more reasonable and additional features can be used to improve the accuracy and diversity of diffusion-based recommender systems. Moreover, this thesis brings a novel perspective in the form of using both opinion dynamics and machine learning to make better recommendations, which is significant to multi-disciplinary integration. Lastly, this thesis is the first to regard group recommendation as a negotiation. This inspired the concept of using virtual coordinators to model user preferences in a way that involves compromise between group members. This is an innovative way to solve the issues of preference conflicts in traditional group recommender systems.

From the perspective of practice, many e-commerce websites and enterprises have their own recommender systems. A good recommender system will exactly predict a customer's requirements and attract a large number of customers to interact with items. In real terms, this is likely to bring economic benefits to the platform. To this end, this thesis focuses on proposing efficient recommendation algorithms for a range of different scenarios, such as movie review websites and social networks. All the methods in the thesis have been tested and validated on real-world datasets, which means practitioners can directly use the methods proposed in their own real-world business scenarios.

1.6 Thesis Structure

The structure of this thesis is shown in Figure 1.1 and the chapters are organized as follows.

Chapter 1 introduces the background and challenges of recommender systems and shows the research questions, objectives and contributions of the thesis.

Chapter 2 presents the research overview along with a literature review on the different kinds of recommender systems.

Chapter 3 presents a mixed similarity diffusion model called MSD that integrates both cosine similarity with explicit feedback data and a resource-allocation index with implicit feedback data into the diffusion process. The proposed MSD considers the balance of different kinds of node degrees in a bipartite network so as to strike a balance between accuracy and diversity. This chapter addresses RQ1 to achieve RO1.

Chapter 4 presents a novel diffusion-based recommendation method called DBRT that extends the resource-allocation process from a bipartite network to a tripartite network. The proposed DBRT simulates a trust diffusion process to integrate explicit trust relations into the resource-allocation process. It also uses a cosine index between nodes to indicate implicit trust, further improving the resource-allocation process. This chapter addresses RQ2 to achieve RO2.

Chapter 5 presents a model-based recommendation method with opinion dynamics called REOD that considers both the dynamic processes of real society and the rating predictions of recommender systems. The proposed REOD model further integrates opinion dynamics and user influence with a matrix factorization framework. This chapter addresses RQ3 to achieve RO3.

Chapter 6 presents a trust-aware group recommendation method called TruGRC that integrates both result and profile aggregation strategies. The method introduces a virtual coordinator to group recommendation, which brings a global perspective on optimizing the evaluation processes of individual user preferences and creates a balanced set of group recommendations. This chapter addresses RQ4 to achieve RO4.

Chapter 7 summarises the findings of this thesis and points to directions for future work.

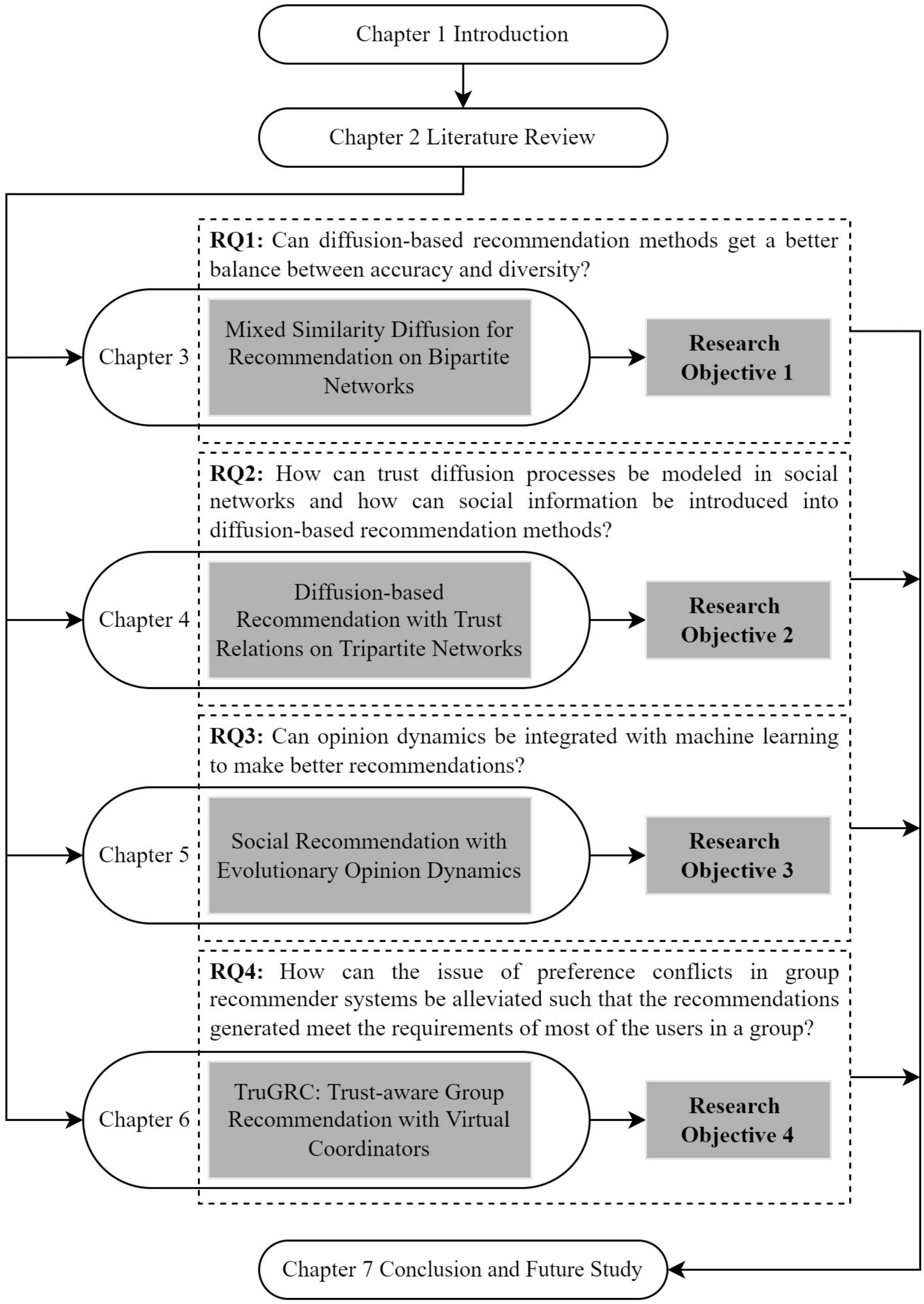


Figure 1.1: Thesis structure.

Chapter 2

Literature Review

The research on recommender systems began in the 1990s. With the rapid development in recent years, recommender systems have gradually become a research content involving multidisciplinary domains (Bobadilla et al., 2013; Zhou et al., 2017; Yang et al., 2021; Lü et al., 2012; Chen et al., 2020b). A lot of disciplines and subjects have made outstanding contributions to the development of recommender systems, such as computer science, physics, management and etc. At present, the research on recommender systems mainly focuses on using complex network theory (Zhou et al., 2007; Zhang et al., 2010; Zhu et al., 2015) and machine learning technologies (Jannach et al., 2021; Portugal et al., 2018; Karatzoglou and Hidasi, 2017; Pan et al., 2019; Jiang et al., 2015) to improve the performance of both personalized and group recommendation tasks. A literature review is presented in this chapter, which is divided into six parts: 1) content-based recommendation, 2) neighborhood-based collaborative filtering, 3) model-based collaborative filtering, 4) diffusion-based recommendation, 5) social recommendation, 6) group recommendation.

2.1 Content-based Recommendation

Content-based recommender systems are one of the earliest types of recommender systems (Lops et al., 2011; Thorat et al., 2015; Shu et al., 2018; Pérez-Almaguer et al., 2021). The principle of content-based recommender systems is that analyzing the description and inherent information of items to find out the items' characteristics, and then recommend other items which are in the same category or similar to the items that have been chosen by users in the past (Li and Kim, 2003; Cantador et al., 2010). For example, a content-based recommender system of online shopping websites can be considered that a user may be more interested in such products similar to the products have purchased many times by the user than the products in a new category. Being different from collaborative filtering, content-based recommender systems only pay attention to items' features and ignore the interactions between users and items (Balabanović and Shoham, 1997).

Content-based recommender systems have the advantage of strong independence between users, which means they only consider the behaviors made by the target user without the impacts of other users. In some situations, using other users' historical data brings noise, because some users make cheat behaviors on the Internet so that their data is not true or reliable. Furthermore, recommendation results made by content-based recommender systems are easier explained to users than those made by collaborative filtering, because the results are based on the description and characteristics of items compared to complex interactive information used in collaborative filtering. Currently, some enhanced models based on the principle of content-based recommender systems have been proposed to handle recommendation tasks in fields with abundant item features (Lops et al., 2019), e.g. news recommendation (Kompan and Bieliková, 2010), music recommendation (Oord et al., 2013) and etc. Rutkowski et al. (2018) considered the rich content of movies, including ratings, tags, genres, years and TMDB, and applied the neuro-fuzzy approach to make recommendations for users. Reddy et al. (2019) assumed movie genres are a very important factor in making recommendations and built a recommender system based on content-based filtering. Zhong et al. (2018) proposed a music recommendation method based on convolutional neural networks to improve the accuracy of recommendations, which uses the representation ability of neural networks to learn music segments from more than 60 thousands songs.

However, the shortcomings of content-based recommender systems cannot be ignored. First of all, the content information of items is generally more difficult to obtain than interactive information. In social networks, the interactions between users and items are plentiful, by contrast, the information about items is limited and sometimes hard to be discovered. In addition, content-based recommender systems only depend on users' preference for the intrinsic content of certain items in the past and the content updates very slowly, which means the recommendations proposed by content-based recommender systems may be hysteretic. Because of these shortcomings, some researchers started to integrate both content and interactive information to make hybrid solutions (Arampatzis and Kalamatianos, 2017). Basilico and Hofmann (2004) was the first attempt to systematically integrate user-item interactions as well as the attributes of items or users to learn an unified prediction function. Yao et al. (2014) applied a recommendation method to make web service recommendations, which simultaneously considers both QoS rating data and semantic content data of web services using a probabilistic generative model. De Campos et al. (2010) handled the issue of combining content-based and collaborative features by using Bayesian networks. Ronen et al. (2013) proposed an algorithmic framework to automated select or generate meaningful informative content-based features, e.g. text and tags, and regraded these features as supplementary information for collaborative filtering in commercial systems.

2.2 Neighborhood-based Collaborative Filtering

Collaborative filtering is the most widely used recommendation method (Bobadilla et al., 2011; Ekstrand et al., 2011; Shi et al., 2014; Yang and Li, 2009; Sarwar et al., 2001) and has attracted lots of attention by researchers starting from the Netflix Prize in 2006 because of its outstanding performance in making recommendations (Bennett et al., 2007; Amatriain, 2013). Collaborative filtering considers that users may be impacted by others' opinions or behaviors when they are making decisions, which fully implements both user-item and user-user interactive information to evaluate user preference to propose suitable recommendations (Mahmood and Ricci, 2009).

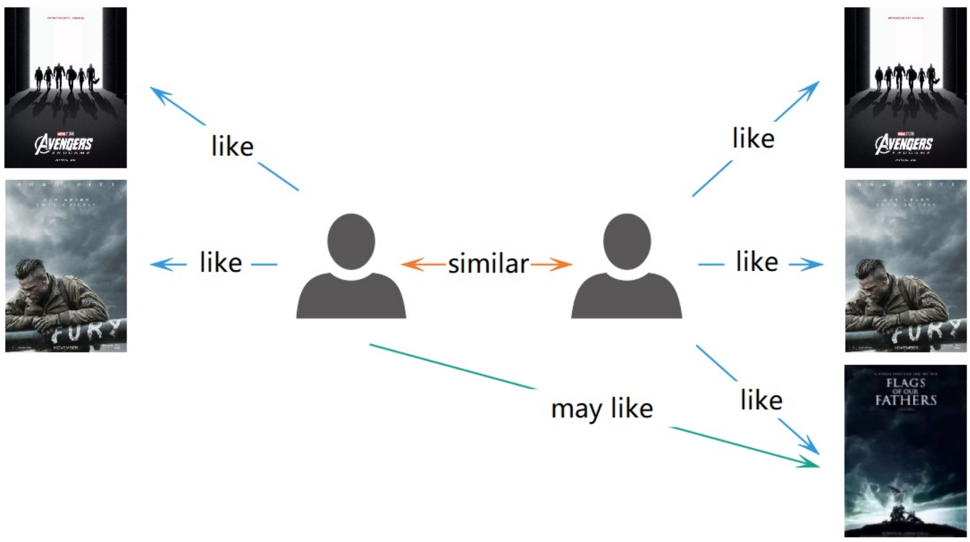


Figure 2.1: The schematic diagram of neighborhood-based collaborative filtering.

Neighborhood-based collaborative filtering is a classical recommendation method that evaluates the relevance between users by calculating the similarity and finds out neighbors for a target user (Aggarwal, 2016). It assumes that if there exists a high similarity between two users, these users may have analogous preferences and each other's opinions will be accepted easily. A schematic diagram of neighborhood-based collaborative filtering is shown in Figure 2.1. It can be seen from the figure, that two users like the same two movies, so they have similar preferences and can be defined as neighboring users. According to the assumption, the user on the left will be more interested in the movie at the bottom chosen by the user on the right.

Neighborhood-based collaborative filtering is commonly used in rating prediction tasks. If we want to predict the rating of an item α given by a target u, we can calculate the similarities between the target user u and other users and then find out top-K similar users who have already rated the item α. The prediction $\hat{r}_{u\alpha}$ means the rating of the item α given by the target u, defined as

$$\hat{r}_{u\alpha} = \overline{r}_u + \frac{\sum_{k=1}^{K} sim(u,k) \times (r_{k\alpha} - \overline{r}_k)}{\sum_{k=1}^{K} \left|sim(u,k)\right|} \tag{2.1}$$

where $\overline{r}_u$ and $\overline{r}_k$ mean the average rating scores of user u and its top k similar users, $sim(u,k)$ means the similarity between users u and k, and $r_{k\alpha}$ means the item α's rating given by user k. It can be seen from Eq. 2.1, user u will get more impacts from other users who are more similar to it.

Table 2.1: Similarity measures frequently used in neighborhood-based collaborative filtering.

Measure	Definition
Cosine similarity	$Cosine(u,k) = \frac{\sum_{\alpha \in I'} r_{u\alpha} \times r_{k\alpha}}{\sqrt{\sum_{\alpha \in I'} r_{u\alpha}^2} \times \sqrt{\sum_{\alpha \in I'} r_{k\alpha}^2}}$
Pearson correlation coefficient	$PCC(u,k) = \frac{\sum_{\alpha \in I'} (r_{u\alpha} - \bar{r}_u) \times (r_{k\alpha} - \bar{r}_u)}{\sqrt{\sum_{\alpha \in I'} (r_{u\alpha} - \bar{r}_u)^2} \times \sqrt{\sum_{\alpha \in I'} (r_{k\alpha} - \bar{r}_u)^2}}$
Constraint Pearson correlation coefficient	$CPCC(u,k) = \frac{\sum_{\alpha \in I'} (r_{u\alpha} - r_{med}) \times (r_{k\alpha} - r_{med})}{\sqrt{\sum_{\alpha \in I'} (r_{u\alpha} - r_{med})^2} \times \sqrt{\sum_{\alpha \in I'} (r_{k\alpha} - r_{med})^2}}$
Jaccard similarity	$Jaccard(u,k) = \frac{\lvert I_u \cap I_k \rvert}{\lvert I_u \cup I_k \rvert}$
Mean square	$Mean(u,k) = 1 - \frac{\sum_{\alpha \in I'} (r_{u\alpha} - r_{k\alpha})^2}{\lvert I' \rvert}$

An important part of neighborhood-based collaborative filtering is similarity measures and those frequently used in recommender systems are shown in Table 2.1, such as cosine similarity, Pearson correlation coefficient, Jaccard similarity and etc. In Table 2.1, I_u and I_k are item sets respectively rated by users u and k, I' is an item set rated by both users u and k at the same time, $r_{u\alpha}$ and $r_{k\alpha}$ mean the ratings of item α given by users u and k respectively, and r_{med} means the median of all ratings.

Based on the above common similarity measures, some further improvements have been proposed by researchers. Ahn (2008) analyzed the distribution of rating data and proposed a heuristic similarity measure with the popularity and impacts of rating objects, which focuses on improving recommendation performance under cold-start conditions. Bobadilla et al. (2010) combined mean square and Jaccard correlation coefficient to propose a similarity measure called JMSD that extends Jaccard correlation coefficient to rating prediction tasks. In the meantime, Bobadilla et al. (2012) integrated both rating scores and the probability distributions of user ratings to enhance the rationality of similarity measures. To alleviate the dilemma of data sparsity, Patra et al. (2015) applied the Bhattacharyya coefficient in signal and image processing domains to calculate similarities between item vectors and regarded these results as global similarities when common rated items by users are very few.

2.3 Model-based Collaborative Filtering

Machine learning is a vital research domain in computer science, which has been widely applied in data mining (Teng and Gong, 2018; Li et al., 2017a) and recommender systems (Weimer et al., 2008; Singh and Gordon, 2008; Hu et al., 2021; Xiao et al., 2022). Inspired by machine learning, model-based collaborative filtering makes recommendations based on mining users' and items' features in latent feature spaces rather than calculating similarities (Koren, 2008). Most model-based approaches are developed on matrix factorization (MF) which is a typical model in machine learning to solve prediction problems. Koren et al. (2009) firstly applied MF in recommender systems and achieved excellent results in the Netflix Prize competition.

The principle of MF is to get the latent feature matrices of users and items by decomposing a user rating matrix, and then predicts rating

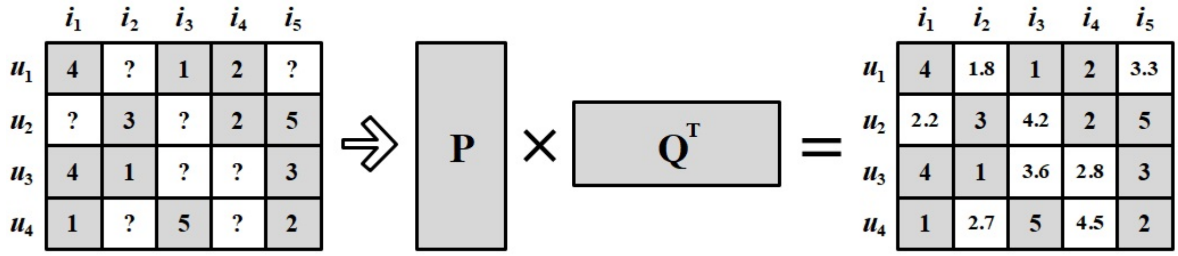


Figure 2.2: The schematic diagram of the matrix factorization algorithm.

scores through the latent feature vectors of users and items (Chen and Peng, 2018). Figure 2.2 shows the progress of MF where **P** and **Q** are low-rank matrices to represent the latent features of users and items. If the number of users and items are m and n and the dimension of the feature space is d, the dimensions of **P** and **Q** can be expressed as $m \times d$ and $n \times d$ respectively and the rating matrix **R** is a $m \times n$ sparse matrix. The target of MF is to predict unknown ratings in the rating matrix **R** using latent features, defined as

$$\mathbf{R} = \mathbf{P}\mathbf{Q}^{\mathrm{T}} \tag{2.2}$$

where T is the symbol of matrix transpose. Each rating prediction in the rating matrix **R** can be defined as

$$\hat{r}_{u,i} = \mathbf{p}_u \mathbf{q}_i^{\mathrm{T}} \tag{2.3}$$

where $\hat{r}_{u,i}$ means the predicted rating of item i given by user u, $\mathbf{p}_u$ and $\mathbf{q}_i$ are user u's and item i's latent feature vectors. In order to learn latent feature vectors, a loss function is necessary to be used for minimizing the gap between observed ratings and predictions (Guo et al., 2016a). Square loss is the most frequent function in MF, defined as

$$\min_{\mathbf{p}_u,\mathbf{q}_i} \mathcal{L} = \frac{1}{2}\sum_{u=1}^{m}\sum_{i=1}^{n}\delta_{u,i}\left(r_{u,i}-\hat{r}_{u,i}\right)^2 \tag{2.4}$$

where $\delta_{u,i}$ is an indicator when $\delta_{u,i}=1$ means $r_{u,i}$ has been observed and $\delta_{u,i}=0$ otherwise. Gradient descent is an useful and efficient optimization method for finding the local optimal values of the loss function (Pan et al., 2015b; Loni et al., 2016; Guo et al., 2016c). According to Eq. (2.4), the gradients of $\mathbf{p}_u$ and $\mathbf{q}_i$ are as follows.

$$\begin{aligned} \frac{\partial\mathcal{L}}{\partial\mathbf{p}_u} &= \sum_{i=1}^{n}\delta_{u,i}\left(r_{u,i}-\hat{r}_{u,i}\right)\mathbf{p}_u \\ \frac{\partial\mathcal{L}}{\partial\mathbf{q}_i} &= \sum_{u=1}^{m}\delta_{u,i}\left(r_{u,i}-\hat{r}_{u,i}\right)\mathbf{q}_i \end{aligned} \tag{2.5}$$

Updating $\mathbf{p}_u$ and $\mathbf{q}_i$ with a certain learning rate η until the algorithm converges, shown in Eq. (2.6).

$$\begin{aligned} \mathbf{p}_u &\leftarrow \mathbf{p}_u - \eta\frac{\partial\mathcal{L}}{\partial\mathbf{p}_u} \\ \mathbf{q}_i &\leftarrow \mathbf{q}_i - \eta\frac{\partial\mathcal{L}}{\partial\mathbf{q}_i} \end{aligned} \tag{2.6}$$

The approximate rating can be calculated by Eq. (2.3) based on the local optimal $\mathbf{p}_u$ and $\mathbf{q}_i$.

Because of the great success of MF in recommender systems, many further improvements have been proposed. Koren (2010) integrated both the ideas of neighborhood-based collaborative filtering and singular value decomposition (SVD) to propose SVD++ that considers users' latent features and explicit feedback. Because of users' preferences are dynamically changed, Koren (2009) introduced the concept of the time window in SVD++ to propose an algorithm named TimeSVD++ based on dynamic temporal series for predicting the user preferences of specific periods. Guo et al. (2016a) introduced the concept of direct and indirect trust in social networks into the SVD algorithm and proposed the TrustSVD model that improves the recommendation effect of both normal and cold-start users. Ma et al. (2011) calculated the similarity between users by Pearson correlation coefficient and assumed that users with a high similarity have strong social influence. Based on this assumption, a social constraint MF algorithm was proposed to apply potential social relations in rating prediction tasks. Li et al. (2019b) analyzed the impact of social network topology and introduced this impact into MF algorithms to evaluate the global influence of users in social networks through topology structures. To enhance the effect of recommendations under data sparse conditions based on the principle of cross-domain recommendation, Ji et al. (2016) divided user scoring modes into several categories and then constructed the mappings between different modes using the CodeBook method. Ranking-based prediction is a vital part of recommender systems, Shi et al. (2010)

proposed a list-wise ranking method that expands MF algorithms from scoring-based predictions to ranking-based predictions. Hu et al. (2008) proposed a weight-based MF algorithm that uses implicit feedback such as shopping records and browsing records to predict users' clicks on websites. He et al. (2016) proposed a fast MF algorithm that greatly reduces the computing complexity of the weight-based MF algorithm. The probabilistic matrix factorization (PMF) explained MF from a probability perspective, which performs very well when datasets are sparse and samples are imbalanced (Mnih and Salakhutdinov, 2007). Furthermore, Salakhutdinov and Mnih (2008) proposed the Bayesian probabilistic matrix factorization (BPMF) model based on PMF and proved the BPMF model can be efficiently trained through the Markov chain Monte Carlo approach. Zheng and Xiong (2018) introduced users' social labels into the PMF model to enhance the rationality and accuracy of recommendation algorithms by extracting potential semantic information. A MF model based on neural network architectures was proposed by Xue et al. (2017), which designs a novel loss function based on binary cross-entropy and provides a new perspective for improving MF algorithms.

In addition to MF models, factorization machines (FM) is also a typical machine learning method based on latent features (Rendle, 2010, 2012). It contains the ideas of both logistic regression and MF, and can be used in solving a variety of problems, e.g., regression, binary classification and ranking tasks (He and Chua, 2017; Juan et al., 2016). FM models are usually applied in combining additional features with recommender systmes and cross-domain recommendation tasks (Chen et al., 2020a; Hu et al., 2018; Loni et al., 2014; Li et al., 2019a). Loni et al. (2014) proposed a FM-based cross-domain recommendation algorithm that implements the information interactions between different domains by building unified latent features of a target domain and auxiliary domains. A fast FM model based on context information was proposed by Rendle et al. (2011), which uses context information for rating prediction tasks. Guo et al. (2016c) proposed a pair-wise learning model to put FM in solving personalized ranking issues. (Pan et al., 2015a) introduced a recommendation algorithm for heterogeneous data, which first compresses each field features and then migrates features from different fields through FM to achieve heterogeneous data interactions for realizing knowledge transformation. DeepFM is an outstanding recommendation method that has been widely used in industry (Guo et al., 2017c). It combined deep learning with FM and used neural networks to finish non-linear representation for features. (Knoll, 2016) extended FM to a higher order, which makes interactions between three features and keeps linear-complexity to realize the goal of improving recommendation effect and ensuring computational complexity at the same time.

Because of the ability of knowledge representation, deep learning has been applied in recommender systems in recent years (Zhang et al., 2019; Da'u and Salim, 2020; Khan et al., 2021). He et al. (2017) introduced a neural architecture that is first to model the interactions between both user and item features with neural networks. Ahmadian et al. (2022) utilized deep neural networks to model the representation of trust relationships and tag information to extract latent features from trust information and user-tag matrices. Bobadilla et al. (2021) extracted demographic information from user and item factors through a deep learning-based feature selection method. Da'u et al. (2021) applied neural attention techniques to learn adaptive user and item representations and fine-grained user-item interactions, which enhances the accuracy of item recommendation tasks. Tahmasebi et al. (2021) proposed a hybrid social recommendation method that utilizes a deep autoencoder network to employ collaborative and content-based filtering, as well as user social information in the recommendation process. Liang et al. (2022) introduced a dynamic heterogeneous graph convolutional network for item recommendation tasks, which consists of two components named graph learner and heterogeneous graph convolution. Specifically, the graph learner considers different kinds of interactions between users and items, and the heterogeneous graph convolution aggregates both graph representations and item content information. Deep learning technologies are also introduced in some special recommendation tasks, such as point-of-interest recommendation (Islam et al., 2022; Liu and Wu, 2021), citation recommendation (Ali et al., 2020; Gupta et al., 2021) and news recommendation (Ji et al., 2021).

2.4 Diffusion-based Recommendation

Complex networks are often used to describe a system formed by the connections between different objects, and the behaviors of users who collect items can be abstracted to connections or relationships in complex networks (Lü and Zhou, 2011). A bipartite network is a special structure in complex networks that distinguishes nodes into two non-intersect vertex sets (Koskinen and Edling, 2012). In recommender systems, users and items can be regarded as the nodes in bipartite networks, and each edge connecting user nodes and item nodes represents the behavior of a user who purchases or selects such items.

Inspired by physical dynamics (Grigull and Sandner, 1984; Sarman and Evans, 1992), many researchers designed diffusion dynamics in bipartite networks to make recommendations. The work proposed by Zhou et al. (2007) is the pioneer of diffusion-based recommendation methods called mass diffusion which is also a physical phenomenon's name. In this work, a diffusion-based method that simulates mass diffusion in physics in a bipartite network to make recommendations for users through evaluating user preferences for non-selected items. In order to enhance the diversity of recommendations, Zhang et al. (2007) inspired by the heat conduction phenomenon in physics to design a diffusion process simulating the heat conduction in a bipartite network. This method brings a good diversity for recommendations which means it can be used in the condition of providing long-tail results for users.

Depending on mass diffusion and heat conduction models, many improvements have been suggested by researchers both in physics and computer science (Qiu et al., 2014; Wang et al., 2016c; Chen et al., 2017a; Li et al., 2017b; Shuang et al., 2019). Jia et al. (2008) considered resource initialization in bipartite networks has a certain impact on recommendation results and changed the way of initialization according to the degrees of item nodes. Zhou et al. (2009) assumed that the correlation resulting from a specific attribute may be repeatedly counted in cumulative recommendations from different objects in the mass diffusion process and designed an improved algorithm to eliminate redundant correlations by considering higher order correlations. Based on the heat conduction model, Liu et al. (2011) introduced a hyperparameter to improve the accuracy and contained the diversity of recommendations. Zeng et al. (2014) proposed a similarity-preferential diffusion process that uses hyperparameters to compress or expand the resources of user nodes in bipartite networks to control the impacts between similar users. Traditional mass diffusion-based models only consider unidirectional processes, which lead to be a biased causal similarity estimation. To solve this issue, Zhu et al. (2015) proposed a consistence-based mass diffusion algorithm via a bidirectional diffusion process against biased causality and the algorithm achieved good performance on several real-world datasets.

2.5 Social Recommendation

Social-based recommendation models often integrate the preferences of neighbors into the prediction of unknown ratings, so that social information is utilized in recommender systems (Mao et al., 2017; Yu et al., 2021; Miao et al., 2022). Latent factors of neighbors may affect the prediction of user ratings or relations. In the work by Ma et al. (2009), users' social trust was combined with the PMF model, and the interests of users and their trusted friends were fused to make a decision on uncollected ratings. Instead of integrating social information for rating prediction, recommendation with social regularization exerts social constraints on MF frameworks (Ma et al., 2011). Users may take the average preference of neighbors, or they may have similar interests with each neighbor. Both trust networks and social networks are considered. Social relations are not homogeneous among different users, and weak dependency connections exist widely on social networks. Weak dependency connections represent the relations among users in a group that have similar tastes. In the work by (Tang et al., 2016b), after community detection, social dimensions that express user tastes were exploited, and a user may be involved in different dimensions. Based on social dimensions, a recommendation framework was proposed, which incorporates the heterogeneity of social relations and weak dependency connections. Social dimensions improve the recommendation effectiveness.

From an analysis of real-world datasets, rating data and social data in social networks are usually complementary. Guo et al. (2016a) incorporated both the explicit and implicit influence of user trust, and both trusted and trusting users were considered in the prediction of ratings for an active user. User preferences do not always remain unchanged, instead, it drifts over time. Zhang et al. (2016b) inferred the latent social network from cascade data, and identified the dynamic changes of users over time using the latest updated social network. A model of implicit dynamic social recommendation was proposed to address the common existing preference drifting issues. Mining social information in time helps to improve recommendations. Tang et al. (2015) leveraged social science theories to develop a methodology for the study of online trust evolution. The dynamics of user preferences was exploited to reveal trust evolution. User relations are not always positive, and social networks also contain negative links. The work by Tang et al. (2016a) exploited signed social networks for recommendation, and leveraged positive and negative links in signed social networks. The preferences of users are likely to be closer to those of their friends than those of their foes. The results proved that negative links in signed social networks were as important as positive links for recommendation.

These aforementioned studies utilized social information directly, and user relations were incorporated with latent interest vectors in recommender systems. Hu et al. (2012) measured user influence from network topology. The work distinguished different social relations among users, and latent user preferences were learned from those who have the most influence in social networks. The Shannon entropy principle was used to optimize an influence factor, and the topological distances of users were calculated for the building of influence. Zhang et al. (2016a) developed the global influential model and the local influential model to find influential users. They carried out Monte-Carlo simulations to obtain an approximate result while handling large-scale user networks. Global and local influence was used as regularization terms in the MF framework. Experiment results proved that these methods which explore user influence from social relations have an advantage in terms of accuracy and stability.

2.6 Group Recommendation

Group recommender systems are designed to generate recommendations for a set of individuals with diverse interests (Wang et al., 2022), which have been implemented in several domains, e.g., tourism (Anagnostopoulos et al., 2017) and TV programs (Yu et al., 2006). The main idea of group recommendation is to aggregate group members' choices and the current group recommendation approaches fall into two categories, i.e., profile aggregation and result aggregation.

Profile aggregation builds a virtual user to represent the group profile by combining each individual profile, or regards the preference of the whole group as a special user. Ortega et al. (2016) applied MF techniques to combine group user profiles in latent feature spaces. Wang et al. (2016b) proposed a group recommendation model based on member contributions which are evaluated by the degrees of user importance via the separable non-negative MF technique. Kagita et al. (2015) defined a virtual user's profile using precedence relations in a group and regarded it as the group's profile. Leng and Yu (2022) used neural networks to represent the impacts of global and local social networks for group members and modeled group-item and user-item interactions to enhance recommendation performance.

Compared to profile aggregation, result aggregation often has more flexibility and has attracted more attention. Research on result aggregation mainly focuses on designing better aggregation functions to integrate group members' preferences. Some studies have indicated the average aggregation function (AVG) gets the best results among navie aggregation functions (Dwivedi and Bharadwaj, 2015; Amer-Yahia et al., 2009). Therefore, applying AVG to combine the results of individual recommendation methods can be regarded as baselines in result aggregation. Castro et al. (2017a) proposed an aggregation strategy that applies opinion dynamics to simulate the information interaction process between group members and, through this opinion exchange, any conflicting preferences between group members are resolved. Abolghasemi et al. (2022) generated implicit feedback and pairwise ratings to make personalized item scores for each group member, and then reached a consensus by a group decision-making model. The clustering group is a typical kind of user groups, Boratto et al. (2016) adopted clustering methods to detect user groups and tested plentiful aggregation functions to find out which function will bring the best performance of accuracy in this scenario. Quijano-Sanchez et al. (2017) proposed a personalized social individual explanation approach that infers social relations from demographic information of group members. Guo et al. (2016b) introduced a computational model to integrate the influence of personality, expertise factor and preference similarities, and demonstrated considering social influence can improve the quality of group recommendation. Some other models to diminish the negative effects of natural noise in group recommender systems have also been studied (Castro et al., 2017b, 2018).

