論文 / 著書情報
Article / Book Information

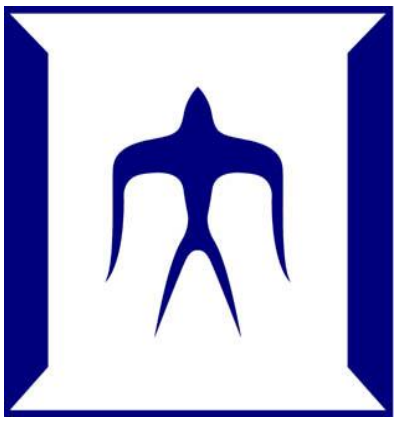
題目(和文)	利用者と項目間の影響を考慮した推薦アルゴリズムの研究
Title(English)	Study of Influence-Centric Recommendation Algorithms
著者(和文)	常娜
Author(English)	Na Chang
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:甲第9658号, 授与年月日:2014年9月25日, 学位の種別:課程博士, 審査員:寺野　隆雄,出口　弘,新田　克己,三宅　美博,小野　功
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:甲第9658号, Conferred date:2014/9/25, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

Study of Influence-Centric Recommendation Algorithms

(利用者と項目間の影響を考慮した推薦アルゴリズムの研究)

Thesis by

Na Chang



Tokyo Institute of Technology

Tokyo, Japan

Table of Contents

List of Figures

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor Prof. Takao Terano for his continuous support of my Ph.D study and research, for his patience, motivation, enthusiasm, and immense knowledge, as well as the support for me attending conferences. His guidance helped me in all the time of research and writing of this thesis.

I would also like to thank the rest of my thesis committee: Prof. Katsumi Nitta, Prof. Yoshihiro Miyake, Prof. Hiroshi Deguchi, and Prof. Isao Ono, for their insightful comments and important suggestions through the process of finishing this dissertation.

I benefit greatly from my fellow labmates from the discussions and correspondence in Terano Lab. Especially, I would like to thank Dr. Mhd Irvan, Chathura Rajapakse and Yibing Xiong for their kind advice on my research work.

I am grateful to Dr. Donald L. Amoroso, who helps me expand my research and gives me many advice not only on what to do research, but also how to do research.

Last but not the least, I would like to thank my family members to encourage and support me to continue my research. Especially, my husband not only supports me economically, but also encourages me to learn and try new things.

Finally, I want to dedicate this thesis to my daughter Nami, and wish her a peaceful, healthy and happy life.

Abstract

This thesis extends traditional user-based collaborative filtering, item-based collaborative filtering and influence-based recommendation algorithms by incorporating social impact theory and linear threshold model as well as related data mining/machine learning techniques. Depending on the prediction subjects, user influence-centric recommendation algorithm and item influence-centric recommendation algorithm are proposed respectively. In each method, users' preferences are predicted based on the direct influences of the neighborhood of users or items and the indirect influences by mining the latent attributes of neighborhood.

To be specific, firstly, the proposed methods use an unbalanced strategy to select several nearest neighbors for each user or item. Secondly, the methods measure each users' or items' influence coefficients on the other users or items, then distinguish the influences of users or items into two categories: persuasive influences and supportive influences. Next, the methods calculate the total influences and take into account its effect in the preference prediction process. Third, the methods also cope with the indirect influences exerted by the structure of users' or items' neighborhoods. Based on these processes, the methods are applicable to mining and discovering meaningful and useful latent attributes of users' or items' neighborhoods. Thus, combined with suitable data mining and/or machine learning techniques, the methods are able to learn users' or items' preferences.

Using the proposed methods, the thesis conducts experiments on two benchmark datasets: MovieLens and Netflix. The experimental results have showed that (1) the unbalanced neighbor selection method improves both the performance of traditional user-based and item-based collaborative filtering methods in terms of Root Mean Squared Error (RMSE) and accuracy, and that (2) the methods have an advantage such that it just needs fewer nearest neighbors to obtain best performance. This study has also investigated the performance of the direct influence model under the condition that the closeness threshold is satisfied. As a result, we found that there is a tradeoff between the performance and the coverage of satisfied cases. With respect to the indirect influence model, the results also have indicated that the tradeoff between the performance and the coverage requires some parameter tuning tasks. That is, the

indirect influence model is able to give much better performance when the Group Size Difference (GSD) threshold is satisfied, but the coverage of satisfied cases decreases with the performance improvements. Finally, this study presents the performance of the combination of the direct influence model and the indirect influence model. Compared with other conventional collaborative filtering algorithms, the proposed methods give better performance in terms of RMSE and accuracy.

In summary, the thesis puts forward a novel and integrated influence-centric framework, which not only predicts users' preference based on neighbors' direct influence but also takes into account indirect influence of neighborhood. Furthermore, the proposed influence-centric recommendation algorithms show better performance than conventional collaborative filtering algorithms. Benefiting the advantage of collaborative filtering, the proposed influence-centric recommendation algorithms can be applied in many areas and also can be integrated with other recommendation algorithms.

1 Introduction

1.1 Research Background

With the development of Internet and E-commerce, recommendation systems are adopted by more and more companies such as Amazon, Netflix, Yahoo music, and so on. At the same time, more and more consumers also consider recommendation systems as useful tools in their online activities such as buying something or gathering some information. As personalized information filtering systems, recommendation systems are designed to provide consumers with preferable items from a large volume of products or information such as books, movies, music, travel packages, news, and so on [1]. The aims of a recommendation system are to predict consumers' preferences based on implicit feedback, such as consumers' clicking or purchasing histories or spent time on items, or explicit feedback, such as the ratings or like/dislike given by consumers, or both of two and then to recommend and present the most favorite items which are likely to be interested by consumers [2].

From consumers' point of view, recommendation systems can help them find information or preferable products more quickly and accurately than a system without recommender. On the other hand, from businesses' point of view, they can benefit from recommendation systems in terms of revenue, attracting consumers, and so on [3]. More specifically, providers can increase their revenues by providing personalized recommendations of products or services. For example, Amazon has reported that over 20% of their sales resulted from personalized recommendations in 2002 [4]. Overall, recommendation systems have been widely applied in various domains to deal with various forms of recommendation uncertainty. In summary, recommendation systems are important in many applications, and have drawn increasing attention from different research communities such as information retrieval, electronic commerce and machine learning, as indicated by the recent emergence of series of the ACM Conference on Recommendation systems (RecSys) since 2007 and the highly publicized Netflix competition [5].

In the area of recommendation systems, recommendation algorithms are important research topics in academia. Generally speaking, recommendation systems are usually classi-

fied into the following categories, based on how recommendations are made [2]:

- Content-based recommendations: the items which are similar to the ones the user preferred in the past will be recommended[6], [7], [8];
- Collaborative filtering recommendations: the items which are preferred by the other users with similar preferences in the past will be recommended [9], [10], [11];
- Hybrid approaches: these methods combine collaborative and content-based methods [12], [13], [14].

On the basis of these traditional recommendation algorithms, many extensions have been proposed by incorporating theories or methodologies from other areas. Among these, influence-based recommendation has attracted researchers' attention recent years. Most of such research focuses on users' influence on neighborhood. For instance, researchers assume that an influential user will easily exert influence or impact on the other users and propose various methods to find influential users [15], [16], [17]. However, the influence of items is rarely mentioned in such area. But the fact is that items also can have influences on others and differs in different influence level [18]. In [18], the author defined item rating frequency as its level of influence and took paper citation as an example: A paper may be influential if it is cited often and perhaps cited by many other influential papers. [19] proposed an item influence-based recommendation algorithms, which not only considered about the influence of items' neighbors, but also took the effect of total influence in the process of preference prediction. Furthermore, most related research only focuses on the neighbors' direct influence on a target user, ignoring the impact exerted by the structure of neighborhood [20]. For instance, a target user's neighbors may form two groups based on their preference for target item. The one group contains users who like the target item or give high rating, and the other group contains those who dislike the target item or give low rating. The effect of the size of each group is also underestimated by traditional user-based collaborative filtering method. These factors, which are not considered in traditional user-based or item-based collaborative filtering method, may actually play important roles in the prediction of target user's preference.

Thus, this thesis proposes an influence-centric recommendation framework, which includes user influence-centric recommendation algorithm and item influence-centric recom-

mendation algorithm. In each method, users' preference can be predicted based on direct influence of user or item's neighborhood and indirect influence of the structure of corresponding neighborhood.

1.2 Influence in Recommendation systems

Recommendation systems find associations among users or items from users' evaluations of items and use this to predict users' preference [18]. For instance, in user-based collaborative filtering, recommendations for a user are computed based on how his/her like-minded users have expressed opinions on the target items. Therefore, implicit relationships among users are formed through the way they help each other find an appropriate product. In turn, these implicit relationships can be considered to have induced social networks in recommendation systems. Moreover, there might be a set of users in such social network who help a large number of other users receive recommendations.

In this thesis, the influence between users or items is classified into direct influence and indirect influence. Direct influence measures the degree that a user or item is affected by neighbors. Indirect influence measures how a user or item is affected by the structure of neighborhood.

Most traditional influence-based recommendation systems consider direct influence of a user or item on another user or item. Among these, Radish [18] list several influence measurement methods, such as the similarity between two users can be utilized to indicate the strength of relationships, the authority score of a user as his/her influence score, and centrality-based measures, which are commonly used in sociology literature to compute a node's importance or influence, are also used in recommendation systems. In this case, users or items are considered as nodes in a social network.

On the other hand, user-based or item-based collaborative filtering takes the integrated preference of neighbors of a target user or item to predict the user's preference, ignoring the structure of neighbors. Let's take user-based collaborative filtering for example, these neighbors may have opposite preferences: some neighbors may like the target item (or give high

ratings); Meanwhile, some neighbors may dislike the target item (or give low ratings). Thus, the neighbors form two groups based on their preference. The one group contains users who like the target item or give high ratings. On the other hand, the other group contains users who dislike the target item or give low ratings. Furthermore, the effect of the size of each group is also underestimated by traditional user-based CF method. For instance, if the number of one group is greater than that of the other group, the target user may tend to take the integrated preference of the group that contains majority users. These factors, which are not considered in tradition user-based CF method, may actually play important roles in the prediction of target user's preference. In the case of item-based collaborative filtering, things are the same as user-based CF.

Based on this fact, this thesis considers that the structure of a user or item's neighborhood may also exert influence on the target user or item. We define this kind of influence as indirect influence. The aim of study on indirect influence is to discover the latent attributes of user or item's neighborhood, and the correlations of these attributes with users' preference.

1.3 Questions Studied in the Thesis

In traditional user-based and item-based collaborative filtering recommendation systems, there are three major issues [10]:

- Neighbors' selection: measuring users' closeness (similarity) and find a user's several most similar neighbors, or measure items' closeness and find an item's several most similar neighbors.
- Weight measurement: measuring the weight between a user or an item and the neighbors. The weight means the importance of a neighbor's preference about the prediction of the target user or item. The weight could be similarity or influence.
- Preference prediction: predicting a target user's preference based on the neighbors' preference for the target item (user-based CF), or based on the user's preference for the target item's neighbors (item-based CF).

This thesis makes extensions to traditional user-based and item-based CF methods by in-

corporating social impact theory [60] and linear threshold model [68]. The framework of this research is as figure 1 shows. On the basis of the major issues above, this thesis studies the following questions:

- Unbalanced strategy for neighbors' selection

In traditional user-based or item-based CF algorithms, there are several methods to measure the closeness of two users or items, such as Pearson correlation, cosine-based similarity, and probability-based similarity, which are often used and discussed. Generally, the closeness between users or items is balanced or undirected, which means that arbitrary two different users or items have the same closeness between them. However, in some cases, two users may have different feelings or perceptions for each other. For example, from Jim's point of view, he and Joe are good friends, but from Joe's point of view, the relationship between them is just normal.

This thesis use such an assumption in neighbors' selection: from a user *u*'s point of view, *u* and *v*'s closeness is not definitely the same as *v*'s closeness to *u* from *v*'s point of view. And, the closeness between a user and another user is not only related to the similarity between them, but also has correlated with the fraction of co-liked and co-disliked items to the total rated items.

- Influence measurement

There are many approaches to measure a user or item's influence on the other users or items. For example, [18] proposed a method to estimate the influence of an item by changing its rating and observing the changes in the approximated ratings of other unrated item. [22] put forward two ways to measure a user's influence over others. One is to exclude a user and measure the net changes in predictions caused by the removal. The other is a predictive model based on several quality factors that may affect users' influence levels, such as number of ratings, degree of agreement with others, rarity of the rated items, and standard deviation in one's rating and so on

This thesis considers influence between two users or items as the weight between them. For instance, user *u*'s influence on another user *v* is defined as a function of the user's influence coefficient on *v* and the closeness between them. Influence coefficient measures the

ability a user or an item to influence another user or item, and persuade to have the similar preference with the user or item.

- Preference prediction based on direct influence

Direct influence measures the degree that a user or item is affected by others. Traditional influence-based recommendation algorithms mainly focus on finding influential users or items. For instance, [15] proposed an approach to find most influential users by decision tree and analyze the behavior of these users. [26] proposed an MIN (most influential users)-based model to solve the new item or new user problems.

On the basis of neighborhood-based collaborative filtering, this thesis proposes that a user's preference is related to influence-weighted preference of the neighborhood and the effect of total influence [19]. Specifically speaking, we distinguish influence into persuasive influence and supportive influence, not only consider the influence-weighted preference of neighbors, but also take into account the effect of total influence based on the social impact theory [23].

- Preference prediction based on indirect influence

Indirect influence measures how a user's preference is affected by the structure of the neighborhood [20]. Traditional neighborhood-based collaborative filtering predicts the user's preference based on the integrated preference of neighborhood, ignoring the structure of these neighbors. For instance, these neighbors might have opposite preference and form two distinct groups: some neighbors may like the target item or give high ratings, while some neighbors may dislike the target item or give low ratings. That is, the method ignores the effect of the size of each group, and this may influence users' choice. For instance, the target user may tend to take the opinion (integrated preference) of majority group.

As an extension of user-based collaborative filtering, this thesis attempts to discover useful and meaningful latent attributes from the structure of neighborhood, and mines corresponding correlations between users' preference and these latent attributes by several popular data mining techniques.

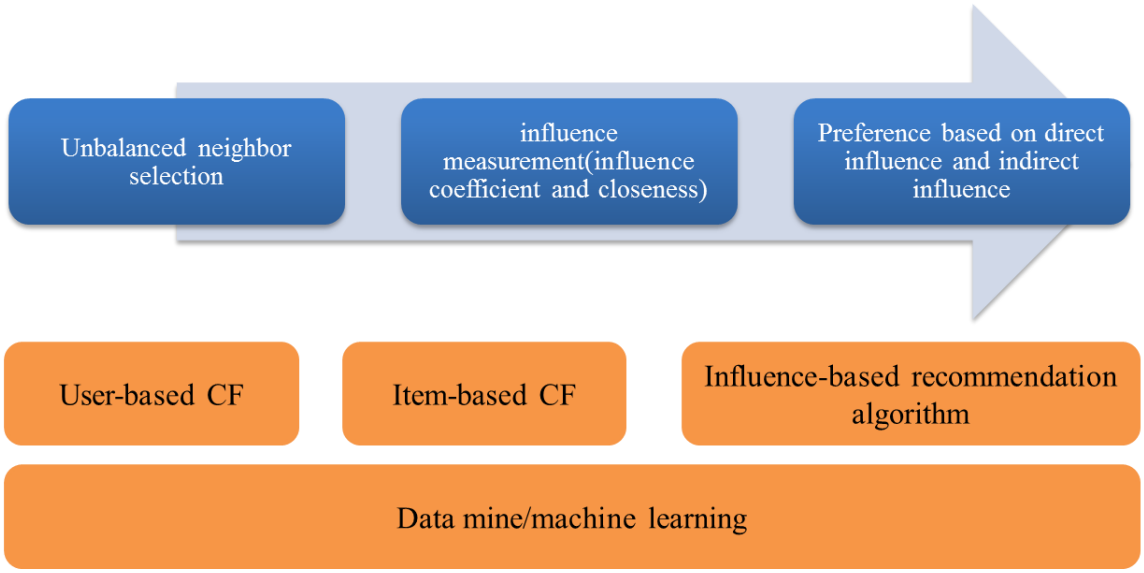


Figure 1 Framework of influence-centric recommendation algorithms

1.4 Major Contributions

The contributions in the thesis are the unbalanced neighbor selection, influence measurement, preference prediction based on direct influence and prediction based on indirect influence. In this section, we briefly summarize the contributions in each dimension.

- Unbalanced strategy for neighbor selection

Unbalanced strategy for neighbor selection considers about users or items intrinsic features and considers the relationships between them are unbalanced. From experiments, we observe that unbalanced neighbor selection method not only improves performance of traditional user-based CF and item-based CF in terms of RMSE and accuracy, but also has an advantage that it just needs fewer nearest neighbors to obtain best performance.

- Influence measurement

This thesis considers influence between two users or items as the weight between them. A user *u*'s influence on another user *v* is defined as a function of the user's influence coefficient on *v* and the closeness between them. This measurement not only focuses on the closeness between users or items, but also takes into account the influence coefficient, which measures the ability a user or an item to influence another user or item, and persuades to have the similar preference with the user or item.

- Preference prediction based on direct influence

Direct influence measures the degree that others affect a user or item. This thesis defines that the preference based on direct influence as the combination of the influence-weighted preference of the neighborhood and the effect of total influence. Specifically speaking, we distinguish influence into persuasive influence and supportive influence, not only consider the influence-weighted preference of neighbors, but also take into account the effect of total influence based on the social impact theory.

- Preference prediction based on indirect influence

Indirect influence measures how a user's preference is affected by the structure of the neighbors of the user or target item. Traditional neighborhood-based collaborative filtering predicts user's preference based on the integrated opinion of the neighborhood ignoring the structure of these neighbors. As an extension of user-based and item-based CF, this thesis attempts to discover useful and meaningful latent attributes from the structure of neighborhood, and mines corresponding correlations between users' preference and these latent attributes by several popular data mining techniques.

1.5 Thesis Overview

Chapter 2 introduces the basic algorithms of recommendation systems, including main recommendation algorithms: content-based recommendation algorithm, collaborative filtering algorithm and hybrid recommendation algorithm. Other influence-based recommendation algorithms are also presented.

Chapter 3 introduces the theoretical foundations of the thesis: (1) similarity measurements, (2) social impact theory, and (3) linear threshold model. Among them, similarity measurements are used to measure the similarity between different users or items, social impact theory and linear threshold model are used as the basics of direct influence model.

Chapter 4 introduces user influence-centric recommendation algorithm. This chapter firstly defines the basic idea of user influence-centric recommendation model, and then presents preference prediction based on neighbors direct influence and the effect of total influ-

ence, and preference prediction based on neighborhood indirect influence, which is based on extraction and selection of the latent attributes of neighborhood.

Chapter 5 introduces item influence-centric recommendation algorithm. This chapter firstly defines the basic idea of item influence-centric recommendation model, and then presents preference prediction based on neighbors' direct influence and the effect of total influence, and preference prediction based on neighborhood indirect influence, which is based on extraction and selection of the latent attributes of neighborhood.

Chapter 6 presents experimental evaluation and discussion of proposed algorithms. The experiments were conducted on MovieLens and Netflix datasets, using RMSE and Accuracy metrics. We firstly study the unbalanced strategy for neighbor selection method, then investigate direct influence model and indirect influence model respectively, and finally evaluate the total performance of the combination of direct influence model and indirect influence model.

Chapter 7 summarizes the main contributions and contents of this thesis: unbalanced neighbor selection method, preference prediction based on direct influence model and indirect influence model. In addition, several future works and interesting directions are also presented.

2 Recommendation systems basics

2.1 Introduction

Recommendation systems employ prediction algorithms to provide users with items that match their interests, thus help them sift through available books, articles, web pages, movies, music, restaurants, jokes, grocery products, and so forth to find the most interesting and valuable information for them. There are many applications of recommendation systems [5] [24], such as:

- Products recommendations: this kind of recommendations, such as Amazon, Taobao, and Netflix, is very popular and important to users. On-line shops provide personalized recommendations with products based on user purchase history or ratings of items, as well as products attributes.
- Search engine personalization: search engines rank items (e.g., documents, and images) from evolving collections, and recommend a ranked list of relevant items, in response to a query for different users. Query relevance and user interests are the sources of uncertainty for search engines.
- Contextual advertising: contextual advertising recommends advertisements based on their relevance to user contexts. Online advertising has become the cornerstone of many sustainable Internet business models, including search engines (e.g., Google), content providers (e.g., Yahoo!), and social networks (e.g., Facebook).
- News Articles: News services have attempted to recommend articles of interest to users based on the articles that they have read in the past. With the development of mobile network and mobile applications, consumers become used to read news articles through their own mobile devices, thus personalized news article recommendations become necessary to them.
- Personalized tourism recommendation systems: personalized tourism recommendation systems process available tourism information related to tourism sites for a city, and make personalized site and activity recommendations according to locations and uncertainty over user preferences. Currently, social networks and trust information

are also being integrated to provide high fidelity recommendations.

2.2 Recommendation Algorithms

Recommendation systems are usually classified into the following categories, based on how recommendations are made [1]:

- Content-based recommendations: the items which are similar to the ones the user preferred in the past will be recommended[6], [7], [8];
- Collaborative filtering recommendations: the items which are preferred by the other users with similar preferences in the past will be recommended [9], [10], [11];
- Hybrid approaches: these methods combine collaborative and content-based methods [12], [13], [14].

2.2.1 Content-based algorithms

The content-based recommendation algorithm has its roots in information retrieval and information filtering research [25], [26]. Generally speaking, systems implementing a content-based recommendation approach analyze a set of items and/or descriptions of items previously rated by a user, and build a model or profile of user interests based on the features of the items rated by that user [6]. And the profile is a structured representation of user interests, adopted to recommend new interesting items. The recommendation process basically consists in matching up the attributes of the user profile against the attributes of item content. The result is a relevance judgment that represents the user's level of interest in that item.

Because of the significant and early advancements made by the information retrieval and filtering communities and because of the importance of several text-based applications, many current content-based systems focus on recommending items containing textual information, such as documents, Web sites (URLs), and Usenet news messages. The improvement over the traditional information retrieval approaches comes from the use of user profiles that contain information about users' tastes, preferences, and needs. The profiling information can be elicited from users explicitly, e.g., through questionnaires, or implicitly – learned from their

transactional behavior over time [1].

Generally, items that can be recommended to the user are often stored in a database table and include a number of features. An example of item (book) representation is as Table 1 shows.

Table 1 An example of item representation

itemID	Title	Author	Price	Publish time
00001	All the Light We Cannot See	Anthony Doerr	$16.2	2014
00002	The Invention of Wings	Sue Monk Kidd	$17	2014
00003	The Book of Unknown Americans	Cristina Henriquez	$15	2014

The features that Table 1 shows are structured data in which there is a small number of features. In actual usage, items, such as web pages, documents, have more information than that is shown in Table 1 and the information is unstructured data [27]. For instance, a news article is example of such item. There are no attribute names with well-defined values. The common method to deal with such items is to create a structured representation of original item [28] by related techniques such as Natural Language Processing [29], [30].

