Information Extraction Method for Capturing User Preferences and Recommender System Application

Xiaohui LI

June 2013

Waseda University Doctoral Dissertation

Information Extraction Method for Capturing User Preferences and Recommender System Application

Xiaohui LI

Graduate School of Information, Production and Systems,
Waseda University

June 2013

Contents

List of Figures

List of Tables

Abstract

Recommender systems are becoming an indispensable application and they are re-shaping the world in the e-commerce scopes. Recommender systems are primarily targeted for individuals who lack sufficient personal experience to evaluate their preferences over potentially overwhelming number of alternative choices. Nowadays, personalized recommender systems have been considered to be one of the most promising approaches for providing individually customized services, which include predicting what would be the most suitable products or services in according to the users' preferences and thus provide invaluable suggestions for an individual user. Although many algorithms have been proposed in the recommender system literatures, and established a sound foundation in the problem domain of recommending academic researches, the quality of recommendation remains a challenging task and to be improved.

This dissertation aims to design innovative recommendation methods that take into the key consideration of improving the quality of recommendation. Three approaches are discussed in this dissertation: A tracking recommendation approach based on Formal Concept Analysis for new user and gray sheep problems that is troubling the existing recommender systems is introduced. An affect-based recommendation approach for eliciting user's preferences is also presented. Finally a hybrid recommendation approach that increases recommendation diversity while maintaining an acceptable accuracy level is described. To demonstrate the effectiveness of the proposed recommendation approaches, a case study on the development of a movie recommendation application based on these approaches was conducted. Moreover, this dissertation employs a system of comprehensive metrics to conduct the evaluation of recommendation quality. The experimental results show that the proposed approach significantly improves the accuracy and diversity of recommendation while compare to the traditional recommendation approaches.

Acknowledgements

Pursuing doctoral degree is the biggest challenge I ever experienced, I am glad that there are many peoples accompany me, encourage me, and help me to achieve it.

First of all, I would like to express my deepest gratitude to my advisor, Professor Tomohiro Murata, for his invaluable instruction and guidance in my Ph.D. program. Professor Murata is a very friendly, open-minded mentor. He not only teaches me how to explore the interests on research area, but also shows me the ways to approach the goal. I always feel honored and fortunate to have him as my supervisor.

I would like to give a heartfelt, special thanks to the committee members, Professor Junzo Watada, Professor Hee-Hyol Lee, and Professor Hao Wen Lin for volunteering their precious time to review this work as well as their insightful comments and constructive suggestions to improve the quality of dissertation.

I wish to express my sincere appreciation for the financial support I received that is the Research Fellowship of the "Ambient SoC Global COE Program of Waseda University" from the Ministry of Education, Culture, Sports, Science and Technology in Japan.

I am indebted to all student colleagues of Murata Lab, who have encouraged me constantly throughout the years I have been studying in Waseda University. They have provided many stimulating and entertaining experiences in the high-pressure research environment in which we learned. Thanks for their brought me the wonderful times. It will remain an indelible impression on my memory.

Last, and most importantly, I owe a great deal to my family and my friends for their unconditional love, support and motivation during my doctoral study in Japan. Thank you for forever standing by me and carries me through always.

Chapter 1

Introduction

1.1 Background

Recommender systems emerged as an independent research area since the appearance of the first papers on collaborative filtering since the mid-1990s [4, 7, 23, 57, 59, 60]. There has been much work done both in the industry and academia on developing new approaches to recommender systems over the last decade. The interest in this area still remains high because it constitutes a problem rich research area and because of the abundance of practical applications that help users to deal with information overload problem and provide personalized recommendations, content and services to them. The suggestions relate to various decision-making processes, such as what items to buy, what music to listen to, or what online news to read. Examples of such applications include recommending books, CDs [1] and other products at Amazon.com [27], movies by MovieLens [23, 28], and news at VERSIFI Technologies (formerly AdaptiveInfo.com) [58].

The study of recommender systems is relatively new compared to research into other classical information system tools and techniques (e.g., databases or search engines). In recent years, the interest in recommender systems has dramatically increased. An investigation from America Silicon Alley showed that with the explosive growth and variety of information available on the Web, recommender systems such as Amazon.com, YouTube, Netflix, Yahoo, Tripadvisor, Last.fm, and IMDB have been played more and more important role in e-commerce services [2], as in Figure 1.1. Moreover many media companies are now developing and deploying recommender systems as part of the services they provide to their subscribers. For example Netflix, the online movie rental service, and awarded a million dollar prize to the team that first succeeded in improving substantially the performance of its recommender system [2, 3, 6].