CHAPTER 3

MIXED SIMILARITY DIFFUSION FOR RECOMMENDATION ON BIPARTITE NETWORKS

3.1 Introduction

With the revolutionary development of the Internet, the quantity of information is growing very quickly and has become out of the capability of human beings (Ma et al., 2015). Information overload appears to be a serious problem in traditional data analytic studies (Zhang et al., 2017; Pan et al., 2017; Wang et al., 2017c), and exploring useful content from rapidly increasing information tends to be a raising trend in modern society (Zhang et al., 2016c; Wang et al., 2017c; Pan et al., 2016a). Recommender systems have been recognized as an effective tool to handle this problem, and play a crucial role in data processing tasks (Lu et al., 2015). Recently, personalized recommendation among numerous potential choices attracts more and more attention (Lü et al., 2012), and have been applied to many actual domains, such as recommending movies (Gogna and Majumdar, 2015; Wang et al., 2016b), content (Verma et al., 2016), citations (Liu et al., 2015), locations (Wang et al., 2017a,d), mobile applications (Yin et al., 2017) and services for e-business (Wu et al., 2015b; Lu et al., 2013) and e-goverment (Wu et al., 2015a; Lu et al., 2010).

Collaborative filtering is a typical and the most popular information filtering technology in recommender systems (Xu et al., 2016). Its main idea is to evaluate user preference through exploiting user feedback data in a collective way. Two kinds of feedback data can be processed, i.e., explicit feedback and implicit feedback. The former, e.g., 5-star ratings, means the level of how a user likes an item, while the latter one, e.g., clicks or purchases, indicates whether a user likes an item or not (Liu et al., 2017). In addition, elements in explicit feedback matrix can be any numeric values while the implicit feedback matrix is a single-valued matrix. In collaborative filtering, diffusion-based recommendation algorithms can act on unary data and make recommendations based on network structures, which are inspired by diffusion phenomenon in physical dynamics and have good interpretability (Xiong and Li, 2017; Wang et al., 2017b). These algorithms use a user-item bipartite network to represent input data, e.g., rating matrix, and links on the bipartite network indicate the collection behaviors between users and items. Some physical processes can be then employed on the bipartite network to make recommendations, such as random walk, mass diffusion (Zhou et al., 2007) and heat conduction (Zhang et al., 2007). Unfortunately, since traditional diffusion-based recommendation methods use binary value to simulate user's collection behaviors, e.g., a user collects or rejects an item, those algorithms only take advantage of implicit feedback but neglect explicit feedback, which is also a crucial feature for precisely evaluating user preference (Pan and Ming, 2017; Wang et al., 2016a).

In this chapter, we propose a two-step resource-allocation process to overcome the above research gap. On social networks, individuals can review or give their ratings on objects, e.g. items, movies, games and etc. These behaviors bring lots of user feedback data that can be discovered to model user preference. In this chapter, a Mixed Similarity Diffusion model (MSD) is designed by involving both explicit feedback data and implicit feedback data. And, we consider the degrees of users and items at the same time in diffusion processes to improve the performance of the model. The main contributions of this chapter are summarized as follows.

1) A mixed similarity diffusion model, named MSD, is proposed to improve the performance of recommendation on bipartite networks, which introduces both the cosine similarity with explicit feedback data and the resource-allocation index with implicit feedback data into diffusion processes.

2) The impacts of node degree on bipartite networks are discovered and

MSD considers the degree balance of different kinds of nodes on networks to make diffusion processes more reasonable.

3) Extensive experiments have been conducted to evaluate the effectiveness of MSD. We compare the proposed method with several state-of-the-art diffusion-based recommendation methods in three real-world datasets and the results show that MSD achieves an accuracy-diversity balance that enhances the accuracy and the diversity of recommendations at the same time.

3.2 The Proposed Method

A recommender system can be represented by a user-item bipartite network which consists of a user set U and an item set I. The user set is defined as $U = \{u_1, u_2, \ldots, u_m\}$ and the item set is defined as $I = \{i_1, i_2, \ldots, i_n\}$, where m and n are the numbers of users and items in the recommender system. A link set $E = \{e_1, e_2, \ldots, e_z\}$ is used to denote relations between users and items and z is the amount of links. In this chapter, to make it easy to understand, we use Greek and Latin letters to express item-related and user-related indices, respectively. An $m \times n$ adjacent matrix A can be utilized to describe the user-item bipartite network, where every element $a_{i\alpha}$ is defined in Eq. (3.1).

$$a_{i\alpha} = \begin{cases} 1, & \text{user } i \text{ collects item } \alpha, \\ 0, & \text{else.} \end{cases} \tag{3.1}$$

Degree is an important concept in complex networks, which is the number of edges linked to a vertex (Zhou et al., 2007). Accordingly, we define the degrees of item α and user i as k_α and k_i that represent the number of users who collect item α and the number of items collected by user i, respectively. The primary purpose of a ranking-based recommender system is to evaluate user preference and provide a recommendation list for a target user. That is to say, a set of items uncollected by the target user with the highest recommendation score would be included in the recommendation list. The length of the recommendation list is defined as L in this chapter.

3.2.1 Mass Diffusion Model

Mass diffusion model (MD) is a successful and popular recommendation algorithm (Zhou et al., 2007), which takes advantage of a resource-allocation process to make recommendations on a bipartite network. In MD, a target user who will receive a recommendation list of items needs to be chosen at first. Then, items linked to the target user on the bipartite network obtain initial resource. Note that assuming the initial resource on each item is one unit for convenient computation in this chapter.

MD can be described as a two-step resource-allocation process. In step 1, the initial resource on item nodes flows to neighboring user nodes based on each item's degree, so the resource on each user node can be calculated

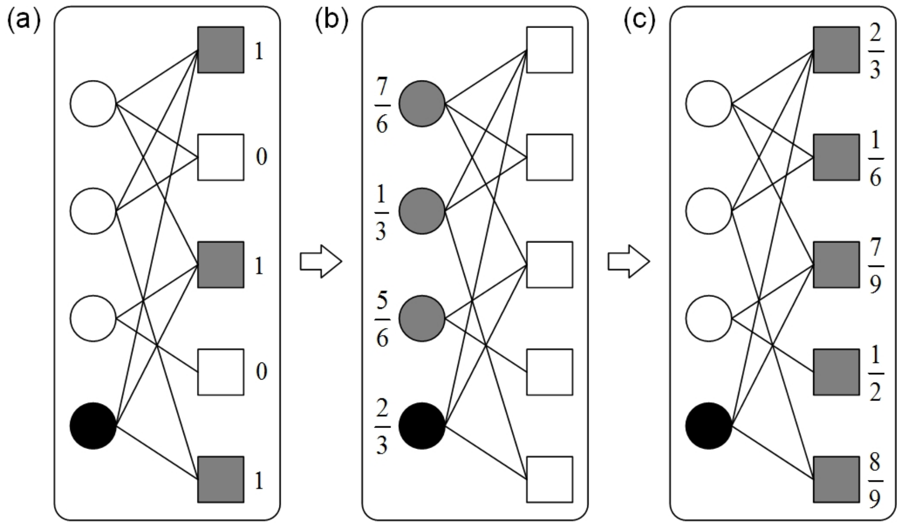


Figure 3.1: An illustration of mass diffusion model (MD). Users and items are represented by circles and squares, respectively. The black circle means the target user, and the circles and squares with grey color indicate the resource is currently distributed on these nodes. Plot (a) is the initial configuration, each item linked to the target user obtains one unit of resource. Plot (b) shows that the resource flows from items to users according to each item's degree and the resource on each user can be calculated by Eq. (3.2). Plot (c) shows the resource flows back to items based on each user's degree and the final resource on each item is calculated by Eq. (3.3).

as

$$f'_{ij} = \sum_{\alpha=1}^{n} \frac{a_{i\alpha}a_{j\alpha}}{k_\alpha} f_\alpha \tag{3.2}$$

where user i is the target user and user j is the user who will get resource in step 1; f_α and f'_{ij} are the initial resource on item α and the resource on user j after step 1, respectively; $a_{i\alpha}$ and $a_{j\alpha}$ are elements in the adjacent matrix A, and k_α is the degree of item α. The distribution strategy in step 2 is based on each user's degree. Therefore, the final resource $f'_{i\beta}$ on item

β is defined as

$$f'_{i\beta} = \sum_{j=1}^{m} \frac{a_{j\beta}}{k_j} f'_{ij} \tag{3.3}$$

where k_j is the degree of user j. After a two-step resource-allocation process, the initial resource is redistributed on items, and then a recommendation list of uncollected items can be arranged for the target user according to the final resource on each item. The uncollected items with most final resource will be placed at the top of the list. An illustration of the resource-allocation process of the mass diffusion model is shown in Figure 3.1.

Although MD is proved to be effective in recommendation tasks, there are still some weaknesses. Firstly, MD makes recommendations with implicit feedback which only includes binary value, such as 1 for a positive example and 0 for a negative example. Explicit feedback is neglected by MD. However, explicit feedback is also very important on social networks. For example, ratings are multivariant in real-world datasets, e.g., a user can give a rating to an item from 1 to 5, which can be regarded as explicit feedback. Furthermore, MD only considers the degree of each user when the resource flows back to items in step 2. In some previous studies (Zhang et al., 2007; Liu et al., 2011), the degree of each item plays a vital role in the resource redistribution process, which can improve the diversity of recommendations.

In next sections, we will improve mass diffusion model through a mixed similarity diffusion strategy that integrates both implicit feedback and explicit feedback. Additionally, we will consider the degree of users and items at the same time when the resource flows on the bipartite network.

3.2.2 Similarity Measurement Methods

The similarity measurement between users is a crucial part in evaluating user preference. In recommender systems, we always assume a user will accept suggestions or choices from other most similar users. Therefore, how to measure the similarity between users obtains a lot of attention recently. In the proposed method, we take advantage of two common similarity measurement methods, i.e., cosine similarity and resource-allocation (RA) index (Chen et al., 2017b), to integrate implicit feedback and explicit feedback into the diffusion process.

The cosine similarity is a widely used approach in evaluating user preference based on explicit feedback, e.g., ratings. Between user i and j, the cosine similarity is defined as

$$Cos(i,j) = \frac{\sum_{\alpha=1}^{n'} R_{i\alpha} R_{j\alpha}}{\sqrt{\sum_{\alpha=1}^{n'} R_{i\alpha}^2}\sqrt{\sum_{\alpha=1}^{n'} R_{j\alpha}^2}} \tag{3.4}$$

where $R_{i\alpha}$ and $R_{j\alpha}$ are rating scores on item α rated by user i and j; n' is the number of co-rated items by both users. The value of cosine similarity is located in $[0,1]$, because the rating scores are greater than 0. The cosine similarity measures the angle between two user vectors of ratings, where a greater value of the cosine similarity indicates the closer relationship between two users (Ahn, 2008).

The RA index is a typical similarity measurement on bipartite networks. The usual configuration of initial resource on each node is binary, i.e., 0 and 1, which resembles implicit examinations such as clicks, browses and collections in real systems. Thus, evaluating the similarity between two nodes via the RA index is to calculate the similarity with implicit feedback in recommender systems. The similarity between two nodes i and j can be defined as

$$RA(i,j) = \sum_{\alpha=1}^{n} \frac{a_{i\alpha} a_{j\alpha}}{k_\alpha} \tag{3.5}$$

where $a_{i\alpha}$ and $a_{j\alpha}$ are the elements in adjacent matrix A; k_α is the degree of node α.

If we assume the nodes i and j represent two users and the node α represents an item, it becomes a part of step 1 in MD model. The resource-allocation process then can be regarded as a one-step random walk on the user-item bipartite network starting from their common neighbors. So step 1 in MD model is equivalent to a similarity measurement process between two users when the initial resource on items is one unit.

3.2.3 Mixed Similarity Diffusion for Recommendation

A mixed similarity diffusion model is proposed by integrating both explicit feedback and implicit feedback. In MD, the resource is distributed based on each node's degree, which leads to non-personalized recommendations. While step 1 in MD only considers implicit feedback, MSD involves explicit feedback together. Similarly, a two-step resource-allocation process for MSD model is given below.

Step 1: We assume each item collected by the target user i is assigned with one unit of initial resource. So, the amount of resource will be distributed to user j is defined as

$$f'_{ij} = \sum_{\alpha=1}^{n} \frac{a_{i\alpha} a_{j\alpha} Cos(i,j)}{\sum_{k=1}^{m} a_{k\alpha} Cos(i,k)} f_{\alpha} \tag{3.6}$$

where $Cos(i,j)$ and $Cos(i,k)$ are the cosine similarity calculated by explicit feedback, e.g., ratings; $\sum_{k=1}^{m} a_{k\alpha} Cos(i,k)$ means the sum of similarity between the target user i and all users who have collected the item α, which is a normalization. In this step, we integrate the cosine similarity and RA index to propose a resource-allocation strategy based on mixed similarity and two kinds of feedback are both used for calculating the mixed similarity between users in the resource-allocation process.

Step 2: The resource allocated on users will flow back to items, in order to finish the resource redistribution process. We intend to consider both user's degree and item's degree to enhance the diversity of recommendations. A parameter λ is introduced into our model to control the impact of user's degree and item's degree in this step. Assuming item β will receive the resource from users, the final resource on item β can be defined as

$$f'_{i\beta} = \sum_{j=1}^{m} \frac{a_{j\beta}}{k_{\beta}^{\lambda} k_{j}^{1-\lambda}} f'_{ij} \tag{3.7}$$

where we substitute Eq. (3.6) into Eq. (3.7) to generate the final model of our proposed method and project the resource-allocation process onto an item-item network, as

$$f'_{i\beta} = \sum_{j=1}^{m} \frac{a_{j\beta}}{k_\beta^\lambda k_j^{1-\lambda}} \sum_{\alpha=1}^{n} \frac{a_{i\alpha}a_{j\alpha}Cos(i,j)}{\sum_{k=1}^{m} a_{k\alpha}Cos(i,k)} f_\alpha \tag{3.8}$$

Finally, all items are sorted by their final resource and then a top-L recommendation list of uncollected items is generated for the target user i. The pseudo-code of mixed similarity diffusion method is shown in Algorithm 3.1.

Algorithm 3.1 The algorithm of the MSD method.

Require: An adjacent matrix $\boldsymbol{A}^{m\times n}$, a cosine similarity matrix $\boldsymbol{Cos}^{m\times m}$ and a parameter λ

Ensure: A recommendation list for the target user i, which is generated by descending order of the final resource on uncollected items in $\boldsymbol{V}$

```
Initialization of a final resource vector V^{1×n} for the target user i
for β = 1,2,...,n do
    Set f'_{iβ} = 0
    for j = 1,2,...,m do
        Set f'_{ij} = 0
        for α = 1,2,...,n do
            Calculate f'_{ij} = f'_{ij} + a_{iα}a_{jα}Cos(i,j) / (Σ_{k=1}^{m} a_{kα}Cos(i,k)) f_α
        end for
        Calculate f'_{iβ} = f'_{iβ} + a_{jβ} / (k_β^λ k_j^{1−λ}) f'_{ij}
    end for
    Set V(β) = f'_{iβ}
end for
```

3.3 Data and Metrics

This section describes the details of three benchmark datasets at first, and then a series of evaluation metrics are presented.

Table 3.1: Statistics of datasets.

Dataset	#User	#Item	#Rating	Sparsity
ML100K	943	1682	100,000	6.30×10^{-2}
ML1M	6040	3706	1,000,209	4.47×10^{-2}
MLlatest	671	4801	94,537	2.93×10^{-2}

3.3.1 Data Description

We use three different versions of MovieLens datasets in our experiments including ML100K, ML1M and MLlatest to evaluate our proposed method in different circumstances. MovieLens datasets are public and real-world datasets crawled from movie review websites, which are widely used for evaluating the performance of algorithms in recommender systems. The ML100K consists of 943 users, 1682 items and 100,000 observed ratings, while the ML1M has 1,000,209 ratings of 6040 users and 3706 items. In addition, we extract 94,537 ratings of 4801 items which are at least collected by three users from the MovieLens latest dataset (MLlatest) published on September, 2016. The range of ratings in MovieLens datasets is $[1,5]$. Statistics of these three datasets are illustrated in Table 3.1.

Following some previous studies (Zhou et al., 2007; Wang et al., 2016c), we convert ratings to binary links to build a bipartite network where we assign 1 as ‘relevant’ for the ratings above 3 and 0 as ‘non-relevant’ for the remaining ratings. Note that the cosine similarity between users is calculated by the original ratings of all datasets, because it evaluates user’s preference by explicit feedback.

A five-fold cross-validation is utilized in our experiments. We randomly divide each dataset into five folds and four are regarded as the training set, with the remaining fold treated as the testing set. Five iterations are arranged to make sure that all folds are tested.

3.3.2 Evaluation Metrics

To present a comprehensive evaluation of the recommendation performance, we take advantage of some widely used evaluation metrics to measure the accuracy and diversity of our proposed method. The following metrics are used to measure the accuracy of recommendations.

Precision (*Pre@L*) is an important evaluation metric for ranking prediction in recommender systems, which measures the fraction of top-L recommended items that are consumed by the target user. Mathematically, the average value of *Pre@L* for all users is defined as

$$Pre@L = \frac{1}{m}\Big(\sum_{i=1}^{m} \frac{D_i(L)}{L}\Big) \tag{3.9}$$

where $D_i(L)$ is the number of recommended items consumed by user i in test set when the length of the recommendation list is L.

Recall (*Rec@L*) is another crucial metric in recommender systems, which calculates the proportion of correct recommended items and the number of total items in the test set for the target user. The average value of *Rec@L* for all users is defined as

$$Rec@L = \frac{1}{m}\Big(\sum_{i=1}^{m} \frac{D_i(L)}{T_i(L)}\Big) \tag{3.10}$$

where $T_i(L)$ is the number of items collected in the test set.

Rank-Biased Precision (*RBP@L*) (Moffat and Zobel, 2008) assumes each user has a fixed probability p to scan next recommended item from the first place in a recommendation list, defined as

$$RBP@L = \frac{1}{m}\Big(\sum_{i=1}^{m}(1-p)\sum_{\alpha=1}^{L} c_{i\alpha}p^{\alpha-1}\Big) \tag{3.11}$$

where $c_{i\alpha} = 1$ means that the αth item in the recommendation list L is collected by user i in test set and $c_{i\alpha} = 0$ is the opposite. *RBP@L* is a significant ranking-based measurement that is very needful, because users always accept the recommended items at the top of a recommendation list and *RBP@L* is very close to an individual's actual habits of collecting items. Here, we assume the probability p is 0.5.

Mean Reciprocal Rank (*MRR*) directly utilizes the reciprocal of the item's position in a recommendation list to measure the performance of recommendation algorithms, defined as

$$MRR = \frac{1}{m}\Big(\sum_{i=1}^{m}\sum_{\alpha\in S(i)}\frac{1}{rank_{\alpha}^{i}}\Big) \tag{3.12}$$

where $S(i)$ is the items collected by user i in test set and $rank_{\alpha}^{i}$ is the position of item i in the recommendation list for user i. Be similar to *RBP@L*, *MRR* is also a ranking-based measurement that supposes the item consumed by the target user in test set placed at the top of the recommendation list obtains a grater score in the evaluation than the item at the bottom. Therefore, a larger value of *MRR* means a better performance.

The diversity also plays an important role in recommender systems, which indicates the ability of pushing out unpopular items for users. A metric used to measure the diversity of recommendations is represented as follows.

Hamming Distance (*Ham@L*) is a common method to evaluate the diversity of recommendations. The definition of *Ham@L* for all users is

$$Ham@L = \frac{1}{m(m-1)} \sum_{i \neq j} \left(1 - \frac{Q_{ij}(L)}{L}\right) \tag{3.13}$$

where $Q_{ij}(L)$ is the number of overlapped items in the recommendation lists for user i and user j. The larger value of *Ham@L* means the higher diversity.

3.4 Experiments

This section introduces the baselines that will be used for comparing with our method, the impact of parameter λ , and the results of the comparative experiments.

3.4.1 Baselines

We intend to compare the performance of our proposed method with some classic baselines to verify the superiority.

PopRank is a basic recommendation algorithm with implicit feedback, which provides a recommendation list based on items' popularity. The most popular item will be arranged at the top. In our experiments, we regard the degree of each item as its popularity. The item with larger degree means it has already collected by more users, i.e., grater popularity.

UserCF is a classic collaborative filtering method based on the cosine similarity between users. This method assumes the target user will accept the opinions from the most similar users.

MD (Zhou et al., 2007) is a pioneer of diffusion-based recommendation algorithms, which uses a resource-allocation process to make recommendations on bipartite networks.

HC (Zhang et al., 2007) employs the heat conduction process of physical dynamics on recommendation tasks. This model is good at pushing out small-degree items, so the results have a high diversity in general.

CosRA (Chen et al., 2017b) is a vertex similarity index on bipartite networks. The recommendation algorithm based on CosRA index is named CosRA-based method that can be regarded as a special situation of hybrid diffusion (Zhou et al., 2010b).

SPMD (Zeng et al., 2014) is an improvement on mass diffusion model. It introduces similarity-preferential diffusion into the recommendation process, which can enhance or suppress the weight of users who are most similar to the target user.

BHC (Liu et al., 2011) is an improvement on heat conduction model. Because heat conduction model sacrifices the accuracy of recommendations to push out small-degree items, BHC proposes a biased resource distribution strategy to enhance the precision of recommendations.

3.4.2 The Impact of Parameter λ

In the mixed similarity diffusion model, the parameter λ controls the impact of user's degree and item's degree in the second step of the resource-allocation process. To determine the optimal value of λ in our method, we adjust the parameter on all three datasets. The precision and recall can comprehensively reflect the accuracy, so we use these two metrics to determine the optimal value of λ for MSD when the recommendation list is $L = 10$.

Figure 3.2 reports the results of *Pre@10* and *Rec@10* for our proposed method when the parameter λ changes from 0 to 1 at a calculative step of 0.05. Figure 3.2(a), (b) and (c) represent the variation of *Pre@10* and

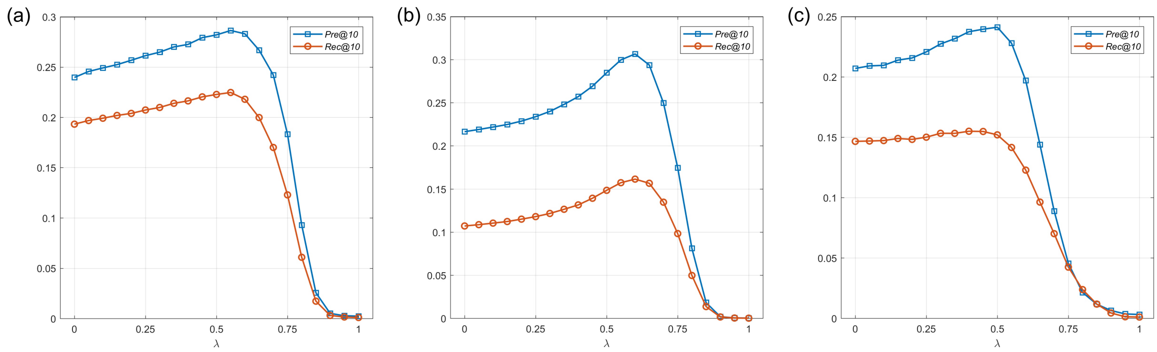


Figure 3.2: The *Pre@10* and *Rec@10* of MSD when changing the parameter λ between 0 and 1 at a calculative step of 0.05 in ML100K, ML1M and MLlatest datasets are represented. (a) The optimal value is $\lambda = 0.55$ in ML100K. (b) The optimal value is $\lambda = 0.6$ in ML1M. (c) The optimal value is $\lambda = 0.5$ in MLlatest.

Rec@10 in ML100K, ML1M and MLlatest datasets, respectively. It can be seen from Figure 3.2, the optimal value of the parameter λ is 0.55, 0.6 and 0.5 for ML100K, ML1M and MLlatest, respectively. Even though the sparsity of these three datasets is totally different, the observation that the optimal λ locates around 0.55 might support the inference that our proposed model has certain practical value. Generally, if $\lambda = 0$, our model becomes a simple one that only combines MD model with the cosine similarity with explicit feedback. According to the experiment results, with the increasing value of parameter λ, the degree of items provides more impact on the final recommendations and improve the accuracy.

3.4.3 Recommendation Performance Evaluation

Here, we use three real-world rating datasets to evaluate our proposed method that is compared with seven baselines. The whole experiment results are presented in Table 3.2 and the optimal parameters of every algorithm in different datasets are also included in this table for result reproducibility.

In ML100K, MSD obtains the best results in all metrics of accuracy, i.e., Precision, Recall and Rank-Biased Precision, when the length of the recommendation list is 5 and 10. Compared to SPMD, *Pre@5*, *Rec@5*, *RBP@5* and *MRR* can be improved 4.8%, 6.5%, 6.6% and 5.2% by MSD, respectively. The results of MSD and BHC are same on *Ham@5*, however, MSD gets a better result on *Ham@10*. In ML1M, all the best results on

Table 3.2: Recommendation performance of MSD and seven baselines in the ML100K, ML1M and MLlatest datasets.

ML100K	$Pre@5$	$Pre@10$	$Rec@5$	$Rec@10$	$RBP@5$	$RBP@10$	MRR	$Ham@5$	$Ham@10$
PopRank	0.1639	0.1560	0.0567	0.1106	0.2014	0.2061	0.7664	0.5126	0.4601
UserCF	0.2698	0.2261	0.1086	0.1743	0.3101	0.3161	1.0607	0.7319	0.6769
MD	0.2781	0.2387	0.1167	0.1932	0.3210	0.3275	1.1114	0.7534	0.7150
HC	0.0025	0.0036	0.0013	0.0032	0.0015	0.0016	0.1702	0.8852	0.8686
CosRA	0.3139	0.2644	0.1267	0.2059	0.3563	0.3634	1.2229	0.8378	0.8082
SPMD ($\theta = 2.25$)	0.3222	0.2693	0.1316	0.2077	0.3601	0.3670	1.2317	0.8608	0.8434
BHC ($\lambda = 0.75$)	0.3194	0.2704	0.1223	0.2058	0.3565	0.3636	1.2380	**0.9109**	0.8774
MSD ($\lambda = 0.55$)	**0.3376**	**0.2863**	**0.1401**	**0.2247**	**0.3840**	**0.3912**	**1.2956**	**0.9109**	**0.8849**
ML1M	$Pre@5$	$Pre@10$	$Rec@5$	$Rec@10$	$RBP@5$	$RBP@10$	MRR	$Ham@5$	$Ham@10$
PopRank	0.1939	0.1681	0.0444	0.0736	0.2091	0.2135	0.8160	0.5128	0.4729
UserCF	0.2448	0.2032	0.0612	0.0970	0.2740	0.2792	0.9927	0.6326	0.5670
MD	0.2656	0.2164	0.0690	0.1070	0.2988	0.3042	1.0688	0.6926	0.6103
HC	0.0007	0.0026	0.0003	0.0020	0.0004	0.0005	0.2248	0.8812	0.8470
CosRA	0.3063	0.2507	0.0856	0.1330	0.3524	0.3586	1.2262	0.7965	0.7197
SPMD ($\theta = 3.05$)	0.3175	0.2626	0.0847	0.1331	0.3560	0.3626	1.2547	0.8670	0.8166
BHC ($\lambda = 0.85$)	0.3334	0.2885	0.0849	0.1458	0.3486	0.3566	1.3089	0.9315	0.9011
MSD ($\lambda = 0.6$)	**0.3674**	**0.3066**	**0.1040**	**0.1614**	**0.4072**	**0.4152**	**1.4234**	**0.9440**	**0.9126**
MLlastest	$Pre@5$	$Pre@10$	$Rec@5$	$Rec@10$	$RBP@5$	$RBP@10$	MRR	$Ham@5$	$Ham@10$
PopRank	0.1939	0.1681	0.0444	0.0736	0.2091	0.2135	0.8160	0.5128	0.4729
UserCF	0.2448	0.2032	0.0612	0.0970	0.2740	0.2792	0.9927	0.6326	0.5670
MD	0.2656	0.2164	0.0690	0.1070	0.2988	0.3042	1.0688	0.6926	0.6103
HC	0.0007	0.0026	0.0003	0.0020	0.0004	0.0005	0.2248	0.8812	0.8470
CosRA	0.3063	0.2507	0.0856	0.1330	0.3524	0.3586	1.2262	0.7965	0.7197
SPMD ($\theta = 3.05$)	0.3175	0.2626	0.0847	0.1331	0.3560	0.3626	1.2547	0.8670	0.8166
BHC ($\lambda = 0.85$)	0.3334	0.2885	0.0849	0.1458	0.3486	0.3566	1.3089	0.9315	0.9011
MSD ($\lambda = 0.6$)	**0.3674**	**0.3066**	**0.1040**	**0.1614**	**0.4072**	**0.4152**	**1.4234**	**0.9440**	**0.9126**

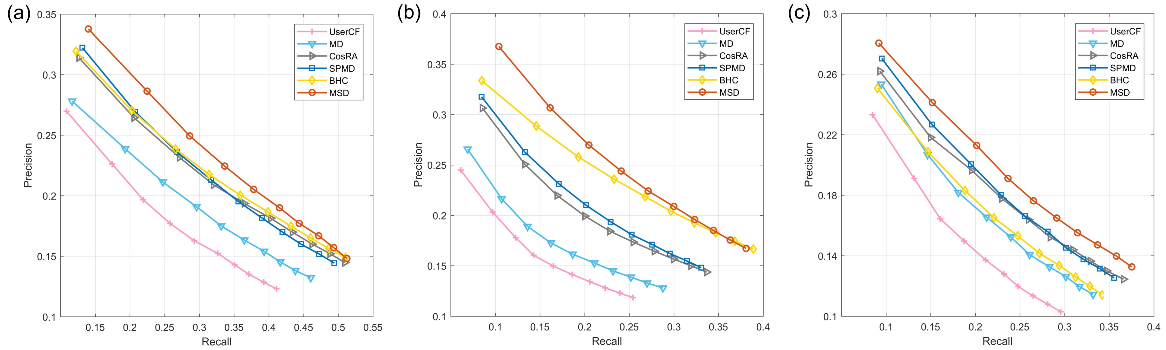


Figure 3.3: The Precision-Recall curves in the ML100K, ML1M and MLlatest datasets are represented in diagrams (a), (b) and (c), respectively, where the length of recommendations is from 5 to 50 at a calculative step of 5. Because of the poor performance of PopRank and HC in these two metrics, we do not show their results in this figure.

the accuracy and the diversity are achieved by MSD that enhances *Pre@5*, *Rec@5*, *RBP@5* and *MRR* by 10.2%, 22.5%, 16.8% and 8.7% than BHC. MSD also brings an 1.3% improvement on *Ham@5* than BHC which gets the best on diversity in seven baselines. Furthermore, MSD achieves the best results on most of metrics in MLlatest dataset.

Figure 3.3 depicts the Precision-Recall curves on three datasets with the length of the recommendation list from 5 to 50 at a calculative step of 5. In Figure 3.3(a) and (c), MSD always has the best results. In Figure 3.3(b), MSD gets the best performance when the recommendation list is short. As the length increases, BHC gradually obtains better performance than MSD. However, a user always pay more attention to the items at the top of a recommendation list, so top-10 recommendations are the most important in evaluating recommender systems, which means the algorithm with the best performance in the short recommendation list is more meaningful

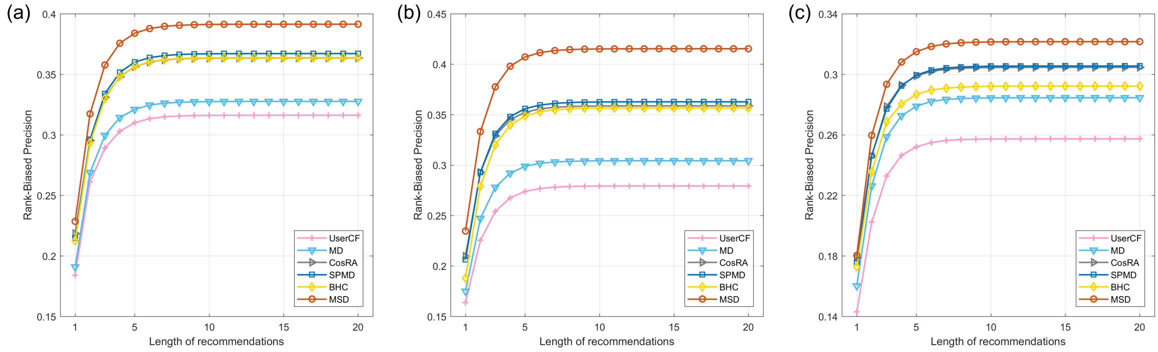


Figure 3.4: The Rank-Biased Precision ($p = 0.5$) of MSD and five baselines in the ML100K, ML1M and MLlatest datasets are represented in diagrams (a), (b) and (c), respectively, where the length of recommendations is from 1 to 20. Because of the poor performance of PopRank and HC in this metric, we do not show their results in this figure.