On the other hand, user's profile can be constructed and updated by the feedback and related item descriptions [6], i.e., user's profiles are also represented by those features in item representation. Table 2 shows an example of user profile. The number 1 indicates a user likes a feature and 0 indicates a user dislike a feature in the table. User feedbacks can be distinguished into two kinds of categories: positive feedback, i.e., inferring features liked by the user, and negative feedback, i.e., inferring features disliked by the user [31].

Table 2 An example of user profile

userID	Feature 1	Feature 2	Feature 3	Feature 4	Feature 5
00001	1	1	0	0	0
00002	0	1	1	1	0
00003	1	1	1	1	0

Usually, given an item representation, the aim of content recommendation is to predict whether it is likely to be of interested by a target user, by comparing features in the item representation to those in the representation of user preferences. Since tastes of users usually change over time, up-to-date information must be maintained and provided in order to automatically update the user profile. Further feedback is gathered on generated recommendations by letting users state their satisfaction or dissatisfaction with items. After gathering that feedback, the learning process is performed again on the new training set, and the resulting profile is adapted to the updated user interests. The iteration of the feedback-learning cycle over time allows the system to take into account the dynamic nature of user preference [6].

In addition to the traditional information retrieval-based approaches, other techniques for content-based recommendation have also been used, such as Bayesian classifiers [32] and various machine learning techniques, including clustering, decision trees, and artificial neural networks [33]. These techniques differ from information retrieval-based approaches in that they calculate utility predictions based on a model learned from the underlying data using statistical learning and machine learning techniques.

Although content-based recommendation algorithms can be implemented in many text-based recommendation systems, there are several limitations that are described [2], [34].

- **Limited content analysis.** The features that are explicitly associated with items are limited. Therefore, in order to have a sufficient set of features, the content must either be in a form that can be parsed automatically by a computer (e.g., text), or the features should be assigned to items manually. While information retrieval techniques work well in extracting features from text documents, some other domains have an inherent problem with automatic feature extraction. For example, automatic feature extraction methods are much harder to apply to the multimedia data, e.g., graphical images, audio and video streams. Moreover, it is not often practical to assign attributes by hand due to limitations of resources.
- **Over-specialization**. When the system can only recommend items that score highly against a user's profile, the user is limited to being recommended items similar to those already rated. For example, a person with no experience with Greek cuisine

would never receive a recommendation for even the greatest Greek restaurant in town. This problem, which has also been studied in other domains, is often addressed by introducing some randomness. For example, the use of genetic algorithms has been proposed as a possible solution in the context of information filtering.

- **New user problem**. The user has to rate a sufficient number of items before a content-based recommendation system can really understand user's preferences and present the user with reliable recommendations. Therefore, a new user, having very few ratings, would not be able to get accurate recommendations.

2.2.2 Collaborative filtering algorithms

Unlike content-based recommendation methods, collaborative recommendation systems (or collaborative filtering systems) try to predict the utility of items for a particular user based on the items previously rated by other users. Collaborative filtering works on user-item ratings matrix as Table 3 shows. Collaborative recommendations can be grouped into two general classes: memory-based and model-based [2].

Table 3 User-item ratings matrix

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1	5	-	1	-	4
User 2	2	4	2	-	5
User 3	-	1	2	5	3

Memory-based CF algorithms use the entire user-item ratings matrix to generate a prediction without knowing items' contents [2], [10], [35], [36], [37]. Based on the way that prediction is done, there are two methods:

- User-based CF algorithm: predicting a target user's preference for a target item using the ratings for this item by the nearest neighbors of the target user [38], [39], [40].
- Item-based CF algorithm: predicting a target user's preference for a target item based on the ratings of the target user for items similar to target item [4], [10], [41].

The main process of user-based CF and item-based CF is as follows:

- Neighbors' selection: measuring users' closeness (similarity) and find a user's several most similar neighbors, or measure items' closeness and find an item's several most similar neighbors.
- Weight measurement: measuring the weight between a user or an item and the neighbors. The weight means the importance of a neighbor's preference on the prediction of the target user or item. The weight could be similarity or influence.
- Preference prediction: predict a user's preference based on the neighbors' preference for the target item (user-based CF), or based on the user's preference for the target item's neighbors (item-based CF).

In contrast to memory-based methods, which use the stored ratings directly in the prediction, model-based algorithms use the collection of ratings to learn a model, which is then used to make rating predictions. The general idea is to model the user-item interactions with factors representing characteristics of the users and items, such as the preference class of users and the category class of items. This model is then trained using the available data, and later used to predict users for new items. Model-based approaches for the task of recommending items are numerous and include Bayesian Clustering [42], Latent Semantic Analysis [43], Support Vector Machines [44], and Singular Value Decomposition [45], [46], [47].

The pure collaborative recommendation systems do not have some of the shortcomings that content-based systems have. In particular, since collaborative systems use other users' recommendations (ratings), they can deal with any kind of content and recommend any items, even the ones that are dissimilar to those seen in the past. However, collaborative systems have their own limitations [2], [48].

- **New user problem.** It is the same problem as for content-based systems. In order to make accurate recommendations, the system must first learn the user's preferences from the ratings that the user makes.
- **New item problem.** New items are added regularly to recommendation systems. Collaborative systems rely solely on users' preferences to make recommendations. Therefore, until the new item is rated by a substantial number of users, the recom-

mendation system would not be able to recommend it. This problem can also be addressed using hybrid recommendation approaches, described in the next section.

- **Sparsity.** In any recommendation system, the number of ratings already obtained is usually very small compared to that of ratings that need to be predicted. Effective prediction of ratings from a small number of examples is important. Also, the success of the collaborative recommendation system depends on the availability of a critical mass of users.

2.2.3 Hybrid algorithms

Hybrid recommendation systems combine two or more recommendation techniques to gain better performance by combining collaborative and content-based methods, which helps to avoid certain limitations of content-based and collaborative systems [2], [49], [50], [51]. Different ways to combine collaborative and content-based methods into a hybrid recommendation system can be classified as follows:

- **Weighted**: A weighted hybrid recommender is such that the score of a recommended item is computed from the results of all of the available recommendation techniques present in the system. The benefit of a weighted hybrid is that all of the system's capabilities are brought to bear on the recommendation process in a straightforward way and it is easy to perform post-hoc credit assignment and adjust the hybrid accordingly.
- **Switching**: A switching hybrid builds in item-level sensitivity to the hybridization strategy: the system uses some criterion to switch between recommendation techniques. Switching hybrids introduce additional complexity into the recommendation process since the switching criteria must be determined, and this introduces another level of parameterization. However, the benefit is that the system can be sensitive to the strengths and weaknesses of its constituent recommenders.
- **Mixed**: Where it is practical to make large number of recommendations simultaneously, it may be possible to use a "mixed" hybrid, where recommendations from

more than one technique are presented together. The mixed hybrid avoids the "new item" start-up problem: the content-based component can be relied on to recommend new shows on the basis of their descriptions even if they have not been rated by anyone.

- **Feature Combination:** Another way to achieve the content/collaborative merger is to treat collaborative information as simply additional feature data associated with each example and use content-based techniques over this augmented data set. The feature combination hybrid lets the system consider collaborative data without relying on it exclusively, so it reduces the sensitivity of the system to the number of users who have rated an item. Conversely, it lets the system have information about the inherent similarity of items that are otherwise opaque to a collaborative system.
- **Cascade**: one recommendation technique is employed firstly to produce a coarse ranking of candidates and a second technique refines the recommendation from among the candidate set. Cascading allows the system to avoid employing the second, lower-priority, technique on items that are already well-differentiated by the first or that are sufficiently poorly-rated that they will never be recommended.

2.3 Influence-based recommendation algorithms

Most recommendation systems based on collaborative filtering assume that users are independent, which ignores the role of social influence in people's buying decisions. Furthermore, social networks and social influence, on the other side, can provide us with extra information on users' preferences, but have received less attention [52]. Recently, the emergence of Online Social Networks (OSN) provides us with an opportunity to reconsider the structure and effects of social networks so as to improve recommendation results [53], [54].

Recently, influence-based recommendation approach is one important branch of recommendation algorithms. The basic idea in these methods is considering that a user or item has influence on another user or item, and predicting target user's preference based on the influence. The influence can be explicit influence, such as social influence among Facebook,

Twitter users, or implicit influence, such as influence from users who have similar tastes or item which have similar acceptance [19], [55], [56], [57].

[52] proposed influence-based recommendation models on the basis of social contagion and social influence network theory. In the first model for individuals, they improved the result of collaborative filtering prediction with social contagion outcome. In the recommendation model for groups, they applied social influence network theory to take interpersonal influence into account to form a settled pattern of disagreement, and then aggregate opinions of group members. [17] put forward a method to select informative items that are not only uncertain but also influential by estimating the influence of an item by changing its rating and observing the changes in the approximated ratings of other unrated items. [18] presented two ways to measure a user's influence over others. One way is to exclude a user and measure the net changes in predictions caused by the removal. The other way is a predictive model based on several quality factors that may affect users' influence levels, such as number of ratings, degree of agreement with others, rarity of the rated items, and so on. [16] defined the influence rating of a user on the community as the degree to which its item ratings match that of an average user's item ratings and then determined the most influential users who have the maximum influence on the evaluation of other user's recommendations.

To summary, traditional influence-based recommendation algorithms mainly focus on the direct influence of a user or item on another user or item. The aim is to discover the most influential users or items, and solve the specific problems of collaborative filtering such as new user, new item, and cold-start.

2.3 Summary

In summary, this chapter introduces several basics of recommendation systems. Firstly, the main definition and applications of recommendation systems are presented. Secondly, Section 2.2 introduces popular recommendation algorithms: content-based algorithms, collaborative filtering algorithms and hybrid algorithms. The main advantages and limitations are also descripted. Section 2.3 presents related research in influence-based recommendation algorithms.

The literature review allows us to formally understand research problems in recommendation systems, especially in influence-based recommendation systems. It provides us with insights and motivations in addressing these problems and develops new methods.

3 Theoretical Foundations

3.1 Introduction

In this chapter, we introduce the following three theories and foundations, which play important roles in this thesis:

- Similarity measurement methods
- Social impact theory
- Linear threshold model

Of the three, similarity measures the closeness between two users or items in terms of preference or acceptance. Social impact theory defines the magnitude of impact (influence) is related to the strength, immediacy and number of other people in the social environment. Besides, by the theory, social influence can be distinguished into persuasive influence and supportive influence. Linear threshold model defines a threshold/condition as whether a user becomes active because of the neighbors' influence.

With respect to the application of these theories in this thesis, similarity measurement is used to in neighbor selection phase. Social impact theory is used in direct influence model, where we distinguish influence into persuasive influence and supportive influence, and take into account the effect of total influence based on the theory. Linear threshold model is also used in direct influence model, where we use it to restrict the number (coverage) of cases predicted by direct influence model.

3.2 Similarity measurements

Various approaches have been used to compute the similarity between users or items in collaborative recommendation systems. In most of these approaches, the similarity between two users or items is based on their user-item ratings matrix $R = \{r_{ui} | 1 \le u \le m; 1 \le i \le n\}$. Here, we present three popular methods: cosine-based similarity, correlation-based similarity and adjusted-cosine similarity as follows [58], [59]:

- Cosine-based similarity

In this case, two users, *u* and *v*, are treated as two vectors in *n*-dimensional space of user-item ratings matrix as Table 3 shows, where *m* is the number of users, *n* is the number of items. The similarity between them is measured by computing the cosine of the angle between these two vectors.

$$s_{uv} = cos(\vec{u}, \vec{v}) = \frac{\vec{u} \times \vec{v}}{\left\|\vec{u}\right\| \times \left\|\vec{v}\right\|} \tag{1}$$

where "•" denotes the dot-product of the two vectors, $\vec{u}$ and $\vec{v}$ are the *m* dimensional vectors corresponding to the *u*th and *v*th column of matrix R. Similarly, the similarity between two arbitrary items *i* and *j* is measured as follows:

$$s_{ij} = cos(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{\left\|\vec{i}\right\| \times \left\|\vec{j}\right\|} \tag{2}$$

where $\vec{i}$ and $\vec{j}$ are the *n* dimensional vectors corresponding to the *i*th and *j*th column of matrix R.

- Correlation-based similarity

In this case, similarity between two users is measured by computing the Pearson correlation. Pearson correlation computes the statistical correlation between two user's common ratings to determine their similarity.

$$s_{uv} = \frac{\sum_{i \in I} (r_{ui} - \overline{r}_u)(r_{vi} - \overline{r}_v)}{\sqrt{\sum_{i \in I} (r_{ui} - \overline{r}_u)^2} \sqrt{\sum_{i \in I} (r_{vi} - \overline{r}_v)^2}} \tag{3}$$

where $\overline{r}_u$ are the average rating of the co-rated items of the *u*th user. Similarly, the similarity between two arbitrary items *i* and *j* is as follows:

$$s_{ij} = \frac{\sum_{u \in U} (r_{ui} - \overline{r}_i)(r_{uj} - \overline{r}_j)}{\sqrt{\sum_{u \in U} (r_{ui} - \overline{r}_i)^2} \sqrt{\sum_{u \in U} (r_{uj} - \overline{r}_j)^2}} \tag{4}$$

where $\overline{r}_i$ denotes the average rating of the *i*th item by users.

- Adjusted-cosine similarity

Cosine similarity computing using basic cosine measure has one important drawback: the differences in rating scale between different items or users are not taken into account. The adjusted cosine similarity offsets this drawback by subtracting the corresponding item or user average from each co-rated pair. Thus, the adjusted-cosine similarity between user *u* and *v* is given by

$$s_{uv} = \frac{\sum_{i \in I} (r_{ui} - \overline{r}_i)(r_{vi} - \overline{r}_i)}{\sqrt{\sum_{i \in I} (r_{ui} - \overline{r}_i)^2} \sqrt{\sum_{i \in I} (r_{vi} - \overline{r}_i)^2}} \tag{5}$$

where $\overline{r}_i$ denotes the average rating of the *i*th item by users. Similarly, the similarity between two arbitrary items *i* and *j* is as follows:

$$s_{ij} = \frac{\sum_{u \in U} (r_{ui} - \overline{r}_u)(r_{uj} - \overline{r}_u)}{\sqrt{\sum_{u \in U} (r_{ui} - \overline{r}_u)^2} \sqrt{\sum_{u \in U} (r_{uj} - \overline{r}_u)^2}} \tag{6}$$

where $\overline{r}_u$ are the average rating of the co-rated items of the *u*th user.

3.3 Social impact theory

In social life, people affect each other in many different ways such as that we are drawn by the attractiveness of others and aroused by their mere presence, stimulated by their activity and embarrassed by their attention. Social impact theory is a theory that uses mathematical equations to predict the level of such social impact created by specific social situations and was defined as any influence on individual user's feelings, thoughts, or behavior that is exerted by the real, implied, or imagined presence or actions of others in [60] and [61]. The theory of social impact is a metatheory that attempts to characterize how the many ways in which individuals affect each other are subject to the constraints of time and space, and specifically, how impact is moderated by the strength, immediacy, and number of other people in the social environment [65].

Social impact theory concerns the magnitude of impact that one or more people or groups (sources) have on an individual, and thus is a static theory of how social processes operate at the level of the individual at a given point in time. One part of the theory deals with how

much impact is experienced by an individual as a function of the strength, immediacy, and number of sources of impact. According to the theory, impact is a multiplicative function of three classes of factors: *f = f(SIN)*. That is, social impact theory is affected by strength (S), immediacy (I) and the number of sources (N). This rule postulates that the greater the number of sources of social impact in a social situation, the greater the impact could be. Within this equation, the strength (S) is a measure of how much influence, power, or intensity the target perceives the source to possess. The amount of influence, power, or intensity is often determined through factors such as age, social class, whether or not a previous relationship had existed, or anticipation of a future relationship existing. Immediacy (I) takes into account how recent the event occurred and whether or not there were other intervening factors. The number of people (N) is the number of sources exerting social influence on the target. The I = f(SIN) equation illustrates that there is more social impact when higher status individuals are the source, when the action is more immediate, and when there are a greater number of sources.

In the extension of social impact theory [62], they distinguished impact into persuasive impact and supportive impact. Persuasive impact tries to convince someone to change opinion, and supportive impact tries to keep opinion. And the total persuasive impact was defined as follows:

$$w_u^p = N_p^{1/2}\left[\frac{\sum(p_i / d_i^2)}{N_p}\right] \tag{7}$$

where N_p denotes the number of sources who have opposite opinion with individual u, p_i is the persuasiveness of source i, and d_i is the distance between source i and u. The interpretation of this formula is that the persuasive impact of a group is the average force (persuasiveness divided by the square of the distance) exerted by each group member multiplied by the square root of the number of group members. Similarly, the total supportive impact was defined as follows:

$$w_u^s = N_s^{1/2}\left[\frac{\sum(s_i / d_i^2)}{N_s}\right] \tag{8}$$

where N_s denotes the number of sources who have the same opinion with individual *u*, s_i is the persuasiveness of source *i*, and d_i is the distance between source *i* and *u*. The interpretation of this formula is that the supportive impact of a group is the average force exerted by each group member multiplied by the square root of the number of group members.

Furthermore, the total impact that an individual *u* experiences from his/her social environment is as follows:

$$w_u = N_p^{1/2}\left[\frac{\sum(p_i / d_i^2)}{N_p}\right] - N_s^{1/2}\left[\frac{\sum(s_i / d_i^2)}{N_s}\right] \quad (9)$$

According to [60], [62], there are two possible situations by taking combined effect of persuasive influence and supportive influence into account:

- If $w_u \geq 0$, this denotes that the pressure in favor of the preference state change overcomes the pressure to keep the current opinion. That is, the individual will change his/her mind.
- If $w_u < 0$, this denotes that the pressure to keep the current preference state overcomes the pressure in favor of the opinion change. That is, the user will keep his/her current mind.

As a simple and effective method, social impact theory has been extended by many research, such as [62], [63], [64], and applied in diverse area [65], [66], [67], such as simulation of language change and pattern recognition.

3.4 Linear Threshold model

In the area of influence/innovation diffusion, linear threshold model [68] is one of the important models formalizing the behavior of influence propagation concerning the spread of new trends and innovations in social networks. In this model, each user is denoted as a node in a graph and can be in one of two states: active (i.e., supports an idea or a product) or inactive as Figure 2 shows.

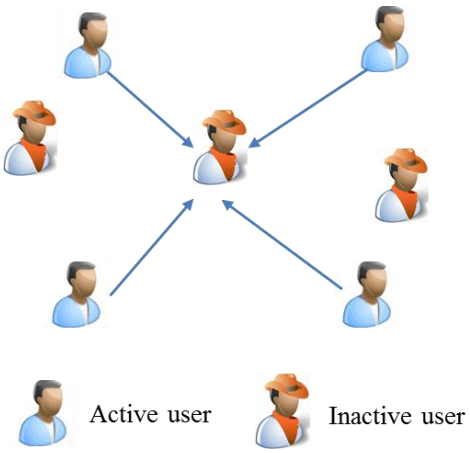


Figure 2 Active users and inactive users

Each user *u* has a threshold, θ_u , which is chosen uniformly at random from the interval [0, 1]. This represents the weighted fraction of *u*'s neighbors that must become active in order for *u* to become active. The user is influenced by each neighbor *v* according to weight w_{uv} , such that:

$$\sum_{v\ neighbor\ of\ u} w_{uv} \leq 1 \tag{10}$$

Then the user *u* will become active if the total weight of its active neighbors reaches its threshold, i.e.

$$\sum_{v\ neighbor\ of\ u} w_{uv} \geq \theta_u \tag{11}$$

Thus, the threshold θ_u intuitively represents the different latent tendency of user *u* to adopt the innovation when their neighbors do. Usually, the fact that these thresholds for users are randomly selected is intended to model the lack of knowledge of their values - we are in effect averaging over possible threshold values for all the nodes [69], [70].

Linear threshold model has been adopted in many researches, such as innovation/idea diffusion [69], [70]. With respect to our influence-centric recommendation framework, the threshold model is used to measure whether a user or item can be influenced by the corresponding neighbors.

3.5 Summary

In summary, this chapter reviews the theoretical foundations of this thesis including similarity measure, social impact theory, as well as linear threshold model. These theories play important roles in our proposed influence-centric recommendation algorithms. For instance, similarity measures the closeness between two users or items in terms of preference or acceptance (Section 4.3.1 and 5.3.1).

Social impact theory is used in direct influence model, where we distinguish influence into persuasive influence and supportive influence, and take into account the effect of total influence based on the theory (Section 4.3.3 and 5.3.3). Linear threshold model is also used in direct influence model, where we use it to restrict the number (coverage) of cases predicted by direct influence model (Section 4.3.4 and 5.3.4).

4 User influence-centric algorithm

4.1 Introduction

This chapter presents user influence-centric recommendation algorithm (user-ICR). User-ICR predicts users preference based on direct influence exerted by the target user's neighbors, and indirect influence because by mining the latent attributes of the neighborhood.

Direct influence algorithm is based on user-based CF, i.e. a user's preference for an item is approximated based on the preference of similar user who has exerted influence on the target user. The main difference between our proposed algorithm and standard user-based algorithm is that each rating is weighted by the corresponding influence of neighbors on target user instead of their similarity. And furthermore, as an extension of influence-based algorithms, we distinguish these influences into persuasive influence and supportive influence, and take into account the total influence of these two types of influence in the preference prediction process. With respect to indirect influence model, it measures how a user's preference is affected by the structure of the neighborhood. Traditional user-based collaborative filtering considers the integrated preference of the neighborhood as target user's preference, ignoring the structure of these neighbors. This thesis attempts to discover useful and meaningful latent attributes from the structure of neighborhood, and mines corresponding correlations between users' preference and these latent attributes by several popular data mining techniques.

In the following sections of this chapter, we develop a user influence-centric model to satisfy the above description. Section 4.1 defines the basic notation used in the algorithm; Section 4.2 presents how to predict user's preference based on direct influence, and Section 4.3 shows how to predict user's preference based on indirect influence.

4.2 Problem definition

Consider that there are m users and n items in a user-based collaborative filtering system. We can define the set of users U as the set of integers $\{1, 2 \ldots m\}$ and the set of items I as the set of integers $\{1, 2 \ldots n\}$. The ratings of users for item r_{ui} are stored in a $m \times n$ rating matrix

$R=\{r_{ui}|1\le u\le m;1\le i\le n,0\le r_{ui}\le r\}$ where r is the rating scale. Except for absolute ratings, we also use binary measurement to indicate user's preference state or item's acceptance state, i.e.

$$o_{ui}=\begin{cases}1, & if\ r_{ui}\ge\lambda\\ -1, & if\ 0<r_{ui}<\lambda\end{cases} \tag{12}$$

Where $0<\lambda\le r$, 1 indicates the user likes the item, and 0 denotes the user dislikes the imte. For instance, if the rating scale of a recommendation system is 5, i.e. r = 5, then we can define $\lambda=4$. This indicates that a user likes an item if he/she give rating 4 or 5 to the item. If u has not rated item i, then we use the initial preference r_{ui}^0 instead of the real rating r_{ui} in formula (12). Initial preference represents a user's preference tendency based on the rating distribution or other recommendation algorithms. In order to compare our algorithms with other standard collaborative filtering algorithms, we use a simple formula to get r_{ui}^0 as follows:

$$r_{ui}^0=\beta\overline{r_u}+(1-\beta)\overline{r_i} \tag{13}$$

where $0\le\beta\le1$, $\overline{r_u}$ is the average rating user u has given and $\overline{r_i}$ is the average rating item i received.