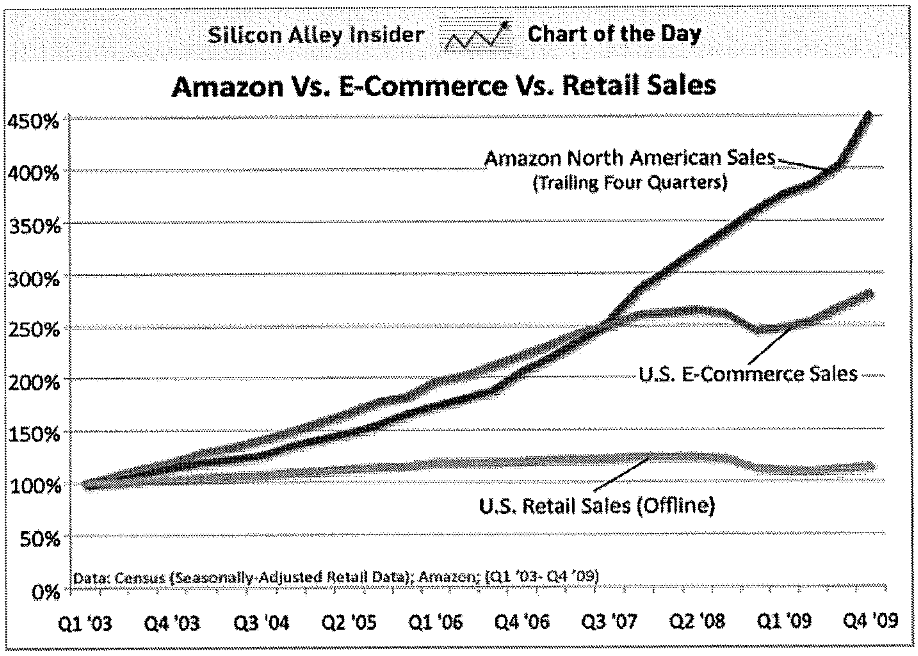


Figure 1.1. Comparison of Amazon, E-Commerce and Retail Sales in recent years.

The functions of recommender systems are described as follow [1, 3, 6]:

- *Increase the number of items sold.* This is probably the most important function for commercial recommender systems, i.e., to be able to sell an additional set of items compared to those usually sold without any kind of recommendation. This goal is achieved because the recommended items are likely to suit the user's needs and wants. Presumably the user will recognize this after having tried several recommendations. Non-commercial applications have similar goals, even if there is no cost for the user that is associated with selecting an item. For instance, a content network aims at increasing the number of news items read on its site. In general, we can say that from the service provider's point of view, the primary goal for introducing recommender systems is to increase the conversion rate, i.e., the number of users that accept the recommendation and consume an item, compared to the number of simple visitors that just browse through the information.

- *Sell more diverse items.* Another major function of recommender systems is to enable the user to select items that might be hard to find without a precise recommendation. For instance, in a movie recommender system such as Netflix, the service provider is interested in renting all the DVDs in the catalogue, not just the most popular ones. This could be difficult without a recommender system since the service provider cannot afford the risk of advertising movies that are not likely to suit a particular user's taste. Therefore, a recommender system suggests or advertises unpopular movies to the right users
- *Increase the user satisfaction.* A well designed recommender system can also improve the experience of the user with the site or the application. The user will find the recommendations interesting, relevant and, with a properly designed human-computer interaction, she will also enjoy using the system. The combination of effective, i.e., accurate, recommendations and a usable interface will increase the user's subjective evaluation of the system. This in turn will increase system usage and the likelihood that the recommendations will be accepted.
- *Increase user fidelity.* A user should be loyal to a Web site which, when visited, recognizes the old customer and treats him as a valuable visitor. This is a normal feature of a recommender system since many recommender systems compute recommendations, leveraging the information acquired from the user in previous interactions, e.g., her ratings of items. Consequently, the longer the user interacts with the site, the more refined her user model becomes, i.e., the system representation

of the user's preferences, and the more the recommender output can be effectively customized to match the user's preferences.

- *Better understand what the user wants.* Another important function of recommender systems, which can be leveraged to many other applications, is the description of the user's preferences, either collected explicitly or predicted by the system. The service provider may then decide to re-use this knowledge for a number of other goals such as improving the management of the item's stock or production. For instance, in the travel domain, destination management organizations can decide to advertise a specific region to new customer sectors or advertise a particular type of promotional message derived by analyzing the data collected by the recommender systems.

1.2 Motivation and Objective

The existent recommender systems depended on three approaches, including content-based filtering, collaborative filtering and hybrid filtering. Nevertheless, several limitations that implied unjustifiable inferences have been still unsolved.

To address the new user and gray sheep problems in existing recommender systems, this study aims to design a heuristic recommendation approach based on Formal Concept Analysis. The goal of our recommender systems is to discover the preferences of individual users and provide personalized services such as help new users reach a decision to meet their demands. To this end, we propose a knowledge-based recommendation model to identify new user preference timely and automatically recommend items for making the new users feel instantly satisfaction. And exploit user behavior tracking strategy conquers gray sheep problem to guide a suitable recommendation.