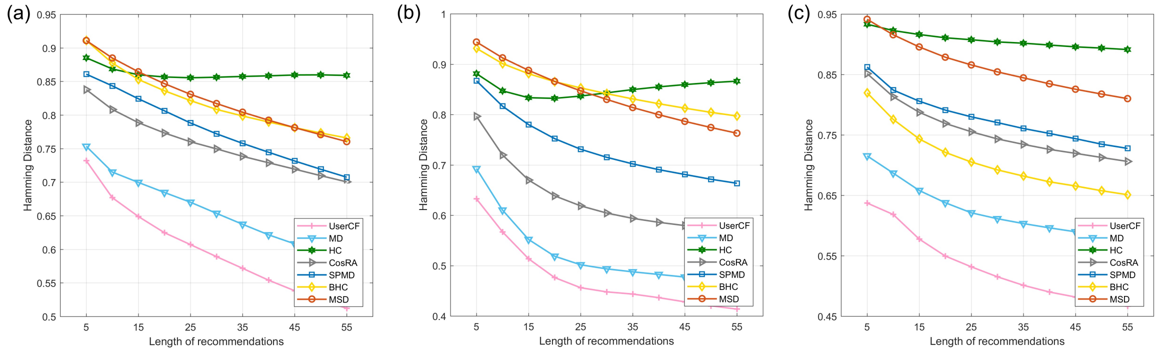


Figure 3.5: The Hamming Distance of MSD and six baselines in the ML100K, ML1M and MLlatest datasets are represented in diagrams (a), (b) and (c), respectively, where the length of recommendations is from 5 to 55 at a calculative step of 5. Because of the poor performance of PopRank in this metric, we do not show its results in this figure.

for practical applications. The results of Rank-Biased Precision are shown in Figure 3.4, MSD always keeps the best results in three benchmark datasets.

The accuracy-diversity dilemma is ubiquitous in recommender systems (Zhou et al., 2010b). A popular item should be accepted by most users, so recommending a list of popular items to a user in accordance with his/her preference may enhance the accuracy but reduce the diversity. Figure 3.5 indicates the results of Hamming Distance. HC focuses on pushing out small-degree items which lead to a high diversity with the increasing of the recommendation list. However, considering the poor performance of HC on the accuracy, it is hard to apply HC in real-world systems. When the recommendation list is short, MSD surpasses HC and proposes recommendations with higher diversity. Therefore, our proposed method improves the accuracy and the diversity at the same time, which means it relieves this dilemma to some extent and makes an accuracy-diversity balance in recommender systems.

3.5 Summary

This chapter proposes a mixed similarity diffusion model to improve the performance of recommendation, which integrates the similarity from both explicit feedback and implicit feedback. We calculate the cosine similarity with explicit feedback data and the resource-allocation index with implicit feedback data, and combine these two kinds of similarity into diffusion processes. Experiments on three real-world datasets demonstrate our method performs better than most baselines. Specifically, MSD brings some significant improvements compared to BHC, SPMD and CosRA which are the state-of-the-art diffusion-based recommendation algorithms.

MSD also proposes recommendations with higher diversity than HC when the recommendation list is short. Therefore, MSD achieves an accuracy-diversity balance, which enhances the accuracy and the diversity at the same time.

CHAPTER 4

DIFFUSION-BASED RECOMMENDATION WITH TRUST RELATIONS ON TRIPARTITE NETWORKS

4.1 Introduction

A number of advanced recommendation algorithms have been proposed by researchers both in physics (Zhou et al., 2010b; Zhang et al., 2007) and computer science (Sarwar et al., 2001). Particularly, physical dynamics, which is employed in complex networks by diffusion-based methods and can make personalized recommendations, have attracted great attention from many researchers (Zhu et al., 2014; Yu et al., 2016). Mass diffusion (MD) (Zhou et al., 2007) and heat conduction (HC) (Zhang et al., 2007) can be regarded as the pioneers of diffusion-based recommendation approaches. They distribute the resource of each node through a two-step resource-allocation process on bipartite networks in different ways. Subsequently, several closely related methods were proposed, such as changing resource-allocation process between nodes to enhance accuracy (Liu et al., 2011) and altering the initial resource distribution of nodes on networks to improve recommendation performance (Jia et al., 2008).

With the development of social networks, it is necessary to consider both users' own behaviors and trust relations between users when modeling user preference (Dong et al., 2022). Because users can be affected by others and may change their opinions and behaviors when communicating with trusted users on social networks, such as friends. However, due to the limitations of bipartite networks, there is a lack of diffusion-based recommendation methods integrated with trust relations on social networks. Some previous research results demonstrate that social information can bring significant improvements to recommendations and a vital additional feature for ratings in recommender systems.

In this chapter, a Diffusion-Based Recommendation method with Trust relations (DBRT) on tripartite networks is proposed to integrate users' social trust into a recommendation process. The tripartite network has been verified as an effective way to combine extra features into the diffusion-based recommendation approach and has already been applied in collaborative tagging systems (Shang et al., 2010; Zhang et al., 2010). However, they only apply the original MD algorithm on two different bipartite networks firstly, and then combine the final resource on tripartite networks.

Compared to existing algorithms on tripartite networks, DBRT provides a consistent and synergetic two-step resource-allocation process that combines the resource from a user-object network and a user trust network in the first step and lets the resource flow back to objects in the second step. Moreover, users' social trust relations, such as implicit and explicit trust, are introduced into the diffusion-based recommendation approach in our method. Extensive experiments on three real-world datasets indicate that DBRT obtains remarkable improvements over most of the benchmark approaches. The main contributions of this chapter are summarized as follows.

1) A novel diffusion-based recommendation method, named DBRT, is proposed to extend resource-allocation processes from bipartite networks to tripartite networks. DBRT simulates a trust diffusion process through a user-user trust network to introduce explicit trust relations into resource-allocation processes.

2) Implicit trust on social networks are explored during resource-allocation processes. DBRT uses cosine index between nodes to implement the similarity calculation of users and assumes users may trust another users who has high similarities. Then, a special resource-allocation process has been designed, which combines both explicit and implicit trust to model user preference exacter.

3) Extensive experiments have been conducted to evaluate the effectiveness of DBRT. In the experiments, we compare the proposed method

with several state-of-the-art recommendation methods and analyze the impacts of parameters in DBRT. Results show that considering trust relations can improve the accuracy and diversity of recommendation results.

4.2 The Proposed Method

In a recommender system, user-object relations can be described on a bipartite network $G(U,O,E)$. The user set is defined as $U = \{u_1, u_2, \ldots, u_m\}$ and the object set is defined as $O = \{o_1, o_2, \ldots, o_n\}$, where m and n are the numbers of users and objects. The link set between users and objects is defined as $E = \{e_1, e_2, \ldots, e_z\}$ and z is the amount of links. We can use an $m \times n$ adjacent matrix A to describe the bipartite network $G(U,O,E)$. To make it easy to understand, we use Greek and Latin letters to express object-related and user-related indices respectively. Accordingly, the element in the adjacent matrix is represented as $a_{i\alpha} = 1$ if there is a link between node o_α and node u_i, which means object α is collected by user i, and $a_{i\alpha} = 0$ otherwise. The degree of the object α and user i are defined as k_α and k_i, which represents the number of users who collect object α and the number of objects collected by user i, respectively. The primary purpose of a recommender system is to provide a recommendation list for a target user. That is to say, a set of objects uncollected by the target user with the highest recommendation scores should be included in the recommendation list. The length of the recommendation list is defined as

L in this chapter.

Before introducing our method, we classify social trust relations in recommender systems into two categories, which are explicit trust and implicit trust. The explicit trust means that trust statements are directly specified by users. For instance, users can add other users to their trust lists or establish friendships with other users on social websites, such as Ciao, Epinions and Facebook, so explicit trust relations can be found between friends. By contrast, the implicit trust is the relationship that cannot be directly observed in social trust networks (Guo et al., 2016a). It is often inferred by other information, such as user similarities in ratings. If two users have a high similarity, we can assume they have implicit trust. In this section, we propose a novel diffusion-based recommendation method integrated with both categories of social trust relations. Additional details about the proposed method are discussed in the following sections.

4.2.1 Implicit Trust between Users

Considering a situation that two users have a lot of related behaviors, such as purchasing the same things or giving similar ratings to movies. However, their trust relation cannot be observed in their trust or friend lists. According to their similar user behaviors, we assume that implicit trust exists between these two users, which means they may share and adopt each other's choices and preference. We use the similarity between users to calculate the implicit trust in the proposed method. The cosine index is a widely used similarity evaluation approach on bipartite networks, and it has already proved effectual for measuring the similarity between objects (Chen et al., 2017b). As a matter of fact, the cosine index calculates an inner product space of two object vectors. For two objects α and β, the cosine index is defined as

$$cos^{o}_{\alpha\beta} = \frac{1}{\sqrt{k_\alpha k_\beta}} \sum_{i=1}^{m} a_{i\alpha} a_{i\beta} \tag{4.1}$$

where k_α and k_β are the degree of objects α and β, $a_{i\alpha}$ and $a_{i\beta}$ are the elements in the adjacent matrix which indicate the links between user i and these two objects α and β. Using Eq. (4.1) the similarity can be measured by the cosine index if the two objects are collected by the same user at least once, otherwise the similarity will be 0.

We suggest using the analogous method to calculate the similarity between users. User rating behaviors can also be expressed as user vectors, e.g., if a user collects an object, the corresponding element in the user vector is 1; otherwise it is 0. An inner product space of two user vectors measures the similarity between these two users, which is defined as

$$cos^{u}_{ij} = \frac{1}{\sqrt{k_i k_j}} \sum_{\alpha=1}^{n} a_{i\alpha} a_{j\alpha} \tag{4.2}$$

where k_i is the degree of user i and k_j is the degree of user j, respectively. We suppose that an implicit trust relation between two users exists if there is a similarity between them. The values of similarity indicate the level of implicit trust. For example, if two users collect many common objects, they will have a large inner product that is equivalent to a large value of similarity between these users, which demonstrates a high level of implicit trust between them. Although the explicit trust of these two users is not observed on trust networks, they could share each other's preference and opinions as well. The implicit trust is a significant feature in recommender systems, and introducing it into the diffusion-based recommendation method can make resource-allocation processes be more reasonable and effective.

4.2.2 Resource-allocation on Explicit Trust Networks

Explicit trust means that the trust relations between users are observed. In social networks, explicit trust can be divided into two categories: symmetric trust and asymmetric trust. Specifically, asymmetric trust is a more common situation and symmetric trust can be represented by two asymmetric trust links. Trust relations on Twitter are a typical example of asymmetric trust that a user could be a follower for other users on Twitter, but the other users may not be followers of the user as well, which means the trust relations are asymmetric.

We propose a resource-allocation process on trust networks where explicit trust relations are observed. The main idea is a user can obtain resource from followers on asymmetric trust networks. The explicit trust network can be defined as a monopartite graph represented on an $m \times m$ adjacent matrix B, where the element $b_{ij} = 1$ if the explicit trust is observed on the trust network, otherwise $b_{ij} = 0$. We only make a one-step resource-allocation process so that users obtain resource from their trusting users, e.g., followers. This is different to the original MD method (Zhou et al., 2007) because we cannot ensure that the trusted user is also a trusting user simultaneously in asymmetric trust networks, so we do not let the resource flow back. The resource-allocation process on the explicit trust network is defined as

$$f_j^t = \sum_{l=1}^{m} \frac{b_{jl}}{k_l^t} f^t(l) \tag{4.3}$$

where f_j^t means the resource obtained by user j on the explicit trust network, b_{jl} is the element in the adjacent matrix and k_l^t is the degree of user l on the explicit trust network. Note that $f^t(l)$ is the initial resource of user l on the explicit trust network and it is one unit of resource for convenient calculation (Shang et al., 2010). The explicit trust is regarded as an auxiliary feature that helps the implicit trust improve the performance of the diffusion-based recommendation approach.

4.2.3 Recommendation with Integrated Social Trust Relations

This section will propose a diffusion-based recommendation method to integrate the implicit trust and explicit trust. One previous study (Chen et al., 2017b) indicates CosRA-based method applies CosRA index between objects to make better recommendations, which is a two-step resource distribution process on a user-object bipartite network. In each step, the

CosRA-based method distributes resource based on the degree of both each object and its neighbouring users at the same time, shown in Eq. (4.4),

$$f^{'(i)} = S^{CosRA} f^{(i)} \tag{4.4}$$

where $S^{CosRA} = \frac{1}{\sqrt{k_\alpha k_\beta}} \sum_{j=1}^{m} \frac{a_{j\alpha} a_{j\beta}}{k_j}$ is CosRA index to measure the similarity between objects α and β. $f^{(i)}$ is a n-dimensional vector indicating the initial resource of all objects given the target user i and $f^{'(i)}$ is the vector recording all the final resource of each object. However, the CosRA-based method only considers the similarity between objects without the implicit trust between users in its recommendation process. To solve this weakness, we distribute resource based on not only each object's degree but also the implicit trust between the target user and the neighbouring users of each object. Our method is described as a two-step resource distribution process.

Step 1: We assume the objects collected by the target user i are assigned with one unit of resource which will be distributed to all the neighbouring users. The resource of user j obtained from objects can be written as

$$f^{'}_{ij} = \sum_{\alpha=1}^{n} \frac{a_{i\alpha} a_{j\alpha}}{\sqrt{k_\alpha \sqrt{k_i k_j}}} f(\alpha) \tag{4.5}$$

where k_α is the degree of object α, k_i is the degree of the target user i and k_j is the degree of user j who obtains resource from the neighbouring objects. $f(\alpha)$ is the initial resource of object α, which is one unit of resource.

It can be seen from Eq. (4.5), the cosine index between is used to evaluate the implicit trust between users. Furthermore, the explicit trust also needs to be utilized in our method to improve the recommendation performance. As mentioned in Section 4.2.2, we consider to employ a one-step resource-allocation process to represent the effect of explicit trust. So, the user j not only gets resource from the user-object network but also obtains resource from the explicit trust network simultaneously. We adopt a simply way to linearly combine the resource from the explicit trust network with the resource from objects, defined as

$$\begin{aligned} f_{ij}^{'} &= \lambda * f_{ij}^{t} + (1-\lambda) * \sum_{\alpha=1}^{n} \frac{a_{i\alpha} a_{j\alpha}}{\sqrt{k_{\alpha}\sqrt{k_i k_j}}} f(\alpha) \\ &= \lambda * \sum_{l=1}^{m} \frac{b_{il} b_{jl}}{k_l^t} f^t(l) + (1-\lambda) * \sum_{\alpha=1}^{n} \frac{a_{i\alpha} a_{j\alpha}}{\sqrt{k_{\alpha}\sqrt{k_i k_j}}} f(\alpha) \end{aligned} \tag{4.6}$$

where $f_{ij}^t = \sum_{l=1}^{m} \frac{b_{il} b_{jl}}{k_l^t} f^t(l)$ is extended from f_j^t in Eq. (4.3), because the target user i is considered in the recommendation process. The parameter $\lambda \in [0,1]$ is a tunable parameter to control the proportion of resource from objects and followers.

Step 2: The resource of users should flow back to objects. Assuming object β will obtain the resource from users, the final resource of object β can be calculated as

$$f_{i\beta}^{'} = \sum_{j=1}^{m} \frac{a_{j\beta}}{\sqrt{k_{\beta}\sqrt{k_i k_j}}} f_{ij}^{'} \tag{4.7}$$

where k_{β} is the degree of object β and $f_{ij}^{'}$ is the resource of user j after the step 1. We substitute Eq. (4.6) into Eq. (4.7) to generate the final model of

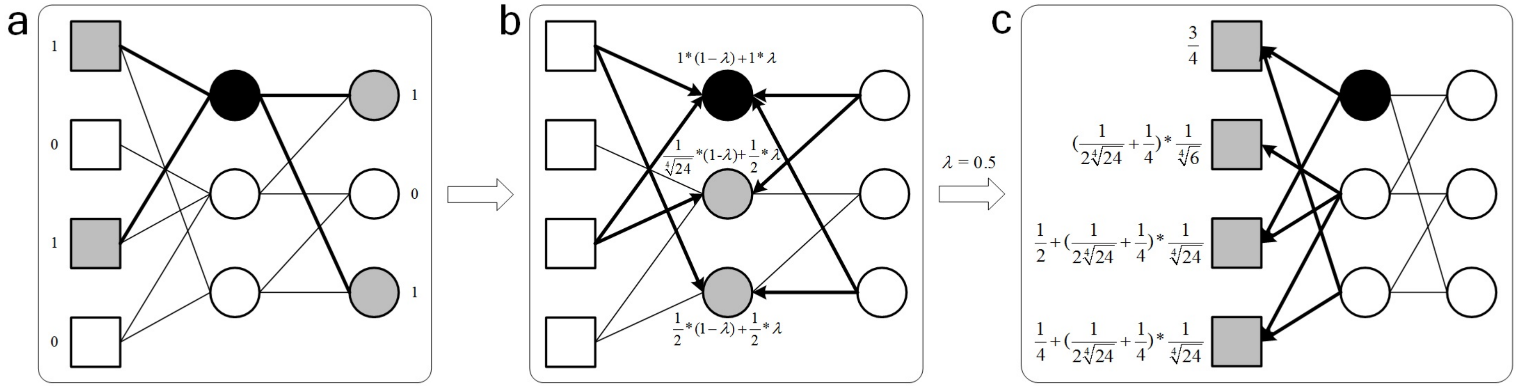


Figure 4.1: An illustration of the diffusion-based recommendation method with trust relations (DBRT). Users and objects are represented by circles and squares, respectively. In each plot, the links between circles and squares on the left side are user-object relations, and the links between circles on the right side are explicit trust relations. The black circle means the target user. The circles and squares with grey color indicate the resource is distributed on these nodes. Plot (a) is the initial configuration, objects and followers linked with the target user obtain one unit of resource. Plot (b) indicates the step 1, in which the resource of objects and followers linked with the target user flows to users on tripartite networks, the resource of each user can be calculated by Eq. (4.6). Plot (c) is the step 2, in which the users distribute resource to all the objects linked with them, the final resource of each object can be calculated by Eq. (4.7). $\lambda = 0.5$ is used as an example.

the proposed method, as presented in Eq. (4.8). An example of our method is shown in Figure 4.1.

$$f'_{i\beta} = \sum_{j=1}^{m} \frac{a_{j\beta}}{\sqrt{k_\beta \sqrt{k_i k_j}}} \left(\lambda * \sum_{l=1}^{m} \frac{b_{il} b_{jl}}{k_l^t} f^t(l) + (1-\lambda) * \sum_{\alpha=1}^{n} \frac{a_{i\alpha} a_{j\alpha}}{\sqrt{k_\alpha \sqrt{k_i k_j}}} f(\alpha) \right) \tag{4.8}$$

Finally, all objects are sorted by their final resource and then a top-L recommendation list of uncollected objects is generated for the target user i. The pseudo-code of DBRT is shown in Algorithm 4.1.

Algorithm 4.1 The algorithm of the DBRT method.

Require: An adjacent matrix $\boldsymbol{A}^{m\times n}$, an explicit trust matrix $\boldsymbol{B}^{m\times m}$ and a parameter λ

Ensure: An objects' resouce vector $\boldsymbol{V}^{1\times n}$ of target user u_i

```
Initialization of a resource vector V^{1×n} for the target user u_i
for β = 1,2,...,n do
    Set f'_{iβ} = 0
    for j = 1,2,...,m do
        Set f'_{ij} = 0
        for α = 1,2,...,n do
            Calculate f'_{ij} = f'_{ij} + (a_{iα} a_{jα} / sqrt(k_α sqrt(k_i k_j))) f(α)
        end for
        for l = 1,2,...,m do
            Calculate f'_{ij} = f'_{ij} + (b_{il} b_{jl} / k_l^t) f^t(l)
        end for
        Calculate f'_{iβ} = f'_{iβ} + (a_{jβ} / sqrt(k_β sqrt(k_i k_j))) f'_{ij}
    end for
    Set V(β) = f'_{iβ}
end for
```

A simpler model is shown in Eq. (4.9), when $\lambda = 0$.

$$f'_{i\beta} = \sum_{j=1}^{m} \frac{a_{j\beta}}{\sqrt{k_\beta \sqrt{k_i k_j}}} \sum_{\alpha=1}^{n} \frac{a_{i\alpha} a_{j\alpha}}{\sqrt{k_\alpha \sqrt{k_i k_j}}} f(\alpha)$$

$$= \sum_{j=1}^{m} \frac{a_{j\beta}}{\sqrt{k_i k_j}} \sum_{\alpha=1}^{n} \frac{a_{i\alpha} a_{j\alpha}}{\sqrt{k_\alpha k_\beta}} f(\alpha) \tag{4.9}$$

Under this circumstance, our model only takes advantage of the implicit trust to improves the recommendation performance.

Note that the resource from the explicit trust network is regarded as additional resource for the resource from the user-object network. Each user obtains the additional resource in step 1 and then transfers it to linked objects in step 2. A user with large degree on the explicit trust network is an influential user who can transfer more additional resource to objects. Therefore, the objects collected by the influential user get more final resource and these objects are more likely to be arranged at the top of a recommendation list. As a result, the recommendation performance will be improved, because the influential user's choices have a higher probability to be accepted by others.

4.3 Datasets and Metrics

In this section, details of the three benchmark datasets are described, and then the evaluation methods used in this chapter are shown.

4.3.1 Data Description

The three benchmark datasets used in our experiments are Ciao, Epinions and Flixster. These three datasets all contain social information that can be used as trust relations in recommender systems (Guo et al., 2016a).

Ciao, Epinions and Flixster are public, real-world datasets, and widely used in the evaluation of previous trust-combined recommender systems. When building bipartite networks, we convert ratings to binary links by assigning 1 as 'relevant' for the ratings above 3 and 0 as 'non-relevant' for the remaining ratings. The Ciao dataset consists of 2960 users, 4394 objects and 77,861 observed rating links, while the Epinions dataset has 70,438 rating links from 5000 users and 3000 objects. In addition, the Flixster dataset contains 6072 users, 5366 objects and 115,840 rating links. There are also 56,998, 139,982 and 167,552 trust links in the Ciao, Epinions and Flixster. Statistics of the datasets are illustrated in Table 4.1.

Table 4.1: Statistics of the Ciao, Epinions and Flixster datasets.

Dataset	#User	#Object	#Rating link	Sparsity	#Trust link
Ciao	2960	4394	77,861	5.99×10^{-3}	56,998
Epinions	5000	3000	70,438	4.70×10^{-3}	139,982
Flixster	6072	5366	115,840	3.56×10^{-3}	167,552

A five-fold cross-validation is used for evaluations in our experiments. Specifically, we randomly divide each dataset into five folds. Four are regarded as the training set, with the remaining fold treated as the testing set. Five iterations are arranged to make sure that all folds are tested.

4.3.2 Metrics

To present a comprehensive evaluation of recommendation performance, some widely investigated evaluation metrics are employed to measure the accuracy and diversity of our proposed method.

Precision (Chen et al., 2017b) is an important metric in recommender systems that measures the proportion of the number of recommended objects appearing in the test set to the length of a recommendation list. Mathematically, the average value of precision for all users is defined as

$$P(L) = \frac{1}{m}\left(\sum_{i=1}^{m} \frac{D_i(L)}{L}\right) \tag{4.10}$$

where $D_i(L)$ is the number of objects appearing in both the recommendation list L and in the test set aimed at user i.

Recall (Zhou et al., 2010b) is also a crucial metric in recommender systems, it measures the proportion of correct recommended objects and the number of total objects in the test set, as

$$R(L) = \frac{1}{m}\left(\sum_{i=1}^{m} \frac{D_i(L)}{T_i(L)}\right) \tag{4.11}$$

where $T(i)$ is the number of objects collected by user i in the test set.

F1 (Wang et al., 2016b) is used to provide a comprehensive assessment of our method. It is a two-dimensional vector, which considers both precision and recall simultaneously and provides a balanced evaluation. The F1 metric is defined as

$$F1(L) = \frac{2P(L) \times R(L)}{P(L) + R(L)} \tag{4.12}$$

where $P(L)$ and $R(L)$ are precision and recall when the length of the recommendation list is L.

Rank-biased precision (RBP) (Moffat and Zobel, 2008) assumes each user scans recommended objects from the first place in a recommendation list and browses the next object with a fixed probability p, defined as

$$RBP(L) = \frac{1}{m}\left(\sum_{i=1}^{m}(1-p)\sum_{\alpha=1}^{L} c_{i\alpha}p^{\alpha-1}\right) \tag{4.13}$$

where $c_{i\alpha} = 1$ means user i has already collected αth object in the recommendation list and $c_{i\alpha} = 0$ is the opposite. Users always accept the recommended objects at the top of a recommendation list so that evaluating the performance of recommendation algorithms based on a recommendation sequence is very necessary. RBP is very close to an individuals' actual habits of collecting objects, which is a more reasonable way to evaluate the accuracy of recommendation methods.

Average reciprocal hit rank (ARHR) (Kabbur et al., 2013) is different from RBP, as it directly uses the reciprocal of the object's position in a recommendation list without a hypothetical collection probability. The rank of the objects in a recommendation list impacts the possibility that they can be collected by users in the top-n recommender systems. ARHR is defined as

$$ARHR = \frac{1}{m}\left(\sum_{i=1}^{m}\sum_{\alpha\in hits}\frac{1}{pos_{i\alpha}}\right) \tag{4.14}$$

where hits means the recommendations are correctly verified in the test sets, accordingly, $pos_{i\alpha}$ is the position of object α in the recommendation list for user i. Note that the object at the bottom of the list should get a low score in this metric, so a higher ARHR denotes better accuracy.

Hamming distance (Zhou et al., 2008) is a common way to measure the diversity of recommendation algorithms, which can be defined as

$$H_{ij}(L) = 1 - \frac{C_{ij}(L)}{L} \tag{4.15}$$

where $C_{ij}(L)$ means the common objects in the recommendation lists for user i and user j. If two users have the same recommendation list, $H_{ij}(L)$ will be 0 and two completely different recommendation lists lead to $H_{ij} = 1$. Finally, the average of H_{ij} over all the user pairs is denoted by the mean distance in Eq. (4.16).

$$H(L) = \frac{1}{m(m-1)} \sum_{i \neq j} H_{ij}(L) \tag{4.16}$$

Novelty (Lü et al., 2012) is a metric to evaluate the algorithm's ability to generate unpopular results. In general, the average popularity of objects in recommendation lists is used to represent the novelty, which is defined as

$$N(L) = \frac{1}{mL} \left(\sum_{i=1}^{m} \sum_{o_\alpha \in o_i^L} k_\alpha \right) \tag{4.17}$$

where k_α is the degree of object α in the recommendation list o_i^L of user i. To some extent, small-degree objects are regarded as unpopular objects. Hence, a small novelty value means the algorithm is good at pushing unpopular objects out.

4.4 Experimental Results

An introduction of benchmark methods is proposed in Section 4.4.1. The impact of the parameter in our method is then analyzed in Section 4.4.2. Comparisons on the performance of accuracy, diversity and novelty between DBRT and eight benchmark approaches are shown in Section 4.4.3.

4.4.1 Benchmark Methods

Global ranking method (GRM) (Ricci et al., 2011): All the objects in the dataset are sorted in the descending order based on their degree firstly. Then, the objects with the largest degree that are not collected by the target user are recommended. The number of proposed objects depends on the length of recommendation list L.

User-based collaborative filtering (UCF) (Liu et al., 2009): For a target user, collaborative filtering mainly focuses on recommending objects from users who have the similar taste with the target user. Generally, the target user prefers to accept the shared opinions of the most similar users. UCF takes advantage of the cosine similarity to evaluate the preference and taste of each user and then quantify which object the target user will collect.

Mass diffusion (MD) (Zhou et al., 2007): MD is an alternative name of network-based inference (NBI), and is a classical resource-allocation process on user-object bipartite networks. In MD, users and objects distribute their resources based on their own degree.

Similarity-preferential mass diffusion (SPMD) (Zeng et al., 2014): SPMD improves mass diffusion by introducing a parameter that is used to enhance or suppress the weight of users who are most similar to the target user.

Heat conduction (HC) (Zhang et al., 2007): HC is another resource-allocation process on bipartite networks. Actually, HC is a physical phenomenon applied in recommender systems in which each node allocates its resource in accordance with the degree of its adjacent nodes. Some previous studies has demonstrated that HC has an outstanding performance in pushing small-degree objects out.

Biased heat conduction (BHC) (Liu et al., 2011): BHC is an improvement of the HC method. It considers the degree effects in the last step of the local heat conduction process, which can greatly enhance the accuracy of the standard HC algorithm.

CosRA-based method (CosRA) (Chen et al., 2017b): Both cosine index and resource-allocation index are integrated into the CosRA-based method, which avoids a strong bias on the degree of objects.

Diffusion-based similarity (DBS) (Shang et al., 2010): DBS combines a user-object network and a user-tag network into tripartite networks, which brings tag information of users to the recommendation process. In DBS, the original NBI process is applied independently in the user-object network and the user-tag network firstly, and then the final resource of users on these two networks are combined linearly.

4.4.2 The Impact of Parameter λ

Our method uses the parameter λ to integrate the resource from objects and the explicit trust network linearly. To determine the optimal value of λ in our method, we adjust the parameter in experiments on different datasets. The F1 metric provides a comprehensive evaluation of precision and recall, which indicates the accuracy of a recommendation algorithm. Hence, using the F1 metric is a fair way to determine the optimal λ for DBRT.

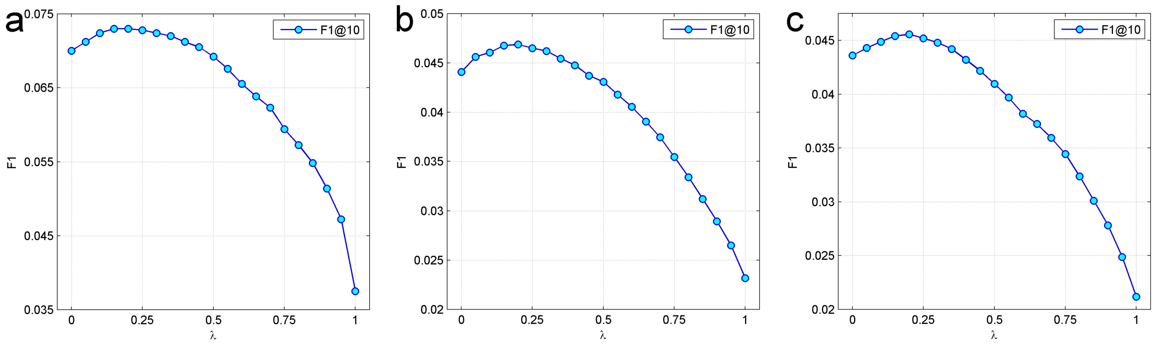


Figure 4.2: The F1 results of DBRT changing the parameter of λ between 0 and 1 at calculation step 0.05 in three datasets when the recommendation list is 10. (a) The optimal value is $\lambda = 0.15$ in the Ciao dataset. (b) The optimal value is $\lambda = 0.2$ in the Epinions dataset. (c) The optimal value is $\lambda = 0.2$ in the Flixster dataset.

Figure 4.2 reports the results of the F1 metric for our method when the parameter λ is changed from 0 to 1 at calculation step of 0.05. The results of the Ciao, Epinions and Flixster datasets are presented in Figure 4.2(a), (b) and (c), respectively. Because users often pay more attention to the top objects in a recommendation list, the most effective length of a recommendation list is $L = 10$ (Pan et al., 2016b). Therefore, the optimal value of λ for the top-10 recommendation can be employed in DBRT. In Figure 2(b) and (c), the optimal λ is 0.2 in the Epinions and Flixster datasets, and $\lambda = 0.15$ is the optimal value in the Ciao dataset in Figure 4.2(a). Although the sparsity of rating links and explicit trust links is totally different, the parameter has an optimal value around 0.2, which means the parameter is not sensitive to the number of users, items and trust links. If $\lambda = 0$, this means that our method only uses the implicit trust calculated by rating data to make recommendations on bipartite networks. As the value of λ increases, the impact of explicit trust becomes larger. The optimal λ in the three datasets indicates the explicit trust actually improves the performance of DBRT. If λ is very large, it leads to a biased evaluation, because the user-object network is the main domain in assessing user preference, and the explicit trust network is regarded as auxiliary data used to improve recommendations. The parameter λ is tested in three

4.4.3 Performance of Recommendations

We use three real-world online rating datasets with trust relations to evaluate our method. In the Ciao dataset, there are fewer trust links than rating links. By contrast, the trust links are more plentiful in the Epinions and Flixster. These different kinds of datasets reflect various social conditions of users that can verify our method in different situations.

Table 4.2: Results of the seven evaluation metrics after applying our method and the eight benchmark methods on the Ciao, Epinions and Flixster datasets. The length of the recommendation list is $L = 50$.

Ciao	$P(50)$	$R(50)$	$F1(50)$	$RBP(50)$	$ARHR$	$H(50)$	$N(50)$
GRM	0.0157	0.1525	0.0285	0.0526	0.1530	0.0926	167
UCF	0.0220	0.2139	0.0399	0.0753	0.2154	0.5396	138
MD	0.0228	0.2161	0.0412	0.0778	0.2227	0.6827	120
SPMD	0.0228	0.2171	0.0413	0.0782	0.2231	0.6926	119
HC	0.0067	0.0556	0.0119	0.0078	0.0318	**0.9467**	**8**
BHC	0.0233	0.2200	0.0421	0.0825	0.2343	0.7792	84
CosRA	0.0243	0.2219	0.0437	0.0817	0.2358	0.8786	74
DBS	0.0229	0.2187	0.0415	0.0778	0.2233	0.6523	125
DBRT	**0.0252**	**0.2346**	**0.0455**	**0.0874**	**0.2501**	0.8419	82
Epinions	$P(50)$	$R(50)$	$F1(50)$	$RBP(50)$	$ARHR$	$H(50)$	$N(50)$
GRM	0.0081	0.1162	0.0151	0.0226	0.0670	0.0505	152
UCF	0.0136	0.1863	0.0254	0.0358	0.1113	0.6868	112
MD	0.0144	0.1897	0.0268	0.0361	0.1141	0.8070	83
SPMD	0.0145	0.1901	0.0269	0.0365	0.1145	0.8170	80
HC	0.0051	0.0649	0.0095	0.0058	0.0251	**0.9235**	**10**
BHC	0.0149	0.1921	0.0277	0.0408	0.1256	0.8431	73
CosRA	0.0151	0.1839	0.0279	0.0388	0.1211	0.9039	51
DBS	0.0144	0.1923	0.0269	0.0366	0.1153	0.7739	100
DBRT	**0.0163**	**0.2088**	**0.0302**	**0.0503**	**0.1471**	0.8703	64
Flixster	$P(50)$	$R(50)$	$F1(50)$	$RBP(50)$	$ARHR$	$H(50)$	$N(50)$
GRM	0.0076	0.0837	0.0139	0.0190	0.0588	0.0477	174
UCF	0.0138	0.1544	0.0253	0.0353	0.1098	0.6912	129
MD	0.0148	0.1581	0.0271	0.0353	0.1134	0.8302	95
SPMD	0.0150	0.1587	0.0274	0.0355	0.1138	0.8240	96
HC	0.0048	0.0501	0.0087	0.0058	0.0242	**0.9511**	**8**
BHC	0.0153	0.1587	0.0279	0.0408	0.1264	0.8901	75
CosRA	0.0154	0.1537	0.0280	0.0339	0.1128	0.9408	50
DBS	0.0148	0.1600	0.0271	0.0353	0.1137	0.8019	109
DBRT	**0.0169**	**0.1800**	**0.0309**	**0.0474**	**0.1441**	0.9001	71

We compare our method with eight benchmark methods using the optimal value of parameter for SPMD, BHC and DBS on all three datasets.