The mechanism for user influence-centric algorithm is that a user's preference is affected by direct influence exerted by neighbors and indirect influence exerted by the structure of neighborhood. Preference prediction based on direct influence is the combination of influence-weighted preference of the target user's neighbors and the effect of the total influence the user received from his/her neighborhood. On the other hand, preference prediction based on indirect influence is learning the correlation of target user's preference and latent attributes of the structure of neighborhood.

4.3 Preference based on direct influence

4.3.1 User neighbor selection

In traditional user-based CF algorithms, there are several methods to measure the closeness of two users such as Pearson correlation, cosine-based similarity, probability-based similarity

which are discussed in Section 3.2. Generally, the closeness between users is balanced or undirected, which means that arbitrary two users have the same closeness between them. However, in some cases, two users may have different feelings or perceptions for each other. For example, from Jim's point of view, he and Joe are good friends, but from Joe's point of view, the relationship between them is just normal (Figure 3).

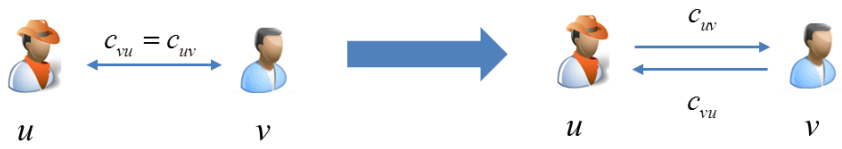


Figure 3 Traditional similarity strategy and unbalanced similarity strategy

This thesis uses such an assumption in neighbors' selection: from a user u's point of view, u and v's closeness is not definitely the same as v's closeness to u from v's point of view. And, the closeness between a user and another user is not only related to the similarity between them, but also has correlated with the fraction of items with the same acceptance state to the total rated items.

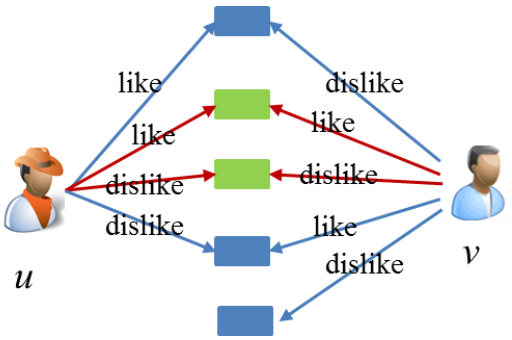
- From a user u's point of view, u and v's closeness is different from v's closeness to u from v's point of view, i.e. $c_{uv} \neq c_{vu}$.
- For user u's several nearest neighbors, we need to rank c_{uv} in descending order and select most nearest neighbors.
- For the prediction of user u' preference for item i, we use weight c_{vu} , which measures user v's influence on user u.

A user's influence on another user is related to the closeness between them. In addition to traditional similarity, we define the following formula to measure the closeness of two users from user v's point of view:

$$c_{uv} = s_{uv} \times |Y_{uv}| / |Y_v| \tag{14}$$

where s_{uv} is the similarity between u and v, $|Y_{uv}|$ is the number of items for which both u and v have the same preference state, and $|Y_v|$ is the number of items which has been rated by

v. For instance, Figure 3 shows how to measure the unbalanced similarity.



$$c_{uv} = s_{uv} \cdot \frac{|Y_{uv}|}{|Y_u|} = s_{uv} \cdot \frac{2}{4}$$

$$c_{vu} = s_{vu} \cdot \frac{|Y_{uv}|}{|Y_v|} = s_{vu} \cdot \frac{2}{5}$$

$$c_{uv} \neq c_{vu}$$

Figure 4 Computation of unbalanced similarity

After measuring the closeness of two arbitrary users, the next step is to select nearest neighbors. In this thesis we have the following definitions:

$$N(u) = \{v \mid c_{uv} > 0, v \in U\} \tag{15}$$

$$N(u;i) = \{v \mid c_{uv} > 0, r_{vi} > 0, v \in U\} \tag{16}$$

where c_{uv} is the closeness from *u* to *v*. Formula (15) means user *u*'s global neighborhood without considering specific item. Formula (16) means user *u*'s neighbors who also rated item *i*. From the definitions, we can see that $N(u)$ is fixed for arbitrary user *u*, but $N(u;i)$ is not static and adaptively varies with the target item. Apparently, $N(u;i) \subseteq N(u)$ holds for arbitrary user.

4.3.2 User influence coefficient

User influence coefficient denotes the importance of a user in predicting other users' preference. In this thesis, we use user influence coefficient to measure a user's ability to influence another user, and persuade to have the same preference rate with the target user. The definition of user influence coefficient of a user *v* on user *u* is as follows:

$$p_{uv} = |N(v)| \cdot |Y_{uv}| / max_{q \in N(u)}(N(q) \cdot |Y_{uq}|) \tag{17}$$

where $|N(v)|$ is the number of *v*'s neighbors and $|Y_{uv}|$ is the number of items for which both *u* and *v* have the same preference state. Apparently, user influence coefficient is normalized,

i.e. $0 \le p_{uv} \le 1$ in our definition. Based on this definition, we can see that the more the number of neighbors of *v* and the more the number of items for which both users have the same preference state, the more ability *v* can influence *u*.

4.3.3 Persuasive influence and supportive influence

On the basis of social impact theory, if a target user *u*'s preference state for an item *i* is the same as a neighbor *v*'s preference state for the target item *i*, we can say that *v* has supportive influence on *u*. On the other hand, if *u*'s preference state for an item *i* is different from a neighbor *v*'s preference state for *i*, then we can say that *v* has persuasive influence on *u*. Thus, a target user *u*'s neighbors can be grouped into: persuasive group, in which neighbors have opposite preference with *u*, and supportive group, in which neighbors have the same preference state as *u*. Except for influence coefficient and closeness defined in [18], we also considert the number of ratings given by the target user, i.e. the more ratings a user gave, the less influence the other user will exert on him/her, and the vice versa. Then we can define user *v*'s persuasive influence or supportive influence on *u* as follows:

$$w_{uv}^{p} = p_{uv} c_{uv} / log(p + |Y_u|) \tag{18}$$

$$w_{uv}^{s} = p_{uv} s_{uv} / log(s + |Y_u|) \tag{19}$$

where $|Y_u|$ indicates the number of ratings user *u* gave, parameter *p* and *s* are used to distinguish the difference between persuasive influence and supportive influence. We call *p* persuasive factor, and *s* supportive factor.

Additionally, we also consider that the combined effect of persuasive influence and supportive influence plays an important role in the prediction of preference. On the basis of social impact theory, the total influence user *u* received from the neighborhood could be calculated in the following forms:

$$w_u = N_p^{\eta} \sum_{v \in N^p(u;i)} w_{uv}^{p} - N_s^{\eta} \sum_{v \in N^s(u;i)} w_{uv}^{s} \tag{20}$$

where $N^p(u;i)$ is the set of users who have opposite preference state with *u* and have rated item *i*, N_p is the number of the set, $N^s(u;i)$ is the set of users who have the same preference state with *u* and have rated item *i*, N_s is the number of the set, and η is a parameter controlling the contribution of numbers of users in persuasive group and supportive group respectively.

There are two possible situations by taking combined effect of persuasive influence and supportive influence into account:

- If $w_u > 0$, this indicates that the pressure in favor of the preference state change overcomes the pressure to keep the current preference state. In this thesis, we define the greater the value of w_u, the higher probability of user *u* changes his/her preference state for item *i*.
- If $w_u < 0$, this shows that the pressure to keep the current preference state overcomes the pressure in favor of the preference state change. In this thesis, we define that the greater the value of $|w_u|$, the higher probability user *u* keeps his/her preference state for item *i*.

4.3.4 Preference prediction

Before introducing the preference prediction model, we introduce a closeness threshold parameter, θ, based on linear threshold model, i.e. a user *u* can be influenced by his/her neighbors only if

$$\sum_{v \in N(u;i)} c_{vu} \geq \theta \tag{21}$$

In the closeness threshold model above, we need to notice that we use c_{vu} instead of c_{uv}. This is because we take unbalanced strategy.

As discussed in Section 4.1, a user's predictive preference based on user direct influence is the combination of influence-weighted preference of the target user's neighbors' preference for the target item and the reinforcement of the total influence effect the target user received from his/her neighborhood. Thus, considering the combined effect of total influence, we use a parameter, ρ, to control the contribution of the effect of the total influence and have the fol-

lowing formula for the prediction of a user *u*'s preference for item *i* based on *u*'s neighbors' preference for *i*:

$$\tilde{r}_{ui} = \frac{\sum_{v \in N^p(u;i)} r_{ui} w_{uv}^p + \sum_{v \in N^s(u;i)} r_{ui} w_{uv}^s}{\sum_{v \in N^p(u;i)} w_{uv}^p + \sum_{v \in N^s(u;i)} w_{uv}^s} - \rho o_{ui} \times \frac{1 - e^{-w_u}}{1 + e^{-w_u}} \tag{22}$$

where the first term indicates the influence-weighted preference of user *u*'s neighborhood, and the second term addresses the effect of total influence received from neighborhood.

With respect to prediction based on direct influence, this thesis use iterative schema to update the preference:

$$\hat{r}_{ui}^k = \begin{cases} \alpha \tilde{r}_{ui}^{k-1} + (1-\alpha)\hat{r}_{ui}^{k-1}, & k \geq 2 \text{ and closeness threshold is satisfied} \\ r_{ui}^0, & k = 1 \text{ or closeness threshold is not satisfied} \end{cases} \tag{23}$$

where $\tilde{r}_{ui}^{k-1}$ is the prediction of user *u*'s preference for item *i* based on *u*'s neighbors' direct influence during the *k*-th iteration. If *k* = 1, we use user's original preference r_{ui}^0.

In summarily, the process of direct influence algorithm is as follows:

1. Measuring users' closeness and selecting a target user's nearest neighbors using un-balanced strategy related to target item.
2. Computation of a user's influence coefficient which denotes the user's ability to influence another user, and persuade to have the similar preference (the same preference state) with the user.
3. Computation and analysis of persuasive influence and supportive influence, and analyzing the total influence effect of these two types of influence.
4. Prediction of preference based on neighborhood and total influence effect by formula (22) and (23) under the condition of closeness threshold is satisfied.
5. If the closeness threshold is not satisfied, then repeat step 3 and 4.

4.4 Preference based on indirect influence

4.4.1 Preference influenced by the structure of neighborhood

Traditional user-based CF methods consider the integrated preference of the neighbors of a target user to predict the user's preference, ignoring the influence of the structure of the neighborhood. For instance, these neighbors may have opposite preference state as Section 4.3 presents, i.e. the neighbors form two groups based on their preference. One group contains users who give persuasive influence on the target user, and we call it persuasive group. On the other hand, the other group contains users who give supportive influence on the target user, and we call it supportive group. Furthermore, the effect of the size of each group is also underestimated by traditional user-based CF method. For example, if the number of persuasive group is greater than the number of supportive group, the target user may tend to take the integrated preference of supportive group, which contains majority users. These factors, which are not considered in tradition user-based CF method, may actually play important roles in the prediction of target user's preference [20].

Thus, in this thesis, we consider the influence of the structure of neighborhood as indirect influence, and try to discover several useful and meaningful latent attributes of the structure, such as group size difference, total influence difference and standard deviation difference, in user-based CF systems. Then, corresponding correlations between users' preference and these attributes by several popular data mining techniques are suggested. The flow of indirect influence algorithm is as Figure 5 shows.

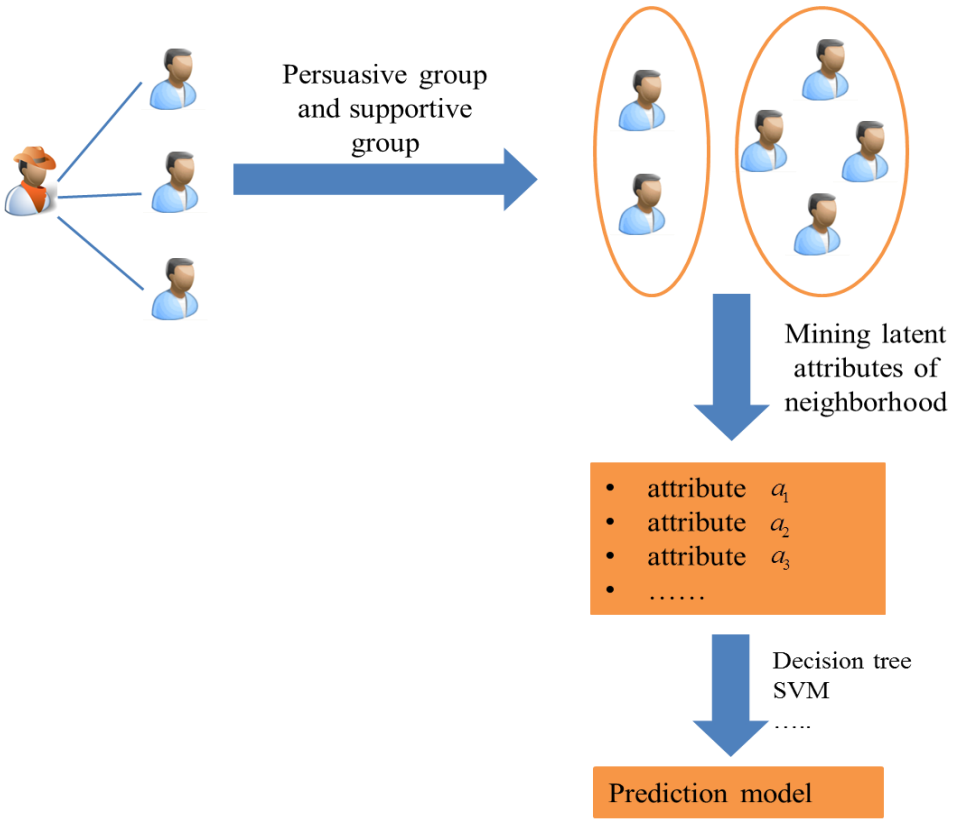


Figure 5 Process of prediction based on indirect influence

Technically speaking, we partition a target user *u*'s neighbors related to item *i* into persuasive group and supportive group. Persuasive group $N^p(u;i)$ contains users who have rated item *i* and have opposite preference state with *u*, i.e.

$$N^p(u;i)=\{v|r_{vi}>0,c_{uv}>0,o_{ui}o_{vi}<0,v\in U\} \tag{24}$$

Supportive group $N^s(u;i)$ contains users who have rated item *i* and have the same preference state as *u*, i.e.

$$N^s(u;i)=\{v|r_{vi}>0,c_{uv}>0,o_{ui}o_{vi}>0,v\in U\} \tag{25}$$

The number of user in each group is $|N^p(u;i)|$ and $|N^s(u;i)|$ respectively. Apparently, $N^p(u;i)\bigcup N^s(u;i)=N(u;i)$ and $N^p(u;i)\bigcap N^s(u;i)=\varnothing$. If we define the integrated preference of each group as the average preference of them, then we have integrated preference of persuasive group as

$$r_{ui}^p=\frac{\sum_{v\in N^p(u;i)}r_{vi}}{|N^p(u;i)|} \tag{26}$$

and integrated preference of supportive group as

$$r_{ui}^{s} = \frac{\sum_{v \in N^{s}(u;i)} r_{vi}}{\left| N^{s}(u;i) \right|} \tag{27}$$

The aim of indirect influence algorithm is to find useful and meaningful latent attributes between the persuasive group and supportive group, and predict whether the target user would like to take the integrated preference of persuasive group or supportive group.

Assuming there is a set of latent attributes $A = \{a_1, a_2, \cdots, a_l\}$ for arbitrary users' neighbors related to item *i*, and then the prediction becomes a classification problem, i.e. each observed user u's latent attributes of neighborhood consists of a pair: a attributes vector $a_{ui} = \{a_{u1}, a_{u2}, \cdots a_{ul}\}$, where $a_{ui} \in R^{l}$, and the associated label y_{ui},where $y_{ui} \in \{r_{ui,}^{p} r_{ui}^{s}\}$. The classification task is to learn the mapping $a_{ui} \mapsto y_{ui}$. In the next section, we define several rated latent attributes of user neighborhood.

4.4.2 Mining latent attributes

In [15], the authors list 24 potential attributes of user-item-rating matrix, such as number of ratings, degree of agreement with others, average of user's rating, standard deviation in user rating and so on. But clearly these attributes are just the statistical features of the matrix. In this thesis, we attempt to discovery several latent attributes of the neighborhood which describe the structure of target user's neighbors and have correlations with user's preference.

We use two data set Movielens and Netflix (see details in Section 6) to conduct experiments and after several trials, we have the following useful and meaningful latent attributes in users' neighborhood:

- Group Size Difference (GSD): the difference of size of persuasive group and supportive group. This attribute measures the difference of two groups in terms of neighbors' number. If GSD>0, the persuasive group has more users than that of supportive group and vice versa.

$$GSD = \frac{\left| N^{p}(u;i) \right| - \left| N^{s}(u;i) \right|}{\left| N^{p}(u;i) \right| + \left| N^{s}(u;i) \right|} \tag{28}$$

- Total Influence Difference (TID): the difference of influence of persuasive group and supportive group on target user. This attribute measures the difference of influence between two groups and the target user. If TID>0, the persuasive group has much more influence on the target user than that of supportive group, and vice versa.

$$TID = \sum_{v \in N(u;i)} w_{uv}^{p} - \sum_{v \in N(u;i)} w_{uv}^{s} \tag{29}$$

- Standard Deviation Difference (SDD): the difference of standard deviation between persuasive group and supportive group. This attribute measures difference of ratings in two groups in terms of standard deviation. SDD>0 indicates that the ratings in persuasive group are closer than that of supportive group, and vice versa. In the following computation for SDD, $\bar{r}_i^{\,p}$ and $\bar{r}_i^{\,s}$ indicate the average rating of persuasive group and supportive group for item *i* respectively.

$$SDD = \sqrt{\frac{\sum_{v \in N^{p}(u;i)} (r_{vi} - \bar{r}_i^{\,p})^2}{\left|N^{p}(u;i)\right|}} - \sqrt{\frac{\sum_{v \in N^{s}(u;i)} (r_{vi} - \bar{r}_i^{\,s})^2}{\left|N^{s}(u;i)\right|}} \tag{30}$$

Taking Netflix data set for example, the correlation of these attributes and users' preference tendency is as Figure 6 shows. The red points mean users would like to take the opinion of persuasive group, and the blue points mean users would like to take the opinion of supportive group. From this figure, we can see that the two classes do not have distinct boundaries, but still are separated into two parts. Furthermore, if we set a threshold to GSD, for example |GSD|>=0.1, then we obtain a result as Figure 7 shows. We can see that the two classes are divided into two parts clearly except several points.

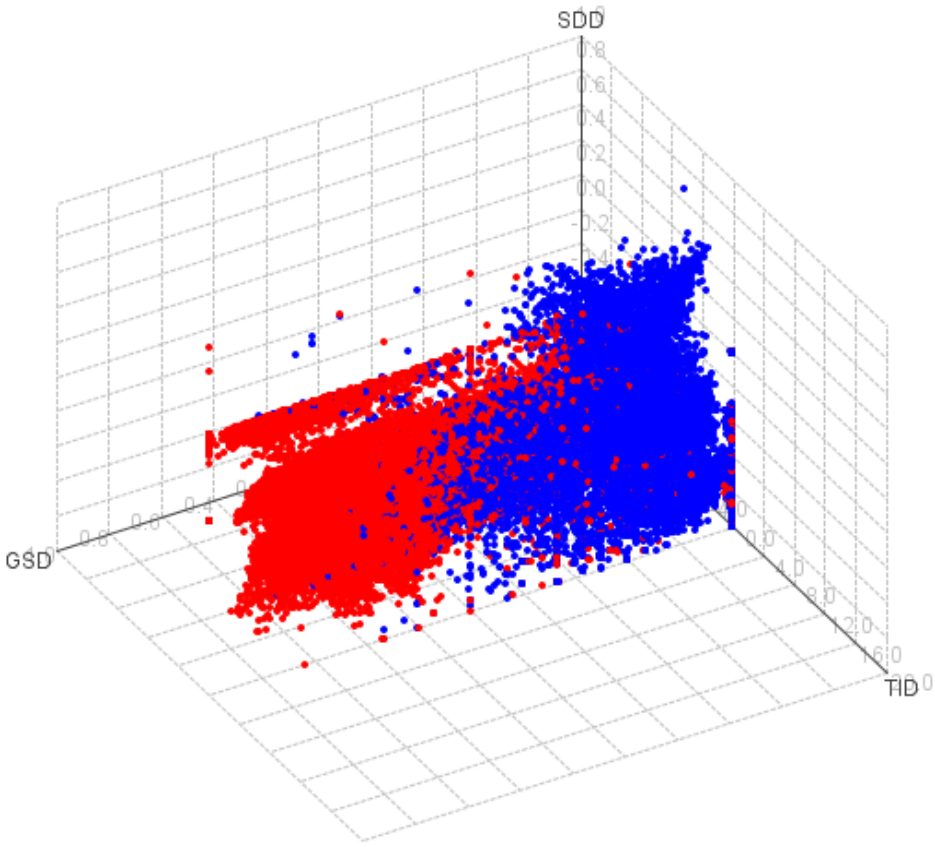


Figure 6 Correlation of latent attributes and preference

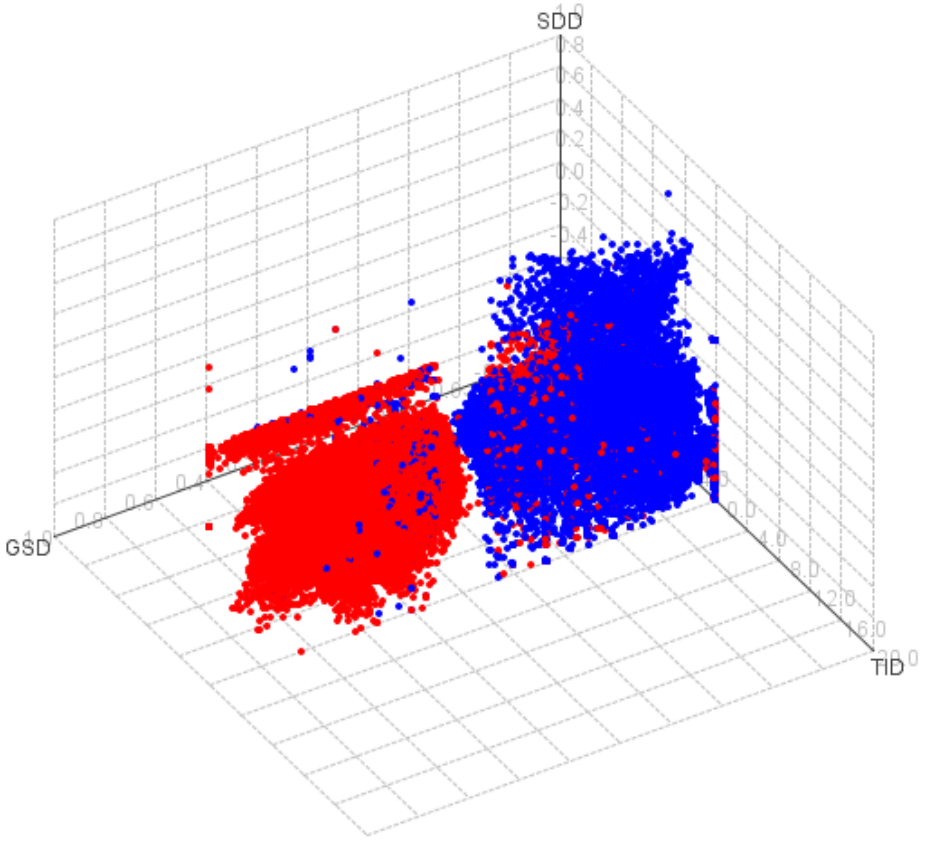


Figure 7 Correlations of latent attributes and preference given threshold to GSD

Moreover, we also investigate the pair correlation of these three attributes and users' preference as Figure 8, 9 and 10 show, where blue lines and points mean probability that users would like to take integrated preference of persuasive group and red ones mean probability that users would like to take integrated preference of supportive group. From these figure, we can see that the greater of the absolute value of these attributes, the high probability that users would like to take the opinion of one of the groups.

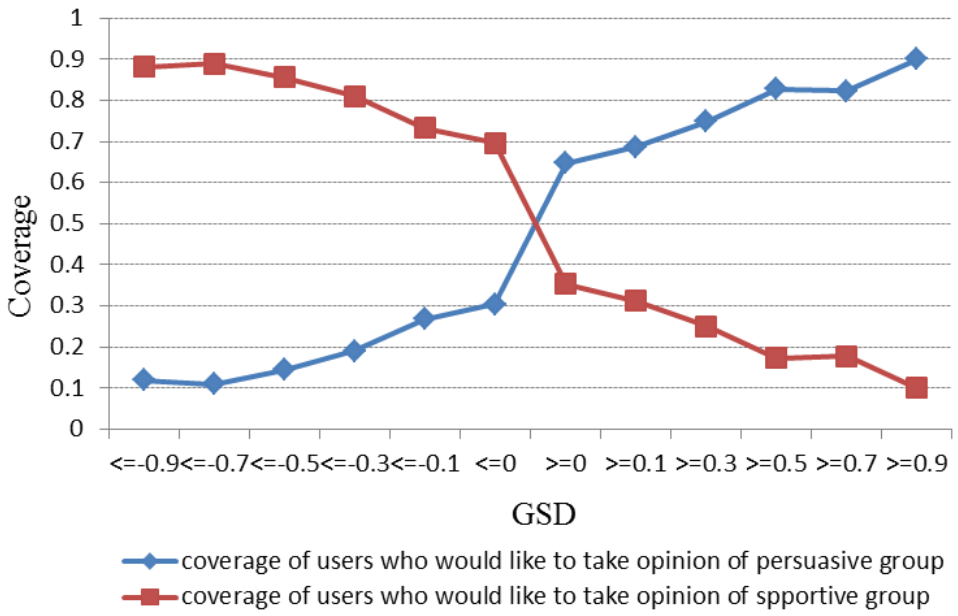


Figure 8 Correlation of GSD and users' preference

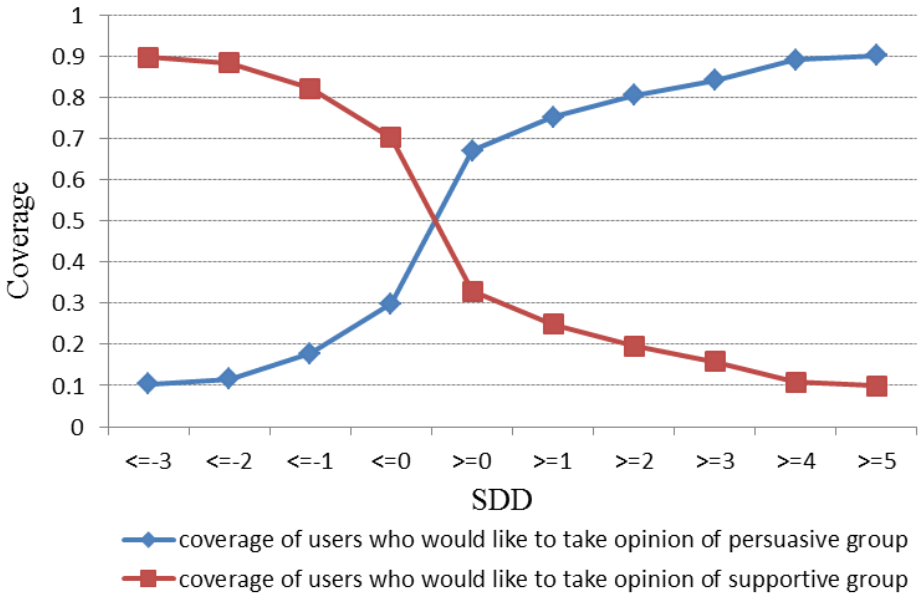


Figure 9 Correlation of TID and users' preference

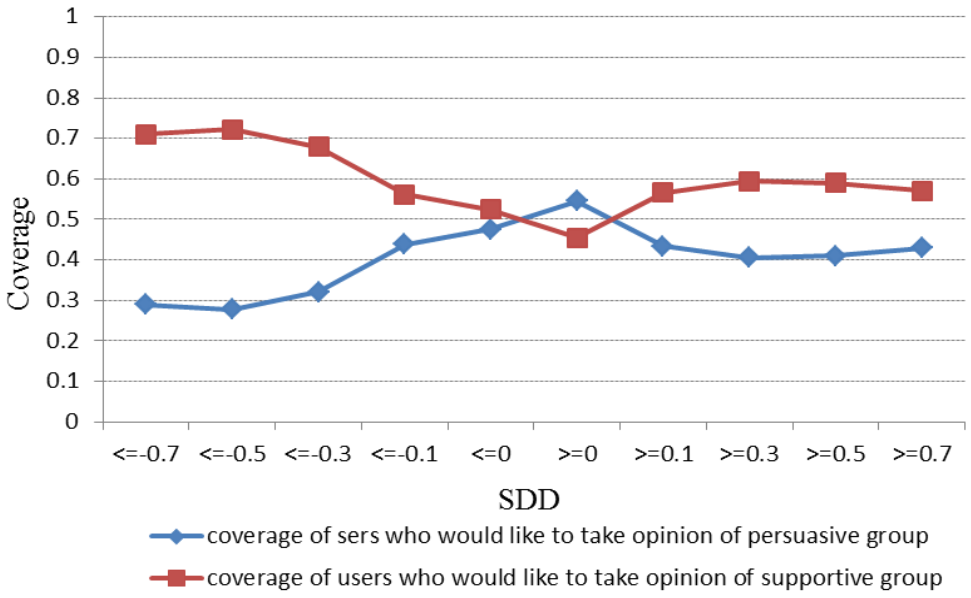


Figure 10 Correlation of SDD and users' preference

Based on these findings, we can see that users' preference is related to these three attributes: group size difference, total influence difference and standard deviation difference. Thus, we can have the following conclusions:

- Users' preference is related to neighbors' group size difference, total influence difference and standard deviation difference.
- Given threshold for those attributes, the boundary of persuasive group and supportive group could be clear.

4.4.3 Implementation

Based on Section 4.4.2, we have three latent attributes, i.e. GSD, TID, SDD, for each user *u*'s neighborhood related to item *i*. That is, $A=\{a_1,a_2,\cdots,a_k\}=\{GSD,TID,SDD\}$. Accordingly, the label is $y_{ui}\in\{r_{ui}^{p}\ r_{ui}^{s}\}$. As discussed in Section 4.4.1, the user indirect influence algorithm is a classification problem. Besides, from Section 4.4.2 we can see that we can obtain better classification if we set thresholds for one or more attributes. Thus, before implementing classification algorithms, we firstly define thresholds for attributes. In order to avoid too complicated, we just use one threshold π for GSD, i.e. $|GSD|\geq\pi$, in this thesis. That is, we only implement classification algorithms on these cases that satisfy GSD threshold.

There are many classification algorithms in artificial intelligence literature, such as Decision tree, Clustering, Logical regression, Bayesian classification, Support Vector Machine, and so on. In this thesis, since the prediction task is a classification task, we use the following data mining/machine learning techniques to predict users' preference based on the three latent attributes discussed in Section 4.3.2.

- Decision Tree: Decision trees are classifiers on a target attribute (or class) in the form of a tree structure [71], [72]. Each node corresponds to one of these attributes, and edges correspond to the possible values of the attributes. Decision trees are produced by algorithms that identify various ways of splitting a data set into branch-like segments. These segments form an inverted decision tree that originates with a root node at the top of the tree. The object of analysis is reflected in this root node as a simple, one-dimensional display in the decision tree interface. The name of the field of data that is the object of analysis is usually displayed, along with the spread or distribution of the values that are contained in that field.
- Naive Bayesian: A Bayesian classifier is a probabilistic framework, which is based on the definition of conditional probability and the Bayes theorem. In simple terms, a naive Bayes classifier assumes that the value of a particular feature is unrelated to the presence or absence of any other feature, given the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 3" in diameter. A naive Bayes classifier considers each of these features to contribute independently to the probability that this fruit is an apple, regardless of the presence or absence of the other features [73].
- Neural Networks: An Artificial Neural Network (ANN) is an assembly of inter-connected nodes and weighted links that is inspired in the architecture of the biological brain. Nodes in an ANN are called neurons as an analogy with biological neurons. These simple functional units are composed into networks that have the ability to learn a classification problem after they are trained with sufficient data [74], [75].
- Support Vector Machines: the goal of a SVM classifier is to find a linear hyperplane (decision boundary) that separates the data in such a way that the margin is maximized.

SVM can be used for both regression and classification problems. For classification problems, an SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on [76], [77].

4.5 Summary

In summary, this chapter presents the process of user influence-centric recommendation algorithm. Firstly, we define the basic idea and notation in user influence-centric recommendation model, and then presents preference prediction based on neighbors direct influence and the effect of total influence, and preference prediction based on neighborhood indirect influence, which is based on extraction and selection of the latent attributes of neighborhood.

5 Item influence-centric algorithm

5.1 Introduction

This chapter presents item influence-centric recommendation algorithm (item-ICR). Item-ICR predicts users preference based on direct influence from the target item's neighbors, and indirect influence because of the structure of item's neighborhood [19].

As well as user direct influence model, item direct influence model is based on item-based collaborative filtering, i.e. a user's preference for an item is approximated based on user's ratings on these items which have influence on the target item. The main difference between our proposed algorithm and standard item-based algorithm is that each rating is weighted by the corresponding influence of rated item on target unrated item instead of their similarity. And furthermore, as an extension of the other influence-based algorithms, we distinguish these influence into persuasive influence and supportive influence, and take into account the total influence of these two types of influence in the preference prediction process. With respect to indirect influence model, it measures how a user's preference is affected by the structure of the target item's neighborhood. Traditional item-based collaborative filtering predicts the user's preference based on the integrated preference of the neighborhood, ignoring the structure of these neighbors. This thesis attempts to discover useful and meaningful latent attributes from the structure of neighborhood, and mines corresponding correlations between users' preference and these latent attributes by several popular data mining techniques.

In the following sections of this chapter, we develop an item influence-centric model to satisfy the above description. Section 5.1 defines the basic notation used in the algorithm; Section 5.2 presents how to predict user's preference based on direct influence and 5.3 shows how to predict user's preference based on indirect influence.

5.2 Problem definition

Consider that there are m users and n items in a user-based collaborative filtering system. We can define the set of users U as the set of integers $\{1, 2 \ldots m\}$ and the set of items I as the set of

integers {*1, 2... n*}. The ratings of users for items r_{ui} are stored in a $m \times n$ rating matrix $R = \{r_{ui} | 1 \le u \le m; 1 \le i \le n, 0 \le r_{ui} \le r\}$ where *r* is the rating scale. Except for absolute ratings, we also use binary measurement to indicate user's preference state or item's acceptance state, i.e.

$$o_{ui} = \begin{cases} 1, & if \ r_{ui} \ge \lambda \\ -1, & if \ 0 < r_{ui} < \lambda \end{cases} \tag{31}$$

where $0 < \lambda \le r$. If *u* has not rated item *i*, then we use the initial preference r_{ui}^0 instead of the real rating r_{ui} in formula (31). Initial preference represents a user's preference tendency based on the rating distribution or other recommendation algorithms. In order to compare our algorithms with other standard collaborative filtering algorithms, we use a simple formula to get r_{ui}^0 as follows:

$$r_{ui}^0 = \beta \overline{r_u} + (1 - \beta) \overline{r_i} \tag{32}$$

where $0 \le \beta \le 1$, $\overline{r_u}$ is the average rating user *u* has given and $\overline{r_i}$ is the average rating item *i* received.

The idea for item influence-centric algorithm (Item-ICA) is that a user's preference for an item is affected by direct influence exerted by the item's neighborhood and indirect influence exerted by the structure of item's neighborhood. Preference prediction based on direct influence is the combination of influence-weighted preference of the target user's preference for target item's neighbors and the effect of the total influence the item received from its neighborhood. On the other hand, preference prediction based on indirect influence is learning the correlation of target user's preference for target item and latent attributes of the structure of target item's neighborhood.

5.3 Preference based on direct influence

5.3.1 Item neighbor selection

In traditional item-based CF algorithms, there are several methods to measure the closeness of two items such as Pearson correlation, cosine-based similarity, probability-based similarity

which are discussed in Section 3.2. Generally speaking, the closeness between items is balanced or undirected, which means that arbitrary two items have the same closeness between them. Similarly to user neighbor selection, this thesis also uses unbalanced strategy in item neighbor selection:

- From an item *i*'s point of view, *i* and *j*'s closeness is different from *j*'s closeness to *i* from *j*'s point of view, i.e. $c_{ij} \neq c_{ji}$.
- For item *i*'s several nearest neighbor, we need to rank c_{ij} in descending order and select most nearest neighbors.
- For the prediction of users' preference for item *i*, we use weight c_{ji}, which measures item *j*'s influence on item *i*.

And, the closeness between an item *i* and another item *j* is not only related to the similarity between them, but also has correlation with the fraction of users with the same preference state for *i* and *j* to the total number of users who give ratings to *i*. Thus, we define the following formula to measure the closeness of two items from user *i*'s point of view:

$$c_{ij} = s_{ij} \times \left|Y_{ij}\right| \Big/ \left|Y_i\right| \tag{33}$$

where s_{ij} is the similarity between *i* and *j*, $\left|Y_{ij}\right|$ is the number of users who have the same preference state for both *i* and *j*, and $|Y_i|$ is the number of users who have rated *i*.

After measuring the closeness of two arbitrary items, the next step is to select nearest neighbors. In this thesis we have the following definitions:

$$N(i) = \left\{ j \mid c_{ij} > 0, j \in I \right\} \tag{34}$$

$$N(i;u) = \left\{ j \mid c_{ij} > 0, r_{uj} > 0, j \in I \right\} \tag{35}$$

where c_{ij} is the closeness from *i* to *j*. Formula (34) means item *i*'s global neighborhood without considering specific user *u*. Formula (35) means item *i*'s neighbors which also are rated by user *u*. From the definitions, we can see that $N(i)$ is fixed for arbitrary item *i*, but $N(i;u)$ is not static and adaptively varies with the target user. Apparently, $N(i;u) \subseteq N(i)$ holds for arbitrary item.

5.3.2 Item influence coefficient

Items' influence coefficient denotes the importance of items in predicting users' preference for other items. In this paper, we use influence coefficient to measure an item's ability to influence another item, and persuade to have the same acceptance rate with the item. We define the influence coefficient of an item *j* on item *i* as follows:

$$p_{ij} = \left|N(j)\right| \cdot \left|Y_{ij}\right| / max_{k \in N(i)} (N(k) \cdot \left|Y_{ik}\right|) \tag{36}$$

where $|N(j)|$ is the number of *j*'s neighbors and $|Y_{ij}|$ is the number of users who have the same preference state for both *i* and *j*. Apparently, items' influence coefficient is normalized, i.e. $0 \le p_{ij} \le 1$ in our definition. Based on this definition, we can see that the more number of neighbors of *j* and the more number of users who have the same preference state for both *i* and *j*, the more ability *v* can influence *u*.

5.3.3 Persuasive influence and supportive influence

On the basis of social impact theory, if a user *u*'s preference state for an item *j* is the same as *u*'s preference state for the target item *i*, we define that *j* have supportive influence on *i*. On the other hand, if *u*'s preference state for an item *j* is different from *u*'s preference state for *i*, then we define that *j* have persuasive influence on *i*. Except for influence coefficient and closeness defined in [60], we also consider the popularity of items, i.e. the more popular an item is, the less influence the other items will have on it, and the vice versa. So if we use the number of ratings item *i* received to measure the popularity of the item, then we can define item *j*'s persuasive influence or supportive influence on *i* with respect to user *u* as follows:

$$w_{ij}^{p} = p_{ij} c_{ij} / log(p + \left|X_i\right|) \tag{37}$$

$$w_{ij}^{s} = p_{ij} s_{ij} / log(s + \left|X_i\right|) \tag{38}$$

where $|X_i|$ indicates the number of ratings item *i* received, parameters *p* and s are used to

distinguish the difference between persuasive influence and supportive influence. Intuitively, the value of *p* is bigger than that of *s*, since it is harder to persuade to change than support to keep the status.

Additionally, we also assume that the combined effect of persuasive influence and supportive influence plays an important role in the prediction of preference. On the basis of social impact theory, the total influence item *i* received with respect to user *u* could be calculated in the following forms:

$$w_i = N_p^{\eta} \sum_{j \in N(i:u)_p} w_{ij}^p - N_s^{\eta} \sum_{j \in N(i:u)_s} w_{ij}^s \tag{39}$$

where $N(i:u)_p$ is the set of items which have opposite acceptance state with *i* by user *u*, N_p is the number of the set, $N(i:u)_s$ is the set of users who have the same acceptance state with *i* by user *u*, N_s is the number of the set, and η is a parameter controlling the contribution of numbers of items who have opposite or the same acceptance state with *i* by *u*.

There are two possible situations by taking combined effect of persuasive influence and supportive influence into account:

- If $w_i \geq 0$, this indicates that the pressure in favor of the preference state change overcomes the pressure to keep the current preference state. Additionally, the bigger the value of w_i, the higher possibility user *u* changes his/her preference state for item *i*.
- If $w_i < 0$, this shows that the pressure to keep the current preference state overcomes the pressure in favor of the preference state change. In addition, the bigger the value of $|w_i|$, the higher possibility user *u* keeps his/her preference state for item *i*.

5.3.4 Preference prediction

Before introducing the preference prediction model, we introduce a closeness threshold parameter μ based on linear threshold model, i.e. an item *i* can be influenced by his/her neighbors if

$$\sum_{j \in N(i;u)} c_{ji} \geq \mu \tag{40}$$

In the closeness threshold model above, we need to notice that we use c_{ji} instead of c_{ij}. This is because we take unbalanced strategy.

As discussed in Section 4.1, a user's predictive preference based on item influence-centric algorithm (item-ICA) is the influence-weighted preference of the target user's preference for the target item's neighbors and the reinforcement of the total influence effect the item received from its neighborhood.

Thus, considering the combined effect of total influence, we use a parameter ρ to control the contribution the effect of the total influence and have the following formula for the prediction of a user *u*'s preference for item *i* based on *u*'s preference for i's neighbors:

$$\tilde{r}_{ui} = \frac{\sum_{j \in N(i;u)_p} r_{uj} w_{ij}^p + \sum_{j \in N(i;u)_s} r_{uj} w_{ij}^s}{\sum_{j \in N(i;u)_p} w_{ij}^p + \sum_{j \in N(i;u)_s} w_{ij}^s} - \rho o_{ui} \times \frac{1 - e^{-w_i}}{1 + e^{-w_i}} \tag{41}$$

where the first term indicates the weighted preference of item *i*'s neighborhood, and the second term addresses the effect of total influence received from neighborhood.

With respect to prediction based on direct influence, this thesis use iterative schema to update the preference:

$$\hat{r}_{ui}^k = \begin{cases} \alpha \tilde{r}_{ui}^{k-1} + (1-\alpha)\hat{r}_{ui}^{k-1}, & k \geq 2 \text{ and closeness threshold is satisfied} \\ r_{ui}^0 & , \ k = 1 \text{ or closeness threshold is not satisfied} \end{cases} \tag{42}$$

where $\tilde{r}_{ui}^{k-1}$is the prediction of user *u*'s preference for item *i* based on *u*'s preference *i*'s neighbors' direct influence during the *k*-th iteration.

In summary, the process of direct influence algorithm is as follows:

1. Measuring items' closeness and selecting a target item's nearest neighbors using unbalanced strategy related to target user.
2. Computation of an item's influence coefficient which denotes the user's ability to influence another user, and persuade to have the similar preference (the same preference state) with the user.
3. Computation and analysis of persuasive influence and supportive influence, and ana-

lyzing the total influence effect of these two types of influence.

4. Prediction of preference based on item's neighborhood and total influence effect by formula (41) and (42) under the condition of closeness threshold is satisfied.
5. If the closeness threshold is not satisfied, then repeat step 3 and 4.

5.4 Preference based on indirect influence

5.4.1 Preference influenced by the structure of neighborhood

In the case of item influence-centric method, indirect influence is the impact of the structure of an item's neighbors. Like user influence-centric method, we consider the influence of the structure of items' neighborhood as indirect influence, and try to discover several useful and meaningful latent attributes of the structure, such as group size difference, total influence difference and average popularity difference, in item-based CF systems. Then, corresponding correlations between users' preference and these attributes by several popular data mining techniques are suggested.

Technically speaking, we partition a target item *i*'s neighbors rated by user *u* into persuasive group and supportive group. Persuasive group $N^p(i;u)$ contains items, which are rated by *u* and have opposite acceptance state with *i*, i.e.

$$N^p(i;u)=\{j\,|\,r_{uj}>0, c_{ij}>0, o_{ui}o_{uj}<0, j\in I\} \tag{43}$$

Supportive group $N^s(i;u)$ contains items, which are rated by user *u* and have the same acceptance state as *i*, i.e.

$$N^s(i;u)=\{j\,|\,r_{uj}>0, c_{ij}>0, o_{ui}o_{uj}>0, j\in I\} \tag{44}$$

The number of items in each group is $|N^p(i;u)|$ and $|N^s(i;u)|$ respectively. Apparently, $N^p(i;u)\bigcup N^s(i;u)=N(i;u)$ and $N^p(i;u)\bigcap N^s(i;u)=\varnothing$. If we define the integrated acceptance of each group as the average acceptance by target user *u*, then we have integrated preference of persuasive group is

$$r_{iu}^p=\frac{\sum_{j\in N^p(i;u)} r_{uj}}{|N^p(i;u)|} \tag{45}$$

and integrated preference of supportive group is

$$r_{iu}^{s} = \frac{\sum_{j \in N^{s}(i;u)} r_{uj}}{\left| N^{s}(i;u) \right|} \tag{46}$$

The aim of indirect influence algorithm is to find useful and meaningful latent attributes between the persuasive group and supportive group, and predict whether the target user would like to take the integrated acceptance of persuasive group or supportive group.