The typical results of all methods on the Ciao, Epinions and Flixster datasets using a recommendation list $L = 50$ are presented in Table 4.2. Accuracy is indicated by precision, recall, F1, RBP and ARHR and the higher value in these five metrics, the better the performance. DBRT achieves the highest value for all accuracy metrics across all datasets. Specifically, compared to the DBS method, which is similar to our method and also takes advantage of the additional user's related feature to improve recommendations on tripartite networks. Our method enhances precision by 10%, 13.2%, and 14.2% for Ciao, Epinions and Flixster, respectively, when $L = 50$. This means our method provides a more reasonable and effective way to integrate the additional feature into the diffusion-based recommendation approach that increases the accuracy of recommendations. Moreover, compared to CosRA and BHC, our method also achieves much better results on all the accuracy metrics.

Figure 4.3 reports the F1 of all nine algorithms with a recommendation list ranging from 1 to 100. DBRT outperforms the comparison methods. ARHR and RBP are two metrics used to assess algorithms based on an object's position in a recommendation list. The higher value in these two metrics, the more likely the recommendation is to be collected by users. RBP assumes users browse the next object from the first place in a recommendation list with a fixed probability $p = 0.5$. The results

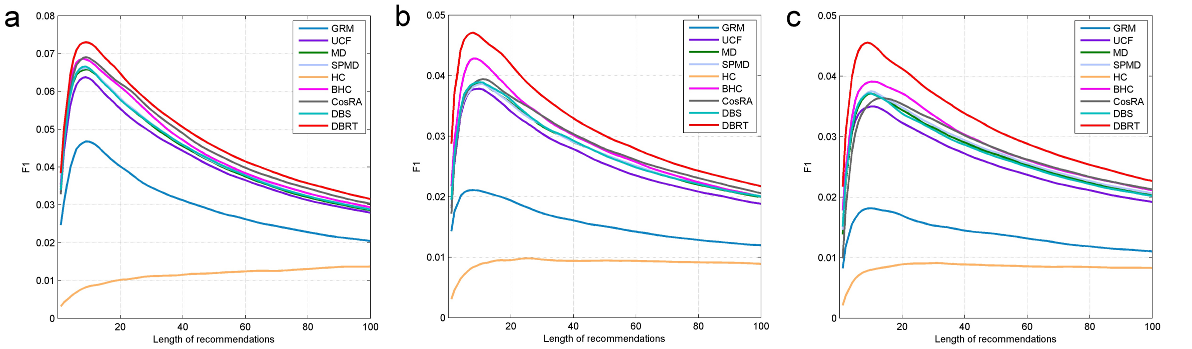


Figure 4.3: The F1 of our method and the eight benchmark methods for the Ciao, Epinions and Flixster are represented in the diagrams (a), (b) and (c), respectively. The length of recommendation list is from 1 to 100.

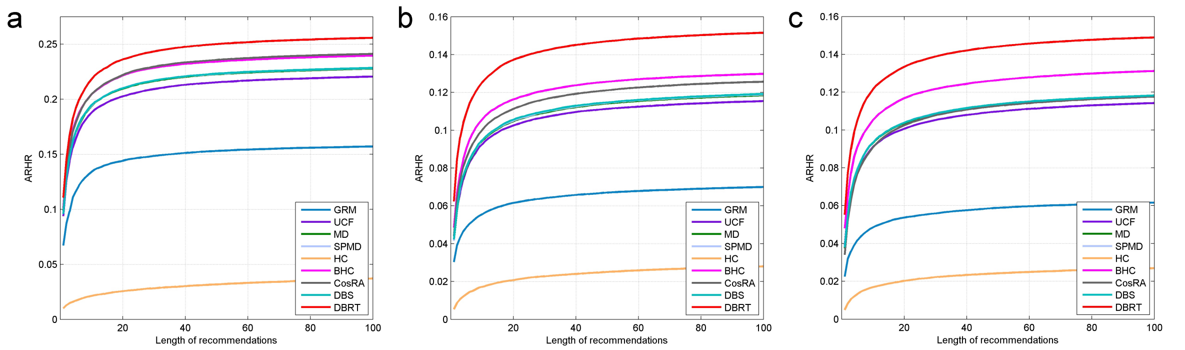


Figure 4.4: The ARHR of our method and the eight benchmark methods for the Ciao, Epinions and Flixster represented in the diagrams (a), (b) and (c), respectively. The length of recommendation list is from 1 to 100.

demonstrate some significant improvements for RBP provided by our method. Similar to RBP, ARHR is also a sequence-based metric, which means the higher the value in this metric, the more rational and efficient the rank of recommendations.

Figure 4.4 reports that DBRT has some remarkable advantages in ARHR when the length of the recommendation list ranged from 1 to 100. Note that although CosRA has a good performance in precision, its results are worse than BHC in RBP and ARHR, which means CosRA lacks the ability to put the user preferred objects at the top of a recommendation list. Conversely, our method simultaneously improves the performance of precision, RBP and ARHR.

The diversity of recommendations is evaluated by the hamming distance, and novelty represents the ability to push out small-degree objects. HC achieves the best performance in both diversity and novelty, which indicates it focuses on recommending small-degree objects. The diversity of DBRT is close to the CosRA method and shows overwhelming advantages over the other benchmark methods. Particularly for the DBS method, DBRT improves the hamming distance by 29.1%, 12.5% and 12.2% in Ciao, Epinions and Flixster, respectively, when $L = 50$. This indicates our method improves the way auxiliary features are combined into the diffusion-based recommendation. When resource is diffused on tripartite networks, the node that has a larger degree plays a more important role. In our method, large-degree users can transfer more resource from the explicit trust network to objects, which means the objects linked with the large-degree users can obtain more final resource and will be placed at the top of a recommendation list. Large-degree users' choices have a greater influence and are more likely to be accepted by other users (Liu et al., 2011), so it is easy to understand why DBRT achieves better performance in accuracy. Paying more attention to large-degree users and large-degree objects will decrease the diversity and novelty, which explains why DBRT does not show better results over HC or CosRA in diversity and novelty.

4.5 Summary

In this chapter, users' social trust relations are divided into two parts: implicit trust relations and explicit trust relations, which have different effects on improving recommendations. A novel combination approach is proposed, called diffusion-based recommendation method with trust relations (DBRT), that integrates trust information into the diffusion-based recommendation on tripartite networks. Experiments on three real-world datasets show our method performs better than most benchmark methods. Specifically, DBRT provides many remarkable improvements in terms of accuracy, diversity and novelty over the DBS, which also uses tripartite networks to combine additional features to make better recommendations. Large-degree objects and users play very important roles in recommender systems. Large-degree objects are preferred by many users, and users with larger degrees may have a greater influence on user-object networks. Therefore, our method pays more attention to large-degree users, which leads to outstanding improvements in accuracy.

CHAPTER 5

Social Recommendation with Evolutionary Opinion Dynamics

5.1 Introduction

An enormous growth in the amount of data presents a significant challenge in terms of finding useful information (Xuan et al., 2016). Recommender systems (Lu et al., 2015) have attracted a lot of attention as a tool for information filtering and have been used in many aspects of people's life. With the development of social networks, social recommendation which is a branch of recommender systems becomes important and has widely been applied in many social websites. Users in online social networks often interact with others. They may observe the actions of other users, and comment on these users. Therefore, they make connections with other users. The connections contain both physical links and virtual trust relations (Wang et al., 2017e). Users' social relations have been combined with recommender systems (Eirinaki et al., 2014). Users who have connections with each other are assumed to have similar tastes. Current studies consider that user ratings are not only determined by their own opinions, but they are also influenced by the tastes of their friends (Wang et al., 2016c). Therefore, opinions of neighbors are incorporated into the product of latent vectors. In addition, social regularization is used to minimize the difference between latent vectors of each user and neighbors (Ma et al., 2011). The network of user relations has also been mapped into latent factor spaces, which explicitly describes feedback on how users affect or follow the opinions of others. Recommender systems often have a better performance than traditional recommendation algorithms due to their inclusion of social or trust information (Mao et al., 2017).

Indeed, when users in social networks make a decision on ratings or reviews, opinions and behaviors about items will be directly or indirectly affected by others (Lu et al., 2016; Ok et al., 2016; Hu et al., 2017). Current studies often model the impact of friends by a linear combination of others' latent vectors (Guo et al., 2017b). The combination of vectors may affect the ratings or social connections among users, including both explicit and implicit influence. However, it is still unclear whether the evolution of a user's opinion follows the superposition of others when it interacts with friends. The evolutionary pattern of opinions has been widely investigated in the research field of social physics and statistical learning (Jiang et al., 2014a; Castro et al., 2017a; Jiang et al., 2014b). In the Deffuant-Weisbuch (DW) opinion model (Deffuant et al., 2000), when two users discuss a topic, their opinions change and become closer to each other. In the model with continuous opinions and discrete actions (Martins, 2008), users change their opinions under the Bayesian rule of how likely their neighbors are to be correct, after they observe the external actions (ratings or reviews) of their neighbors. This model promotes the appearance of extreme opinions and forces opinions to cluster together. In the work by Cao and Li (2008), users update their opinions according to the birth-death and death-birth process during interactions. These models often originate from real physical phenomena, and have been verified in the interactions of real society. Further exploration is needed to determine whether these opinion models can be applied to characterize real opinion interactions in online social networks.

In this chapter, we propose a recommendation model that includes opinion interactions and user influence. Evolutionary opinion dynamics are introduced to recommender systems. We characterize the impact of neighbors on user opinions by evolutionary game theory. We define the strategies during an interaction of two users, i.e., changing or keeping their opinions, and give the payoff for each strategy. Users choose a better strategy to maximize their payoffs when they discuss an item with another user. Opinion interactions are conducted with the matrix factorization

(MF), and therefore, user ratings are affected by the opinions of others. In addition, user influence which measures the status of a node in the network, is added to the recommendation model, so that the ratings of each user are weighted. We conduct experiments on two real-world datasets, and the results demonstrate that our method works better than state-of-the-art recommendation models. Furthermore, our method has much less computational complexity than its counterparts. Our work reveals that studies in other research fields, such as social physics and statistics, can be incorporated in recommender systems, to improve the recommendation performance. This work makes the following contributions.

1) A Rcommendation method with Evolutionary Opinion Dynamics (REOD) is proposed to introduce the evolutionary game theory into recommendation tasks. The payoffs of strategies during an interaction are associated with latent item factors and observed ratings. Users update their opinions to reduce rating errors and the distances between their opinions. This model considers both the dynamic process in real society and the rating prediction of recommender systems.

2) We introduce opinion dynamics and user influence to the MF framework, and improve the recommendation. During the training of MF, users update their opinions according to the payoff matrix of the game. When users make decisions on items, they are affected by others, so the opinions of others contribute to the ratings. In addition, user influence that originates from the trust network is added to the recommendation. The

method which combines MF and random dynamics is general.

3) We conduct extensive experiments to evaluate the effectiveness of our method for all users and cold-start users. We compare our method with several state-of-the-art recommendation models, and analyze the computational complexity of the proposed method. Results show that our method outperforms its counterparts and encourages users to reduce the divergence of their opinions, in accordance with real dynamics.

5.2 Regularized Matrix Factorization

MF is an effective approach for recommender systems to predict missing ratings. This method assumes that user decisions are determined by a few latent factors, and a rating is estimated according to how an item meets a user's preference toward the latent factors.

We define the set of users as $\{u_1, u_2, ..., u_m\}$, and the set of items as $\{v_1, v_2, ..., v_n\}$. m denotes the number of users, and n denotes the number of items. The ratings are given by a matrix $\boldsymbol{R} \in \mathbb{R}^{m\times n}$. MF decomposes the $m \times n$ rating matrix into two low-rank matrices $\boldsymbol{U} \in \mathbb{R}^{m\times d}$ and $\boldsymbol{V} \in \mathbb{R}^{n\times d}$, obviously $d < min(m,n)$. The rating matrix is expressed by $\boldsymbol{R} = \boldsymbol{U}\boldsymbol{V}^T$, meaning that the target matrix $\boldsymbol{R}$ can be approximated by the product of two low-rank matrices. For an accurate description, we rewrite the approximation process as

$$\boldsymbol{R} = \boldsymbol{U}\boldsymbol{V}^T + \boldsymbol{e} \tag{5.1}$$

where $\boldsymbol{e}$ is the error matrix. One can find suitable $\boldsymbol{U}$ and $\boldsymbol{V}$ to make the error as small as possible. Thus, we approximate the rating matrix by minimizing

$$\mathcal{L} = \frac{1}{2} \left\| \boldsymbol{R} - \boldsymbol{U}\boldsymbol{V}^T \right\|^2 \tag{5.2}$$

where $\|\cdot\|$ denotes the Frobenius norm. $\boldsymbol{U}$ and $\boldsymbol{V}$ are obtained from the observed ratings, and they can be utilized to predict the missing ratings. Considering the observed ratings, Eq. (5.2) is changed to

$$\min_{U,V} \mathcal{L} = \min_{U,V} \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij} \left(R_{ij} - U_i V_j^T \right)^2 \tag{5.3}$$

where I is a binary function standing for whether user i has rated item j. To avoid over fitting, quadratic regularization terms are added to the sum-of-squared-errors objective function as

$$\min_{U,V} \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} I_{ij} \left(R_{ij} - U_i V_j^T \right)^2 + \frac{\lambda}{2} \|\boldsymbol{U}\|^2 + \frac{\lambda}{2} \|\boldsymbol{V}\|^2 \tag{5.4}$$

where λ is the extent of regularization, and $\lambda > 0$. Stochastic gradient descent (SGD) is applied to optimize the objective function and find a local minimum. In each iteration of training, all the observed ratings are estimated by latent vectors, and the corresponding vectors are updated as follows

$$
\begin{aligned}
&\frac{\partial \mathcal{L}}{\partial U_i} = -\sum_j (R_{ij} - U_i V_j^T) V_j + \lambda U_i \\
&\frac{\partial \mathcal{L}}{\partial V_j} = -\sum_i (R_{ij} - U_i V_j^T) U_i + \lambda V_j \\
&U_i \leftarrow U_i - \gamma \frac{\partial \mathcal{L}}{\partial U_i} \\
&V_j \leftarrow V_j - \gamma \frac{\partial \mathcal{L}}{\partial V_j}
\end{aligned}
\tag{5.5}
$$

where γ denotes the learning rate. MF is one of the most popular methods in model-based collaborative filtering.

5.3 The Proposed Method

In this section, we introduce our recommendation method in detail. We define the influence of each user, and use it to weight the objective function of MF. Opinion interactions are characterized by evolutionary game theory, and they are incorporated into the SGD training of MF. In the following, we first introduce the game theory model of opinion dynamics. Then, we describe the method of MF with user influence. Last, we detail the whole training algorithm and analyze the computational complexity of the proposed method.

5.3.1 Game Theory Model of Opinion Evolution

User i's latent vector U_i is treated as user i's opinion toward latent factors, revealing how each factor applies to the user. Opinions of users do not always remain unchanged. Users try to persuade others to adopt their opinions, and therefore, opinions are dynamic. Users on online social networks may interact with other users, and exchange opinions. On product review websites, when a user publishes a rating or comment on an item, some other users may read the comment and discuss the item with the user. After the interaction, they may change their opinions. Therefore, opinions evolve during the dynamics.

Many opinion models were proposed to characterize the process of opinion interactions (Li et al., 2013; Jamali and Ester, 2010). As a typical representative of continuous opinion models, the DW model describes pairwise interactions between users who have similar opinions. In each update event, two agents i and f are selected at random, and they start a conversation. Meanwhile, the assumption of bound confidence is introduced to the opinion model. When the opinions of these two agents are close enough, they will change their opinions. Therefore, if the opinions of user i and f satisfy $\|U_i - U_f\| < \varepsilon$ ($\varepsilon > 0$), each opinion moves in the direction of the other as

$$U_i \leftarrow U_i + \mu \cdot \left(U_f - U_i\right) \tag{5.6}$$

$$U_f \leftarrow U_f + \mu \cdot \left(U_i - U_f\right) \tag{5.7}$$

where μ ($0 < \mu \leq 0.5$) is the trust parameter of users, and ε is the tolerance threshold. In the DW model, ε and μ are constants during the evolution. For a special case in which opinions have only one dimension, i.e., $d = 1$

, if $\varepsilon > 0.5$, all opinions converge to a single central one, and the system reaches consensus. If $\varepsilon < 0.5$, the system reaches a state of fragmentation, in which a final number of opinion clusters occur, scaling with the number of users. The number of clusters is in proportion to $1/\varepsilon$.

The DW model characterizes user interaction behaviors, but the impact of item factors and observed ratings are ignored during the evolution. In addition, the tolerance threshold ε is fixed for each user, however, users in real society often have different thresholds. Now, we use evolutionary game theory to model the process of user interactions with item and rating information. Game theory investigates the process of decision making when two players struggle to maximize their own payoffs. Meanwhile, game theory can also be used to explore user behaviors in opinion dynamics.

We present the opinion dynamic model through the framework of evolutionary game theory as follows. In each interaction, two users i and f are selected at random, and are regarded as players in a game. An item j is randomly selected, and is treated as a topic. Users generally try to persuade others or reach agreement on the topic. The interaction strategies available to each player are either to change their opinions or maintain their opinions. The payoffs that the players receive depend on the strategies they implement in the game. A strategy with a higher payoff is preferred by players (Luo et al., 2016). In real interactions, each player wants to convince the other one that its opinion is correct. Meanwhile, each player tends to adopt the strategy that can decrease errors of estimated ratings. Therefore, user opinions and ratings should be included in payoffs. Assume that in an interaction, user i changes its opinion U_i, and then its opinion will be updated to $U_{i,new}$ following Eq. (5.6). Considering the observed rating R_{ij} and item j's latent vector, the payoffs for the strategies are defined as follows.

1) If user i changes its opinion, the payoff that user i obtains is $\left|R_{ij} - U_i V_j^T\right| - \left|R_{ij} - U_{i,new} V_j^T\right|$. Users should adapt their opinions to reduce the errors between the observed ratings and estimated ratings. Therefore, if the error for the estimated rating decreases after the opinion update, user i will obtain a positive payoff and it is willing to change its opinion. We suppose that the payoff for the strategy, i.e., the user changing its opinion, depends on the difference between the original error and that after this strategy is adopted.

2) If user i retains its opinion, the payoff for i is $\beta \cdot \left|U_i V_j^T - U_f V_j^T\right|$ where β ($\beta > 0$) is used to control the contribution of this strategy which represents individual stubbornness. Users generally prefer to persuade their opponents rather than changing their own opinions, since changing an opinion may incur a cost. The payoff correlates with the difference between user i's and f's estimated ratings on item j, and a large difference between ratings leads to a large cost when changing opinions. If users decide to maintain their opinions, they will receive a positive payoff.

3) If user f changes its opinion, user i receives the payoff $\beta \cdot \left|U_i V_j^T - U_f V_j^T\right|$.

If a user succeeds in persuading its opponent to change an opinion, it will obtain a positive payoff.

When user i and f interact in relation to topic j, the payoffs for user i are shown in Table 5.1. The Nash equilibrium point of the aforementioned game is related to latent item vector V_j and rating R_{ij}. We can infer the Nash equilibrium point from Table 5.1 as follows.

1) When $\left|R_{ij} - U_i V_j^T\right| - \left|R_{ij} - U_{i,new} V_j^T\right| - \beta \cdot \left|U_i V_j^T - U_f V_j^T\right| > 0$, the Nash equilibrium strategy for user i is changing its opinion.

2) When $\left|R_{ij} - U_i V_j^T\right| - \left|R_{ij} - U_{i,new} V_j^T\right| - \beta \cdot \left|U_i V_j^T - U_f V_j^T\right| \leq 0$, the Nash equilibrium strategy for user i is maintaining its opinion.

The analogous Nash equilibrium strategy can be found for user f. For the aforementioned evolutionary game model, the condition for opinion updates varies with time. The model does not have a fixed tolerance threshold ε. Inserting Eq. (5.6) into the Nash equilibrium condition, we have

$$\begin{aligned} \left|R_{ij} - U_i V_j^T\right| - \left|R_{ij} - U_i V_j^T - \mu \cdot (U_f - U_i) V_j^T\right| \\ - \beta \cdot \left|U_i V_j^T - U_f V_j^T\right| > 0 \end{aligned} \tag{5.8}$$

From Eq. (5.8), if user i changes its opinion in an interaction, it holds true that $U_i V_j^T < R_{ij}$ & $U_f V_j^T > U_i V_j^T$, or $U_i V_j^T > R_{ij}$ & $U_f V_j^T < U_i V_j^T$. In addition, if we do not consider the impact of observed ratings, so the condition for opinion updates in Eq. (5.8) reduces to $\left|U_i V_j^T - U_f V_j^T\right| < \varepsilon$. As in the work by Xiong et al. (2017), when the system in homogeneous networks converges, the initial average value of $\left|U_i V_j^T - U_f V_j^T\right|$ over all

Table 5.1: Payoffs for user i.

	User f changes its opinion	User f maintains its opinion
User i changes its opinion	$\left\|R_{ij}-U_iV_j^T\right\|-\left\|R_{ij}-U_{i,new}V_j^T\right\|+\beta\cdot\left\|U_iV_j^T-U_fV_j^T\right\|$	$\left\|R_{ij}-U_iV_j^T\right\|-\left\|R_{ij}-U_{i,new}V_j^T\right\|$
User i maintains its opinion	$2\beta\cdot\left\|U_iV_j^T-U_fV_j^T\right\|$	$\beta\cdot\left\|U_iV_j^T-U_fV_j^T\right\|$

users should be below ε. U_i and V_j are d-dimensional vectors, and each dimension in the beginning is randomly distributed from $[0,1]$. The expectation of initial U_iV_j is $d/4$. It can be inferred that the expectation of initial $\left|U_iV_j^T - U_fV_j^T\right|$ is $d/18$. A large number of latent factors d leads to a large divergence of opinions and prevents the system from converging.

In each iteration of SGD during the training process, we implement opinion interactions of users following the evolutionary game model. In multi-agent opinion dynamics, a Monte-Carlo time step contains m times of opinion interactions for a population of m users, and hence we introduce m such interactions into an iteration of SGD. In an opinion interaction, two users are selected at random, and they employ different strategies according to their payoffs in relation to a randomly selected item. In an iteration of SGD, opinion interactions are asynchronously implemented m times. When the objective function reaches convergence, for a majority of user-item pairs, the product of latent vectors $U_iV_j^T$ approaches R_{ij}, so that in the stable state, $|R_{ij} - U_iV_j{}^T|$ for these user-item pairs is small. Therefore, for most of users, the payoff of changing their opinions $|R_{ij} - U_iV_j{}^T| - |R_{ij} - U_{i,new}V_j{}^T|$ is often smaller than that of maintaining their opinions $\beta \cdot |U_iV_j{}^T - U_fV_j{}^T|$. In the stable state of the network, the strategy of maintaining one's opinion dominates in opinion interactions.

5.3.2 Matrix Factorization with User Influence

User influence represents the role of a user in a network. This influence is often regarded as a contribution in the process of information diffusion. With large influence, a user may diffuse its information to a greater number of other users, and information recommended by this user is readily accepted by neighbors. Therefore, it has a large impact on others' preferences. Some features of the underlying topology can be used to measure influence, such as the degree, betweenness, k-core index, average clustering coefficient and so on (Xiong and Li, 2017). Here, for the sake of simplicity, we choose node degree as the indicator of user influence.

In the real world, users generally consult their friends before making decisions on items, since they tend to trust the preferences of their friends. From trust relations found on movie or product review websites, a trust network can be obtained and then user influence can be calculated. We define the number of users that trust user i as F_i^-. F_i^- is quite heterogeneous for different users, and therefore, we should renormalize it. User i's influence is given by

$$\varphi_i = \frac{\log\left(F_i^-/\alpha_1 + \alpha_2\right)}{\log\left(\max_f F_f^-/\alpha_1 + \alpha_2\right)} \tag{5.9}$$

The offset α_2 in the logarithmic function increases user influence to greater than 0, as some users do not have any trust relations. The value of α_2 should be set in the interval $(2, 10)$, since too large α_2 reduces the effect of user influence. The denominator in Eq. (5.9) renormalizes the influence and limits the value of φ_i in the interval $(0,1]$. The parameter α_1 is used to control the decay of user influence. If a user has a larger influence, its preference makes a greater contribution in the sum-of-squared-errors objective function. The objective function of Eq. (5.4) is rectified as

$$\min_{U,V} \frac{1}{2}\sum_{i=1}^{m}\varphi_i\sum_{j=1}^{n}I_{ij}\left(R_{ij}-U_iV_j^T\right)^2+\frac{\lambda}{2}\|\boldsymbol{U}\|^2+\frac{\lambda}{2}\|\boldsymbol{V}\|^2 \tag{5.10}$$

Then, the derivatives of the corresponding latent vectors in SGD are calculated as follows.

$$\frac{\partial\mathcal{L}}{\partial U_i}=-\sum_j\varphi_i\left(R_{ij}-U_iV_j^T\right)V_j+\lambda U_i \tag{5.11}$$

$$\frac{\partial\mathcal{L}}{\partial V_j}=-\sum_i\varphi_i\left(R_{ij}-U_iV_j^T\right)U_i+\lambda V_j \tag{5.12}$$

5.3.3 Model Learning

Here, we present our recommendation method with opinion interactions and user influence. The whole training algorithm is shown in Algorithm 5.1. The method is based on the framework of MF, and opinion dynamics are introduced to the process of SGD. Our method comprises two steps in an iteration of SGD.

1) For each observed rating, SGD is used to update latent user vector U_i and item vector V_j. User influence given in Eq. (5.9) from the trust network is included.

2) Opinion interactions are implemented in each iteration of SGD. In each interaction, two users i and f are selected at random. User i

Algorithm 5.1 The proposed recommendation method with evolutionary opinion dynamics.

Require: List of tuples $\Omega = (users, items, ratings)$, list of tuples $SNS = (users, trusted\ users)$, the number of latent factors d, the learning rate γ, regularization parameter λ, user influence parameter α_1, α_2, trust parameter μ, and payoff parameter β

Ensure: Latent user matrix $\boldsymbol{U}$ and latent item matrix $\boldsymbol{V}$

Task 1: Generating user influence
for $i \leftarrow 1, 2, \ldots, m$ **do**
  Calculate φ_i according to Eq. (5.9)
end for

Task 2: Learning user matrix $\boldsymbol{U}$ and item matrix $\boldsymbol{V}$
Initialize $\boldsymbol{U}$ and $\boldsymbol{V}$ randomly
while not convergence **do**
  1) SGD training
  Calculate $\partial\mathcal{L}/\partial\boldsymbol{U}$ according to Eq. (5.11)
  Calculate $\partial\mathcal{L}/\partial\boldsymbol{V}$ according to Eq. (5.12)
  Update $\boldsymbol{U} \leftarrow \boldsymbol{U} - \gamma \cdot \partial\mathcal{L}/\partial\boldsymbol{U}$
  Update $\boldsymbol{V} \leftarrow \boldsymbol{V} - \gamma \cdot \partial\mathcal{L}/\partial\boldsymbol{V}$
  2) Opinion interactions
  for $i \leftarrow 1, 2, \ldots, m$ **do**
    Select two users i and f at random
    Select an item j randomly that user i has rated
    if Eq. (5.8) holds **then**
      $U_i \leftarrow U_i + \mu \cdot (U_f - U_i)$
    end if
  end for
end while

randomly selects an item j that i has rated in the training data. Then, user i interacts with f for item j according to the Nash equilibrium of the game shown in Table 5.1. If the condition of Eq. (5.8) holds true, user i changes its opinion following the first equation of Eq. (5.6). In each iteration, m interactions are implemented. Here, we do not consider the trust network, since a user on online social networks can exchange its opinion with any other users even if it does not have any trust relation with them. Users' comments and ratings are accessible to all other users, so that they can have a discussion on the item.

5.3.4 Complexity Analysis

We analyse the computational complexity for the proposed method. We define the number of observed ratings in the training data as $|R|$, and the number of iterations as N. The computational complexity of SGD in MF is $O\big(d \cdot N \cdot |R|\big)$, where d is the number of latent factors. As previously mentioned, m is the number of users. Thus, the average number of observed ratings for each user is $|R|/m$. To calculate user similarities, the computational complexity $O\big(m^2 \cdot |R|/m\big) = O\big(m \cdot |R|\big)$ is required. The computational complexity for SoReg is $O\big(d \cdot N \cdot \big(|R| + 2m \cdot |f|\big) + m \cdot |R|\big)$, where $|f|$ is the average number of trusted friends for each user. Since m is often much larger than $d \cdot N$, the computation of user similarities in SoReg accounts for a greater proportion than that of SGD for MF. For TrustSVD (Guo et al., 2016a), the computational complexity is $O\big(d \cdot N \cdot \big(|R| + |T|\big) \cdot \max\big(|f|, k\big)\big)$, where $|T|$ is the number of observed relations and k is the average number of ratings received by an item.

For REOD, the computation is mainly caused by SGD training and opinion dynamics. In the process of opinion dynamics during an iteration, m opinion interactions are implemented, each of which only contains one opinion update. An opinion update takes the computational complexity $O(d)$. Therefore, opinion dynamics results in the computational complexity $O(d \cdot N \cdot m)$. Overall, the computational complexity for REOD is $O(d \cdot N \cdot (|R| + m))$. Since $|R|$ is much larger than m, the complexity of our method approximates MF which costs $O(d \cdot N \cdot |R|)$, and REOD involves much less computation than state-of-the-art models.

5.4 Experiments

In this section, we address the following questions: 1) Does the proposed method with evolutionary opinion dynamics and user influence improves the accuracy of recommendation? 2) What is the contribution of opinion interactions and user influence for recommendation? 3) How does the intrinsic parameters of opinion interactions and user influence affect the recommendation results? First, we use two real-world datasets to evaluate our method, and compare the recommendation results of our approach with other state-of-the-art recommendation models to answer the first question. Then, we investigate the effects of the components in our method to answer the second question. Last, we vary the parameters of opinion interactions to explore their effects to answer the third question.

5.4.1 Datasets and Metrics

Table 5.2: Statistics of datasets.

	Ciao	Epinions
#User	7267	7411
#Item	11,211	8728
#Rating	149,147	276,116
#Social relation	110,755	52,982

To evaluate the proposed recommendation method, we collected two datasets, which were taken from the popular social networking websites Ciao and Epinions. Statistics on these datasets are presented in Table 5.2. Users of these social networking services can rate items, browse/write reviews, discuss with others, and add trusted friends. Therefore, we can obtain rating and social relation data from these websites.

Ciao and Epinions are well-known product review websites, where users can rate a product using one of five discrete ratings from 1-5. Ratings imply the sentiment of users towards a given item. If a user is not satisfied with a product, it will give the product a rating of 1; if a user appreciates a product, it will give the product a rating of 5. Each user maintains a 'trust' list which includes the user's social relations. For Ciao, we collected 7267 users, 11,211 items, and 149,147 ratings. The density of the user-item rating matrix is 0.183%. For Epinions, we collected 7411 users, 8728 items, and 276,116 ratings, and the density of the user-item rating matrix is about 0.427%.

In both datasets, F_i^- follows a power-law distribution. The power exponent in the Ciao dataset is -1.076 ± 0.023, and that in Epinions is -0.991 ± 0.021. The average and maximal values of F_i^- in Ciao are 15.2408 and 796, while those in Epinions are 7.1491 and 336, respectively.

For each dataset, we choose $x\%$ as the training set to learn the parameters and use the remaining $1-x\%$ as the test set. We set x at 60, 70 and 80, respectively, and obtain the results. The experiments are conducted five times independently, and we give the average performance.

We use two metrics to evaluate the performance, i.e., Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). MAE is defined as

$$MAE = \frac{1}{|R_{test}|} \sum\nolimits_{R_{ij} \in R_{test}} \left| R_{ij} - U_i V_j^T \right| \tag{5.13}$$

where R_{test} refers to the test set, and $|R_{test}|$ refers to the number of ratings in R_{test}. RMSE is defined as

$$RMSE = \sqrt{\frac{1}{|R_{test}|} \sum\nolimits_{R_{ij} \in R_{test}} \left(R_{ij} - U_i V_j^T \right)^2} \tag{5.14}$$

It has been proved that a smaller MAE or RMSE value means a better performance.

5.4.2 Baselines

In this section, to demonstrate the effectiveness of the proposed method, we compare it with the following representative recommendation models.

PMF (Mnih and Salakhutdinov, 2007): This method only utilizes the user-item rating matrix, and performs probabilistic matrix factorization to make recommendations.

LLORMA (Lee et al., 2016): this method relaxes the low-rank assumption, and approximates the observed matrix as a weighted sum of local low-rank matrices.

SocialMF (Jamali and Ester, 2010): This method introduces the mechanism of trust propagation into the model.

SoRec (Ma et al., 2008): This method is based on probabilistic matrix factorization, and performs a co-factorization on the user-term rating matrix and user-user social relation matrix.

RSTE (Ma et al., 2009): This method makes social recommendation by using social trust ensemble and naturally fusing the preferences of users and their trusted friends together.

TrustMF (Yang et al., 2017): This method performs matrix factorization to map users into low-dimensional latent spaces in terms of their trust relations.

SoReg (Ma et al., 2011): This method incorporates social regularization into matrix factorization, and social regularization represents the social constraints on recommender systems.

TrustSVD (Guo et al., 2016a): This method incorporates the explicit and implicit influence of rated items and trusted users on the prediction of items.

To focus on model evaluation and a fair comparison, for all methods, we set the same number of latent factors $d = 20$. For different parameters in baseline models, we employ cross-validation to determine the optimal values. For REOD, we set the payoff parameter $\beta = 0.05$, the trust parameter $\mu = 0.12$, the learning rate $\gamma = 0.001$, and the influence offset $\alpha_2 = 6$ in both datasets. The influence decay is $\alpha_1 = 30$ in Ciao, and it is $\alpha_1 = 80$ in Epinions. Degrees of the trust network have a heavy tailed distribution, and many users have a tiny F_i^- in Eq. (5.9). Therefore, the ratings of these users make little contribution to the objective function, so Eq. (5.10) may cause over fitting. To alleviate this problem, we multiply φ_i by a positive random number with normal distribution for each iteration.

5.4.3 Performance Comparisons

Tables 5.3 and Table 5.4 compare the results of the different methods for all users. More training data lead to higher recommendation accuracy, especially in Ciao which has sparser rating data. PMF performs worse than all social recommendation models except in the case where TrustMF performs the worst when 80% of data in Epinions are used for training. The reason for this is that the dataset of Epinions has much sparser user relations. Directly factorizing the matrix of the sparse trust network may harm the prediction accuracy on unknown ratings for recommender systems. LLORMA has low accuracy in Ciao, but it performs well in Epinions and even outperforms some social recommendation methods. LLORMA

Table 5.3: Results of recommendation on MAE and RMSE in Ciao.

		PMF	LLORMA	SocialMF	SoRec	RSTE	TrustMF	SoReg	TrustSVD	REOD
60%	MAE	0.9767	0.8592	0.7762	0.7908	0.7971	0.7883	0.7626	0.7515	**0.7359**
	RMSE	1.2401	1.2339	1.0036	1.1337	1.1097	1.0858	1.0116	**0.9844**	0.9858
70%	MAE	0.9078	0.8055	0.7702	0.7855	0.7897	0.7838	0.7539	0.7434	**0.7294**
	RMSE	1.1572	1.1251	0.9988	1.1179	1.0989	1.0685	1.0002	0.9768	**0.9766**
80%	MAE	0.8696	0.7795	0.7640	0.7809	0.7786	0.7792	0.7472	0.7376	**0.7262**
	RMSE	1.1130	1.0654	0.9919	1.1052	1.0859	1.0560	0.9899	0.9704	**0.9701**

Table 5.4: Results of recommendation on MAE and RMSE in Epinions.

		PMF	LLORMA	SocialMF	SoRec	RSTE	TrustMF	SoReg	TrustSVD	REOD
60%	MAE	0.8969	0.8271	0.8552	0.8574	0.8689	0.8788	0.8269	0.8096	**0.7974**
	RMSE	1.1334	1.1277	1.1231	1.1066	1.1705	1.1616	1.0721	1.0425	**1.0396**
70%	MAE	0.8759	0.8119	0.8510	0.8530	0.8611	0.8747	0.8230	0.8041	**0.7938**
	RMSE	1.1129	1.0998	1.1115	1.0947	1.1558	1.1438	1.0674	1.0379	**1.0331**
80%	MAE	0.8624	0.8041	0.8487	0.8493	0.8564	0.8729	0.8211	0.8004	**0.7910**
	RMSE	1.1004	1.0862	1.1004	1.0861	1.1475	1.1313	1.0647	1.0360	**1.0308**

obtains Low-rank matrices that are limited to a local region of the observed matrix, so that it achieves a high performance in denser rating data. In Epinions, SocialMF and SoRec almost perform the same, but when more user relations are available, SocialMF has higher accuracy in the Ciao dataset. RSTE uses social trust ensemble and requires more relation data, so that it performs worse than SoRec in Epinions. SoReg has a smaller MAE and RMSE than SocialMF, SoRec, RSTE and TrustMF, since SoReg uses better social regularization terms. TrustSVD incorporates the implicit influence of user trust and item ratings, so recommendation accuracy is improved and it performs the best of the state-of-the-art methods. Clearly, our method REOD outperforms the other models. When 60% of the training data of Ciao are used, REOD decreases MAE as high as 3.501% in contrast to SoReg, and 2.076% in contrast to TrustSVD; in Epinions, the corresponding improvement is 3.568% in contrast to SoReg, and 1.507% in contrast to TrustSVD. Although in Ciao, the RMSE of REOD approaches that of TrustSVD, REOD can obtain a better performance with sparse social connections. Therefore, we draw the conclusion that REOD improves the accuracy of recommendation.

Recommender systems often suffer from cold start problems, degrading the recommendation performance. We address the accuracy of these models for cold start users who have only rated a few items (equal to or less than five ratings). Figure 5.1 shows the performance of SoReg, TrustSVD and our approach for cold start users. The parameters are the

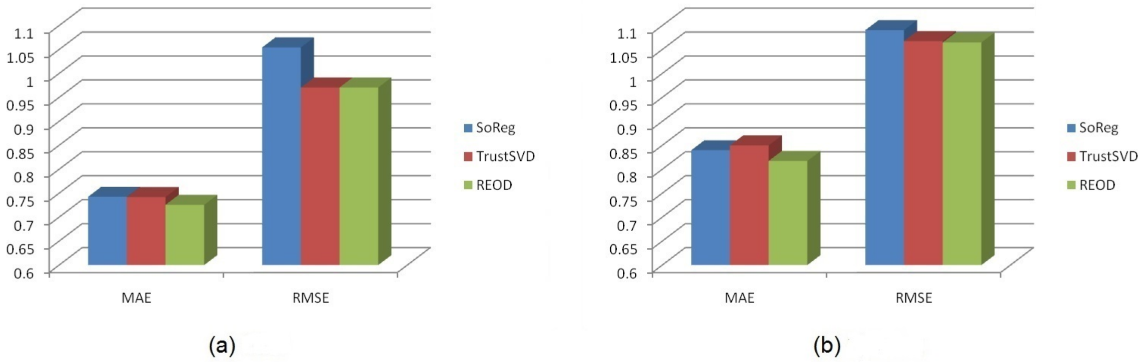


Figure 5.1: Performance comparison of SoReg, TrustSVD and REOD for cold-start users. (a) Ciao. (b) Epinions.

same as above. We select cold start users from the test data, and evaluate the MAE and RMSE on these users. Here, we use 80% of the data for the training data, and the results are similar for different proportions of training data. Figure 5.1 shows that REOD still outperforms the other methods for cold start users, although the RMSE of TrustSVD approximates our method. In both datasets, SoReg has a similar MAE with TrustSVD, but has a larger RMSE than the other two methods. The results demonstrate that incorporating evolutionary opinion dynamics can help recommender systems cope with cold start situations.

Now, we focus on the second issue of examining the effects of user influence and opinion interactions. It has been proven above that recommendation with both effects outperforms the representative recommendation models. We investigate which aspect plays a more significant role in social recommendation. We eliminate the effect of opinion interactions or user influence separately by defining the following algorithmic variants.

1) REOD\UI - Eliminating the effect of user influence. Evolutionary opinion dynamics are considered. The objective function of Eq. (5.10) reduces to that of traditional MF in Eq. (5.4).

2) REOD\OP - Eliminating the effect of opinion interactions. User influence is calculated from the trust network. Opinion interactions are removed from each iteration of SGD training.

3) REOD\UI&OP - Eliminating both the effects of opinion interactions and user influence.

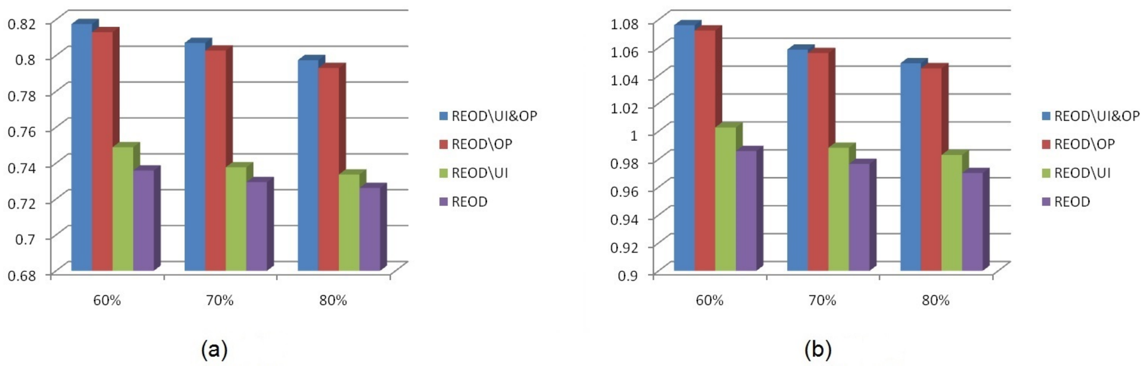


Figure 5.2: Impact of user influence and opinion interactions on recommendation in the Ciao dataset. (a) MAE. (b) RMSE.

Figure 5.2 and Figure 5.3 show the accuracy of these variants in Ciao and Epinions, respectively. In general, each component in our method contributes to better recommendation, and eliminating the effect of opinion interactions or user influence degrades the performance. In both datasets, opinion interactions play a far more significant role in the prediction of unknown ratings, compared with user influence. Therefore, when 60% of the data are used for training, REOD\OP has a 7.884% relative perfor-

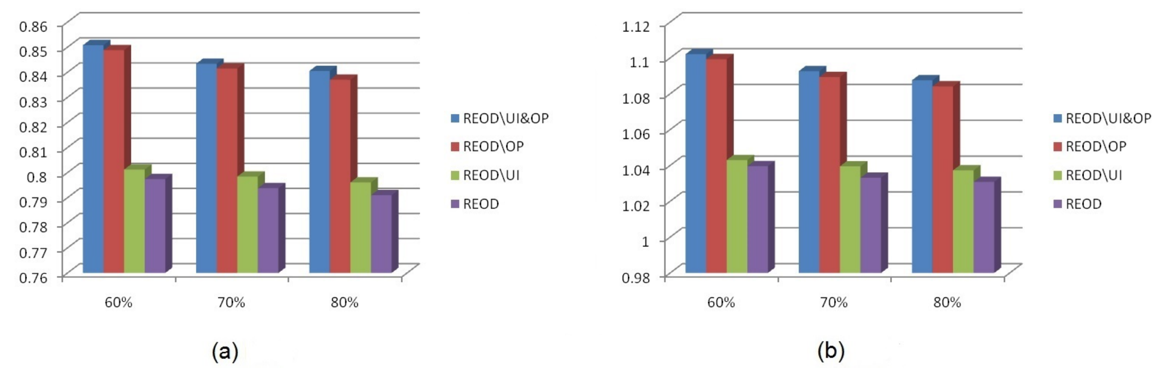


Figure 5.3: Impact of user influence and opinion interactions on recommendation in the Epinions dataset. (a) MAE. (b) RMSE.

mance reduction for MAE in Ciao data, and 5.608% in Epinions data. The procedure of opinion interactions in each iteration of SGD does not need the trust network, therefore, it will not suffer from the sparsity problem of trust relations. User influence slightly reduces MAE and RMSE in both datasets whether the effect of opinion interactions is included or not. Furthermore, the improvement of the performance under the action of user influence is more obvious in Ciao data than in Epinions data, as a result of denser user relations in Ciao, especially when less training data are applied.

We are concerned about the evolution of user opinions during the SGD training of our method. We use the average squared distance of individual opinions to reflect the divergence of opinions. The average squared distance is defined as

$$D(t) = \frac{\sum_i \left\| U_i(t) - E\big(U(t)\big) \right\|^2}{m} \tag{5.15}$$

where $E(\cdot)$ means the expectation operation. A larger squared distance

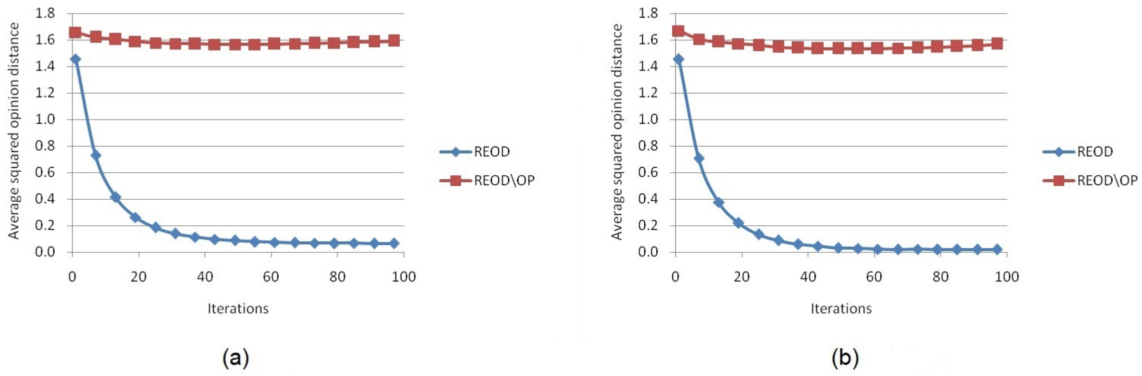


Figure 5.4: Average squared distance of user opinions versus iteration number with 70% training data. (a) Ciao. (b) Epinions.

means more disordering exists in user opinions. Figure 5.4 shows the average squared opinion distance versus the iteration number with or without opinion interactions, when 70% of the data is used for training. We also find that with different training data, the evolution of opinions is analogous. For REOD, the average squared distance drops to a very low value and gradually becomes stable in about 50 iterations. Consensus is almost achieved, especially in the Epinions data, implying very small divergence among user opinions. Due to the existence of opinion interactions, users tend to adapt their opinions so that they are close to each other. This phenomenon is in accordance with real situations in social networks (Li et al., 2013), since users tend to persuade others to trust their opinions during opinion interactions. As a result, opinion interactions clearly improve the recommendation accuracy. However, for REOD\OP, the average squared distance only marginally decreases, and user opinions are quite divergent.

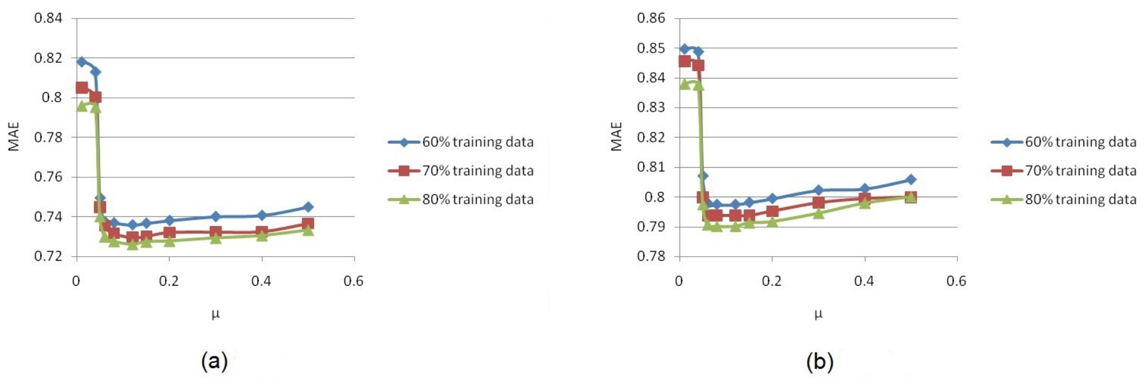
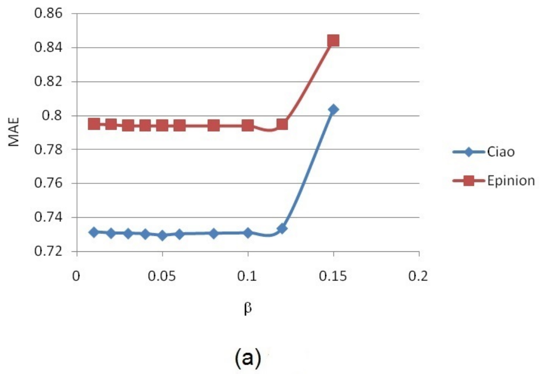


Figure 5.5: Performance variations of REOD versus the trust parameter μ. (a) Ciao. (b) Epinions.

Now, we address the third issue: the effects of parameters for opinion interactions and user influence on the performance. The parameter μ determines the rate of opinion exchanges. We change the value of μ, and investigate the corresponding recommendation accuracy. Since users generally update their opinions so that they are close to their neighbors' opinions, the value of μ does not exceed 0.5. Figure 5.5 shows the impact of μ with different training data. The variations of RMSE are similar to those of MAE, and therefore, we do not depict RMSE here. We can clearly observe a transition at $\mu = 0.05$ below which the method has larger MAE. MAE starts a precipitous decline around $\mu = 0.05$, and reaches a plateau in the interval $[0.05, 0.2]$. We investigate opinion evolution with different μ. For $\mu < 0.05$, the final average squared opinion distance is larger than 1.5, while that for $\mu > 0.05$ is below 0.08. Therefore, if $\mu < 0.05$, user opinions have few changes during opinion interactions, and opinion interactions do not take effect in recommendation. Then, our method



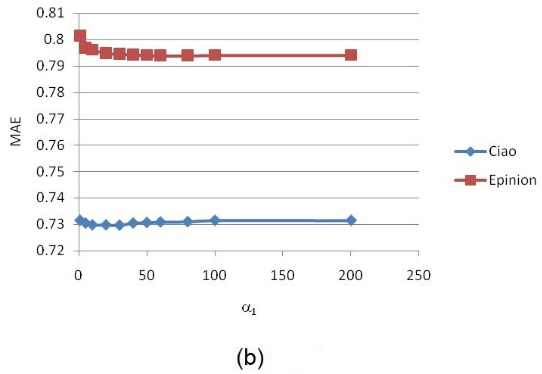


Figure 5.6: Performance variations of REOD versus the payoff parameter β and the influence decay parameter α_1 with 70% training data. (a) the payoff parameter β. (b) the influence decay parameter α_1.

reduces to REOD\OP. If $\mu > 0.2$, MAE increases slowly with μ. For large μ, the variation of an estimated rating $\left|\mu \cdot (U_f - U_i)V_j^T\right|$ may be larger than $2 \cdot \left|R_{ij} - U_i V_j{}^T\right|$, so that we have $\left|R_{ij} - U_{i,new} V_j{}^T\right| > \left|R_{ij} - U_i V_j{}^T\right|$ and errors of estimated ratings may increase. Generally, our method achieves lower MAE in the interval $[0.05, 0.2]$ of μ in both datasets, irrespectively of the proportion of training data. Thus, we can typically set $\mu = 0.12$ without loss of generality. Since the impact of μ does not depend on the datasets, the complexity of our method can be reduced.

The parameter β controls the equilibrium between the strategy of changing an opinion or maintaining an opinion in the evolutionary game model. Here, we only consider one case with 70% training data. In other cases with a different amount of training data, the performance variations are similar. Figure 5.6(a) shows the impact of β on MAE. When $0 < \beta < 0.1$, MAE remains relatively stable in both datasets as β varies. Around $\beta = 0.05$, MAE reaches the lowest value. When $\beta > 0.1$, MAE increases rapidly, demonstrating that the error for the estimated rating should preferentially be considered in the evolutionary game of opinion interactions. From Eq. (5.8), when β approaches μ, increasing β makes users choose the strategy of maintaining their opinions, and restrains the effect of opinion interactions in recommendation. In addition, the impact of the payoff parameter β is also independent of the datasets, reducing the complexity of our method. In different datasets, we can empirically set the value of β. Figure 5.6(b) shows the impact of the influence decay parameter α_1. It is obvious that MAE in both datasets does not have a close correlation with α_1. Although the best performance in different datasets varies with α_1, the change of MAE is small and we obtain relatively low MAE in the interval (20,90) of α_1. Most of degrees F_i^- in both datasets are less than 50. When $\alpha_1 > 20$, the variation of user influence versus α_1 is very small. The aforementioned properties of parameters are useful from a practical point of view because they make it easier to set parameters in using our method.

We use other topological descriptors to measure user influence, such as betweenness centrality, clustering coefficients and k-core index, and incorporate the influence into recommender systems. We use 80% data as training data, and evaluate the recommendation performance with different forms of user influence. All parameters are determined by cross-validation. Results of recommendation accuracy with different user influence are shown in the Table 5.5. Although topological descriptors have different capabilities of measuring user influence, their effects on the recommendation performance are similar in both datasets. MAE and RMSE of degree centrality, clustering coefficients and k-core index are approximately the same, but the descriptor of betweenness centrality has a lower performance. The reason is that betweenness in these networks is more heterogeneous than other descriptors, so that users with large betweenness play an excessively important role in the sum-of-squared-errors objective function. Ratings of users that have very small betweenness have limited contribution to the objective function, but these users may have many ratings and cannot be ignored in recommendation. In addition, we also measure user influence by tie strength, and incorporate user influence and social regularization into recommender systems, but the accuracy cannot be improved.

Table 5.5: Results of recommendation accuracy with different user influence in Ciao and Epinions.

Ciao	Degree centrality	Betweenness centrality	Clustering coefficient	k-core index
MAE	0.7262	0.7325	0.7261	0.7259
RMSE	0.9701	0.9821	0.9697	0.9698
Epinions	**Degree centrality**	**Betweenness centrality**	**Clustering coefficient**	**k-core index**
MAE	0.7262	0.7325	0.7261	0.7259
RMSE	0.9701	0.9821	0.9697	0.9698

5.5 Summary

When users on online social networks interact with their friends, their opinions are influenced by others. User interactions can be applied in recommender systems to improve performance. Social recommendation models utilize social relations, and introduce neighbors' impact into the MF framework. In this chapter, we investigated the impact pattern of other users on latent preferences, and studied its effect on recommendation. We proposed an evolutionary game model to characterize opinion interactions. We defined two interaction behaviors, i.e., maintaining one's opinion or changing one's opinion, and determined the payoff for each behavior according to the rating on a given item. Users choose the behavior which maximizes their payoffs. Then, we measured user influence according to node ingoing degrees in the social network. We further used user influence to weight the objective function of MF, and conducted dynamic opinion interactions during each iteration of training. Experiment results on real-world datasets demonstrated that our method performs better than state-of-the-art recommendation models for all user and cold start users. Meanwhile, our method has much less computational complexity than the other models. Opinion dynamics drive user opinions to converge and reduce the divergence, coinciding with the real situation in online interactions. Moreover, our method does not have a significant dependence of opinion interaction or user influence parameters.

Chapter 6

TruGRC: Trust-aware Group Recommendation with Virtual Coordinators

6.1 Introduction

Recommender systems have attracted much attention for their ability to model user preferences and generate personalized predictions based on historical behaviors (Lu et al., 2015). As such, they have become a useful tool for disposing the information overload problem in e-commerce systems (Ren and Wang, 2018). Most previous studies have focused on personal recommender systems. However, individuals are not isolated entities and they are usually part of some sort of organizations or groups that revolve around shared activities or similar interests (Ji et al., 2018). The behaviors of a group of individuals sharing similar interests can be considered as group activities, for example, travelling, seeing movies, and dining out with a number of friends. As group activities on websites increase, many studies on group recommender systems (Guo et al., 2016b; Castro et al., 2017b; Ortega et al., 2016; Wang et al., 2016b; Castro et al., 2017a) have been conducted in recent years to provide recommendations to a given group of users.

Unlike individual recommender systems, group recommender systems often contain a diverse set of preferences and group recommendation is to make a single set of recommendations for a group of users with different preferences. Hence, the challenge in group recommendation is how to integrate individual's preferences into a unique recommendation list which will be satisfied by all group users. There are two main ways for aggregating personal preferences into group preferences: result aggregation (Castro et al., 2017a; Wang et al., 2018) and profile aggregation (Wang et al., 2016b; Kagita et al., 2015). Result aggregation applies personal recommendation methods to generate recommendations for every user and then makes all these recommendations into a combined recommendation list for the whole group (Baltrunas et al., 2010; Castro et al., 2015). Studies on these types of group recommendation system mainly focus on enhancing the rationality and effectiveness of aggregation functions to achieve better accuracy (Seo et al., 2018). Profile aggregation strategies create a virtual user to represent the combined profile of group members, then predictions are made for the virtual user by applying personal recommendation methods. This alternative is called virtual user-based approaches (Ortega et al., 2016; Kagita et al., 2015). However, most existing studies consider the above two strategies separately. Very few consider ways to benefit from both. Yet, solely relying on one strategy creates problems in complex recommendation scenarios and, generally, does not produce satisfactory recommendations for every group member. Further, websites and social networks contain a great deal of auxiliary information, e.g., trust links (Ma et al., 2011), which have not yet been considered in group recommendation. Although some social-aware group recommender systems have been proposed (Guo et al., 2016b; Quijano-Sanchez et al., 2013), but most of them only simply infer users' social relationships and influence through the Thomas-Kilmann conflict mode instrument (TKI) (Thomas, 2008). Actual and explicit trust links between users on social networks are still ignored.

In this chapter, we propose a Trust-aware Group Recommendation method with virtual Coordinators (TruGRC), which integrates the benefits of both the result and profile aggregation strategies. Moreover, we introduce personal influence and trust links into group recommendation tasks. Group recommendation processes can be considered as a negotiation in which every member of the group hopes the group's preferences will match their own personal preferences as much as possible. Yet, when group members hold conflicting preferences, it can be difficult to aggregate individual preferences using simple aggregation functions, such as the average and least-misery strategies. Some aggregation functions, such as GROD (Castro et al., 2017a), ASI (Guo et al., 2016b) and MC-GR (Wang et al., 2016b), can increase the accuracy of group recommendations, but these functions are complicated, and their afforded improvement is limited. Hence, to maximize the benefits for all group members, while overcoming conflicts in preferences, it is necessary to introduce a coordinator. In our method, we regard the virtual user as a virtual coordinator that is introduced into the process of modeling each group user's preferences. The virtual coordinator provides a global view of all user preferences and harmonizes their benefits by negotiating with them. These negotiations with the coordinator involve compromise but ultimately generate recommendations that are reasonable to each member of the group. Further, the resulting recommendations can be easily distilled using the average aggregation. Thus, the main contributions can be summarized as follows.

1) We propose TruGRC method that integrates both the result and the profile aggregation strategies. Specifically, we introduce a virtual coordinator into group recommendation, which brings a global perspective for optimizing the evaluation process of individual user preferences and creating a balanced set of group recommendations. With the contribution of the virtual coordinator, applying the average aggregation method to generate a satisfactory recommendation list is easy.

2) We introduce trust information into group recommender systems, such as explicit trust links on social networks. The personal influence is inferred from user's trust, and each group member impacts the virtual coordinator based on its personal influence. We also consider the interactions between group users to represent the negotiation process.

3) We implement extensive experiments to evaluate the proposed TruGRC model. The comparisons between TruGRC and several cutting-edge methods indicate TruGRC outperforms its counterparts in four common evaluation metrics at various group sizes.

6.2 The Proposed Method

To formalize the group recommendation problem, the user set of group g is defined as $\mathcal{U}$ and the set of items is defined as $\mathcal{I}$, where $u \in \mathcal{U}$ means each individual in group g, and $i \in \mathcal{I}$ is each item in the item set. We define $|\mathcal{U}| = m$ as the size of the user group and $|\mathcal{I}| = n$ as the number of all items. In this group recommendation scenario, only the users in a specific group are considered, and each group has no impact on another. The user-item rating matrix is denoted as $\mathcal{R} \in \mathbb{R}^{m\times n}$, and the observed rating of user u on item i is defined as $r_{u,i} \in \mathcal{R}$. The aggregation function is the vital part of integrating users' preferences, defined as

$$\widehat{r}_{g,i} = \Phi(\widehat{r}_{u,i}) \tag{6.1}$$

where $\Phi(\cdot)$ is the aggregation function that generates the group preferences from each user's preferences. For item i, $\widehat{r}_{g,i}$ and $\widehat{r}_{u,i}$ are the combined rating of group g and the predicted rating of user u, respectively. For the group, a recommendation list is proposed in accordance with the top-L highest scores of $\widehat{r}_{g,i}$, which is defined as follows.

$$top(g,L) = \arg\max_{i \in \mathcal{I}}^{L} \widehat{r}_{g,i} \tag{6.2}$$

The full set of notations is given in Table 6.1.

Table 6.1: Notations in this work.

Symbol	Description
$\mathcal{U}$	user set in group g, $u \in \mathcal{U}$, $\|\mathcal{U}\| = m$
$\mathcal{I}$	item set, $i \in \mathcal{I}$, $\|\mathcal{I}\| = n$
$\mathcal{R} \in \mathbb{R}^{m \times n}$	user-item rating matrix
$r_{u,i} \in \mathcal{R}$	the observed rating of item i given by user u
$\widehat{r}_{u,i}$	the prediction of user u on item i
$\widehat{r}_{c,i}$	the prediction of virtual coordinator c on item i
$\widetilde{r}_{c,i}$	the interaction representation of virtual coordinator c on item i
$\widehat{r}_{g,i}$	the prediction of group g on item i
d	the dimension of feature space
$\mathbf{p}_u \in \mathbb{R}^{d \times 1}$	individual user latent feature vector
$\mathbf{p}_c \in \mathbb{R}^{d \times 1}$	virtual coordinator latent feature vector
$\mathbf{q}_{i1} \in \mathbb{R}^{d \times 1}$	item-user latent feature vector
$\mathbf{q}_{i2} \in \mathbb{R}^{d \times 1}$	item-coordinator latent feature vector
b_{i1}	item-user bias
b_{i2}	item-coordinator bias
s_u	the personal social impact of user u
$\mathcal{T} \in \mathbb{R}^{m \times m}$	user-user trust matrix
$t_{u,l} \in \mathcal{T}$	the trust relation between user u and user l
λ, λ_α, λ_β	trade-off parameters

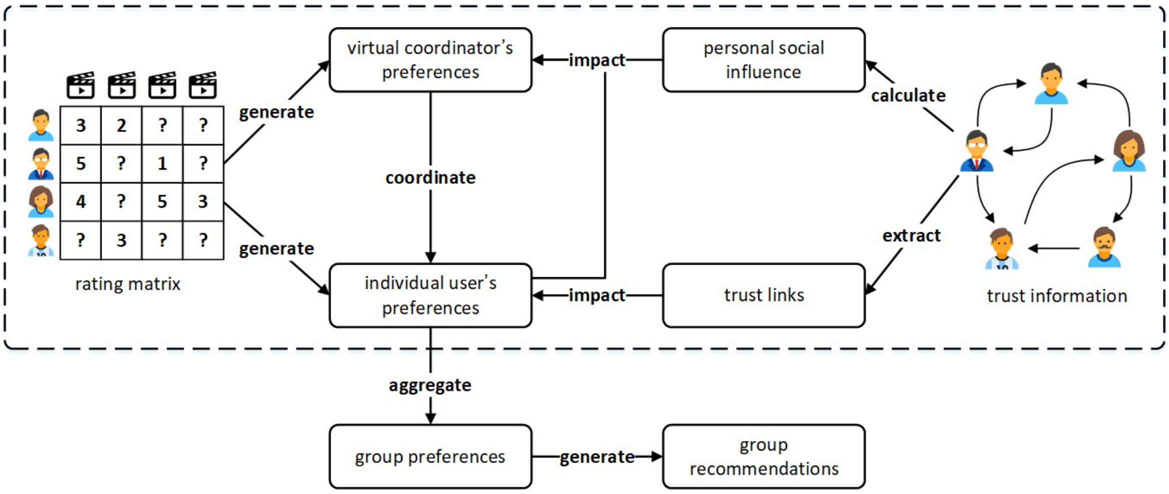


Figure 6.1: The framework of the proposed TruGRC method. A virtual coordinator is introduced to harmonize each individual user's preferences in the group, and each group member simultaneously impacts the virtual coordinator's preferences according to its personal social inference calculated by trust information. Group recommendations are generated in accordance with group preferences aggregated by each group member's preferences.

6.2.1 Overall Framework

This section introduces the framework of the proposed method which improves group recommendation by harmonizing each user's preferences and considering the trust information. We can regard group recommendation as a negotiation process (Salamó et al., 2012) where the aim is to generate a single recommendation list that meets the requirements of most users in the group. In our framework, this is accomplished by integrating a virtual coordinator into a traditional group recommendation framework. This virtual coordinator plays a significant role in the negotiation between individuals, balancing individual preferences with group preferences. An overview of the framework appears in Figure 6.1. We generate each individual user's preferences from the ratings they have already marked over items. In addition, we assume the virtual coordinator can observe all the historical ratings of users in the group and its preferences are generated from all group users' ratings. The virtual coordinator provides an overall perspective for modeling and modifying individual preferences. However, each group member has a level of influence over the virtual coordinator, which is inferred from its influence on trust networks. And this influence can affect negotiations with the virtual coordinator. Group preferences will be generated by simply aggregating each user's preferences once negotiations are complete, and then a group recommendation list will be proposed.