Assuming there is a set of latent attributes $A = \{a_1, a_2, \cdots, a_l\}$ for arbitrary items' neighbors related to user *u*, and then the prediction becomes a classification problem, i.e. each observed item *i*'s latent attributes of neighborhood consists of a pair: a attributes vector $a_{iu} = \{a_{i1}, a_{i2}, \cdots a_{il}\}$, where $a_{iu} \in R^{l}$, and the associated label y_{iu},where $y_{iu} \in \{r_{iu,}^{p} r_{iu}^{s}\}$. The classification task is to learn the mapping $a_{iu} \mapsto y_{iu}$. In the next section, we define several rated latent attributes of item neighborhood.

5.4.2 Mining latent attributes

We use the same data sets as Section 5.2.2 to conduct experiments, and after several trials, we have the following latent attributes in items' neighborhood:

- Group Size Difference (GSD): the difference of size of persuasive group and supportive group. This attribute measures the difference of two groups in terms of neighbors' number. If GSD>0, it indicates that the persuasive group contains more items who are liked by the target user than that of supportive group, and vice versa.

$$GSD = \frac{\left| N^{+}(i;u) \right| - \left| N^{-}(i;u) \right|}{\left| N^{+}(i;u) \right| + \left| N^{-}(i;u) \right|} \tag{47}$$

- Total Influence Difference (TID): the difference of influence of persuasive group and supportive group on target item. This attribute measures the difference of closeness between two groups and the target item. If TID>0, the persuasive group is much closer to the target item than that of supportive group, and vice versa.

$$TID = \sum_{j \in N^{+}(i;u)} w_{ij}^{p} - \sum_{j \in N^{-}(i;u)} w_{ij}^{s} \tag{48}$$

- Average Popularity Difference (APD): the difference of average popularity between persuasive group and supportive group. This attribute measures difference of two groups in terms of the average popularity. APD>0 indicates that the persuasive group is much more popular than that of supportive group on average, and vice versa. In the following computation for APD, $|Y_j|$ denotes the ratings received by item j.

$$APD = \frac{\sum_{j \in N^+(i;u)} |Y_j|}{|N^+(i;u)|} - \frac{\sum_{j \in N^-(i;u)} |Y_j|}{|N^-(i;u)|} \tag{49}$$

Take Netflix dataset for example, the correlation of these attributes and users' preference tendency is as Figure 11 shows. The red points mean the target user would like to take the opinion of persuasive group for the target item, and the blue points mean users would like to take the opinion of supportive group for the target item. From this figure, we can see that the two classes do not have distinct boundaries, but still are separated into two parts. Furthermore, if we set a threshold to GSD, for example |GSD|>=0.1, then we obtain a result as Figure 12 shows. We can see that the two classes are divided into two parts clearly except several points.

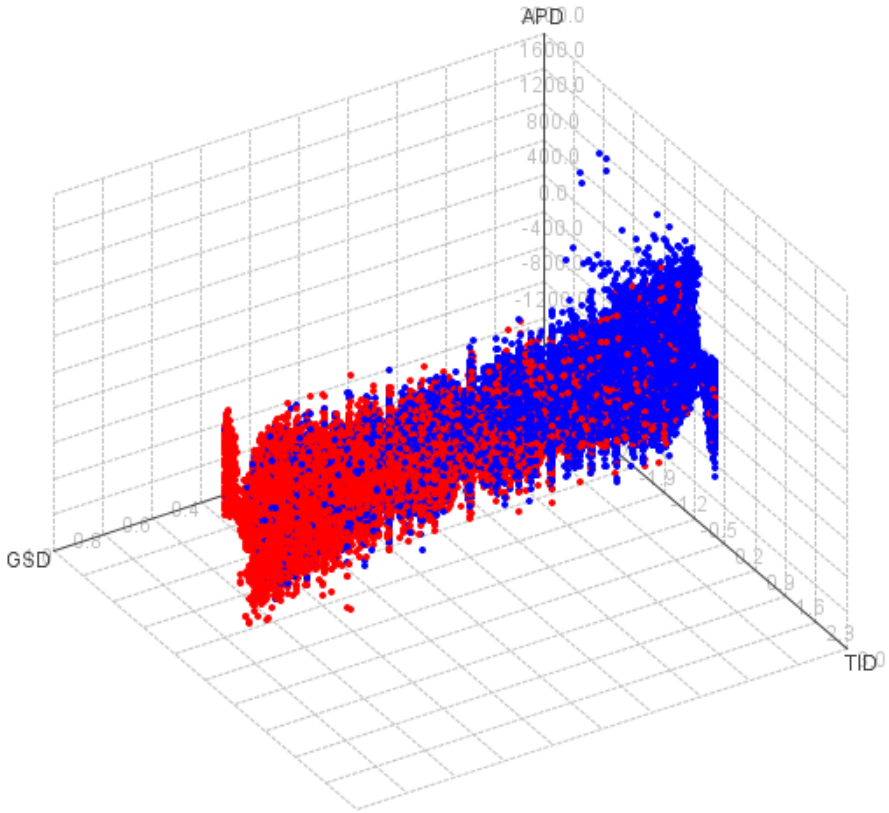


Figure 11 Correlations of latent attributes of target item's neighbors and user's preference

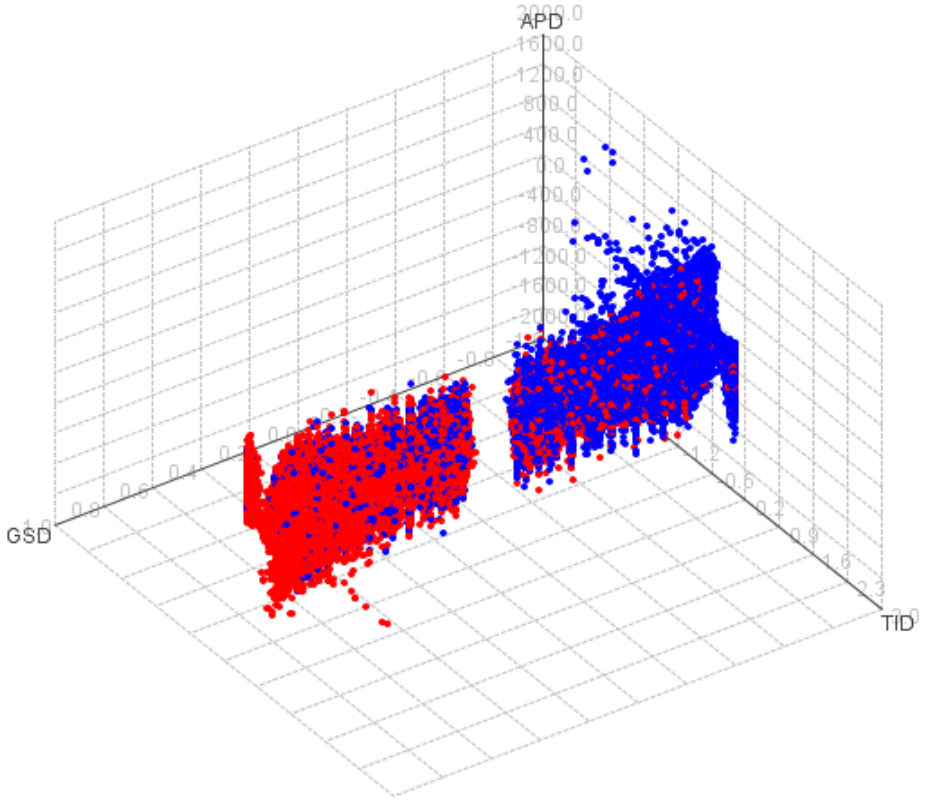


Figure 12 Correlations of latent attributes of target item's neighbors and user's preference given |GSD|>=0.1

Moreover, we also investigate the pair correlation of these three attributes and users' preference as Figure 13, 14 and 15 show. From these figure, we can see that the greater of the absolute value of these attributes, the higher coverage that target user would like to take integrated opinion of one of the groups.

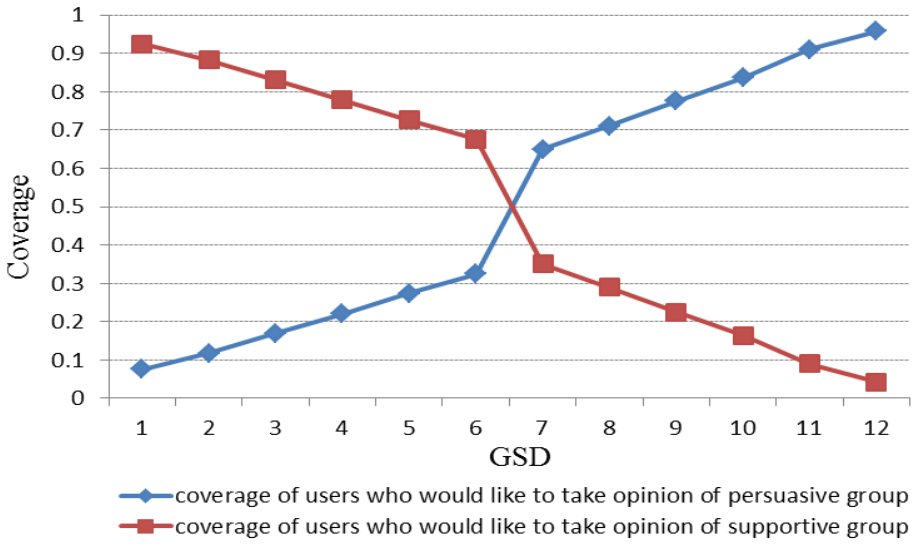


Figure 13 Correlations of GSD and user's preference

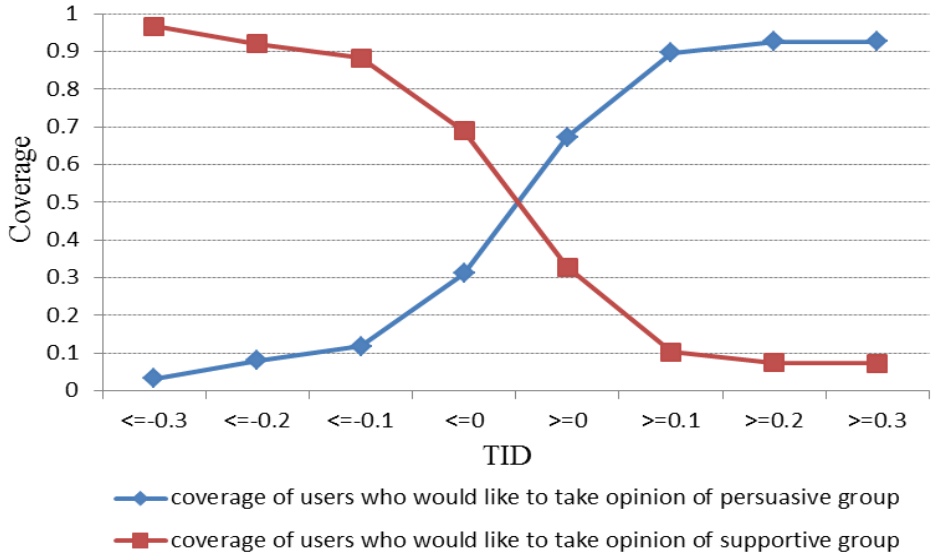


Figure 14 Correlations of TID and user's preference

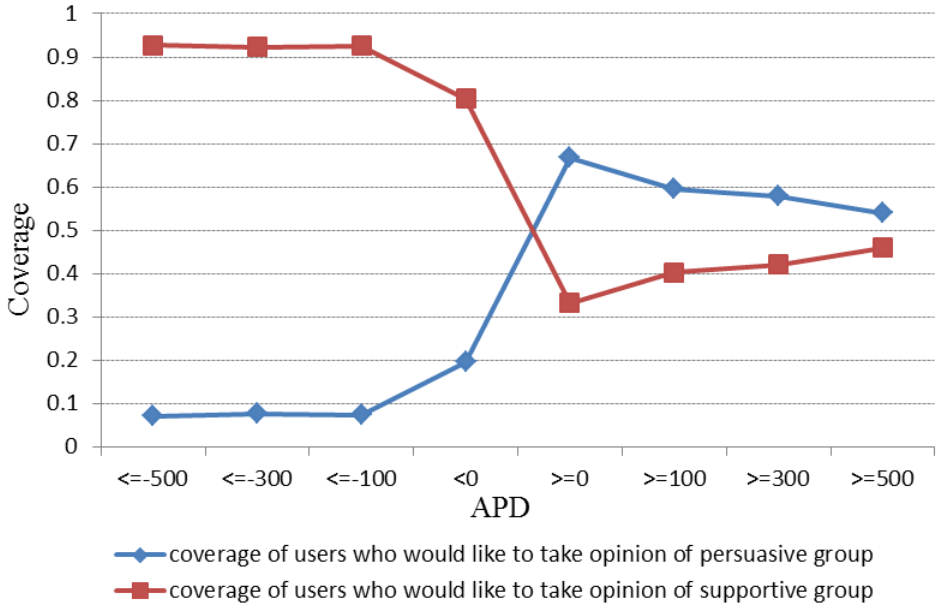


Figure 15 Correlations of APD and user's preference

5.4.3 Implementation

Based on Section 5.4.2, we have three latent attributes, i.e. GSD, TID, APD, for each item *i*'s neighborhood related to user *u*. That is, $A = \{a_1, a_2, \cdots, a_k\} = \{GSD, TID, APD\}$. Accordingly, the label is $y_{iu} \in \{r_{iu}^p, r_{iu}^s\}$. As discussed in Section 5.4.1, the user indirect influence algorithm is a classification problem. Besides, from Section 5.4.2 we can see that we can obtain better classi-

fication if we set thresholds for one or more attributes. Thus, before implementing classification algorithms, we firstly define thresholds for attributes. In order to avoid too complicated, we just use one threshold π for GSD, i.e., $|GSD| \geq \pi$, in this thesis. That is, we only implement classification algorithms on these cases that satisfy GSD threshold.

In this thesis, since the prediction task item based on indirect influence model is also a classification task, we use the following data mining/machine learning techniques, which are shown details in Section 4.4.3, to predict users' preference based on the three latent attributes discussed in Section 5.4.2.

- Decision Tree
- Naive Bayesian
- Neural Networks
- Support Vector Machines

5.5 Summary

In summary, this chapter presents the process of item influence-centric recommendation algorithm. Firstly, we defines the basic idea of item influence-centric recommendation model, and then present preference prediction based on direct influence of item neighbors and the effect of total influence, and finally show how to predict users preference based on indirect influence of items' neighborhood by mining and selecting the latent attributes of neighborhood.

6 Experimental Evaluation

6.1 Introduction

This chapter presents the experimental evaluation methods and discusses correspondence results. We use two popular benchmark dataset: MovieLens dataset and Netflix dataset, to conduct experiment. With respect to evaluation metrics, we use Root Mean Squared Error and accuracy. RMSE measures how accurate an algorithm predicts ratings users will rate items, and accuracy measures whether an algorithm properly predicts that the user like or dislike recommended items.

In the experiments, we investigated the performance of unbalanced strategy for neighbor selection, direct influence model, indirect influence model and the total performance of the combination of the two models.

6.2 Data sets

In this chapter, we use the following benchmark data sets, which are often used in the area of recommendation systems, to verify our research:

- **MovieLens dataset**. The data was collected through the MovieLens web site (movielens.umn.edu) during the seven-month period from September 19th, 1997 through April 22nd, 1998. This data has been cleaned up – users who had less than 20 ratings or did not have complete demographic information were removed from this data set. The dataset contains 100,000 ratings (user-item-rating) with rating scale 1-5 and consists of 943 users and 1,682 movies.
- **Netflix dataset**. Netflix is an online-DVD rental company and held a competition for the best collaborative filtering algorithm to predict user ratings for movies in 2007. The dataset provided by Netflix had a training dataset of 100,480,507 ratings that 480189 users gave to 17,770 movies. We randomly select 119,259 ratings gave by 3,492 users to 673 movies from the original dataset.

Both Moivelens dataset and Netflix dataset are real and provided by online users. As standard dataset, these two datasets have been used by a plenty of research in this area such as [10], [19], [20], [42], [45], [46]. The advantage of using such benchmark datasets is that it helps researchers compare their work with others, as well as the traditional recommendation algorithms. But, unfortunately, the main problem of the datasets is sparsity. For instance, the density of NetFlix dataset is only 5.1%, which means there are 5.1% none-empty entries in user-item-rating matrix, and the rest are empty. Table 4 shows the statistical characteristics of the datasets.

Table 4 Statistical characteristic of datasets

	Users	**Items**	**Ratings**	**Density**
MovieLens	943	1682	100000	6.3%
NetFlix	3492	673	119259	5.1%

6.3 Evaluation metrics

In this thesis, we use Root Mean Squared Error (RMSE) and Accuracy to verify the performance of proposed algorithms [78]. RMSE is the most used metric to measure rating prediction accuracy. The RMSE between the predicted and actual ratings is given by:

$$RMSE = \sqrt{\frac{\sum (\hat{r}_{ui} - r_{ui})^2}{|testset|}} \tag{50}$$

Accuracy indicates the fraction of true predictions to the actual value. That is, the accuracy is the proportion of true results (both true positives and true negatives) in the prediction. The definition is as follows:

	Condition	
Test outcome	True positive (tp)	False positive (fp)
	False negative (fn)	True negative (tn)

$$Accuracy = \frac{\#tp + \#tn}{\#tp + \#fp + \#fn + \#tn} \tag{51}$$

6.4 Experimental results and analysis

6.4.1 Evaluation of unbalanced strategy for neighbor selection

In this thesis we use unbalanced strategy for neighbor selection. This section evaluates the unbalanced strategy for neighbor selection. We use 5-fold cross validation in the experiment. That is, we randomly divide the dataset into 5 equal size subsets. In each time, a single subset is retained as the validation data for testing the performance of prediction, and the remaining 4 (5-1) subsets are used as training data. Then the cross-validation is repeated 5 times. The 5 results from the folds then are averaged to produce the final estimation of performance.

Table 5 Method of validation

1th fold	Train	Train	Train	Train	Test
2th fold	Train	Train	Train	Test	Train
3th fold	Train	Train	Test	Train	Train
4th fold	Train	Test	Train	Train	Train
5th fold	Test	Train	Train	Train	Train

Based on the method above, the experiments results are shown in Figure 15 and16. We observed that the unbalanced neighbor selection method can give better performance in terms of RMSE and accuracy than traditional neighbor selection method on both dataset. In addition, it is clear that it obtains the best RMSE performance just needing 20 nearest neighbors. However, it needs 50 nearest neighbors to get the best RMSE performance on MovieLens data by using traditional neighbor selection method. With respect to Accuracy, the results are more interesting and show different types of sensitivity. From Figure 16, we can see that it just needs the first nearest neighbor to get the best Accuracy performance, and with the increasing of number of nearest neighbors, the accuracy keeps decreasing by using unbalanced neighbor selection method. On the other hand, traditional neighbor selection method improves as we increase the neighborhood size from 1 to 30, after that the rate of increase diminishes and the curve tends to decline.

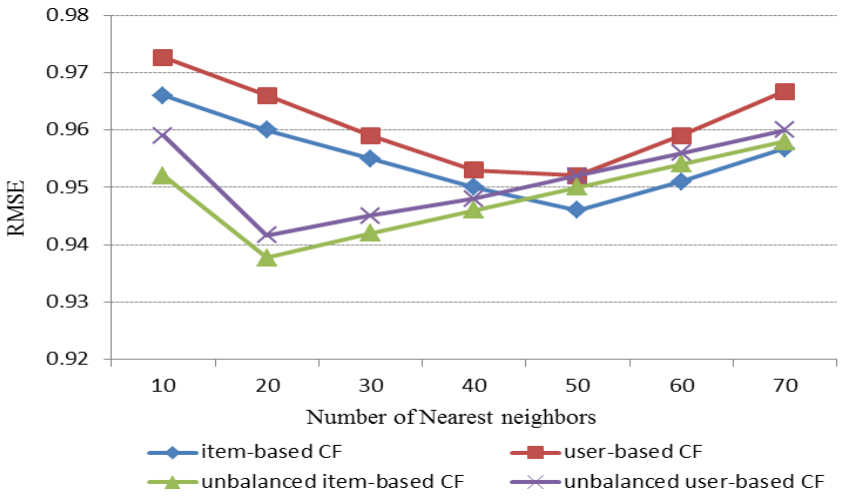


Figure 16 Performance of neighbor selection methods on Movielens in terms of RMSE

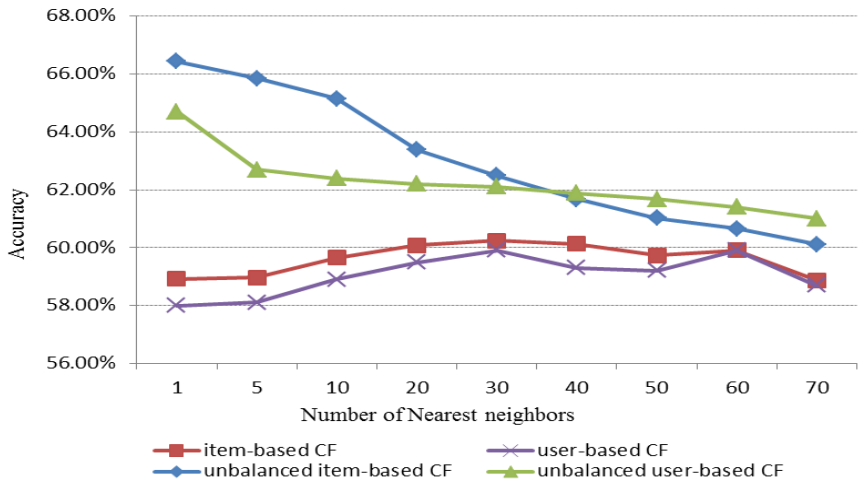


Figure 17 Performance of neighbor selection methods on Movielens in terms of Accuracy

The experiment shows similar results on Netflix dataset in terms of RMSE and Accuracy as figure 17 and 18 show. Obviously, the unbalanced neighbor selection method can give better performance in terms of RMSE and Accuracy than traditional neighbor selection method on both dataset. It obtains best RMSE performance just needing 30 nearest neighbors. However, it needs 60 nearest neighbors to get the best RMSE performance by using traditional neighbor selection method. With respect to Accuracy, the results are also interesting and show different types of sensitivity. From figure 18, we can see that it just needs the first nearest neighbor to get the best Accuracy performance, and with the increasing of number of nearest neighbors, the accuracy keeps decreasing by using unbalanced neighbor selection

method. On the other hand, traditional neighbor selection method improves as we increase the neighborhood size from 1 to 60, after that the rate of increase diminishes and the curve tends to decline.