6.2.2 The Individual Recommendation Method

Given the matrix factorization (MF) model (Koren et al., 2009) has demonstrated good performance for modeling user preferences and predicting missing ratings, we have selected this model as the individual recommendation method for our framework. The MF model applies latent feature vectors to represent the preferences of users and items, then the ratings can be fairly estimated from these latent feature vectors, defined as

$$\widehat{r}_{u,i} = b_{i1} + \mathbf{p}_u^\top \mathbf{q}_{i1} \tag{6.3}$$

where $\widehat{r}_{u,i}$ is the predicted rating that user u gives item i, $\mathbf{p}_u \in \mathbb{R}^{d\times 1}$ is the user-specific latent feature vector, $\mathbf{q}_{i1} \in \mathbb{R}^{d\times 1}$ is the item-specific latent feature vector, d is the dimension of latent features, and b_{i1} is the bias of item i. In recommendation, there is an important task which is to ensure the predicted rating $\widehat{r}_{u,i}$ is as close to the observed rating $r_{u,i}$ as possible (Guo et al., 2016a). To accomplish this goal, following the MF model (Koren et al., 2009), the latent feature vectors and the bias can be learned by minimizing the following loss function

(6.4)
$$\min_{\mathbf{p}_u,\mathbf{q}_{i1},b_{i1}} \mathcal{L}_u = \frac{1}{2}\sum_{u=1}^{m}\sum_{i=1}^{n}\delta_{u,i}\left(r_{u,i}-\widehat{r}_{u,i}\right)^2+\frac{\lambda}{2}\Big(\sum_{u=1}^{m}||\mathbf{p}_u||_F^2+\sum_{i=1}^{n}||\mathbf{q}_{i1}||_F^2+\sum_{i=1}^{n}b_{i1}^2\Big)$$

where $\delta_{u,i}$ represents an indicator that $\delta_{u,i}=1$ means $r_{u,i}$ is observed, and $\delta_{u,i}=0$ otherwise. $||\cdot||_F$ is the Frobenius norm to avoid over-fitting and λ is a trade-off parameter to regulate the impact of the regularization terms.

In term of the definition in Eq. (6.1), the results of group recommendation is generated by aggregating every individual's predictions in the group. Previous studies (Ortega et al., 2016; Castro et al., 2017a) have mainly focused on improving the aggregation functions. However, they have neglected to adjust the individual's preferences to suit the final set of group recommendations. By contrast, we propose a new group recommendation model that introduces a virtual coordinator to enhance performance and considers the trust information associated with each individual. This model is presented in the next sections.

6.2.3 Modeling the Virtual Coordinator in the Group

Group recommendation can be regarded as a negotiation process, where every group member wants to maximize its benefits. However, usually, the results of any negotiation do not entirely meet the needs of all group members. Hence, the coordinator plays an important role in harmonizing each member's opinions and requirements to ensure the benefits are spread across the whole group. Based on this idea, the aim is to model a virtual coordinator that alters the preference estimations for each group member when making predictions. We assume that the virtual coordinator is capable of observing the historical data of all group members, e.g., rating information or purchasing records, so it can form a global perspective on the preferences of the entire group. This approach is unlike typical virtual user-based group recommendation methods (Kagita et al., 2015) where the virtual user only makes recommendations based on its own profile. Whereas, our model combines multiple sources of feedback from both group members and the virtual coordinator to enhance the accuracy of group recommendation.

Similar to individual recommendation methods, we also map the virtual coordinator and each item into the same feature space, defined as $\mathbf{p}_c \in \mathbb{R}^{d\times 1}$ and $\mathbf{q}_{i2} \in \mathbb{R}^{d\times 1}$, respectively. Therefore, the predicted ratings of the virtual coordinator can be defined as

$$\widehat{r}_{c,i} = b_{i2} + \mathbf{p}_c^{\top}\mathbf{q}_{i2} \tag{6.5}$$

where $\widehat{r}_{c,i}$ denotes the predicted rating of the virtual coordinator c and b_{i2} is the bias. $\mathbf{p}_c$, $\mathbf{q}_{i2}$ and b_{i2} are learned through a square loss function because the virtual coordinator needs to observe all group members' ratings. Therefore, the loss function is defined as

$$\min_{\mathbf{p}_c,\mathbf{q}_{i2},b_{i2}} \mathcal{L}_c = \frac{1}{2}\sum_{u=1}^{m}\sum_{i=1}^{n}\delta_{u,i}\left(r_{u,i}-\widehat{r}_{c,i}\right)^2+\frac{\lambda}{2}\Big(||\mathbf{p}_c||_F^2+\sum_{i=1}^{n}||\mathbf{q}_{i2}||_F^2+\sum_{i=1}^{n}b_{i2}^2\Big) \tag{6.6}$$

Be different from Eq. (6.4), Eq. (6.6) is used to learn the preferences of the virtual coordinator c rather than individual's preferences. Moreover, the virtual coordinator's latent feature vector $\mathbf{p}_c$ interacts with all latent feature vectors of items collected by group members in Eq. (6.6), whereas in Eq. (6.4), users only interact with items rated by themselves. Eq. (6.4) and Eq. (6.6) are combined linearly to construct a new loss function that integrates multiple feedbacks as

$$\begin{aligned}\min_{\Theta}\mathcal{L} &= \mathcal{L}_u+\mathcal{L}_c\\ &= \frac{1}{2}\sum_{u=1}^{m}\sum_{i=1}^{n}\delta_{u,i}\left(r_{u,i}-\widehat{r}_{u,i}\right)^2+\frac{1}{2}\sum_{u=1}^{m}\sum_{i=1}^{n}\delta_{u,i}\left(r_{u,i}-\widehat{r}_{c,i}\right)^2\\ &\quad+\frac{\lambda}{2}\Big(\sum_{u=1}^{m}||\mathbf{p}_u||_F^2+||\mathbf{p}_c||_F^2+\sum_{i=1}^{n}||\mathbf{q}_{i1}||_F^2+\sum_{i=1}^{n}||\mathbf{q}_{i2}||_F^2+\sum_{i=1}^{n}b_{i1}^2+\sum_{i=1}^{n}b_{i2}^2\Big)\end{aligned} \tag{6.7}$$

where $\Theta=\{b_{i1}, b_{i2}, \mathbf{p}_u, \mathbf{p}_c, \mathbf{q}_{i1}, \mathbf{q}_{i2}\}$. Two latent feature vectors have been defined for each item, i.e., $\mathbf{q}_{i1}$ and $\mathbf{q}_{i2}$, along with two biases for each item, i.e., b_{i1} and b_{i2}. These vectors are used to make predictions for individuals and the virtual coordinator in Eq. (6.3) and Eq. (6.5), respectively. Although $\mathbf{q}_{i1}$ and $\mathbf{q}_{i2}$ serve for different objects, they indicate the latent features of the identical item, so they need to have the same intrinsic properties. Hence, a regularization term constrains these two latent feature vectors as follows

$$\frac{1}{2}\sum_{i=1}^{n}||\mathbf{q}_{i1}-\mathbf{q}_{i2}||_F^2 \tag{6.8}$$

Once the virtual coordinator has been modeled, the interactions between it and the other group members need to be modeled. The virtual coordinator plays a coordinating role in modeling each user's preferences based on its global perspective. That is, each user's preferences are adjusted so that subsequent predictions create a balance between each user and the entire group. In order to model these interactions, we use $\widetilde{r}_{c,i}$ to define another similar representation of the virtual coordinator's predictions as follows

$$\widetilde{r}_{c,i} = b_{i1} + \mathbf{p}_c^{\top}\mathbf{q}_{i1} \tag{6.9}$$

where $\mathbf{q}_{i1}$ is the item-specific latent feature vector for making the individual's predictions in Eq. (6.3). This representation can be regarded as the interactions between $\mathbf{p}_c$ and $\mathbf{q}_{i1}$ in the latent feature space, which affects $\mathbf{q}_{i1}$ with the virtual coordinator's impact. In addition, a constraint is placed on the two representations of the virtual coordinator's predictions. This constraint is defined as

$$\frac{1}{2}\sum_{i=1}^{n}\left(\widehat{r}_{c,i}-\widetilde{r}_{c,i}\right)^2 \tag{6.10}$$

where the distance between these two representations should be as short as possible because they have the same goal of modeling the virtual coordinator's preferences. These two regularization terms are incorporated into the loss function as

(6.11)

$$\begin{aligned}\min_{\Theta} \mathcal{L} = & \frac{1}{2}\sum_{u=1}^{m}\sum_{i=1}^{n}\delta_{u,i}\left(r_{u,i}-\widehat{r}_{u,i}\right)^2 + \frac{1}{2}\sum_{u=1}^{m}\sum_{i=1}^{n}\delta_{u,i}\left(r_{u,i}-\widehat{r}_{c,i}\right)^2 \\ & + \frac{\lambda_\alpha}{2}\sum_{i=1}^{n}||\mathbf{q}_{i1}-\mathbf{q}_{i2}||_F^2 + \frac{\lambda_\alpha}{2}\sum_{i=1}^{n}\left(\widehat{r}_{c,i}-\widetilde{r}_{c,i}\right)^2 \\ & + \frac{\lambda}{2}\Big(\sum_{u=1}^{m}||\mathbf{p}_u||_F^2 + ||\mathbf{p}_c||_F^2 + \sum_{i=1}^{n}||\mathbf{q}_{i1}||_F^2 + \sum_{i=1}^{n}||\mathbf{q}_{i2}||_F^2 + \sum_{i=1}^{n}b_{i1}^2 + \sum_{i=1}^{n}b_{i2}^2\Big)\end{aligned}$$

where λ_α is a trade-off parameter to regulate the impact of above two item-related regularization terms.

6.2.4 Trust-aware Group Recommendation with the Virtual Coordinator

Trust is an important feature on social networks as it indicates the relationships between users. In practice, asymmetric trust is more general than symmetric trust (Wang et al., 2017e). For example, one user following another user on Twitter can be seen as a trust link between these two users, but an asymmetric one, because the trust is not mutual. Several studies on trust-aware recommendation for individuals have been conducted (Ma et al., 2011; Guo et al., 2016a; Wang et al., 2017e), but, to date, trust information has not been exploited in group recommender systems.

During the negotiation process, the coordinator needs to communicate with group members, which means each individual is not only managed, they also have an impact on the coordinator. Based on the assumption that the virtual coordinator can be affected by the group members, the impact of each user is defined by their personal social influence. Determining a user's influence is similar to identifying the vital nodes in a complex network. Neighborhood-based centralities, e.g., degree centrality, localRank, and coreness, are widely used for identifying vital nodes because of their low computational complexity (Lü et al., 2016). Here, we use degree centrality to express the personal impact of every user on social networks, defined as

$$dc(u) = \frac{k_u}{m-1} \tag{6.12}$$

where the range of $dc(u)$ is from 0 to 1, k_u indicates the degree of user u, and $m-1$ is the largest possible degree. Further, in a group, personal influence is not absolute but rather relative. It depends on a comparison with the influence held by other group members. Therefore, to properly evaluate personal influence within a group, the degree centrality must be normalized. The normalizing function is defined as

$$s_u = \frac{dc(u)}{\sum_{u=1}^{m} dc(u)} \tag{6.13}$$

where $s_u > 0$ indicates user u's personal influence, which is captured and calculated by trust information. $s_u = 0$ is caused by $dc(u) = 0$, which means user u does not have any trust links on the network. Hence, for a user with no observable trust links, the value of personal influence is randomly generated. In general, users with high personal influence, such as actors or public figures, will be very active on social networks. Therefore, when

$dc(u) = 0$, s_u is set as a random number $r \in (0, 0.5]$ because it is highly likely that a user with no trust links will not have great influence. Thus, s_u is altered as follows

$$s_u = \begin{cases} \frac{dc(u)}{\sum_{u=1}^{m} dc(u)}, dc(u) > 0 \\ r \in (0, 0.5], dc(u) = 0 \end{cases} \tag{6.14}$$

With personal influence defined, each group member impacts the virtual coordinator according to its influence level, as defined below

$$\frac{1}{2} \sum_{u=1}^{m} s_u \left|\left|\mathbf{p}_u - \mathbf{p}_c\right|\right|_F^2 \tag{6.15}$$

In addition, the interactions between two group members are modeled based on their trust links on social networks. If user l follows user u, a trust link $t_{u,l}$ between them is identified. This interaction is defined as

$$\frac{1}{2} \sum_{u=1}^{m} \sum_{l=1}^{m} t_{u,l} \left|\left|\mathbf{p}_u - \mathbf{p}_l\right|\right|_F^2 \tag{6.16}$$

Mathematically, Eq. (6.15) and Eq. (6.16) introduce two constraints into the loss function that minimize the distance between two parameters. Effectively, this forces the preferences of the two users to be close together. This makes sense because each user wants the virtual coordinator to represent its preferences better and one user will probably trust another user with similar preferences. Therefore, Eq. (6.15) and Eq. (6.16) are combined with Eq. (6.11) to generate the final loss function as follows

(6.17)

$$\min_{\Theta} \mathcal{L} = \frac{1}{2} \sum_{u=1}^{m} \sum_{i=1}^{n} \delta_{u,i} \left(r_{u,i} - \widehat{r}_{u,i}\right)^2 + \frac{1}{2} \sum_{u=1}^{m} \sum_{i=1}^{n} \delta_{u,i} \left(r_{u,i} - \widehat{r}_{c,i}\right)^2$$

$$+\frac{\lambda_\alpha}{2}\sum_{i=1}^{n}||\mathbf{q}_{i1}-\mathbf{q}_{i2}||_F^2+\frac{\lambda_\alpha}{2}\sum_{i=1}^{n}\left(\widehat{r}_{c,i}-\widetilde{r}_{c,i}\right)^2$$
$$+\frac{\lambda_\beta}{2}\sum_{u=1}^{m}s_u\,||\mathbf{p}_u-\mathbf{p}_c||_F^2+\frac{\lambda_\beta}{2}\sum_{u=1}^{m}\sum_{l=1}^{m}t_{u,l}\,||\mathbf{p}_u-\mathbf{p}_l||_F^2$$
$$+\frac{\lambda}{2}\Big(\sum_{u=1}^{m}||\mathbf{p}_u||_F^2+||\mathbf{p}_c||_F^2+\sum_{i=1}^{n}||\mathbf{q}_{i1}||_F^2+\sum_{i=1}^{n}||\mathbf{q}_{i2}||_F^2+\sum_{i=1}^{n}b_{i1}^2+\sum_{i=1}^{n}b_{i2}^2\Big)$$

where λ_β is a trade-off parameter to regulate the impact of above two trust-related regularization terms.

6.2.5 Learning and Prediction

To learn the parameters Θ of our proposed model in Eq. (6.17), we use the gradient descent to reach a local minimization of the loss function. Gradient descent is an effective way for minimization when objective functions are differentiable and non-convex, and is also the most commonly used algorithm in MF-based recommender systems (Koren et al., 2009; Koren, 2010; Guo et al., 2016a, 2017a; Ma, 2013; Xiong et al., 2020). The gradients of the parameters Θ are performed as follows.

(6.18)

$$\frac{\partial\mathcal{L}}{\partial b_{i1}}=\sum_{u=1}^{m}\delta_{u,i}\left(\widehat{r}_{u,i}-r_{u,i}\right)+\lambda_\alpha\left(\widetilde{r}_{c,i}-\widehat{r}_{c,i}\right)+\lambda b_{i1}$$
$$\frac{\partial\mathcal{L}}{\partial b_{i2}}=\sum_{u=1}^{m}\delta_{u,i}\left(\widehat{r}_{c,i}-r_{u,i}\right)+\lambda_\alpha\left(\widehat{r}_{c,i}-\widetilde{r}_{c,i}\right)+\lambda b_{i2}$$
$$\frac{\partial\mathcal{L}}{\partial \mathbf{p}_u}=\sum_{i=1}^{n}\delta_{u,i}\left(\widehat{r}_{u,i}-r_{u,i}\right)\mathbf{q}_{i1}+\lambda_\beta s_u\left(\mathbf{p}_u-\mathbf{p}_c\right)+\lambda_\beta\sum_{l=1}^{m}t_{u,l}\left(\mathbf{p}_u-\mathbf{p}_l\right)+\lambda\mathbf{p}_u$$
$$\frac{\partial\mathcal{L}}{\partial \mathbf{q}_{i1}}=\sum_{u=1}^{m}\delta_{u,i}\left(\widehat{r}_{u,i}-r_{u,i}\right)\mathbf{p}_u+\lambda_\alpha\left(\mathbf{q}_{i1}-\mathbf{q}_{i2}\right)+\lambda_\alpha\left(\widetilde{r}_{c,i}-\widehat{r}_{c,i}\right)\mathbf{p}_c+\lambda\mathbf{q}_{i1}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{p}_c} = \sum_{i=1}^{n} \delta_{u,i} \left(\widehat{r}_{c,i} - r_{u,i}\right) \mathbf{q}_{i2} + \lambda_\alpha \left(\widehat{r}_{c,i} - \widetilde{r}_{c,i}\right) \left(\mathbf{q}_{i2} - \mathbf{q}_{i1}\right) + \lambda_\beta s_u \left(\mathbf{p}_c - \mathbf{p}_u\right) + \lambda \mathbf{p}_c$$
$$\frac{\partial \mathcal{L}}{\partial \mathbf{q}_{i2}} = \sum_{u=1}^{m} \delta_{u,i} \left(\widehat{r}_{c,i} - r_{u,i}\right) \mathbf{p}_c + \lambda_\alpha \left(\mathbf{q}_{i2} - \mathbf{q}_{i1}\right) + \lambda_\alpha \left(\widehat{r}_{c,i} - \widetilde{r}_{c,i}\right) \mathbf{p}_c + \lambda \mathbf{q}_{i2}$$

The pseudocode for learning the parameters is provided in Algorithm 6.1. The inputs include the user-item rating matrix $\mathcal{R}$, the user-user trust matrix $\mathcal{T}$, the dimension of the feature space d, the regularization parameters λ, λ_α and λ_β, and the learning rate η. The parameters Θ form the output of the algorithm.

Algorithm 6.1 Learning the parameters in TruGRC method.

Require: $\mathcal{R}$, $\mathcal{T}$, d, λ, λ_α, λ_β, η
Ensure: $\Theta = \{b_{i1}, b_{i2}, \mathbf{p}_u, \mathbf{p}_c, \mathbf{q}_{i1}, \mathbf{q}_{i2}\}$

```
Initialize the parameters Θ ~ N(0, 0.01)
Calculate the personal influence s_u according to Eq. (6.14)
while L is not coveraged do
    Calculate gradients according to Eq. (6.18)
    b_i1 ← b_i1 − η ∂L/∂b_i1
    b_i2 ← b_i2 − η ∂L/∂b_i2
    p_u ← p_u − η ∂L/∂p_u
    q_i1 ← q_i1 − η ∂L/∂q_i1
    p_c ← p_c − η ∂L/∂p_c
    q_i2 ← q_i2 − η ∂L/∂q_i2
end while
return Θ
```

Once the parameters Θ are optimized, each user's predictions are generated using Eq. (6.3) and the group's predictions are generated using Eq. (6.1). Note that, in this chapter, the average aggregation method has been used as the aggregation function $\Phi(\cdot)$. The list of recommendations for the whole group is then arranged using Eq. (6.2).

6.2.6 Complexity Analysis

Most of the computation complexity lies in optimizing the loss function and calculating the corresponding gradients. The time to compute the loss function $\mathcal{L}$ is $O(2d\,|\mathcal{R}|)$, where d is the dimension of feature space and $|\mathcal{R}|$ is the number of observed records. The value for $|\mathcal{R}|$ should be much smaller than the matrix cardinality because of the sparsity. Further, the time to compute the gradients in Eq. (6.18) also needs to be considered. The complexity of $\frac{\partial \mathcal{L}}{\partial \mathbf{p}_u}$ is $O(d\,|\mathcal{R}| + d\,|\mathcal{T}|)$, and the other gradients have the same complexity, i.e., $O(d\,|\mathcal{R}|)$. Because $|\mathcal{T}|$ is usually much smaller than $|\mathcal{R}|$, the overall computational complexity for each iteration is $O(2d\,|\mathcal{R}|)$. It follows that the computational time of our model is linear with respect to the number of observed records in the rating matrix and, therefore, has the potential to be used with large-scale datasets.

6.3 Experimental Results

In this section, we introduce datasets, evaluation metrics, baselines and parameter settings at first, and then compare our method with all the baselines. Finally, we analyse the impact of parameters.

6.3.1 Datasets and Metrics

We select two public available datasets collected from product review websites: Ciao and Epinions to evaluate the proposed method. These two datasets have been widely used in social recommendation (Xiong et al., 2020; Tang et al., 2016b). Table 6.2 lists the statistics for these two datasets. A five-fold cross-validation (Wang et al., 2017f) is utilized in our experiments. Specifically, we randomly split each dataset into five folds. In each iteration, four folds are used as the training set, with the remaining fold treated as the testing set.

Table 6.2: Statistics of datasets.

	Ciao	Epinions
#User	2960	5155
#Item	4394	3432
#Rating	86,990	164,994
Rating sparsity	6.69×10^{-3}	9.33×10^{-3}
#Trust link	56,988	133,605

Our experiments focus on occasional groups where the members have no explicitly shared preference relevance (Castro et al., 2017a). Occasional groups are practical because they appear in many application scenarios, such as recommendations for tour groups. Most previous studies have paid attention to small groups that usually contain less than 20 members (Castro et al., 2017b; Ortega et al., 2016; Kaššák et al., 2016). In these situations, opinions are relatively easy to reach a consensus. However, it is essential to assess the feasibility of mehtods in large groups. Therefore, we randomly form groups of different sizes from 10 to 50 with an interval of 10 and test our proposed method on each group size.

Following previous studies (Ji et al., 2018; Ortega et al., 2016), four common evaluation metrics are utilized in our experiments to evaluate the proposed method and baselines, including precision (Ortega et al., 2016), recall (Ji et al., 2018), F1 (Wang et al., 2017e), and mean reciprocal rank (Wang et al., 2017f).

According to precision and recall, when the recommendation length is L, then $Pre@L$ and $Rec@L$ are defined as

$$Pre@L = \frac{1}{m}\left(\sum_{u=1}^{m}\frac{D_u(L)}{L}\right) \tag{6.19}$$

$$Rec@L = \frac{1}{m}\left(\sum_{u=1}^{m}\frac{D_u(L)}{C_u(L)}\right) \tag{6.20}$$

where $D_u(L)$ denotes the number of recommended items collected by user u and $C_u(L)$ means the number of items collected in the test set.

F1 is a comprehensive metric, defined as

$$F1@L = \frac{2Pre@L \times Rec@L}{Pre@L + Rec@L} \tag{6.21}$$

Mean Reciprocal Rank (MRR) is given by

$$MRR = \frac{1}{m}\left(\sum_{u=1}^{m}\sum_{i\in C(u)}\frac{1}{pos_i^u}\right) \tag{6.22}$$

where pos_i^u is the recommendation position of item i. A larger MRR value means better performance.

6.3.2 Baselines and Parameter Settings

To demonstrate the improvements made by the proposed method, we compare its performance with several representative baselines. These baselines span both similarity-based and matrix factorization-based group recommendation models.

User-based CF with the averaging strategy (UCF-AVG): UCF-AVG employs the user-based CF method (Herlocker et al., 1999) to make a predicted score of each item for every user in the group. Then the average aggregation function is used to generate a group recommendation score for each item. Cosine measurement is used to estimate the similarity between users.

User-based CF with the least-misery strategy (UCF-LM): Similar to UCF-AVG, UCF-LM also adpots the user-based CF method to compute estimated scores for items. However, we take the lowest predicted score as the group recommendation score for every item across all users in the least-misery strategy.

Matrix factorization with the averaging strategy (MF-AVG): MF-AVG employs the popular MF model (Koren et al., 2009) to produce individual recommendations. Then, we generate group recommendations by the average aggregation function.

Matrix factorization with the least-misery strategy (MF-LM): MF-LM also uses the MF model to make predictions for each group member, but this model employs the least-misery strategy instead of the averaging strategy when aggregating the group recommendations.

After factorization approach (AF) (Ortega et al., 2016): AF computes a group latent feature vector by combining all the user-specific vectors of group members. Then, group recommendations are made for all items through the group vector dot products with every item vector.

Before factorization approach (BF) (Ortega et al., 2016): BF models a group of users by building a virtual user that represents the item preferences of the users in the group. The recommendations for the virtual user are then used as the group recommendations through the MF model.

TruGRC: This is our proposed MF-based group recommendation method as demonstrated in Figure 6.1. TruGRC method incorporates trust information into group recommendation tasks and models a virtual coordinator to make a balance of the preferences between each user and the entire group.

Table 6.3: The parameter settings of MF-based methods.

Method	Parameter	Ciao	Epinions	Description
MF-AVG	η	0.01	0.01	Learning rate
MF-LM	λ	0.01	0.01	Avoiding over-fitting
AF	η	0.01	0.01	Learning rate
	λ	0.01	0.001	Avoiding over-fitting
BF	η	0.001	0.001	Learning rate
	λ	0.01	0.01	Avoiding over-fitting
TruGRC	η	0.001	0.001	Learning rate
	λ	0.01	0.01	Avoiding over-fitting
	λ_α	1	1	Controlling item-related regularization
	λ_β	0.01	0.001	Controlling trust-related regularization

To propose an equitable comparison, the latent features are set to the same dimension $d = 10$ for all MF-based methods. In addition, we adopt the cross validation to determine the optimal parameter values of each MF-based method in Ciao and Epinions datasets reported in Table 6.3 where MF-AVG and MF-LM share the same parameter settings, because they are both based on the same MF model (Koren et al., 2009).

6.3.3 Performance Evaluation

Table 6.4 and Table 6.5 report the comparisons of $Pre@5$, $Pre@10$, $Rec@5$, $Rec@10$, $F1@5$, $F1@10$ and MRR for the proposed TruGRC and all baseline methods with the group sizes of 10 and 20 that are the most common sizes in daily lives and widely chosen by previous studies (Wang et al., 2016b; Castro et al., 2017b).

According to the results, TruGRC demonstrates the best performance according to most metrics. Specifically, in Ciao, compared to BF, AF, MF-AVG and MF-LM, the proposed TruGRC shows an improvement of 10%, 16%, 22% and 38% on $Pre@10$ and 13%, 15%, 22% and 26% on $Rec@10$ with a group size of 10. In addition, TruGRC also enhances $F1@10$ with 11% and 22% over BF and MF-AVG which are the most competitive methods in baselines. MRR is a comprehensive metric that tests the accuracy of the whole recommendation list. TruGRC improves upon BF and MF-AVG in this metric by 5% and 10%, respectively. Similar improvements are observed with the group size of 20 in Ciao. In Epinions, MF-AVG

Table 6.4: The comparisons between TruGRC and all baselines in Ciao with the group sizes of 10 and 20.

$Group\ size = 10$	Pre@5	Pre@10	Rec@5	Rec@10	$F1$@5	$F1$@10	MRR
UCF-AVG	0.0078	0.0080	0.0064	0.0133	0.0070	0.0100	0.0439
UCF-LM	0.0073	0.0076	0.0062	0.0126	0.0067	0.0095	0.0407
MF-AVG	0.0135	0.0099	0.0115	0.0163	0.0124	0.0123	0.0565
MF-LM	0.0105	0.0088	0.0084	0.0158	0.0094	0.0113	0.0500
AF	0.0135	0.0104	0.0113	0.0173	0.0123	0.0130	0.0573
BF	0.0157	0.0110	0.0125	0.0176	0.0139	0.0135	0.0593
TruGRC	**0.0160**	**0.0121**	**0.0137**	**0.0199**	**0.0147**	**0.0150**	**0.0620**
$Group\ size = 20$	Pre@5	Pre@10	Rec@5	Rec@10	$F1$@5	$F1$@10	MRR
UCF-AVG	0.0057	0.0055	0.0050	0.0094	0.0054	0.0069	0.0323
UCF-LM	0.0047	0.0050	0.0042	0.0089	0.0044	0.0089	0.0296
MF-AVG	0.0229	0.0182	0.0213	0.0340	0.0221	0.0237	0.0841
MF-LM	0.0180	0.0139	0.0165	0.0250	0.0172	0.0179	0.0742
AF	0.0224	0.0169	0.0205	0.0305	0.0214	0.0218	0.0814
BF	0.0244	0.0184	0.0213	0.0329	0.0228	0.0236	0.0881
TruGRC	**0.0255**	**0.0194**	**0.0233**	**0.0356**	**0.0244**	**0.0251**	**0.0940**

has the best performance with the group size of 10 on Pre@5 and $F1$@5 because consistency is easy to achieve with the average aggregation in a small group size and a short recommendation list. Beyond these two metrics, TruGRC obtains the best overall results in comparison to the other baselines. With the group size of 20, BF surpasses MF-AVG in most metrics, which means that BF is more suitable for larger groups. Even so, TruGRC shows further improvements over BF at 5%, 8% and 6% on Pre@10, Rec@10 and $F1$@10. Fewer improvements are found with Epinions than Ciao because the Epinions dataset has more users, which leads to a higher probability of preference conflicts in occasional groups.

We also evaluate TruGRC with other group sizes, ranging from 10 to

Table 6.5: The comparisons between TruGRC and all baselines in Epinions when the group sizes are 10 and 20.

$Group\ size = 10$	$Pre@5$	$Pre@10$	$Rec@5$	$Rec@10$	$F1@5$	$F1@10$	MRR
UCF-AVG	0.0075	**0.0076**	0.0062	0.0123	0.0068	0.0093	0.0378
UCF-LM	0.0078	**0.0076**	0.0062	0.0120	0.0069	0.0093	0.0381
MF-AVG	**0.0092**	0.0070	**0.0081**	0.0124	**0.0087**	0.0090	0.0433
MF-LM	0.0073	0.0061	0.0067	0.0100	0.0070	0.0076	0.0393
AF	0.0084	0.0072	0.0074	0.0125	0.0078	0.0092	0.0426
BF	0.0084	0.0069	0.0076	0.0114	0.0080	0.0086	0.0408
TruGRC	0.0091	**0.0076**	**0.0081**	**0.0133**	0.0086	**0.0097**	**0.0440**
$Group\ size = 20$	$Pre@5$	$Pre@10$	$Rec@5$	$Rec@10$	$F1@5$	$F1@10$	MRR
UCF-AVG	0.0045	0.0060	0.0036	0.0104	0.0040	0.0076	0.0330
UCF-LM	0.0052	0.0064	0.0042	0.0104	0.0046	0.0079	0.0331
MF-AVG	0.0123	0.0103	0.0099	0.0180	0.0110	0.0131	0.0564
MF-LM	0.0105	0.0089	0.0090	0.0149	0.0097	0.0112	0.0499
AF	0.0110	0.0097	0.0090	0.0163	0.0099	0.0122	0.0541
BF	**0.0128**	0.0102	0.0107	0.0169	**0.0117**	0.0127	0.0576
TruGRC	**0.0128**	**0.0107**	**0.0108**	**0.0182**	**0.0117**	**0.0135**	**0.0584**

50. Figure 6.2 reports the results for all four metrics with all methods in Ciao. It is interesting to note that for UCF-AVG and UCF-LM, the performance declines as the group size increases, which means these kinds of methods are not suitable for scenarios with large groups. By contrast, the MF-based methods indicate much better performance in large group scenarios. The difference in accuracy for each MF-based method is not obvious, when the group size is small. However, these differences become more distinct as the group size grows. Although TruGRC yields to BF on $Pre@5$ and $F1@5$ with the group size of 30, it produces the best performance for most other metrics at different group sizes. In terms of MRR, TruGRC consistently demonstrates the best results. Specifically, it shows

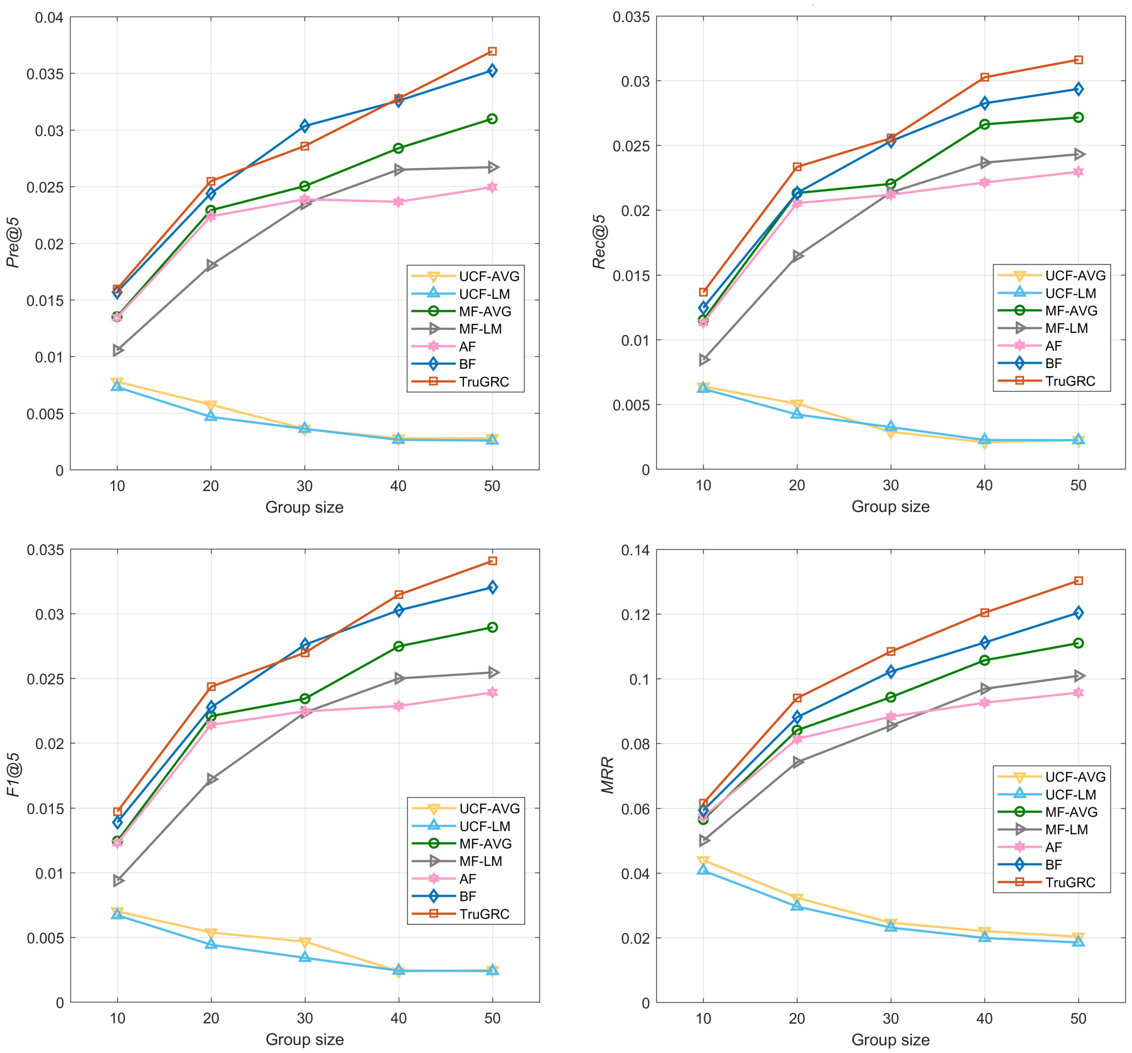


Figure 6.2: The comparisons on $Pre@5$, $Rec@5$, $F1@5$ and MRR with different group sizes in Ciao.

an improvement of 8% and 14% over BF and MF-AVG with the group size of 40 and an improvement of 8% and 17% with the group size of 50. Similar results are apparent in Figure 6.3. Compared to BF and MF-AVG, TruGRC enhances MRR by 6% and 9% with the group size of 40 and 50, respectively. These results support TruGRC's ability to address conflicts and generate consensus recommendations with large groups.

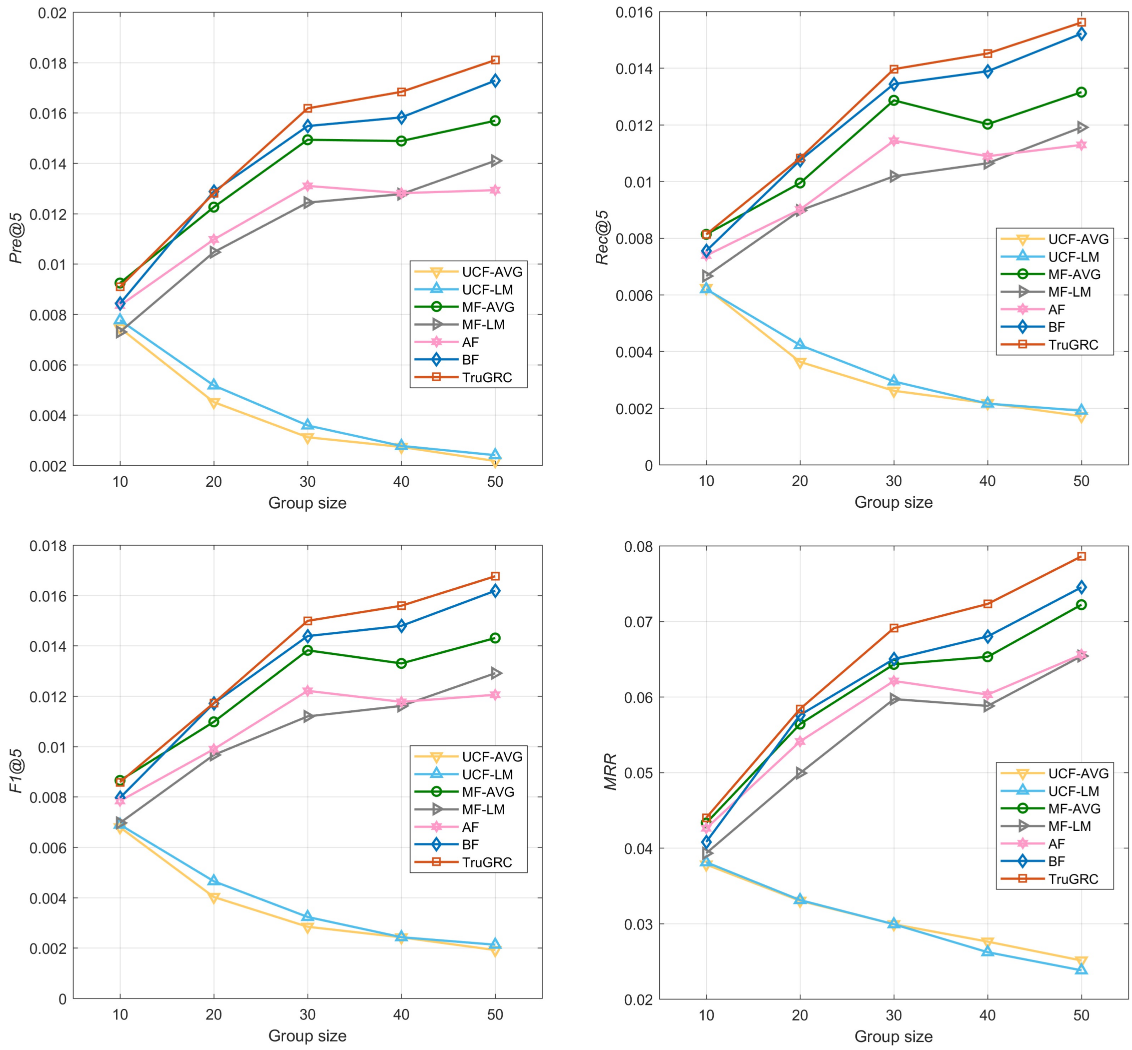


Figure 6.3: The comparisons on $Pre@5$, $Rec@5$, $F1@5$ and MRR with different group sizes in Epinions.

To assess recommendation length, we fix the group size at 20 and vary recommendation length from 5 to 50 in steps of 5 and test each method. Figure 6.4 and Figure 6.5 report the comparisons on precision, recall, and F1 in both Ciao and Epinions. From these two figures, we observe that the accuracy of BF declines with the recommendation length grows, which means BF prioritizes the top items in the recommendation list.

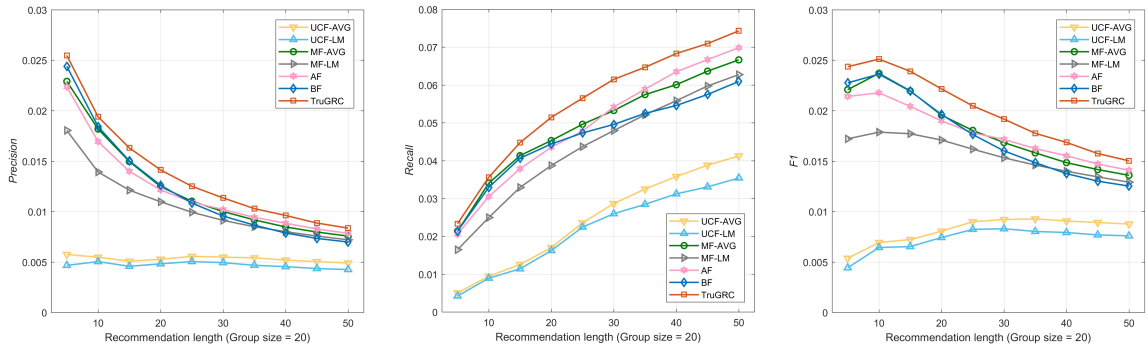


Figure 6.4: The comparisons on $Pre@5$, $Rec@5$ and $F1@5$ with different recommendation length in Ciao.

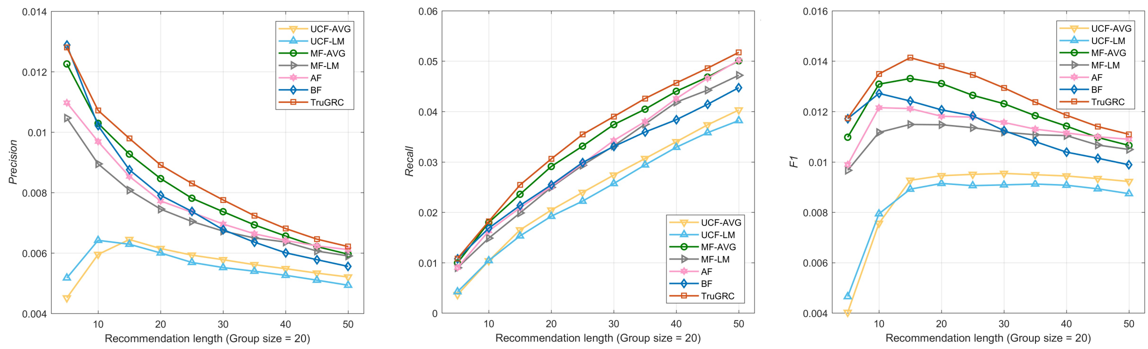


Figure 6.5: The comparisons on $Pre@5$, $Rec@5$ and $F1@5$ with different recommendation length in Epinions.

This means BF has a poor ability to generate long recommendation lists. In comparison, AF's performance overtakes the other baselines as the recommendation list becomes longer. Overall, the differences between each method's performance are reduced as the recommendation length increases. However, TruGRC still shows the better performance than AF, with a respective improvement in terms of $Pre@50$, $Rec@50$ and $F1@50$ by 7%, 6% and 7% in Ciao, and 2%, 3% and 2% in Epinions. Based on these combined experimental results, TruGRC demonstrates superior accuracy with both different group sizes and different recommendation list lengths.

6.3.4 The Impact of Parameters

TruGRC contains three parameters, i.e., λ, λ_α and λ_β. To determine these values for each dataset, we verify them in terms of MRR, and tune each parameter in the range $\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}\}$ while fixing the other two parameters. The results are reported in Figs. 6.6 and 6.7. The parameter λ avoids over-fitting, which should be very small in general, e.g. 0.001 or 0.01 (Guo et al., 2016a; Pan and Ming, 2017). The two figures show that TruGRC produces the best performance when $\lambda = 0.01$ in Ciao and $\lambda = 0.001$ in Epinions, which is reasonable.

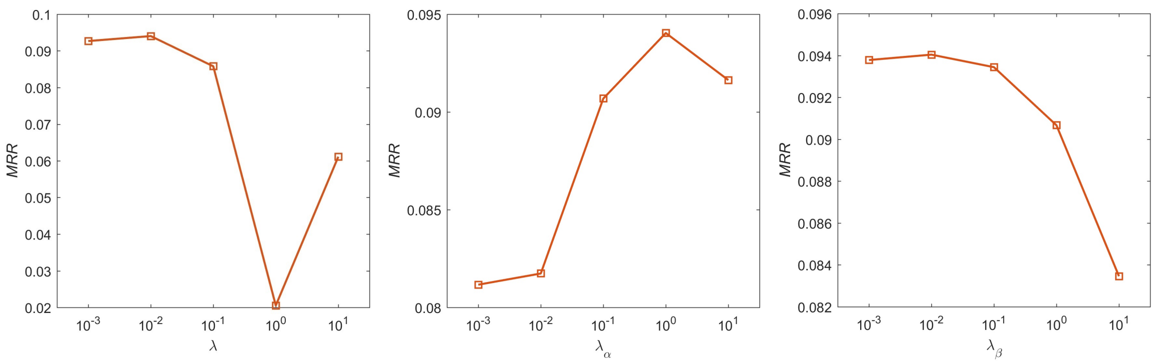


Figure 6.6: The impact of parameters λ, λ_α and λ_β on MRR in Ciao.

The parameter λ_α controls the importance of the item-related and virtual coordinator-related regularization terms. The results clearly indicate that a proper value, i.e., $\lambda_\alpha = 1$, can improve the recommendation performance in these two datasets and also demonstrate that the regularization terms in Eq. (6.8) and Eq. (6.10) are very helpful to our model. The pa-

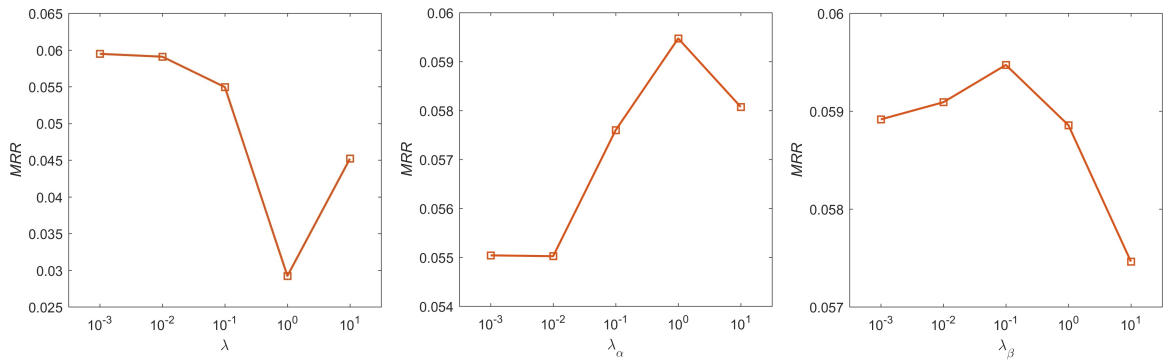


Figure 6.7: The impact of parameters λ, λ_α and λ_β on MRR in Epinions.

rameter λ_β regulates the trust-related regularization terms. The optimal values are 0.01 and 0.1 for Ciao and Epinions, respectively. When tuning this parameter, the performance changes significantly in Epinions compared to Ciao because the Epinions dataset contains more observed trust links, which means personal influence can be calculated more accurately in this dataset.

6.4 System Architecture and Potential Applications

Aiming to illustrate how to make group recommendations, we design an architecture of the proposed trust-aware group recommender system shown in Figure 6.8. There are three components in our designed system, including a system interface, a data server and a group recommender engine.

In the system interface, users in each group can interact with the

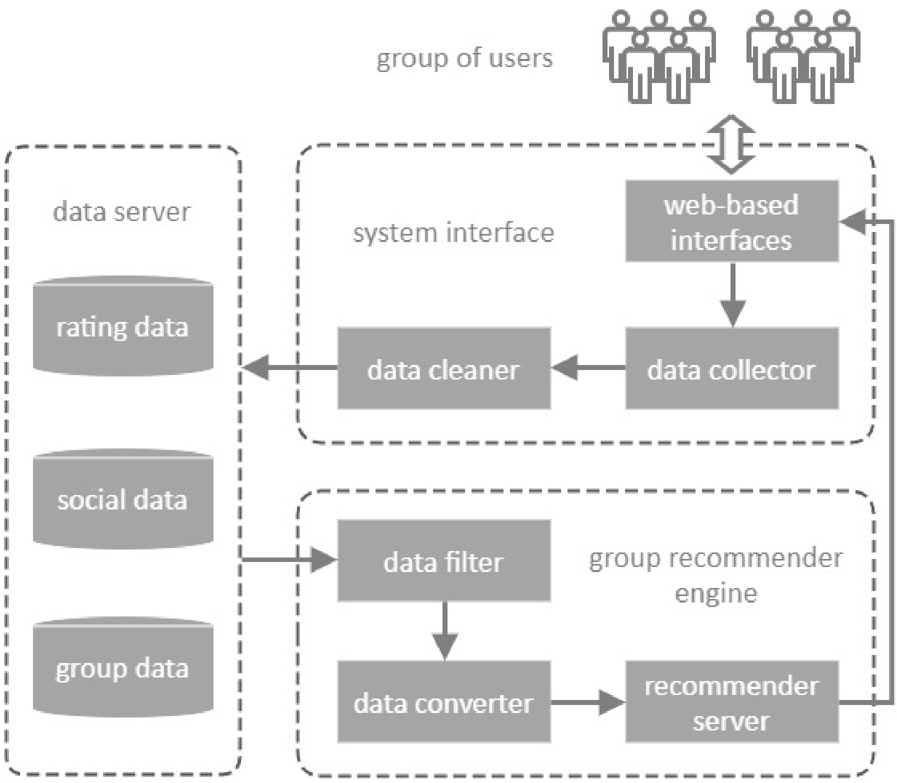


Figure 6.8: The architecture of our designed trust-aware group recommender system. This system includes three components, i.e., a system interface, a data server and a group recommender engine.

system through web-based interfaces, e.g., websites. Each user's behaviors on websites indicate user's personalized preferences, such as rating items and building friendship with other users. A data collector extracts user preference data and historical browsing information from web-based interfaces, and then a data cleanser is arranged to find out useful data and transform it into structural data, e.g., XML.

The data server is responsible for storing user preference data and divides data into three categories that include rating data, social data and group data. Note that, the data server may obtain user preference data from the system interface more than once in order to achieve comprehensive user preferences.

User preference data is regarded as input for the group recommender engine from the data server. A data filter is deployed to eliminate negative feedback in rating data, because we only consider positive feedback in the proposed TruGRC method. Rating data is transformed to rating vectors by an data converter, and then transfers into the recommender server. The proposed TruGRC method is applied in a recommender server which models overall group preferences and makes group recommendations with integrated rating, social and group data. Finally, group recommendation results are reported to web-based interfaces where users can view them.

The trust-aware group recommender system can be used to generate recommendations in several group activity scenarios.

1) Tourism recommendation is a typical application scenario of group recommender systems. All members in a touring party only can select a series of unique destinations.

2) Restaurant recommendation. A group of users only can choose one restaurant for dining, which accords with group recommendation situations.

3) Movie recommendation. If some friends want to see a movie together, these friends are regarded as a specific group and a group recommender system can be used to make recommendations for them.

6.5 Summary

Group recommendation is a significant issue in many domains based on group activities. This chapter proposes a trust-aware group recommendation method, called TruGRC, that integrates benefits from both the result and the profile aggregations to increase the satisfaction rate of group recommendation. A virtual coordinator provides a global view of the overall preferences for a group of users. This coordinator interacts with each group member to relieve conflicting personal preferences within the group. The explicit trust relations within social networks are leveraged to calculate the personal influence of group members, which is then used to model the impact between each group member and the virtual coordinator. With this information, the virtual coordinator can easily generate group preferences using the average aggregation method. The results from experiments on two benchmark datasets indicate TruGRC outperforms the baselines in terms of most metrics with a range of different group sizes.

CHAPTER 7

CONCLUSION AND FUTURE STUDY

7.1 Conclusion

Internet technology has completely changed people's daily lives. Internet information platforms, such as e-commerce websites and social networks have risen rapidly, launching people into the information era. However, with the growth in data, problems with information overload have become increasingly serious. As an effective tool for solving information overload, recommender systems have received extensive attention from both academia and industry. Over the past two decades, research into recommender systems has seen significant progress, and many e-commerce websites have installed recommender systems.

However, among the challenges facing recommender systems today, user behaviors on the Internet have become more diverse, such that it is now much harder to accurately capture user preferences. Additionally, some challenges, such as data sparsity and the accuracy-diversity dilemma, are still hindering the develop of recommender systems. This thesis is motivated by the challenges and practical new trends of recommender systems. To this end, this thesis presents several novel recommendation methods based on diffusion dynamics and machine learning for both individual and group recommendation. The main contributions of this study are summarized as follows.

1) To meet RO1, this study proposes a mixed similarity diffusion model called MSD based on a bipartite network that forces a balance between accuracy and diversity in the recommendations generated.

A two-step resource-allocation process integrates both cosine similarity with explicit feedback and a resource-allocation index with implicit feedback into the diffusion process. Further, the approach reveals the impact of node degrees in a bipartite network, which has a significant influence on the accuracy and diversity of the recommendation results. A parameter-based method is developed to control the impact of the diffusion process and strike a balance between accuracy and diversity. Extensive experiments with real-world datasets show that MSD simultaneously enhances both accuracy and diversity.

2) To meet RO2, a diffusion-based recommendation method called DBRT is developed. The method works with trust relations on a tripartite

network to introduce social information into the diffusion process.

The resource-allocation process is extended from a bipartite network to a tripartite network and the trust diffusion process is simulated through a user-user trust network to introduce explicit trust relations into the resource-allocation process. In addition, implicit trust between users is calculated according to a cosine index on the assumption that users are likely to trust other users if they are highly similar. Experiments show that considering both explicit and implicit trust in the diffusion process can improve the performance of recommendations with social networks.

3) To meet RO3, a model-based recommendation method called REOD is designed. REOD applies opinion dynamics to improve matrix factorization for social recommendation tasks.

Both the dynamic processes of real society and the rating predictions of recommender systems are considered in the framework. The impact of neighbors on user opinions is characterized by evolutionary game theory and the payoffs of strategies during an interaction are associated with latent item factors and observed ratings. Users update their opinions according to the payoff matrix of the game during matrix factorization training. When users make decisions on items, they are affected by others, so the opinions of others contribute to the ratings. Experiments reveal that incorporating opinion dynamics into the social physics of recommender systems can improve performance.

4) To meet RO4, a trust-aware group recommendation model called

TruGRC is devised. TruGRC introduces the concept of virtual coordinators to relieve the issue of preference conflicts in group recommender systems.

The framework's design takes advantages the benefits of both result and profile aggregation strategies. Additionally, it considers the process of group recommendation as a negotiation in which every member of the group hopes the group's preferences will match their own personal preferences as much as possible. Thus, the virtual coordinator provides a global view of all user preferences and harmonizes their benefits by negotiating with them. Extensive experiments show that the balanced recommendation results created by the TruGRC can meet most user requirements at a range of group sizes.

7.2 Future Study

Although this thesis contains several contributions to the advancement of recommender systems, there are still many improvements that need to be made in special recommendation scenarios. Hence, the following research directions would serve as worthwhile future work.

1) Recommendation with deep learning. Deep learning, one of the most popular recent directions in computer science, has had a disruptive impact on machine learning. Deep learning has some advantages when it comes to feature representation in that it can extract knowledge from features at a higher level and with greater dimensionality. Previously, applying

deep learning to recommender systems has been limited by a lack of data. However, as e-commerce sites and social networks advance, increases in the amount of data available are providing opportunities to apply deep learning to recommender systems. By leveraging the outstanding feature representation capabilities of deep learning, it may be possible to drastically improve recommender systems.

2) Recommendation with transfer learning. Transfer learning is a way to improve learning tasks by transferring knowledge from a related domain(s) to a target domain. As mentioned, data sparsity is a long-term problem in recommender systems. However, transfer learning could provide an effective solution for alleviating this problem. In addition, e-commerce applications, such as Douban, include multiple domains, where users can interact in many fields at the same time, including movies, books and music. Through transfer learning, behavioral user data could be gathered from multiple domains, and knowledge could be extracted to complete recommendation tasks in the target domain.

3) Recommendation with the features of time series. User preferences inevitably change over time. Hence, time series features could be introduced into recommender systems to model patterns of user preference drift. Moreover, in some scenarios, such as streaming media and web sessions, time series features are a crucial part of the streaming or session data. As a result, it would be meaningful to consider the features of time series in recommender systems.

Bibliography

Abolghasemi, R., Engelstad, P., Herrera-Viedma, E. & Yazidi, A., 2022, 'A personality-aware group recommendation system based on pairwise preferences', *Information Sciences*, vol. 595, pp. 1–17.

Adomavicius, G. & Tuzhilin, A., 2005, 'Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions', *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749.

Aggarwal, C. C., 2016, 'Neighborhood-based collaborative filtering', *Recommender Systems*, Springer, pp. 29–70.

Ahmadian, S., Ahmadian, M. & Jalili, M., 2022, 'A deep learning based trust-and tag-aware recommender system', *Neurocomputing*, vol. 488, pp. 557–571.

Ahn, H. J., 2008, 'A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem', *Information Sciences*, vol. 178, no. 1, pp. 37–51.

Ali, Z., Kefalas, P., Muhammad, K., Ali, B. & Imran, M., 2020, 'Deep learning in citation recommendation models survey', *Expert Systems with Applications*, vol. 162, p. 113790.

Amatriain, X., 2013, 'Big & personal: data and models behind netflix recommendations', *Proceedings of the 2nd International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications*, pp. 1–6.

Amatriain, X. & Basilico, J., 2015, 'Recommender systems in industry: A netflix case study', *Recommender Systems Handbook*, Springer, pp. 385–419.

Amer-Yahia, S., Roy, S. B., Chawlat, A., Das, G. & Yu, C., 2009, 'Group recommendation: Semantics and efficiency', *Proceedings of the VLDB Endowment*, vol. 2, no. 1, pp. 754–765.

Anagnostopoulos, A., Atassi, R., Becchetti, L., Fazzone, A. & Silvestri, F., 2017, 'Tour recommendation for groups', *Data Mining and Knowledge Discovery*, vol. 31, no. 5, pp. 1157–1188.

Arampatzis, A. & Kalamatianos, G., 2017, 'Suggesting points-of-interest via content-based, collaborative, and hybrid fusion methods in mobile devices', *ACM Transactions on Information Systems*, vol. 36, no. 3, pp. 1–28.

Bakos, Y. & Katsamakas, E., 2008, 'Design and ownership of two-sided networks: Implications for internet platforms', *Journal of Management Information Systems*, vol. 25, no. 2, pp. 171–202.

Balabanović, M. & Shoham, Y., 1997, 'Fab: Content-based, collaborative recommendation', *Communications of the ACM*, vol. 40, no. 3, pp. 66–72.

Baltrunas, L., Makcinskas, T. & Ricci, F., 2010, 'Group recommendations with rank aggregation and collaborative filtering', *Proceedings of the 4th ACM Conference on Recommender Systems*, pp. 119–126.

Basilico, J. & Hofmann, T., 2004, 'Unifying collaborative and content-based filtering', *Proceedings of the 21st International Conference on Machine Learning*, p. 9.

Bennett, J., Lanning, S. et al., 2007, 'The netflix prize', *Proceedings of the 2007 KDD Cup and Workshop*, p. 35.

Bobadilla, J., González-Prieto, Á., Ortega, F. & Lara-Cabrera, R., 2021, 'Deep learning feature selection to unhide demographic recommender systems factors', *Neural Computing and Applications*, vol. 33, no. 12, pp. 7291–7308.

Bobadilla, J., Hernando, A., Ortega, F. & Bernal, J., 2011, 'A framework for collaborative filtering recommender systems', *Expert Systems with Applications*, vol. 38, no. 12, pp. 14609–14623.

Bobadilla, J., Ortega, F., Hernando, A. & Bernal, J., 2012, 'A collaborative filtering approach to mitigate the new user cold start problem', *Knowledge-Based Systems*, vol. 26, pp. 225–238.

Bobadilla, J., Ortega, F., Hernando, A. & Gutiérrez, A., 2013, 'Recommender systems survey', *Knowledge-Based Systems*, vol. 46, pp. 109–132.

Bobadilla, J., Serradilla, F. & Bernal, J., 2010, 'A new collaborative filtering metric that improves the behavior of recommender systems', *Knowledge-Based Systems*, vol. 23, no. 6, pp. 520–528.

Boratto, L., Carta, S. & Fenu, G., 2016, 'Discovery and representation of the preferences of automatically detected groups: Exploiting the link between group modeling and clustering', *Future Generation Computer Systems*, vol. 64, pp. 165–174.

Cantador, I., Bellogín, A. & Vallet, D., 2010, 'Content-based recommendation in social tagging systems', *Proceedings of the 4th ACM Conference on Recommender Systems*, pp. 237–240.

Cao, L. & Li, X., 2008, 'Mixed evolutionary strategies imply coexisting opinions on networks', *Physical Review E*, vol. 77, no. 1, p. 016108.

Castro, J., Lu, J., Zhang, G., Dong, Y. & Martínez, L., 2017a, 'Opinion dynamics-based group recommender systems', *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 12, pp. 2394–2406.

Castro, J., Quesada, F. J., Palomares, I. & Martínez, L., 2015, 'A consensus-driven group recommender system', *International Journal of Intelligent Systems*, vol. 30, no. 8, pp. 887–906.

Castro, J., Yera, R. & Martínez, L., 2017b, 'An empirical study of natural noise management in group recommendation systems', *Decision Support Systems*, vol. 94, pp. 1–11.

Castro, J., Yera, R. & Martinez, L., 2018, 'A fuzzy approach for natural noise management in group recommender systems', *Expert Systems with Applications*, vol. 94, pp. 237–249.

Chen, G., Gao, T., Zhu, X., Tian, H. & Yang, Z., 2017a, 'Personalized recommendation based on preferential bidirectional mass diffusion', *Physica A: Statistical Mechanics and its Applications*, vol. 469, pp. 397–404.

Chen, L.-J., Zhang, Z.-K., Liu, J.-H., Gao, J. & Zhou, T., 2017b, 'A vertex similarity index for better personalized recommendation', *Physica A: Statistical Mechanics and its Applications*, vol. 466, pp. 607–615.

Chen, S. & Peng, Y., 2018, 'Matrix factorization for recommendation with explicit and implicit feedback', *Knowledge-Based Systems*, vol. 158, pp. 109–117.

Chen, T., Yin, H., Nguyen, Q. V. H., Peng, W.-C., Li, X. & Zhou, X., 2020a, 'Sequence-aware factorization machines for temporal predictive analytics', *Proceedings of the 36th IEEE International Conference on Data Engineering*, pp. 1405–1416.

Chen, X., Li, L., Pan, W. & Ming, Z., 2020b, 'A survey on heterogeneous one-class collaborative filtering', *ACM Transactions on Information Systems*, vol. 38, no. 4, pp. 1–54.

Da'u, A. & Salim, N., 2020, 'Recommendation system based on deep learning methods: A systematic review and new directions', *Artificial Intelligence Review*, vol. 53, no. 4, pp. 2709–2748.

Da'u, A., Salim, N. & Idris, R., 2021, 'An adaptive deep learning method for item recommendation system', *Knowledge-Based Systems*, vol. 213, p. 106681.

De Campos, L. M., Fernández-Luna, J. M., Huete, J. F. & Rueda-Morales, M. A., 2010, 'Combining content-based and collaborative recommendations: A hybrid approach based on bayesian networks', *International Journal of Approximate Reasoning*, vol. 51, no. 7, pp. 785–799.

Deffuant, G., Neau, D., Amblard, F. & Weisbuch, G., 2000, 'Mixing beliefs among interacting agents', *Advances in Complex Systems*, vol. 3, no. 01n04, pp. 87–98.

Dong, M., Yuan, F., Yao, L., Wang, X., Xu, X. & Zhu, L., 2022, 'A survey for trust-aware recommender systems: A deep learning perspective', *Knowledge-Based Systems*, vol. 249, p. 108954.

Dwivedi, P. & Bharadwaj, K. K., 2015, 'e-learning recommender system for a group of learners based on the unified learner profile approach', *Expert Systems*, vol. 32, no. 2, pp. 264–276.

Eirinaki, M., Louta, M. D. & Varlamis, I., 2014, 'A trust-aware system for personalized user recommendations in social networks', *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 4, pp. 409–421.

Ekstrand, M. D., Riedl, J. T., Konstan, J. A. et al., 2011, 'Collaborative filtering recommender systems', *Foundations and Trends in Human–Computer Interaction*, vol. 4, no. 2, pp. 81–173.

Ghorab, M. R., Zhou, D., O‚Äôconnor, A. & Wade, V., 2013, 'Personalised information retrieval: Survey and classification', *User Modeling and User-Adapted Interaction*, vol. 23, no. 4, pp. 381–443.

Gogna, A. & Majumdar, A., 2015, 'A comprehensive recommender system model: Improving accuracy for both warm and cold start users', *IEEE Access*, vol. 3, pp. 2803–2813.

Grigull, U. & Sandner, H., 1984, *Heat Conduction*, Springer.

Guo, G., Qiu, H., Tan, Z., Liu, Y., Ma, J. & Wang, X., 2017a, 'Resolving data sparsity by multi-type auxiliary implicit feedback for recommender systems', *Knowledge-Based Systems*, vol. 138, pp. 202–207.

Guo, G., Zhang, J. & Thalmann, D., 2014, 'Merging trust in collaborative filtering to alleviate data sparsity and cold start', *Knowledge-Based Systems*, vol. 57, pp. 57–68.

Guo, G., Zhang, J. & Yorke-Smith, N., 2016a, 'A novel recommendation model regularized with user trust and item ratings', *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 7, pp. 1607–1620.

Guo, G., Zhang, J., Zhu, F. & Wang, X., 2017b, 'Factored similarity models with social trust for top-n item recommendation', *Knowledge-Based Systems*, vol. 122, pp. 17–25.

Guo, H., Tang, R., Ye, Y., Li, Z. & He, X., 2017c, 'Deepfm: A factorization-machine based neural network for ctr prediction', *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 1725–1731.

Guo, J., Zhu, Y., Li, A., Wang, Q. & Han, W., 2016b, 'A social influence approach for group user modeling in group recommendation systems', *IEEE Intelligent Systems*, vol. 31, no. 5, pp. 40–48.

Guo, W., Wu, S., Wang, L. & Tan, T., 2016c, 'Personalized ranking with pairwise factorization machines', *Neurocomputing*, vol. 214, pp. 191–200.

Gupta, S., Wang, H., Lipton, Z. & Wang, Y., 2021, 'Correcting exposure bias for link recommendation', *Proceedings of the 38th International Conference on Machine Learning*, pp. 3953–3963.

He, X. & Chua, T.-S., 2017, 'Neural factorization machines for sparse predictive analytics', *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 355–364.

He, X., Liao, L., Zhang, H., Nie, L., Hu, X. & Chua, T.-S., 2017, 'Neural collaborative filtering', *Proceedings of the 26th International Conference on World Wide Web*, pp. 173–182.

He, X., Zhang, H., Kan, M.-Y. & Chua, T.-S., 2016, 'Fast matrix factorization for online recommendation with implicit feedback', *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 549–558.

Herlocker, J. L., Konstan, J. A., Borchers, A. & Riedl, J., 1999, 'An algorithmic framework for performing collaborative filtering', *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 230–237.