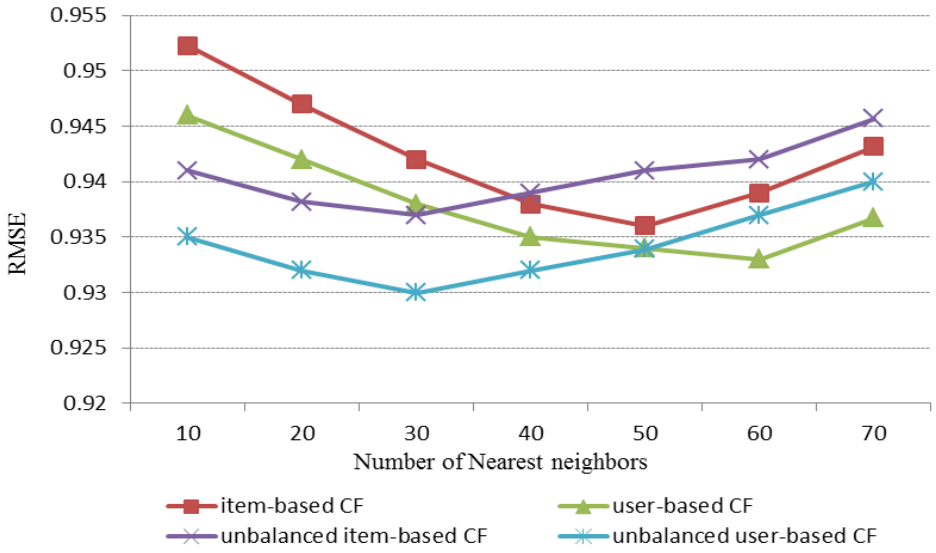


Figure 18 Performance of neighbor selection methods on Netflix in terms of RMSE

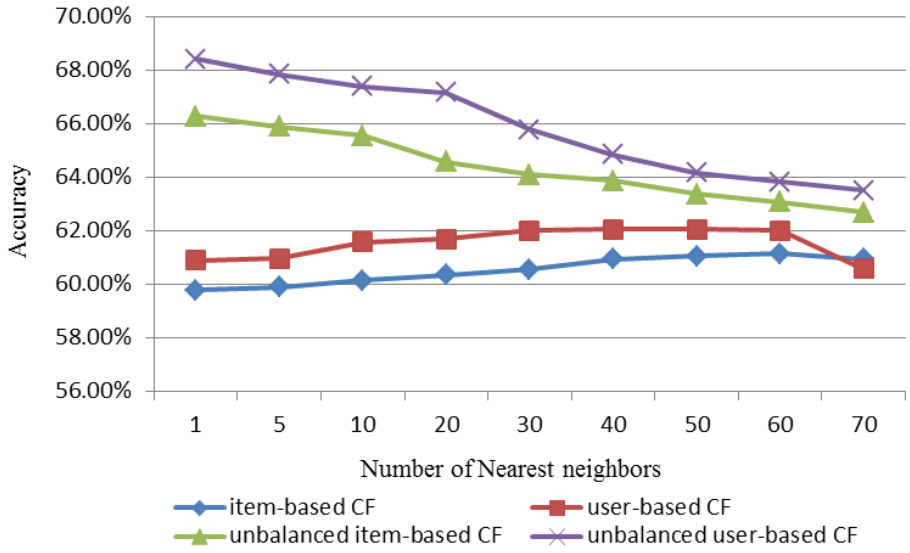


Figure 19 Performance of neighbor selection methods on Netflix in terms of Accuracy

The possible reason for these results might be the application of unbalanced strategy for neighbor selection. By using unbalanced strategy, we firstly select several most nearest neighbors from the target user or item's point of view, and then compute closeness-weighted preference of the neighborhood, in which the closeness is from the neighbors' point of view.

This strategy can measure the relationships between users or items much more accurate and efficient compared to traditional methods,

6.4.2 Evaluation of direct influence model

In this section, we investigated the performance of item direct influence model on MoiveLens dataset. Since only the cases that satisfy the requirement of closeness threshold can be predicted by direct influence model, so the experimental method is a little different. In order to study the relationship between threshold of closeness and RMSE, accuracy, coverage of satisfied cases on different training/test ratio (*y*), we randomly divided the dataset into 10 equal size subsets. In each time, *x* subsets is retained as the validation data for testing the performance of prediction, and the remaining (10-*x*) subsets are used as training data. Besides, if we assume there are *z* cases satisfy the threshold of closeness, then the performance of prediction is defined as (52) and (53) shows.

Table 6 Evaluation method of direct influence model

y=9/1	Train	Train	Train	Train	Train	Train	Train	Train	Train	Test
y=8/2	Train	Train	Train	Train	Train	Train	Train	Train	Test	Test
y=7/3	Train	Train	Train	Train	Train	Train	Train	Test	Test	Test
y=6/4	Train	Train	Train	Train	Train	Train	Test	Test	Test	Test
y=5/5	Train	Train	Train	Train	Train	Test	Test	Test	Test	Test
y=4/6	Train	Train	Train	Train	Test	Test	Test	Test	Test	Test
y=3/7	Train	Train	Train	Test	Test	Test	Test	Test	Test	Test

$$RMSE = \sqrt{\frac{\sum (\hat{r}_{ui} - r_{ui})^2}{z}} \tag{52}$$

$$Accuracy = \frac{\#tp + \#tn}{z} \tag{53}$$

The parameters results of Movielens dataset are shown in Figure 20, 21 and 22. Figure 20 shows the relationship of RMSE and threshold of closeness on different train/test ratio. From the figure, we observe that the RMSE decreases as we increase the closeness threshold θ

from 0 to θ^o, where θ^o < 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, and after that the value of RMSE increases. For instance, with regard to train/test ratio *y*=4/6, the method get best RMSE performance when θ^o = 3. On the other hand, when the train/test ratio *y* = 8/2, the method get best RMSE performance at θ^o = 7. Similarly, with respect to the relationship of closeness threshold and accuracy, we observe that the accuracy improves as we increase the closeness threshold from 0 to θ^o, where θ^o <1,2,3,4,5,6,7,8,9,10, and after that the value of accuracy decreases. For example, when *y* = 4/6, the method gets best accuracy performance at θ^o = 6, which is different from that of RMSE.

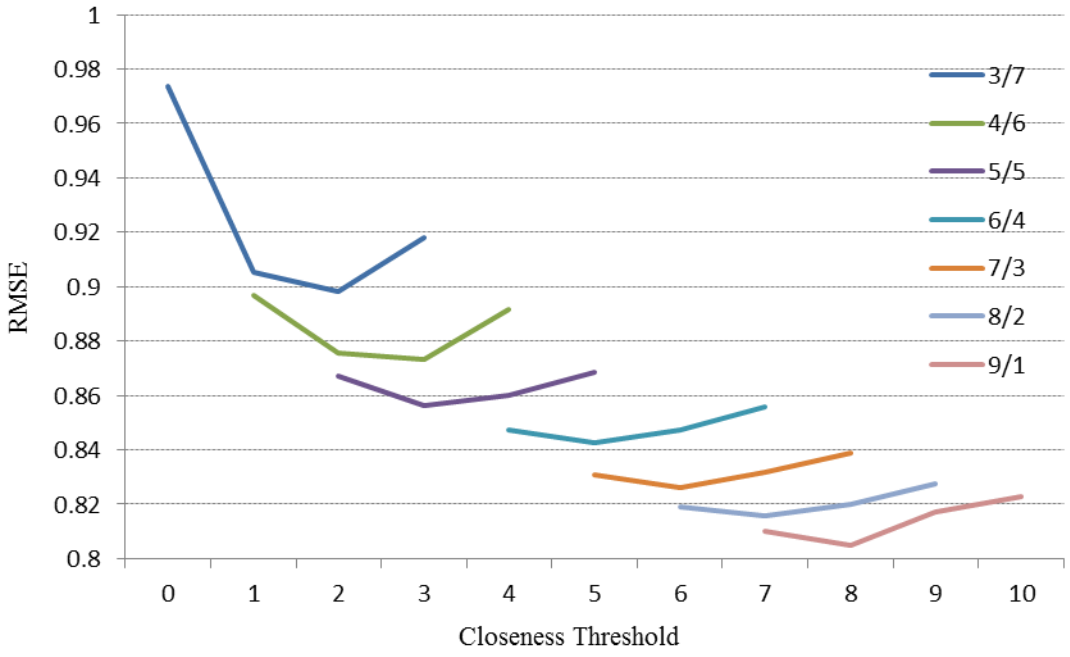


Figure 20 The correlation of threshold of closeness and RMSE on Movielens data

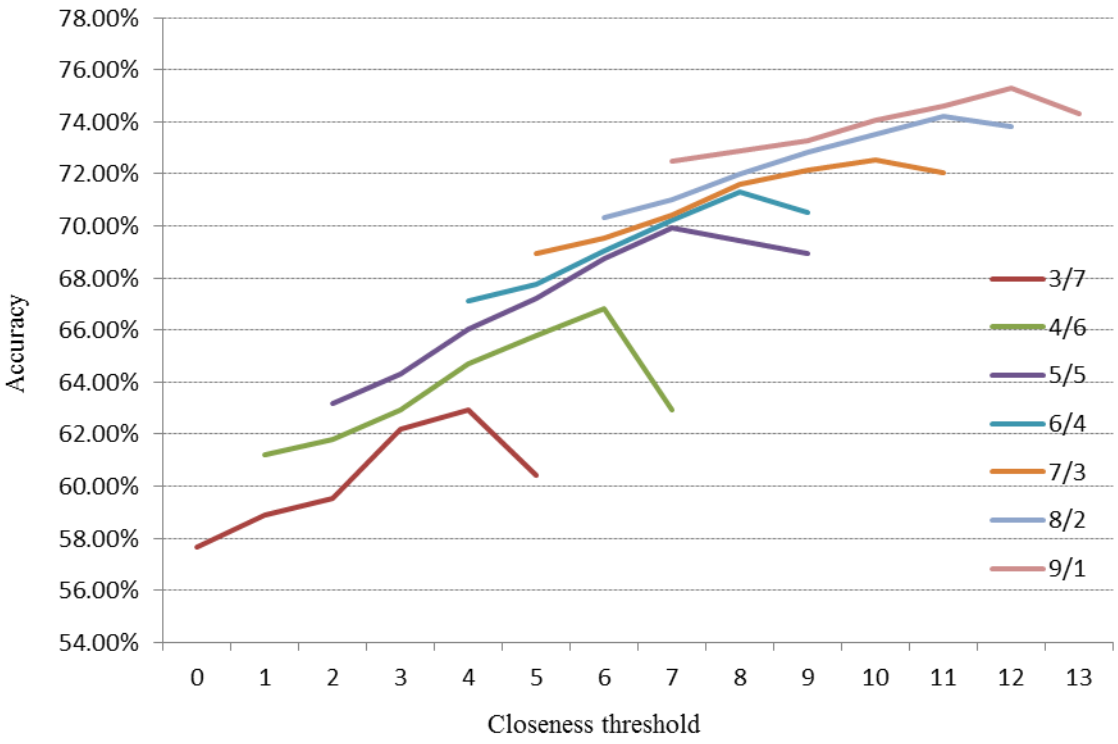


Figure 21 The correlation of persuasive factor and supportive factor and Accuracy

From figure 20 and 21, we can see that the direct influence model has a clear advantage for the cases fulfilling the closeness threshold. But from Figure 22, we find that the coverage of satisfied cases sharply decreases as the closeness threshold increases. For example, when y=8/2 and θ^o = 3, the ratio of satisfied cases is about 0.12, that is, there are only a very small fraction of cases satisfy the closeness threshold and can be predicted by direct influence model.

Hence, there is a tradeoff between prediction performance and coverage of satisfied cases. When the training/test ratio is smaller, the performance is worse than that that is greater. On the other hand, when we increase the coverage of satisfied cases, the performance improves as the increasing of closeness threshold.

The analysis above indicates the rule of closeness threshold. That is, when the density of a data set is sparse, where ratio of data set is smaller, it is better to choose a smaller value of closeness threshold. With the increasing of data density, a moderate value is a good choice. Since if the value of closeness threshold too greater, the ratio of satisfied cases will very low, although the prediction performance improves for these satisfied cases, there are a plenty of

cases that cannot be predicted by direct influence model, then the final performance may not be good.

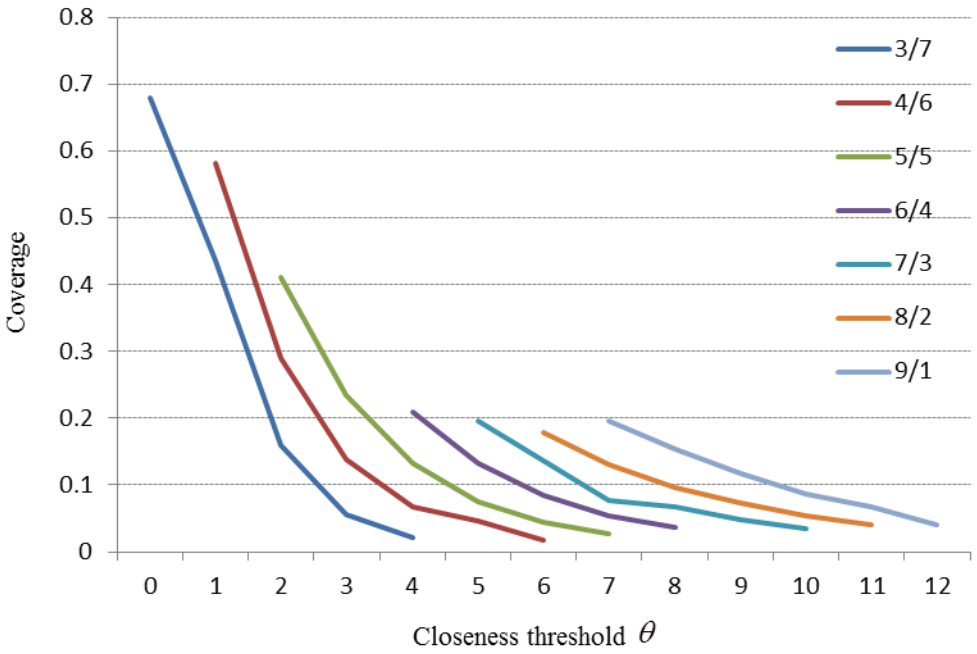


Figure 22 The correlation of coverage and closeness threshold

In addition, we also investigate the relationship of parameter persuasive factor *p* and supportive factor *s*. Figure 22 and 23 show results at training/test ratio *y* = 8/2 and closeness threshold θ = 7. From Figure 22 we observe that the RMSE performance improves when we increase persuasive factor *p*, given a fixed supportive factor *s*. On the contrary, the RMSE performance decreases when we increase supportive factor *s*, given a fixed persuasive factor *p*. As a whole, the result shows better performance when *s*<*p*, which means persuasive factor *p* is greater than supportive factor *s*.

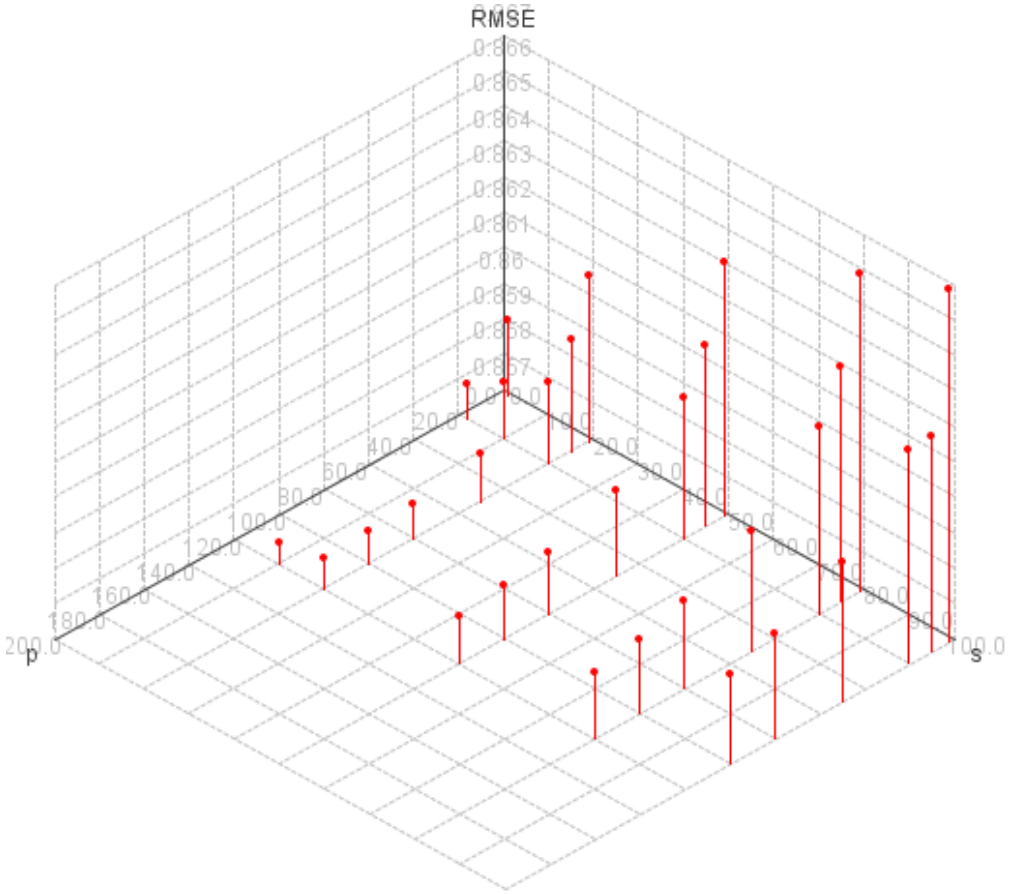


Figure 23 RMSE for different persuasive factor (*p*) and supportive factor (*s*)

With respect to accuracy performance, the experiment shows similar results (Figure 24). From Figure 24, we observe that the accuracy performance improves when we increase persuasive factor *p*, given a fixed supportive factor *s*. On the contrary, the accuracy performance decreases when we increase supportive factor *s*, given a fixed persuasive factor *p*. As a whole, the result shows better performance when $s<p$, which means persuasive factor *p* is greater than supportive factor *s*.

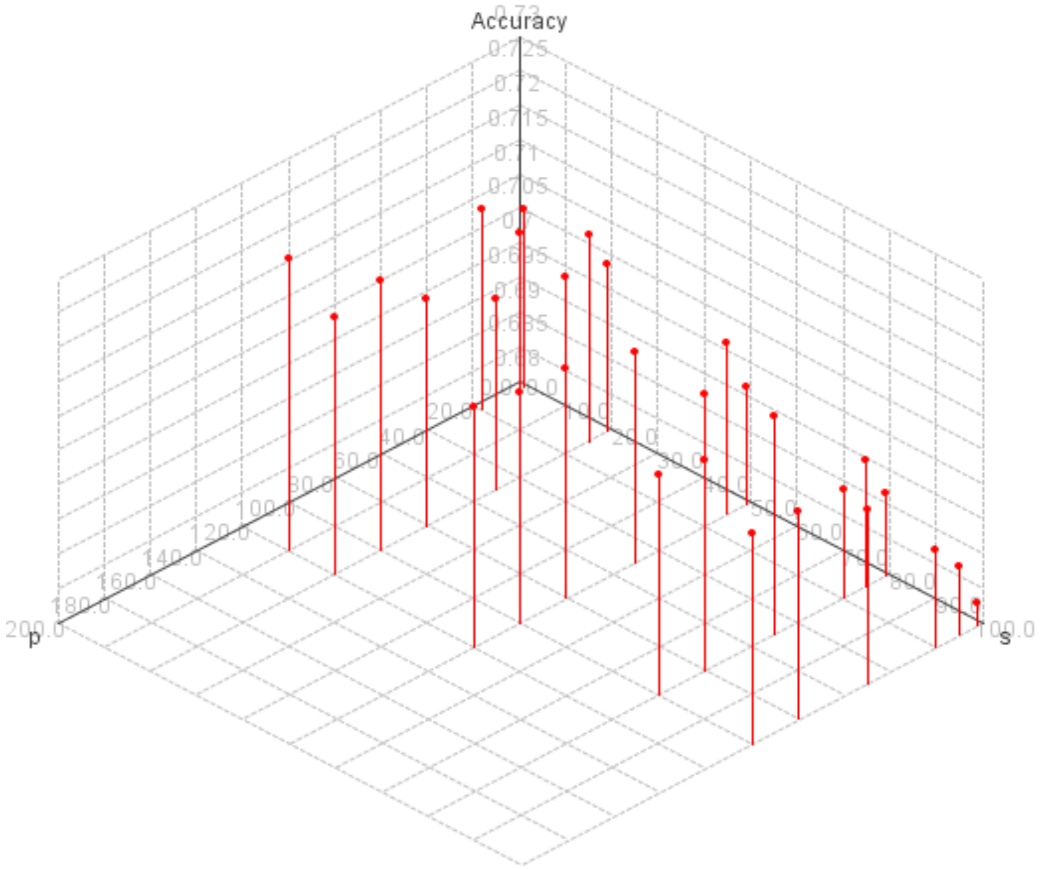


Figure 24 The correlation of persuasive factor and supportive factor and RMSE

Applying these results to the computation of persuasive influence and supportive influence, we can see that the persuasive influence is less than supportive influence given the same value of influence coefficient, closeness and popularity of item. This indicates that it is harder to impose persuasive influence than to impose supportive influence. Considering the meanings of persuasive influence and supportive influence, it reveals that a neighbor gives more influence, if he/she has similar opinion with the target user, than that if they have opposite opinion.

6.4.3 Evaluation of indirect influence model

This part investigates the performance of indirect influence model under different GSD threshold on MovieLens dataset. The experiment method is similar to that in Section 6.4.1. We use RapidMiner[1], which is a popular and powerful tool for data integration, transfor-

[1] http://rapidminer.com/

mation, modeling and visualization, to lean indirect influence model and predict users' preference. The learning process of RapidMiner is shown in Figure 25.

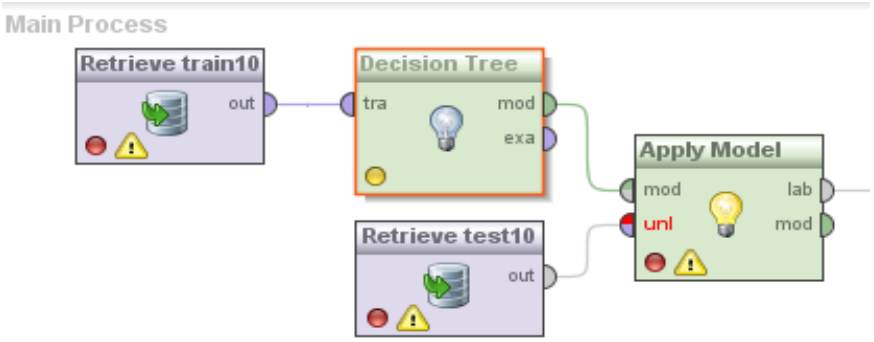


Figure 25 Rapidminer learning process

Figure 26 and 27 show the scatter plot of preference distribution of user indirect influence model with GSD threshold = 0 and 0.3 on training data. The blue points mean user would like to take the integrated opinion of persuasive group, and red point mean user would like to take the integrated opinion of supportive group. Based on the three latent attributes GSD, TID, SDD, the data points (preferences) roughly distribute on two areas. In Figure 26, where the GSD threshold = 0, the areas are overlapped and there is no clear boundary between them. But in figure x, we observe that the areas are divided clearly with GSD threshold = 0.3 except a few points.

These results reveal that there are correlations of preference and latent attributes GSD, TID, SDD. In addition, it is apparent that the greater the value of GSD, the more blue points, which means that users would like to take the integrated opinion of persuasive group. On the contrary, the lower the value of GSD, the more red points, which means that users would like to take the integrated opinion of supportive group.

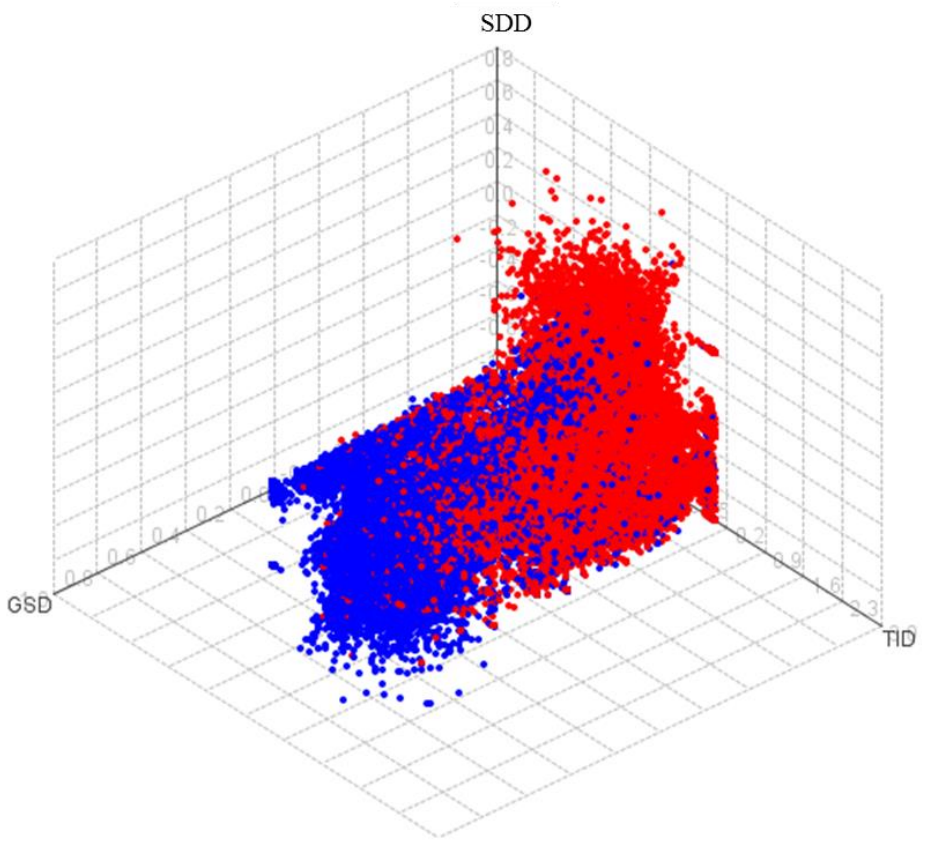


Figure 26 User indirect influence model with GSD threshold = 0

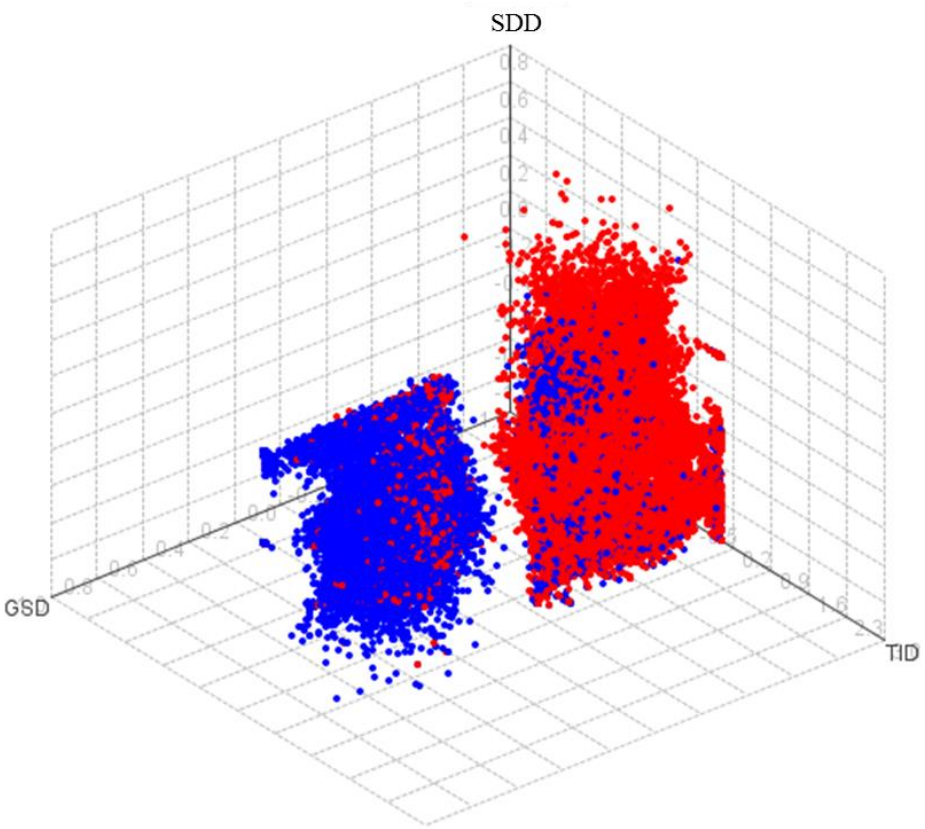


Figure 27 User indirect influence model with GSD threshold = 0.3

Similarly, Figure 28 and 29 show the scatter plot of preference distribution of item indirect influence model with GSD threshold = 0 and 0.1 on training data. The blue points mean user would like to take the integrated opinion of persuasive group, and red point mean user would like to take the integrated opinion of supportive group. Based on the three latent attributes GSD, TID, APD, the data points (preferences) roughly distribute on two areas. In Figure 28, where the GSD threshold = 0, the areas are overlapped and there is no clear boundary between them. But in Figure 29, we observe that the areas are divided clearly with GSD threshold = 0.1 except a few points.

These results reveal that there are correlations of preference and latent attributes, GSD, TID and APD. In addition, it is apparent that the greater the value of GSD, the more blue points, which means that users would like to take the integrated opinion of persuasive group. On the contrary, the lower the value of GSD, the more red points, which means that users would like to take the integrated opinion of supportive group.

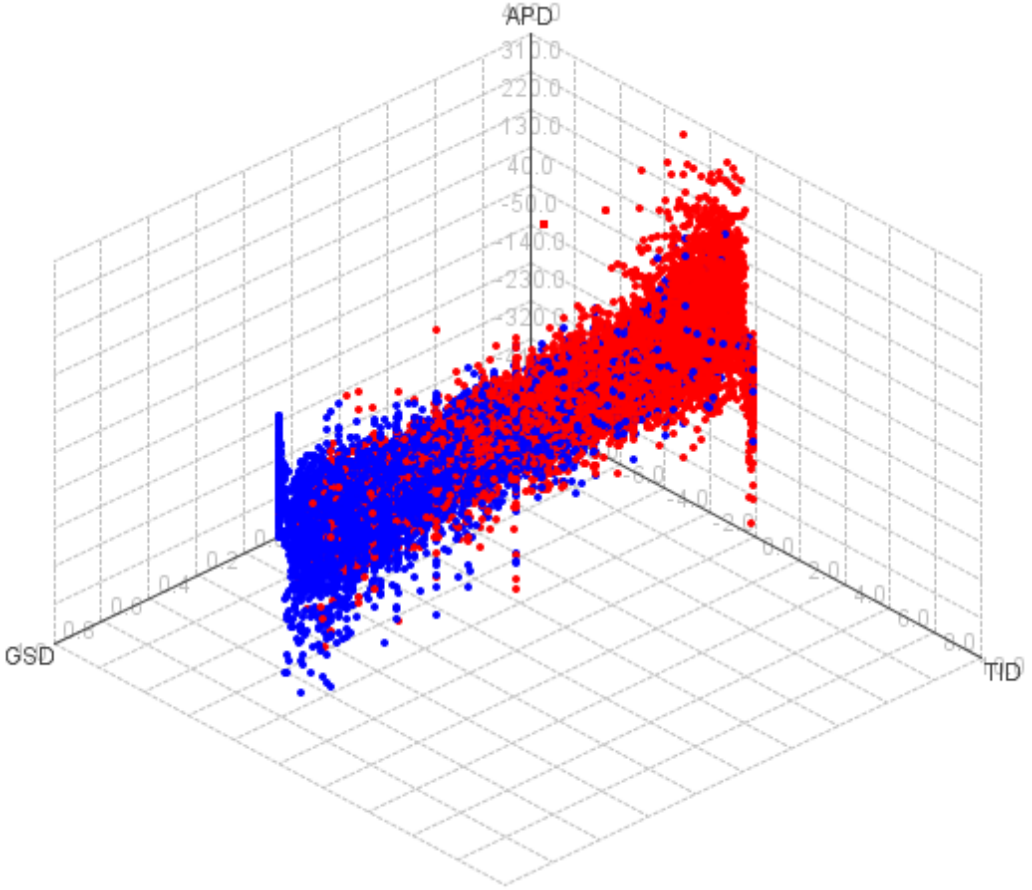


Figure 28 Item indirect influence model with GSD threshold = 0

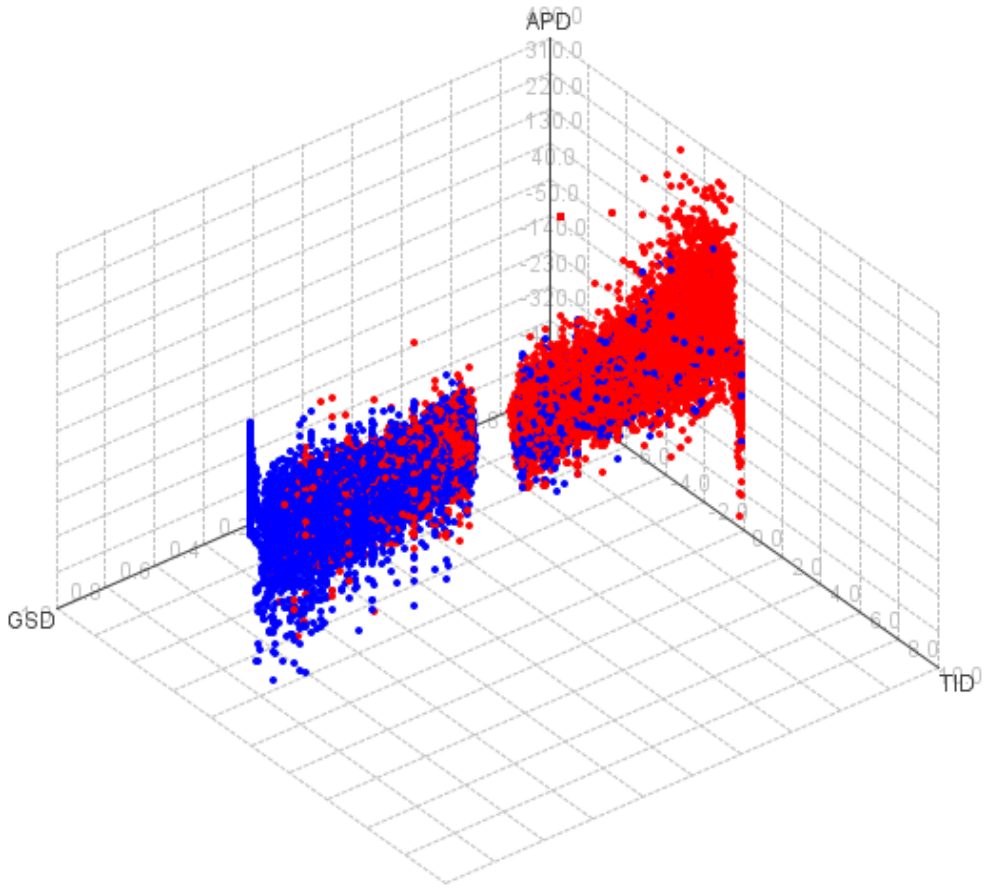


Figure 29 Item indirect influence model with GSD threshold = 0.1

Figure 30, 31, and 32 show the performance of item indirect influence model at train/test ratio = 8/2. From Figure 30, we observe that the RMSE performance improves with the increasing of GSD threshold. Similarly, Accuracy performance shows the same tendency with increased value of GSD threshold. This indicates that the indirect influence model can give more accurate predictions for those cases that satisfied the GSD threshold with the increased value of threshold.

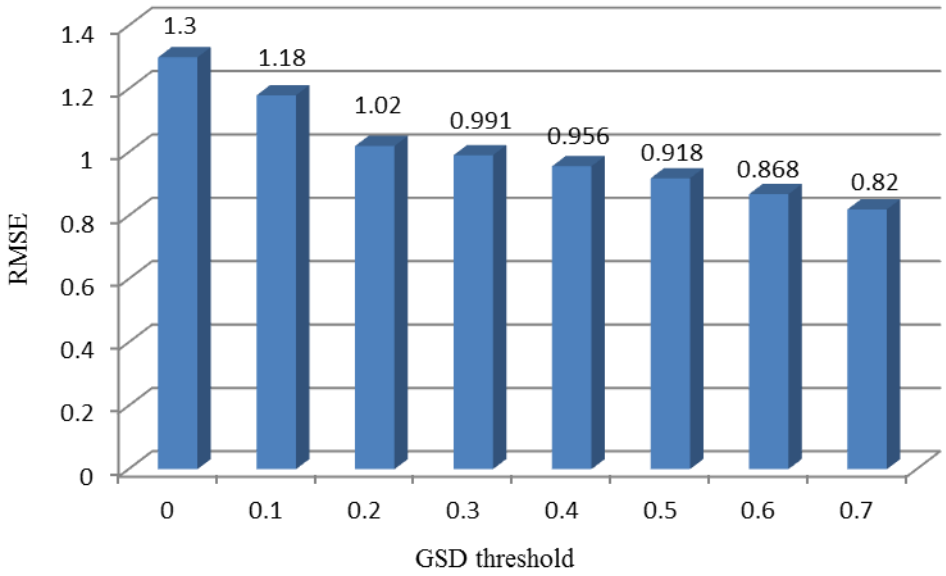


Figure 30 The performance of RMSE at difference GSD threshold

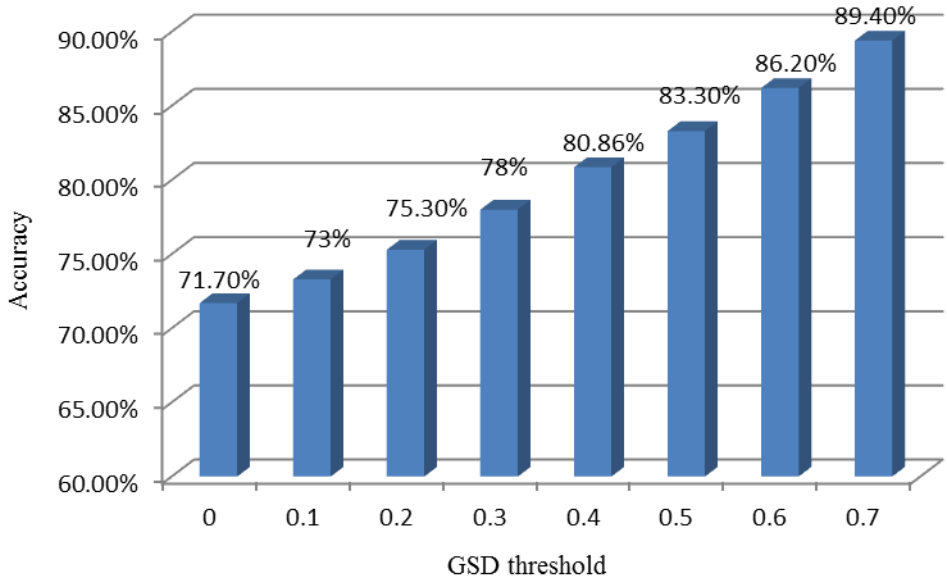


Figure 31 The performance of Accuracy at difference GSD threshold

Similar to direct influence model, from Figure 30 and 31, we can see that the indirect influence model has a clear advantage for the cases satisfying the GSD threshold. But from Figure 31, we find that the coverage of satisfied cases sharply decreases as the GSD threshold increases. For example, when t = 0.7, the coverage of satisfied cases is about 0.09, that is, there are only a very small fraction of cases satisfied the GSD threshold and can be predicted by direct influence model.

Therefore, there is also a tradeoff between prediction performance and coverage of satis-

fied cases. When the coverage is smaller, the performance is worse than that the coverage is greater. On the other hand, if we increase the coverage of satisfied cases, the performance will decline. This indicates that if the value of GSD threshold too greater, the ratio of satisfied cases will be very low, although the prediction performance improves for these satisfied cases, there are a plenty of cases that cannot be predicted by indirect influence model, then the final performance may be not good.

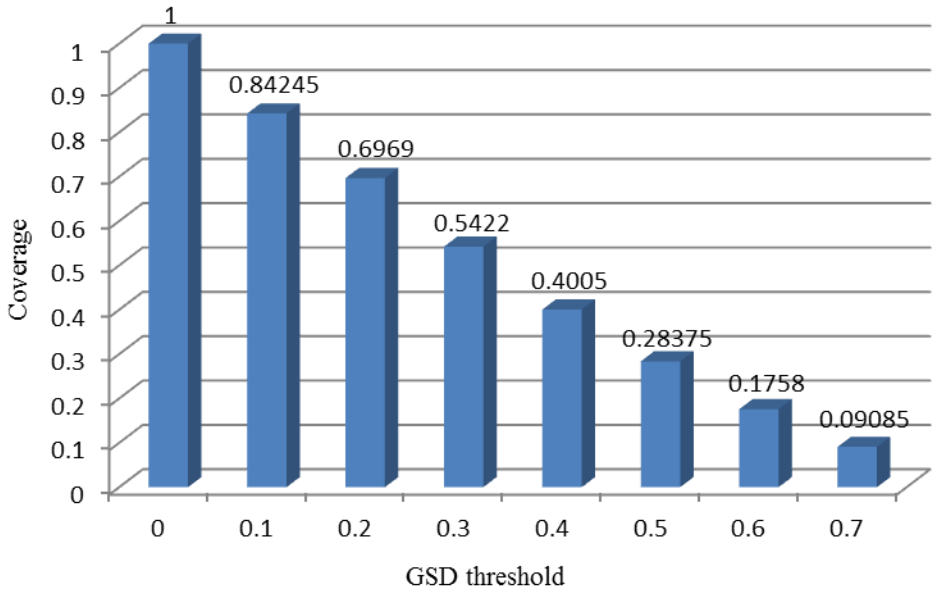


Figure 32 The coverage at difference GSD threshold

6.4.4 Evaluation of total performance

Since the proposed influence-centric recommendation algorithms integrate direct influence and indirect influence model, this section presents the total performance of proposed methods, which combine direct influence and indirect influence model. Table 7 and Table 8 show the main optimal parameters in combined methods on MovieLens dataset and Netflix dataset respectively. These optimal parameters are determined by empirical evaluation.

Table 7 Parameters in combined methods on MovieLens

Algorithms	Parameters	Description	Value
User-ICR	η	contribution of numbers of users	1.2
	p	Persuasive factor	150

Algorithms	Parameters	Description	Value
	s	Supportive factor	20
	θ	Closeness threshold	0.1
	π	GSD threshold	0.2
	ρ	contribution of the effect of the total influence	1.5
Item-ICR	η	contribution of numbers of items	0.7
	p	Persuasive factor	120
	s	Supportive factor	10
	θ	Closeness threshold	0.1
	π	GSD threshold	0.15
	ρ	contribution of the effect of the total influence	1.2

Table 8 Parameters in combined methods on Netflix

Algorithms	Parameters	Description	Value
User-ICR	η	contribution of numbers of users	1
	p	Persuasive factor	200
	s	Supportive factor	20
	θ	Closeness threshold	0.25
	π	GSD threshold	0.3
	ρ	contribution of the effect of the total influence	1.5
Item-ICR	η	contribution of numbers of items	1.2
	p	Persuasive factor	170
	s	Supportive factor	20
	θ	Closeness threshold	0.2
	π	GSD threshold	0.2
	ρ	contribution of the effect of the total influence	1.3

Figure 33 and 34 show the RMSE performance of proposed algorithms and the other standard collaborative filtering methods on different training/test ratio of each dataset. From

the two figures, we can see that the proposed algorithms item-ICA and user-ICA, as well as the combination of item-ICA and user-ICA (shown as user-ICA+item-ICA in the figures), have better performance than the other standard collaborative filtering methods: user-based CF, item-based CF, and SVD, on each training/test ratio. But it is clear that the performance gets worse when the training/test ratio becomes smaller. This indicates that the proposed algorithms still confront sparsity problem of collaborative filtering.