Hu, G., Zhang, Y. & Yang, Q., 2018, 'Conet: Collaborative cross networks for cross-domain recommendation', *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pp. 667–676.

Hu, J., Wang, B., Liu, Y. & Li, D.-Y., 2012, 'Personalized tag recommendation using social influence', *Journal of Computer Science and Technology*, vol. 27, no. 3, pp. 527–540.

Hu, Y., Koren, Y. & Volinsky, C., 2008, 'Collaborative filtering for implicit feedback datasets', *Proceedings of the 8th IEEE International Conference on Data Mining*, pp. 263–272.

Hu, Y., Song, R. J. & Chen, M., 2017, 'Modeling for information diffusion in online social networks via hydrodynamics', *IEEE Access*, vol. 5, pp. 128–135.

Hu, Y., Xiong, F., Pan, S., Xiong, X., Wang, L. & Chen, H., 2021, 'Bayesian personalized ranking based on multiple-layer neighborhoods', *Information Sciences*, vol. 542, pp. 156–176.

Hyun, D., Park, C., Cho, J. & Yu, H., 2021, 'Learning to utilize auxiliary reviews for recommendation', *Information Sciences*, vol. 545, pp. 595–607.

Islam, M. A., Mohammad, M. M., Das, S. S. S. & Ali, M. E., 2022, 'A survey on deep learning based point-of-interest (POI) recommendations', *Neurocomputing*, vol. 472, pp. 306–325.

Jamali, M. & Ester, M., 2010, 'A matrix factorization technique with trust propagation for recommendation in social networks', *Proceedings of the 4th ACM Conference on Recommender Systems*, pp. 135–142.

Jannach, D., Manzoor, A., Cai, W. & Chen, L., 2021, 'A survey on conversational recommender systems', *ACM Computing Surveys*, vol. 54, no. 5, pp. 1–36.

Ji, K., Chen, Z., Sun, R., Ma, K., Yuan, Z. & Xu, G., 2018, 'Gist: A generative model with individual and subgroup-based topics for group recommendation', *Expert Systems with Applications*, vol. 94, pp. 81–93.

Ji, K., Sun, R., Li, X. & Shu, W., 2016, 'Improving matrix approximation for recommendation via a clustering-based reconstructive method', *Neurocomputing*, vol. 173, pp. 912–920.

Ji, Z., Wu, M., Yang, H. & Íñigo, J. E. A., 2021, 'Temporal sensitive heterogeneous graph neural network for news recommendation', *Future Generation Computer Systems*, vol. 125, pp. 324–333.

Jia, C.-X., Liu, R.-R., Sun, D. & Wang, B.-H., 2008, 'A new weighting method in network-based recommendation', *Physica A: Statistical Mechanics and its Applications*, vol. 387, no. 23, pp. 5887–5891.

Jiang, C., Chen, Y. & Liu, K. R., 2014a, 'Evolutionary dynamics of information diffusion over social networks', *IEEE Transactions on Signal Processing*, vol. 62, no. 17, pp. 4573–4586.

Jiang, C., Chen, Y. & Liu, K. R., 2014b, 'Graphical evolutionary game for information diffusion over social networks', *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 4, pp. 524–536.

Jiang, P., Zhu, Y., Zhang, Y. & Yuan, Q., 2015, 'Life-stage prediction for product recommendation in e-commerce', *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1879–‚Äì1888.

Juan, Y., Zhuang, Y., Chin, W.-S. & Lin, C.-J., 2016, 'Field-aware factorization machines for ctr prediction', *Proceedings of the 10th ACM Conference on Recommender Systems*, pp. 43–50.

Kabbur, S., Ning, X. & Karypis, G., 2013, 'Fism: Factored item similarity models for top-n recommender systems', *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 659–667.

Kagita, V. R., Pujari, A. K. & Padmanabhan, V., 2015, 'Virtual user approach for group recommender systems using precedence relations', *Information Sciences*, vol. 294, pp. 15–30.

Karatzoglou, A. & Hidasi, B., 2017, 'Deep learning for recommender systems', *Proceedings of the 11th ACM Conference on Recommender Systems*, pp. 396–397.

Kaššák, O., Kompan, M. & Bieliková, M., 2016, 'Personalized hybrid recommendation for group of users: Top-n multimedia recommender', *Information Processing & Management*, vol. 52, no. 3, pp. 459–477.

Khan, Z. Y., Niu, Z., Sandiwarno, S. & Prince, R., 2021, 'Deep learning techniques for rating prediction: A survey of the state-of-the-art', *Artificial Intelligence Review*, vol. 54, no. 1, pp. 95–135.

Knoll, J., 2016, 'Recommending with higher-order factorization machines', *Proceedings of the 2016 International Conference on Innovative Techniques and Applications of Artificial Intelligence*, Springer, pp. 103–116.

Kompan, M. & Bieliková, M., 2010, 'Content-based news recommendation', *Proceedings of the 11th International Conference on Electronic Commerce and Web Technologies*, pp. 61–72.

Koren, Y., 2008, 'Factorization meets the neighborhood: A multifaceted collaborative filtering model', *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 426–434.

Koren, Y., 2009, 'Collaborative filtering with temporal dynamics', *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 447–456.

Koren, Y., 2010, 'Factor in the neighbors: Scalable and accurate collaborative filtering', *ACM Transactions on Knowledge Discovery from Data*, vol. 4, no. 1, p. 1.

Koren, Y., Bell, R. & Volinsky, C., 2009, 'Matrix factorization techniques for recommender systems', *Computer*, vol. 42, no. 8.

Koskinen, J. & Edling, C., 2012, 'Modelling the evolution of a bipartite network,Äîpeer referral in interlocking directorates', *Social Networks*, vol. 34, no. 3, pp. 309–322.

Langville, A. N. & Meyer, C. D., 2011, *Google's PageRank and Beyond: The Science of Search Engine Rankings*, Princeton University Press.

Lee, J., Kim, S., Lebanon, G., Singer, Y. & Bengio, S., 2016, 'Llorma: Local low-rank matrix approximation', *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 442–465.

Leng, Y. & Yu, L., 2022, 'Incorporating global and local social networks for group recommendations', *Pattern Recognition*, vol. 127, p. 108601.

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J. & Liu, H., 2017a, 'Feature selection: A data perspective', *ACM Computing Surveys*, vol. 50, no. 6, pp. 1–45.

Li, L., Do, Q. & Liu, W., 2019a, 'Cross-domain recommendation via coupled factorization machines', *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, pp. 9965–9966.

Li, L., Scaglione, A., Swami, A. & Zhao, Q., 2013, 'Consensus, polarization and clustering of opinions in social networks', *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 6, pp. 1072–1083.

Li, Q. & Kim, B. M., 2003, 'An approach for combining content-based and collaborative filters', *Proceedings of the 6th International Workshop on Information Retrieval with Asian Languages*, pp. 17–24.

Li, W.-J., Dong, Q., Shi, Y.-B., Fu, Y. & He, J.-L., 2017b, 'Effect of recent popularity on heat-conduction based recommendation models', *Physica A: Statistical Mechanics and its Applications*, vol. 474, pp. 334–343.

Li, Z., Xiong, F., Wang, X., Chen, H. & Xiong, X., 2019b, 'Topological influence-aware recommendation on social networks', *Complexity*, vol. 2019.

Liang, T., Ma, L., Zhang, W., Xu, H., Xia, C. & Yin, Y., 2022, 'Content-aware recommendation via dynamic heterogeneous graph convolutional network', *Knowledge-Based Systems*, vol. 251, p. 109185.

Liu, H., Kong, X., Bai, X., Wang, W., Bekele, T. M. & Xia, F., 2015, 'Context-based collaborative filtering for citation recommendation', *IEEE Access*, vol. 3, pp. 1695–1703.

Liu, H., Lu, J., Yang, H., Zhao, X., Xu, S., Peng, H., Zhang, Z., Niu, W., Zhu, X., Bao, Y. & Yan, W., 2020, 'Category-specific CNN for visual-aware CTR prediction at JD.com', *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 2686–2696.

Liu, J.-G., Zhou, T. & Guo, Q., 2011, 'Information filtering via biased heat conduction', *Physical Review E*, vol. 84, no. 3, p. 037101.

Liu, M., Pan, W., Liu, M., Chen, Y., Peng, X. & Ming, Z., 2017, 'Mixed similarity learning for recommendation with implicit feedback', *Knowledge-Based Systems*, vol. 119, pp. 178–185.

Liu, R.-R., Jia, C.-X., Zhou, T., Sun, D. & Wang, B.-H., 2009, 'Personal recommendation via modified collaborative filtering', *Physica A: Statistical Mechanics and its Applications*, vol. 388, no. 4, pp. 462–468.

Liu, Y. & Wu, A.-b., 2021, 'POI recommendation method using deep learning in location-based social networks', *Wireless Communications and Mobile Computing*, vol. 2021, p. 9120864.

Loni, B., Pagano, R., Larson, M. & Hanjalic, A., 2016, 'Bayesian personalized ranking with multi-channel user feedback', *Proceedings of the 10th ACM Conference on Recommender Systems*, pp. 361–364.

Loni, B., Shi, Y., Larson, M. & Hanjalic, A., 2014, 'Cross-domain collaborative filtering with factorization machines', *Proceedings of the 2014 European Conference on Information Retrieval*, Springer, pp. 656–661.

Lops, P., Gemmis, M. d. & Semeraro, G., 2011, 'Content-based recommender systems: State of the art and trends', *Recommender Systems Handbook*, pp. 73–105.

Lops, P., Jannach, D., Musto, C., Bogers, T. & Koolen, M., 2019, 'Trends in content-based recommendation', *User Modeling and User-Adapted Interaction*, vol. 29, no. 2, pp. 239–249.

Lu, J., Shambour, Q., Xu, Y., Lin, Q. & Zhang, G., 2010, 'Bizseeker: A hybrid semantic recommendation system for personalized government-to-business e-services', *Internet Research*, vol. 20, no. 3, pp. 342–365.

Lu, J., Shambour, Q., Xu, Y., Lin, Q. & Zhang, G., 2013, 'A web-based personalized business partner recommendation system using fuzzy semantic techniques', *Computational Intelligence*, vol. 29, no. 1, pp. 37–69.

Lu, J., Wu, D., Mao, M., Wang, W. & Zhang, G., 2015, 'Recommender system application developments: A survey', *Decision Support Systems*, vol. 74, pp. 12–32.

Lü, L., Chen, D., Ren, X.-L., Zhang, Q.-M., Zhang, Y.-C. & Zhou, T., 2016, 'Vital nodes identification in complex networks', *Physics Reports*, vol. 650, pp. 1–63.

Lü, L. & Liu, W., 2011, 'Information filtering via preferential diffusion', *Physical Review E*, vol. 83, no. 6, p. 066119.

Lü, L., Medo, M., Yeung, C. H., Zhang, Y.-C., Zhang, Z.-K. & Zhou, T., 2012, 'Recommender systems', *Physics Reports*, vol. 519, no. 1, pp. 1–49.

Lü, L. & Zhou, T., 2011, 'Link prediction in complex networks: A survey', *Physica A: Statistical Mechanics and its Applications*, vol. 390, no. 6, pp. 1150–1170.

Lu, Z., Wen, Y., Zhang, W., Zheng, Q. & Cao, G., 2016, 'Towards information diffusion in mobile social networks', *IEEE Transactions on Mobile Computing*, vol. 15, no. 5, pp. 1292–1304.

Luo, W., Tay, W.-P. & Leng, M., 2016, 'Infection spreading and source identification: A hide and seek game.', *IEEE Transactions on Signal Processing*, vol. 64, no. 16, pp. 4228–4243.

Ma, H., 2013, 'An experimental study on implicit social recommendation', *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 73–82.

Ma, H., King, I. & Lyu, M. R., 2009, 'Learning to recommend with social trust ensemble', *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Iinformation Retrieval*, pp. 203–210.

Ma, H., Yang, H., Lyu, M. R. & King, I., 2008, 'Sorec: social recommendation using probabilistic matrix factorization', *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pp. 931–940.

Ma, H., Zhou, D., Liu, C., Lyu, M. R. & King, I., 2011, 'Recommender systems with social regularization', *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*, pp. 287–296.

Ma, R. T., Lui, J. & Misra, V., 2015, 'Evolution of the internet economic ecosystem', *IEEE/ACM Transactions on Networking*, vol. 23, no. 1, pp. 85–98.

Mahmood, T. & Ricci, F., 2009, 'Improving recommender systems with adaptive conversational strategies', *Proceedings of the 20th ACM conference on Hypertext and hypermedia*, pp. 73–82.

Mao, M., Lu, J., Zhang, G. & Zhang, J., 2017, 'Multirelational social recommendations via multigraph ranking', *IEEE transactions on cybernetics*, vol. 47, no. 12, pp. 4049–4061.

Martins, A. C., 2008, 'Continuous opinions and discrete actions in opinion dynamics problems', *International Journal of Modern Physics C*, vol. 19, no. 04, pp. 617–624.

Miao, H., Li, A. & Yang, B., 2022, 'Meta-path enhanced lightweight graph neural network for social recommendation', *Proceedings of the 27th International Conference on Database Systems for Advanced Applications*, pp. 134–149.

Mnih, A. & Salakhutdinov, R. R., 2007, 'Probabilistic matrix factorization', *Proceedings of the 20th International Conference on Neural Information Processing Systems*, pp. 1257–1264.

Moffat, A. & Zobel, J., 2008, 'Rank-biased precision for measurement of retrieval effectiveness', *ACM Transactions on Information Systems*, vol. 27, no. 1, pp. 2–27.

Morita, M. & Shinoda, Y., 1994, 'Information filtering based on user behavior analysis and best match text retrieval', *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 272–281.

Ni, J., Huang, Z., Hu, Y. & Lin, C., 2022a, 'A two-stage embedding model for recommendation with multimodal auxiliary information', *Information Sciences*, vol. 582, pp. 22–37.

Ni, X., Xiong, F., Hu, Y., Pan, S., Chen, H. & Wang, L., 2022b, 'Cyclic transfer learning for recommender systems with heterogeneous feedbacks', *Proceedings of the 2022 SIAM International Conference on Data Mining*, pp. 567–575.

Ok, J., Jin, Y., Shin, J. & Yi, Y., 2016, 'On maximizing diffusion speed over social networks with strategic users', *IEEE/ACM Transactions on Networking*, vol. 24, no. 6, pp. 3798–3811.

Oord, A. v. d., Dieleman, S. & Schrauwen, B., 2013, 'Deep content-based music recommendation', *Proceedings of the 26th International Conference on Neural Information Processing*, pp. 2643–2651.

Ortega, F., Hernando, A., Bobadilla, J. & Kang, J. H., 2016, 'Recommending items to group of users using matrix factorization based collaborative filtering', *Information Sciences*, vol. 345, pp. 313–324.

Pan, S., Wu, J., Zhu, X., Long, G. & Zhang, C., 2017, 'Task sensitive feature exploration and learning for multitask graph classification', *IEEE Transactions on Cybernetics*, vol. 47, no. 3, pp. 744–758.

Pan, S., Wu, J., Zhu, X., Zhang, C. & Wang, Y., 2016a, 'Tri-party deep network representation', *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, pp. 1895–1901.

Pan, W., Chen, L. & Ming, Z., 2019, 'Personalized recommendation with implicit feedback via learning pairwise preferences over item-sets', *Knowledge and Information Systems*, vol. 58, no. 2, pp. 295–318.

Pan, W., Liu, M. & Ming, Z., 2016b, 'Transfer learning for heterogeneous one-class collaborative filtering', *IEEE Intelligent Systems*, vol. 31, no. 4, pp. 43–49.

Pan, W., Liu, Z., Ming, Z., Zhong, H., Wang, X. & Xu, C., 2015a, 'Compressed knowledge transfer via factorization machine for heterogeneous collaborative recommendation', *Knowledge-Based Systems*, vol. 85, pp. 234–244.

Pan, W. & Ming, Z., 2017, 'Collaborative recommendation with multiclass preference context', *IEEE Intelligent Systems*, vol. 32, no. 2, pp. 45–51.

Pan, W., Zhong, H., Xu, C. & Ming, Z., 2015b, 'Adaptive bayesian personalized ranking for heterogeneous implicit feedbacks', *Knowledge-Based Systems*, vol. 73, pp. 173–180.

Patra, B. K., Launonen, R., Ollikainen, V. & Nandi, S., 2015, 'A new similarity measure using bhattacharyya coefficient for collaborative filtering in sparse data', *Knowledge-Based Systems*, vol. 82, pp. 163–177.

Pérez-Almaguer, Y., Yera, R., Alzahrani, A. A. & Martínez, L., 2021, 'Content-based group recommender systems: A general taxonomy and further improvements', *Expert Systems with Applications*, vol. 184, p. 115444.

Portugal, I., Alencar, P. & Cowan, D., 2018, 'The use of machine learning algorithms in recommender systems: A systematic review', *Expert Systems with Applications*, vol. 97, pp. 205–227.

Qiu, T., Han, T.-Y., Zhong, L.-X., Zhang, Z.-K. & Chen, G., 2014, 'Redundant correlation effect on personalized recommendation', *Computer Physics Communications*, vol. 185, no. 2, pp. 489–494.

Quijano-Sanchez, L., Recio-Garcia, J. A., Diaz-Agudo, B. & Jimenez-Diaz, G., 2013, 'Social factors in group recommender systems', *ACM Transactions on Intelligent Systems and Technology*, vol. 4, no. 1, p. 8.

Quijano-Sanchez, L., Sauer, C., Recio-Garcia, J. A. & Diaz-Agudo, B., 2017, 'Make it personal: a social explanation system applied to group recommendations', *Expert Systems with Applications*, vol. 76, pp. 36–48.

Reddy, S., Nalluri, S., Kunisetti, S., Ashok, S. & Venkatesh, B., 2019, 'Content-based movie recommendation system using genre correlation', *Smart Intelligent Computing and Applications*, Springer, pp. 391–397.

Ren, L. & Wang, W., 2018, 'An svm-based collaborative filtering approach for top-n web services recommendation', *Future Generation Computer Systems*, vol. 78, pp. 531–543.

Rendle, S., 2010, 'Factorization machines', *Proceedings of the 10th IEEE International Conference on Data Mining*, pp. 995–1000.

Rendle, S., 2012, 'Factorization machines with libfm', *ACM Transactions on Intelligent Systems and Technology*, vol. 3, no. 3, pp. 1–22.

Rendle, S., Gantner, Z., Freudenthaler, C. & Schmidt-Thieme, L., 2011, 'Fast context-aware recommendations with factorization machines', *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 635–644.

Ricci, F., Rokach, L. & Shapira, B., 2011, 'Introduction to recommender systems handbook', *Recommender systems handbook*, Springer, pp. 1–35.

Ronen, R., Koenigstein, N., Ziklik, E. & Nice, N., 2013, 'Selecting content-based features for collaborative filtering recommenders', *Proceedings of the 7th ACM Conference on Recommender Systems*, pp. 407–410.

Rutkowski, A.-F. & Saunders, C. S., 2010, 'Growing pains with information overload', *Computer*, vol. 43, no. 6, pp. 96–94.

Rutkowski, T., Romanowski, J., Woldan, P., Staszewski, P., Nielek, R. & Rutkowski, L., 2018, 'A content-based recommendation system using neuro-fuzzy approach', *Proceedings of the 2018 IEEE International Conference on Fuzzy Systems*, pp. 1–8.

Salakhutdinov, R. & Mnih, A., 2008, 'Bayesian probabilistic matrix factorization using markov chain monte carlo', *Proceedings of the 25th International Conference on Machine Learning*, pp. 880–887.

Salamó, M., McCarthy, K. & Smyth, B., 2012, 'Generating recommendations for consensus negotiation in group personalization services', *Personal and Ubiquitous Computing*, vol. 16, no. 5, pp. 597–610.

Sarman, S. & Evans, D. J., 1992, 'Heat flow and mass diffusion in binary lennard-jones mixtures', *Physical Review A*, vol. 45, no. 4, p. 2370.

Sarwar, B., Karypis, G., Konstan, J. & Riedl, J., 2001, 'Item-based collaborative filtering recommendation algorithms', *Proceedings of the 10th International Conference on World Wide Web*, pp. 285–295.

Seo, Y.-D., Kim, Y.-G., Lee, E., Seol, K.-S. & Baik, D.-K., 2018, 'An enhanced aggregation method considering deviations for a group recommendation', *Expert Systems with Applications*, vol. 93, pp. 299–312.

Shang, M.-S., Zhang, Z.-K., Zhou, T. & Zhang, Y.-C., 2010, 'Collaborative filtering with diffusion-based similarity on tripartite graphs', *Physica A: Statistical Mechanics and its Applications*, vol. 389, no. 6, pp. 1259–1264.

Shi, Y., Larson, M. & Hanjalic, A., 2010, 'List-wise learning to rank with matrix factorization for collaborative filtering', *Proceedings of the 4th ACM Conference on Recommender Systems*, pp. 269–272.

Shi, Y., Larson, M. & Hanjalic, A., 2014, 'Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges', *ACM Computing Surveys*, vol. 47, no. 1, pp. 1–45.

Shu, J., Shen, X., Liu, H., Yi, B. & Zhang, Z., 2018, 'A content-based recommendation algorithm for learning resources', *Multimedia Systems*, vol. 24, no. 2, pp. 163–173.

Shuang, Z., Sheng, B. & Gengxin, S., 2019, 'Mass diffusion recommendation algorithm based on multi-subnet composited complex network model', *Proceedings of the 2019 International Conference on Artificial Intelligence and Security*, pp. 263–274.

Singh, A. P. & Gordon, G. J., 2008, 'Relational learning via collective matrix factorization', *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 650–658.

Smith, B. & Linden, G., 2017, 'Two decades of recommender systems at amazon.com', *IEEE Internet Computing*, vol. 21, no. 3, pp. 12–18.

Sun, R., Cao, X., Zhao, Y., Wan, J., Zhou, K., Zhang, F., Wang, Z. & Zheng, K., 2020, 'Multi-modal knowledge graphs for recommender systems', *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*, pp. 1405–1414.

Tahmasebi, H., Ravanmehr, R. & Mohamadrezaei, R., 2021, 'Social movie recommender system based on deep autoencoder network using Twitter data', *Neural Computing and Applications*, vol. 33, no. 5, pp. 1607–1623.

Tang, J., Aggarwal, C. & Liu, H., 2016a, 'Recommendations in signed social networks', *Proceedings of the 25th International Conference on World Wide Web*, pp. 31–40.

Tang, J., Gao, H., Sarma, A. D., Bi, Y. & Liu, H., 2015, 'Trust evolution: Modeling and its applications', *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 6, pp. 1724–1738.

Tang, J., Wang, S., Hu, X., Yin, D., Bi, Y., Chang, Y. & Liu, H., 2016b, 'Recommendation with social dimensions.', *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pp. 251–257.

Teng, X. & Gong, Y., 2018, 'Research on application of machine learning in data mining', *IOP Conference Series: Materials Science and Engineering*, p. 062202.

Thomas, K. W., 2008, 'Thomas-kilmann conflict mode', *TKI Profile and Interpretive Report*, pp. 1–11.

Thorat, P. B., Goudar, R. & Barve, S., 2015, 'Survey on collaborative filtering, content-based filtering and hybrid recommendation system', *International Journal of Computer Applications*, vol. 110, no. 4, pp. 31–36.

Verma, C., Hart, M., Bhatkar, S., Parker-Wood, A. & Dey, S., 2016, 'Improving scalability of personalized recommendation systems for enterprise knowledge workers', *IEEE Access*, vol. 4, pp. 204–215.

Wang, H., Fu, Y., Wang, Q., Yin, H., Du, C. & Xiong, H., 2017a, 'A location-sentiment-aware recommender system for both home-town and out-of-town users', *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1143.

Wang, H., Wu, J., Pan, S., Zhang, P. & Chen, L., 2017b, 'Towards large-scale social networks with online diffusion provenance detection', *Computer Networks*, vol. 114, pp. 154–166.

Wang, H., Wu, J., Zhou, C., Ji, Z. & Wu, J., 2016a, 'Mining subcascade features for cascade outbreak prediction in big networks', *Proceedings of the 29th International Joint Conference on Neural Networks*, pp. 3942–3949.

Wang, H., Zhang, P., Zhu, X., Tsang, I. W.-H., Chen, L., Zhang, C. & Wu, X., 2017c, 'Incremental subgraph feature selection for graph classification', *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 1, pp. 128–142.

Wang, J., Jiang, Y., Sun, J., Liu, Y. & Liu, X., 2018, 'Group recommendation based on a bidirectional tensor factorization model', *World Wide Web*, vol. 21, no. 4, pp. 961–984.

Wang, L., Yu, Z., Guo, B., Ku, T. & Yi, F., 2017d, 'Moving destination prediction using sparse dataset: A mobility gradient descent approach', *ACM Transactions on Knowledge Discovery from Data*, vol. 11, no. 3, p. 37.

Wang, P., Li, L., Xie, Q., Wang, R. & Xu, G., 2022, 'Social dual-effect driven group modeling for neural group recommendation', *Neurocomputing*, vol. 481, pp. 258–269.

Wang, W., Zhang, G. & Lu, J., 2016b, 'Member contribution-based group recommender system', *Decision Support Systems*, vol. 87, pp. 80–93.

Wang, X., Liu, Y. & Xiong, F., 2016c, 'Improved personalized recommendation based on a similarity network', *Physica A: Statistical Mechanics and its Applications*, vol. 456, pp. 271–280.

Wang, X., Liu, Y., Zhang, G., Xiong, F. & Lu, J., 2017e, 'Diffusion-based recommendation with trust relations on tripartite graphs', *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2017, no. 8, p. 083405.

Wang, X., Liu, Y., Zhang, G., Zhang, Y., Chen, H. & Lu, J., 2017f, 'Mixed similarity diffusion for recommendation on bipartite networks', *IEEE Access*, vol. 5, pp. 21029–21038.

Weimer, M., Karatzoglou, A. & Smola, A., 2008, 'Improving maximum margin matrix factorization', *Machine Learning*, vol. 72, no. 3, pp. 263–276.

Wu, D., Lu, J. & Zhang, G., 2015a, 'A fuzzy tree matching-based personalized e-learning recommender system', *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 6, pp. 2412–2426.

Wu, D., Zhang, G. & Lu, J., 2015b, 'A fuzzy preference tree-based recommender system for personalized business-to-business e-services', *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 1, pp. 29–43.

Xiao, X., Wen, J., Zhou, W., Luo, F., Gao, M. & Zeng, J., 2022, 'Multi-interaction fusion collaborative filtering for social recommendation', *Expert Systems with Applications*, vol. 205, p. 117610.

Xiong, F. & Li, Z.-Y., 2017, 'Effective methods of restraining diffusion in terms of epidemic dynamics', *Scientific Reports*, vol. 7, p. 6013.

Xiong, F., Liu, Y., Wang, L. & Wang, X., 2017, 'Analysis and application of opinion model with multiple topic interactions', *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 27, p. 083113.

Xiong, F., Wang, X., Pan, S., Yang, H., Wang, H. & Zhang, C., 2020, 'Social recommendation with evolutionary opinion dynamics', *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 10, pp. 3804–3816.

Xu, Z., Zhang, F., Wang, W., Liu, H. & Kong, X., 2016, 'Exploiting trust and usage context for cross-domain recommendation', *IEEE Access*, vol. 4, pp. 2398–2407.

Xuan, J., Luo, X., Zhang, G., Lu, J. & Xu, Z., 2016, 'Uncertainty analysis for the keyword system of web events', *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 6, pp. 829–842.

Xue, H.-J., Dai, X.-Y., Zhang, J., Huang, S. & Chen, J., 2017, 'Deep matrix factorization models for recommender systems', *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 3203–3209.

Yang, B., Lei, Y., Liu, J. & Li, W., 2017, 'Social collaborative filtering by trust', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 8, pp. 1633–1647.

Yang, E., Huang, Y., Liang, F., Pan, W. & Ming, Z., 2021, 'Fcmf: Federated collective matrix factorization for heterogeneous collaborative filtering', *Knowledge-Based Systems*, vol. 220, p. 106946.

Yang, J.-M. & Li, K. F., 2009, 'Recommendation based on rational inferences in collaborative filtering', *Knowledge-Based Systems*, vol. 22, no. 1, pp. 105–114.

Yao, L., Sheng, Q. Z., Ngu, A. H., Yu, J. & Segev, A., 2014, 'Unified collaborative and content-based web service recommendation', *IEEE Transactions on Services Computing*, vol. 8, no. 3, pp. 453–466.

Yin, H., Chen, L., Wang, W., Du, X., Nguyen, Q. V. H. & Zhou, X., 2017, 'Mobi-sage: A sparse additive generative model for mobile app recommendation', *Proceedings of the 33rd International Conference on Data Engineering*, pp. 75–78.

Yu, F., Zeng, A., Gillard, S. & Medo, M., 2016, 'Network-based recommendation algorithms: A review', *Physica A: Statistical Mechanics and its Applications*, vol. 452, pp. 192–208.

Yu, J., Yin, H., Li, J., Wang, Q., Hung, N. Q. V. & Zhang, X., 2021, 'Selfsupervised multi-channel hypergraph convolutional network for social recommendation', *Proceedings of the Web Conference 2021*, pp. 413–424.

Yu, X., Jiang, F., Du, J. & Gong, D., 2019, 'A cross-domain collaborative filtering algorithm with expanding user and item features via the latent factor space of auxiliary domains', *Pattern Recognition*, vol. 94, pp. 96–109.

Yu, Z., Zhou, X., Hao, Y. & Gu, J., 2006, 'Tv program recommendation for multiple viewers based on user profile merging', *User Modeling and User-Adapted Interaction*, vol. 16, no. 1, pp. 63–82.

Zeng, A., Vidmer, A., Medo, M. & Zhang, Y.-C., 2014, 'Information filtering by similarity-preferential diffusion processes', *Europhysics Letters*, vol. 105, no. 5, p. 58002.

Zhang, Q., Wu, J., Yang, H., Lu, W., Long, G. & Zhang, C., 2016a, 'Global and local influence-based social recommendation', *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pp. 1917–1920.

Zhang, Q., Wu, J., Zhang, P., Long, G., Tsang, I. W. & Zhang, C., 2016b, 'Inferring latent network from cascade data for dynamic social recommendation', *Proceedings of the 16th International Conference on Data Mining*, pp. 669–678.

Zhang, S., Yao, L., Sun, A. & Tay, Y., 2019, 'Deep learning based recommender system: A survey and new perspectives', *ACM Computing Surveys*, vol. 52, no. 1, pp. 1–38.

Zhang, W., Zhao, S., Wan, X. & Yao, Y., 2021, 'Study on the effect of digital economy on high-quality economic development in china', *PloS One*, vol. 16, no. 9, p. e0257365.

Zhang, Y., Zhang, G., Chen, H., Porter, A. L., Zhu, D. & Lu, J., 2016c, 'Topic analysis and forecasting for science, technology and innovation: Methodology with a case study focusing on big data research', *Technological Forecasting and Social Change*, vol. 105, pp. 179–191.

Zhang, Y., Zhang, G., Zhu, D. & Lu, J., 2017, 'Scientific evolutionary pathways: Identifying and visualizing relationships for scientific topics', *Journal of the Association for Information Science and Technology*, vol. 68, no. 8, pp. 1925–1939.

Zhang, Y.-C., Blattner, M. & Yu, Y.-K., 2007, 'Heat conduction process on community networks as a recommendation model', *Physical Review Letters*, vol. 99, no. 15, p. 154301.

Zhang, Z.-K., Zhou, T. & Zhang, Y.-C., 2010, 'Personalized recommendation via integrated diffusion on user–item–tag tripartite graphs', *Physica A: Statistical Mechanics and its Applications*, vol. 389, no. 1, pp. 179–186.

Zheng, D. & Xiong, Y., 2018, 'A unified probabilistic matrix factorization recommendation algorithm', *Proceedings of the 2018 International Conference on Robots and Intelligent System*, pp. 246–249.

Zhong, G., Wang, H. & Jiao, W., 2018, 'Musiccnns: A new benchmark on content-based music recommendation', *Proceedings of the 25th International Conference on Neural Information Processing*, pp. 394–405.

Zhou, R., Khemmarat, S. & Gao, L., 2010a, 'The impact of youtube recommendation system on video views', *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, pp. 404–410.

Zhou, T., Jiang, L.-L., Su, R.-Q. & Zhang, Y.-C., 2008, 'Effect of initial configuration on network-based recommendation', *Europhysics Letters*, vol. 81, no. 5, p. 58004.

Zhou, T., Kuscsik, Z., Liu, J.-G., Medo, M., Wakeling, J. R. & Zhang, Y.-C., 2010b, 'Solving the apparent diversity-accuracy dilemma of recommender systems', *Proceedings of the National Academy of Sciences*, vol. 107, no. 10, pp. 4511–4515.

Zhou, T., Ren, J., Medo, M. & Zhang, Y.-C., 2007, 'Bipartite network projection and personal recommendation', *Physical Review E*, vol. 76, no. 4, p. 046115.

Zhou, T., Su, R.-Q., Liu, R.-R., Jiang, L.-L., Wang, B.-H. & Zhang, Y.-C., 2009, ‘Accurate and diverse recommendations via eliminating redundant correlations’, *New Journal of Physics*, vol. 11, no. 12, p. 123008.

Zhou, X., Chen, L., Zhang, Y., Qin, D., Cao, L., Huang, G. & Wang, C., 2017, ‘Enhancing online video recommendation using social user interactions’, *The VLDB Journal*, vol. 26, no. 5, pp. 637–656.

Zhu, X., Tian, H. & Cai, S., 2014, ‘Personalized recommendation with corrected similarity’, *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2014, no. 7, p. P07004.

Zhu, X., Tian, H., Zhang, P., Hu, Z. & Zhou, T., 2015, ‘Personalized recommendation based on unbiased consistence’, *Europhysics Letters*, vol. 111, no. 4, p. 48007.