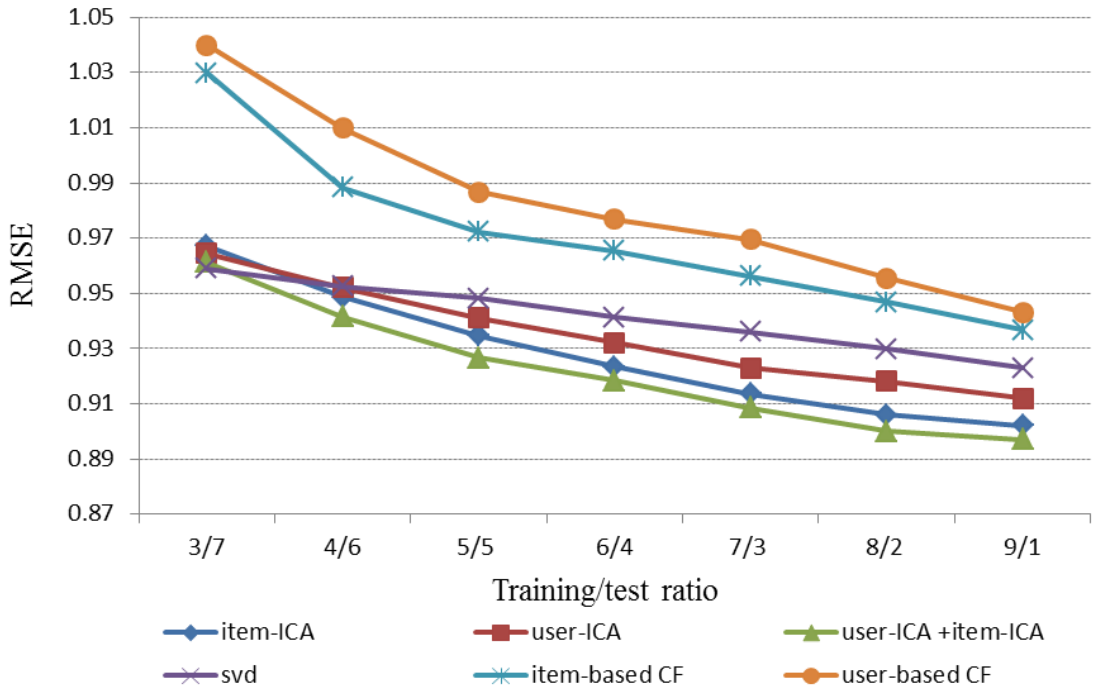


Figure 33 Performances of proposed algorithms and other standard collaborative filtering methods on MovieLens dataset

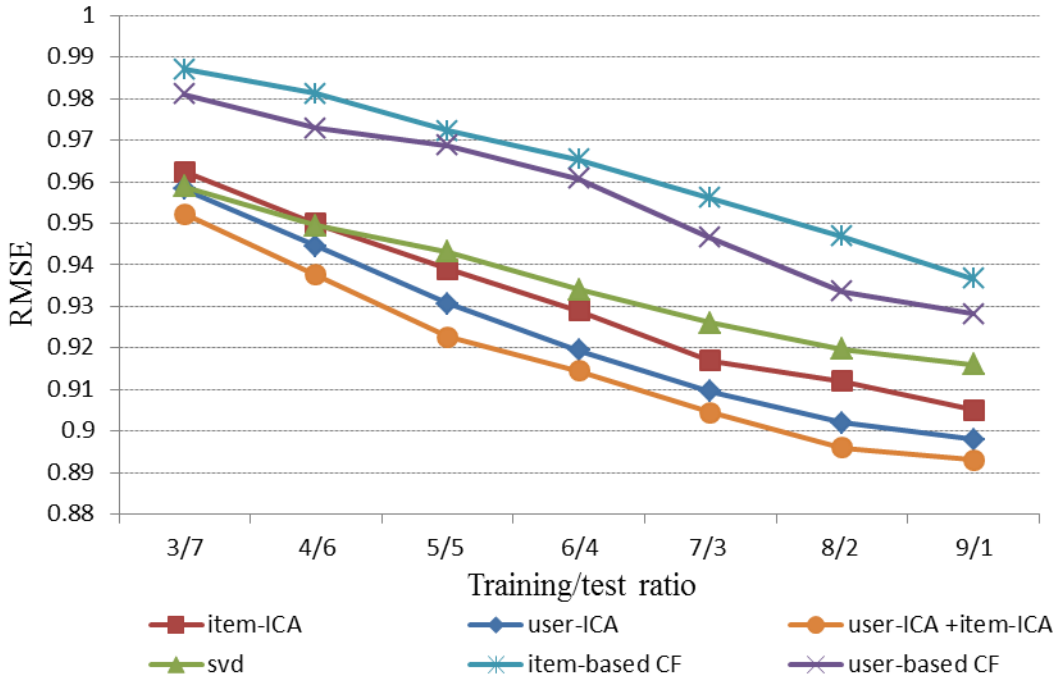


Figure 34 Performances of proposed algorithms and other standard collaborative filtering methods on Netflix dataset

6.5 Summary

In summary, this chapter presents the experimental evaluation methods and discusses correspondence results: the performance of unbalanced strategy for neighbor selection, direct influence model, indirect influence model and the total performance of the combination of the two models.

In Section 6.4.1, we observe that unbalanced neighbor selection method not only improves performance of traditional user-based CF and item-based CF in terms of RMSE and accuracy, but also has an advantage that it just needs fewer nearest neighbors obtaining best performance.

In Section 6.4.2, we investigate the performance of direct influence model on the cases which satisfy closeness threshold. The results are much better than most recommendation algorithms, but we also notice that the coverage of satisfied cases decreases with the performance improves. Hence, we come to the conclusion that there is a tradeoff between performance and coverage.

In Section 6.4.3, we have the similar results with those in Section 6.4.2. That is, the indi-

rect influence model can give much better performance on the cases, which satisfy GSD threshold, but the coverage of satisfied cases decreases with the performance improves.

In Section 6.4.4, we observe the final performance of combination of direct influence model and indirect influence model. Compared with other standard CF algorithms, it is clear that the proposed methods give better performance.

7 Conclusion

7.1 Summary of contributions

The contributions in the thesis are the unbalanced neighbor selection, influence measurement, preference prediction based on direct influence and prediction based on indirect influence. Specifically, the main contributions of this thesis are as follows:

- Unbalanced strategy for neighbors' selection

The unbalanced neighbor selection method not only improves performance of traditional user-based CF and item-based CF in terms of RMSE and accuracy, but also has an advantage that it just needs fewer nearest neighbors obtaining best performance.

- Influence measurement

This thesis considers influence between two users or items as the weight between them. A user *u*'s influence on another user *v* is defined as a function of the user's influence coefficient on *v* and the closeness between them. Influence coefficient measures the ability a user or an item to influence another user or item, and persuade to have the similar preference with the user or item. Influence coefficient could be related to a user's reputation, position and so on. Closeness indicates the similarity between two users or items in terms of interests or acceptance.

- Preference prediction based on direct influence

The performance of direct influence model on the cases, which satisfy closeness threshold, is much better than most recommendation algorithms, but we also notice that the coverage of satisfied cases decreases with the performance improves. Hence, we come to the conclusion that there is a tradeoff between performance and coverage.

- Preference prediction based on indirect influence

Indirect influence measures how a user's preference is affected by the structure of the neighbors of the user. The indirect influence model can give much better performance on the cases which satisfy GSD threshold, but the coverage of satisfied cases decreases with the performance improves.

7.2 Future work

As for our future work, we would like to describe different goals in two categories that are short and long term targets.

7.2.1 Short term targets

- Mining more useful and meaningful latent attributes of neighborhood

In this thesis, we selected three latent attributes for user indirect influence model and item indirect influence model respectively. Although it shows better performance, but we believe that there should be much more useful and meaningful latent attributes to be mined.

- Resolution of sparsity problem

Although the proposed algorithms could leverage the sparsity problem to some extent, there still confront this problem when the data density is very sparse. Incorporating the other techniques, resolution of sparsity problem is one import work in a near future.

7.2.2 Long term targets

- Mining useful and meaningful latent attributes of user or item except for neighborhood

This thesis proposed a method to extract latent attributes of user or item's neighborhood and obtain better performance. In the future, we will extract and select more latent attributes of users or items, not only based on user-item-rating matrix, but also based on the features of them, such as user age, income, item genre, keywords.

- Applying the proposed algorithms in other areas such as friends recommendations

As extensions of collaborative filtering, the proposed algorithms also have the advantage of such approach. For example, it just needs user-item-rating matrix, so it can be applied in many areas. As a long term target, we will apply the proposed algorithms to friends recommendation scenario.

7.3 Final remarks

The thesis puts forward a novel and integrated influence-centric framework, which not only predicts users' preference based on neighbors' direct influence but also takes into account indirect influence of neighborhood. Although the proposed recommendation algorithms can give better performance in terms of RMSE and accuracy compared with other standard collaborative filtering algorithms, challenge still exists in providing recommendations when the dataset is sparse. It is hoped that the sparsity problem could be resolved by focusing on the indirect model in the future work.

References

[1] Ansari, A., Essegaier, S., Kohli R. “Internet Recommendations Systems,” Journal of Marketing Research, 37(3), pp. 363-375, 2000

[2] Adomavicius, G., Tuzhili, A. “Toward the next generation of recommendation systems: a survey of the state-of-the-art and possible extensions” IEEE Transaction on Knowledge and Data Engineering, 17(6), pp. 734-749, 2000

[3] Pu, P., Chen, L. and Hu, R. “A Consumer-Centric Evaluation Framework for Recommender Systems,” Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI), Barcelona, Spain, 2010

[4] Linden, G., Smith, B. and York, J. “Amazon.com recommendations: item-to-item collaborative filtering,” IEEE Internet Computing, 7(1), pp. 76–80, 2003

[5] Guo, S. “Bayesian recommender systems: Models and Algorithms,” Australian National University, 2011

[6] Lops, P., Gemmis, M., Semeraro, G. “Content-based recommendation systems: state of the art and trends,” Recommendation systems Handbook, Springer Science and Business Media, 2011

[7] Pazzani, M. J., & Billsus, D. “Content-based recommendation systems. In The adaptive web, pp. 325-341, Springer Berlin Heidelberg, 2007

[8] Van Meteren, R., & Van Someren, M. “Using content-based filtering for recommendation," In Proceedings of the Machine Learning in the New Information Age: MLnet/ECML2000 Workshop.

[9] Resnick, P. Iacovou, N., Suchak, M., Bergstrom, P. and Riedl, J. “Grouplens: an open architecture for collaborative filtering of netnews,” in Proceedings of the ACM Conference on Computer Supported Cooperative Work, pp. 175–186, New York, NY, USA, 1994.

[10] Sarwar, B., Kar, Y.G., Konstan, J. “Item-based collaborative filtering recommendation algorithms,” In Proceedings of the 10th International World Wide Web Conference, New York, USA, 2001

[11] Grcar, M., Fortuna, B., Mladenic, D., Grobelnik, M. “k-NN versus SVM in the collaborative filtering framework,” Data Science and Classification, pp. 251–260, 2006

[12] Tran, T., & Cohen, R. “Hybrid recommender systems for electronic commerce,” In Proc. Knowledge-Based Electronic Markets, Papers from the AAAI Workshop, Technical Report WS-00-04, AAAI Press, 2000

[13] Burke, R. “Hybrid web recommender systems,” In The adaptive web, pp. 377-408, Springer Berlin Heidelberg, 2007

[14] Li, Q., and Kim, B.M. (2003). “Clustering Approach for Hybrid Recommender System,” Proceedings of the IEEE/WIC International Conference on Web Intelligence (WI’03)

[15] Morid, M.A., Shajari M., Golpayegni A.H., “Who are the Most Influential Users in a Recommender System?” In Proceedings of the 13th International Conference on Electronic Commerce, ACM, New York, 2012

[16] Arora, I., Panchal, V.K. “An MIU [Most Influential Users]-Based Model for Recommendation systems,” In IEEE 24th International Conference on Advanced Information Networking and Applications Workshops, pp. 638-643, 2010

[17] Rubens, N., Sugiyama, M. “Influence-based collaborative active learning,” In Proceedings of the 2007 ACM conference on Recommendation systems, pp. 145-148, 2007

[18] Rashid, A.M. “Mining influence in recommendation systems,” Doctoral Dissertation, University of Minnesota Minneapolis (2007)

[19] Chang, N., Irvan, M., & Terano, T. “An Item Influence-Centric Algorithm for Recommender Systems,” In Distributed Computing and Artificial Intelligence, 11th International Conference (pp. 553-560). Springer International Publishing, 2014

[20] Chang, N. and Terano, T. “Improving the Performance of User-based Collaborative Filtering by Mining Latent Attributes of Neighborhood, (Accepted by 2014 international conference mathematics and computers in sciences and industry)

[21] Chen, D., Lu, L., Shang, M.S., Zhang, Y.C, & Zhou, T. “Identifying influential nodes in complex networks,” Statistical Mechanics and its applications, 391(4), pp.1777-1787, 2011

[22] Rashid, A.M., Karypis, G., Riedl, J. “Influence in Ratings-based Recommendation systems: An Algorithm-Independent Approach,” In Proceedings of 5th SIAM International Conference on Data Mining, Newport Beach, California , 2005

[23] Latané, B. “The psychology of social impact,” American Psychologist, 36, pp. 343-356, 1981

[24] Schafer, J. B., Konstan, J. A., & Riedl, J. “E-commerce recommendation applications,” In Applications of Data Mining to Electronic Commerce, pp. 115-153, Springer US, 2001

[25] Baeza-Yates, R., B. Ribeiro-Neto. “Modern Information Retrieval,” Addison-Wesley, 1999.

[26] Belkin, N. and B. “Croft. Information filtering and information retrieval,” Communications of the ACM, 35(12), pp. 29-37, 1992.

[27] Pazzani, M. J., & Billsus, D. “Content-based recommendation systems,” In The adaptive web, pp. 325-341, Springer Berlin Heidelberg, 2007

[28] Salton, G. Automatic Text Processing. Addison-Wesley (1989)

[29] Fleischman, M., & Hovy, E. “Recommendations without user preferences: a natural language processing approach,” In Proceedings of the 8th international conference on Intelligent user interfaces, pp. 242-244, ACM, 2003

[30] Papangelis, A., Galatas, G., & Makedon, F. “A recommender system for assistive environments,” In Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments (p. 6), ACM, 2011

[31] Holte, R.C., Yan, J.N.Y. “Inferring What a User Is Not Interested in,” In: G.I. McCalla (ed.) Advances in Artificial Intelligence, Lecture Notes in Computer Science, vol. 1081, pp. 159-171 (1996). ISBN 3-540-61291-2

[32] Mooney, R. J., P. N. Bennett, and L. Roy. “Book recommending using text categorization with extracted information” In Recommender Systems. Papers from 1998 Workshop. Technical Report WS-98-08. AAAI Press, 1998.

[33] Pazzani, M. and D. Billsus. “Learning and revising user profiles: The identification of interesting web sites,” Machine Learning, 27, pp. 313-331, 1997.

[34] Balabanovic, M. and Y. Shoham. “Fab: Content-based, collaborative recommendation,” Communications of the ACM, 40(3), pp. 66-72, 1997

[35] Su, X. and Khoshgoftaar, T. M. “A survey of collaborative filtering techniques,” Advances in Artificial Intelligence, Volume 2009

[36] Melville, P., & Sindhwani, V. “Recommender systems,” In Encyclopedia of machine learning (pp. 829-838). Springer US, 2010

[37] Desrosiers, C., & Karypis, G. “A comprehensive survey of neighborhood-based recommendation methods,” In Recommender systems handbook, pp. 107-144, Springer US, 2011

[38] Konstan, J.A., Miller, B.N., Maltz, D., Herlocker, J.L., Gordon, L.R., Riedl, J.: “GroupLens: applying collaborative filtering to usenet news,” Communications of the ACM 40(3), pp. 77–87, 1997

[39] Zhao, Z. D., & Shang, M. S. “User-based collaborative-filtering recommendation algorithms on Hadoop,” In Knowledge Discovery and Data Mining, 2010. WKDD'10. Third International Conference on, pp. 478-481, IEEE, 2010

[40] Ge, F. “A User-Based Collaborative Filtering Recommendation Algorithm Based on Folksonomy Smoothing,” In Advances in Computer Science and Education Applications, pp. 514-518, Springer Berlin Heidelberg, 2011

[41] King, I., Li, J., & Chan, K. T. “A brief survey of computational approaches in social computing,” In Neural Networks, 2009. IJCNN 2009. International Joint Conference on, pp. 1625-1632, IEEE, 2009

[42] Breese, J.S., Heckerman, D., Kadie, C. “Empirical analysis of predictive algorithms for collaborative filtering,” In Proceedings of the 14th Annual conference on uncertainty in artificial intelligence, pp. 43-52, 1998

[43] Hofmann, T. “Collaborative filtering via Gaussian probabilistic latent semantic analysis,” In SIGIR ’03: Proc. of the 26th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, pp. 259–266. ACM, New York, NY, USA, 2003

[44] Grcar, M., Fortuna, B., Mladenic, D., Grobelnik, M. “k-NN versus SVM in the collaborative filtering framework,” Data Science and Classification, pp. 251–260, 2006

[45] Bell, R., Koren, Y., Volinsky, C. “Modeling relationships at multiple scales to improve accuracy of large recommender systems,” In KDD’07: Proceedings of the 13th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, pp. 95-104. ACM, New York, NY, USA, 2007

[46] Koren, Y. “Factorization meets the neighborhood: a multifaceted collaborative filtering model,” In: KDD’08 Proceeding of the 14th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, pp. 426–434. ACM, New York, NY, USA, 2008

[47] Takacs, G., Pilaszy, I., Nemeth, B., Tikk, D. “Investigation of various matrix factorization ´methods for large recommender systems,” In Proc. of the 2nd KDD Workshop on Large Scale Recommender Systems and the Netflix Prize Competition, 2008

[48] Lee, W. S. “Collaborative learning for recommender systems,” In Proccedings of the International Conference on Machine Learning, 2001.

[49] Pazzani, M. “A framework for collaborative, content-based and demographic filtering,” Artificial Intelligence Review, pp. 393-408, December 1999.

[50] Soboroff, I. and C. Nicholas. “Combining content and collaboration in text filtering,” In IJCAI'99 Workshop: Machine Learning for Information Filtering, August 1999

[51] Burke, R. “Hybrid recommender systems: Survey and experiments,” User modeling and user-adapted interaction, 12(4), pp. 331-370, 2002

[52] Shang, S., Hui, P., Kulkarni, S. R., & Cuff, P. W. “Wisdom of the crowd: Incorporating social influence in recommendation models.” In Parallel and Distributed Systems (ICPADS), 2011 IEEE 17th International Conference on , pp. 835-840, 2011

[53] Jianming He and Wesley W. Chu, “A social network based recommender system,” Annals of Information systems: special issue on data mining for social network data, 2010

[54] Rong Zheng, Foster Provost and Anindya Ghose, “Social Network Collaborative Filtering: Preliminary Results”, In Proceedings of the Sixth Workshop on eBusiness(WEB2007), Dec. 2007

[55] Chang, N. and Terano, T. “Recommendation systems Based on Doubly Structural Network,” Proceedings of the 8th International Conference on Innovation & Management, pp. 975-981, 2011

[56] Chang, N. and Terano, T. “A Dynamic Prediction Model for Recommendation systems Based on the Doubly Structural Network,” International Journal of Engineering Innovation and Management, 2(1), pp. 1-8, 2012

[57] Chang, N. and Terano, T. “A User Preference Model based on Influence for Recommendation systems,” Asian Science and Technology Pioneering Institutes of Research and Education ASPIRE LEAGUE, November 8-9, 2012

[58] Herlocker, J. L., Konstan, J. A., Borchers, A. and Riedl, J. “An algorithmic framework for performing collaborative filtering,” in Proceedings of the Conference on Research and Development in Information Retrieval (SIGIR '99), pp. 230–237, 1999.

[59] McLaughlin, M. R. and Herlocker, J. L. “A collaborative filtering algorithm and evaluation metric that accurately model the user experience,” in Proceedings of 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '04)

[60] Latané, B. “The psychology of social impact,” American Psychologist, 36, pp. 343-356, 1981

[61] Sedikides, C., & Jackson, J. M. “Social impact theory: A field test of source strength, source immediacy and number of targets,” Basic and applied social psychology, 11(3), pp. 273-281, 1990

[62] Nowak, A., Szamrej, J., & Latané, B. “From private attitude to public opinion: A dynamic theory of social impact.” Psychological Review, 97(3), 1990

[63] Latané, B. “Dynamic social impact: The creation of culture by communication,” Journal of Communication, 46(4), pp. 13-25, 1996

[64] Ni, S., Weng, W., & Zhang, H. “Modeling the effects of social impact on epidemic spreading in complex networks,” Physica A: Statistical Mechanics and its Applications, 390(23), pp. 4528-4534, 2011

[65] Nettle, D. Using social impact theory to simulate language change. Lingua, 108(2), pp. 95-117, 1999

[66] Bhondekar, A. P., Kaur, R., Kumar, R., Vig, R., & Kapur, P. "A novel approach using Dynamic Social Impact Theory for optimization of impedance-Tongue (iTongue)," Chemometrics and Intelligent Laboratory Systems, 109(1), pp. 65-76, 2011

[67] Macaš, M., Bhondekar, A. P., Kumar, R., Kaur, R., Kuzilek, J., Gerla, V., & Kapur, P. "Binary social impact theory based optimization and its applications in pattern recognition," Neurocomputing, 132, pp. 85-96, 2014

[68] Granovetter, M. "Threshold models of collective behavior," American Journal of Sociology 83(6):1420-1443, 1978.

[69] Kempe, D., Kleinberg, J., & Tardos, É. "Maximizing the spread of influence through a social network," In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 137-146). ACM, 2003

[70] Peyton Young, H. "The Diffusion of Innovations in Social Networks," Santa Fe Institute Working Paper 02-04-018, 2002

[71] Quinlan, J. R. "Induction of decision trees," Machine Learning, 1(1), pp.81–106, March 1986.

[72] Rokach, L. "Data mining with decision trees: theory and applications," Vol. 69. World scientific, 2008.

[73] Miyahara, K. and Pazzani, M.J. "Collaborative filtering with the simple bayesian classifier," In Pacific Rim International Conference on Artificial Intelligence, 2000.

[74] Zurada, J. "Introduction to artificial neural systems," West Publishing Co., St. Paul, MN, USA, 1992.

[75] Berka, T. Behrendt, W. and Gams, E. "A trail based internet-domain recommender system using artificial neural networks," In Proceedings of the Int. Conf. on Adaptive Hypermedia and Adaptive Web Based Systems, 2002

[76] Cristianini, N. and Shawe-Taylor, J. "An Introduction to Support Vector Machines and Other Kernel-based Learning Methods," Cambridge University Press, March 2000.

[77] Kang, H. and Yoo, S. “SVM and collaborative filtering-based prediction of user preference for digital fashion recommendation systems. IEICE Transactions on Inf & Syst, 2007.

[78] Shani, G. and Gunawardana, A. "Evaluating recommendation systems." Recommender systems handbook, Springer US, 2011. pp. 257-297

[79] T. Berka, W. Behrendt and E. Gams, “A trail based internet-domain recommender system using artificial neural networks,” In Proceedings of the Int. Conf. on Adaptive Hypermedia and Adaptive Web Based Systems, 2002

Publications

Journal/Book:

1. Na Chang, Takao Terano, A Dynamic Prediction Model for Recommendation systems Based on the Doubly Structural Network, International Journal of Engineering Innovation and Management, 2(1), pp. 1-8, 2012
2. Na Chang, Mhd Irvan, Takao Terano, Development of a Hybrid TV Recommender System, Knowledge-based Information Systems in Practice (post KES contribution to Springer), (accepted)

Conference:

3. Na Chang, Takao Terano, Recommendation systems Based on Doubly Structural Network, Proceedings of the 8th International Conference on Innovation & Management, pp. 975-981, 2011
4. Mhd Irvan, Na Chang, Takao Terano, Designing a Situation-aware Movie Recommender System for Smart Devices, The Sixth International Conference on Information, Process, and Knowledge Management, pp. 24-27, 2014
5. Na Chang, Mhd Irvan, Takao Terano, A TV Program Recommender Framework, 17th International Conference in Knowledge Based and Intelligent Information and Engineering Systems Vol. 22 pp. 561-570, 2013
6. Na Chang, Takao Terano, A User Preference Model based on Influence for Recommendation systems, Asian Science and Technology Pioneering Institutes of Research and Education ASPIRE LEAGUE, November 8-9, 2012
7. Na Chang, Takao Terano, Development of a hybrid Recommendation Approach based on item content and user social influence, 2014 International Conference on Information Management and Management Engineering (IMME 2014)

8. Na Chang, Mhd Irvan, Takao Terano, An Item Influence-Centric Algorithm for Recommendation systems, 11th International Conference on Distributed Computing and Artificial Intelligence, Advances in Intelligent Systems and Computing, Vol.290, pp. 553-560, 2014
9. Na Chang, Takao Terano, Improving the Performance of User-based Collaborative Filtering by Mining Latent Attributes. (Accepted by 2014 international conference mathematics and computers in sciences and industry)