Despite the extent of improvements necessary to create more effective, accurate and individualized recommendation methods that represent user preferences and behaviors and are applicable to a broader range of real-life applications is still considerable. The question is not whether intelligent recommender systems can mimic emotions, but whether recommender systems can mimic intelligent behavior without emotions. This approach is necessary to make the recommendation process more transparent to the users. To achieve peak performance, a recent and notable question in this research area focuses on how best to learn about changing of user preferences with relevant contextual information from their surroundings and to provide vivid recommendations. Therefore, we must take into account not only information about the users and the products but also must incorporate information about the situational context into the decision-making environment; such user information as time, location and emotional state is crucial to the quality of recommendation results. In other words, if the systems are disconnected from the recommendation environment or situational context, it is quite difficult to make valid decisions for recommendations.

The electronic commerce recommender systems require an emphasis on personalized services that must recommend the right product in the right environment to the right user. For example, recommending a suspenseful movie to a user who has had a busy day is unadvisable. To this end, it is essential to understand the individual user preferences and emotions and to take into consideration the contextual multidimensional information required for systems that recommend products to users. With this motivation, our study focuses on integrating psychological factors and chromatics into the decision-making process. Our profile model not only relies on the traditional two-dimensional factors (i.e., user and item) but also utilizes contextual factors via mood and color to solve the movie recommendation problem. Furthermore, understanding how best to incorporate these multidimensional factors and exploit them in the recommendation model is the major purpose of our study.

Most algorithms proposed in recommender systems literature have focused on improving the aspects of recommendation quality. However, they have often overlooked the importance of recommendation diversity. High accuracy is often obtained by recommending the most popular items to a user, which reduces diversity. More diverse recommendations would be helpful for users and could be beneficial for some businesses. With this motivation, we focus on developing a new recommendation method that can increase the diversity of recommendation sets for a given individual user, often measured by an average dissimilarity between all pairs of recommended items, while maintaining an acceptable accuracy level.

If the work presented here sheds some light on the path to truly personalized systems, it will have been worthwhile.

1.3 Organization of Dissertation

The contents of this thesis consist of 7 chapters which are introduced as follows.

Chapter 1 introduces the research background, motivation, objective, and the structure of this thesis.

Chapter 2 reviews the existing research work that is related to for the scope of this work. The literature review includes an in-depth investigation on existing recommendation techniques and their drawbacks by examining the actual performance of recommender systems.

Chapter 3 describes three key problems to be solved in this research, including cold-start recommendation problem, which attempts to present items that are likely of interest to a new user; context-aware recommendation problem, which aims at improving the accuracy of recommendations by customizing matching techniques to a particular context background; and resolving the diversity recommendation problem, which avoids providing too similar items for each individual user.

Chapter 4 explores a tracking recommendation approach to provide the relevant items for new users, a process which facing the cold-start problem. A new recommendation model based on the synergistic usage of knowledge from repository, which includes user's limited history records and up to two-levels of item's property, was constructed. The hierarchical structure map based on the classificatory attributes of items generating from the Formal Concept Analysis technique is used to discover the candidate items that the user may be interested. The appropriate recommendation returned to a new user utilizes the filtering techniques based on the corresponding descriptive attribute ranking of these candidates. The experimental study includes simulating a prototype recommender system for implementing the proposed approach and its performance is extensively evaluated. Experiments using two datasets demonstrated our strategy was more robust against the drawbacks and preponderate over traditional recommendation approaches under new user cold-start conditions.

Chapter 5 focuses on deriving an affect-based approach for solving the context-aware recommendation problem. The presented approach attempts to catch users' transitional preferences and guide timely recommendations based on these transformations. Firstly, the conceptual network of each items based on their descriptive keywords are created using WordNet and color theory. Secondly, color orders of each item are generated via the spreading activation. Thirdly, a characteristic sequence consisting of color nodes was extracted from the color orders of users' record, and it is used for inferring the contextual information; this information is subsequently used for perceiving the changing of user preferences. The performance of the proposed approach was illustrated using the example of a movie recommendation application. The experiment analysis results show that the proposed approach outperformed the traditional approaches in terms of accuracy.

Chapter 6 discusses a new hybrid recommendation approach that incorporates multidimensional clustering technique into a collaborative filtering recommendation process to overcome the diversity recommendation problem. The proposed approach aims to help users reach a decision to meet their diverse demands and provide the corresponding user with highly idiosyncratic or more diversified recommendations. The proposed hybrid recommendation procedure has been partitioned into three main phases: i) data preprocessing and multidimensional clustering; ii) choosing the appropriate clusters; iii) collaborative filtering recommendation process for the target user. The performance of the proposed approach is evaluated using a public movie dataset and it was compared with two representative recommendation algorithms. The empirical results demonstrate that our proposed approach performs superiorly on increasing recommendation diversity while maintaining an acceptable accuracy level.

Chapter 7 concludes this thesis with possible extensions for future research.

Chapter 2

State-of-the-Art in Recommender Systems

In this chapter, a review of the state of the art in recommender systems is conducted, both for commercial systems and those published within the research literature. Recommender systems are categorized according to the previously defined characteristics. We firstly survey a bunch of recommendation approach in different domains, by providing a brief overview of the most important aspects characterizing various recommender systems. Based on how recommendations are made, recommendation techniques are usually classified into three general categories [8]:

- Traditional recommendation techniques: mainly refers to content-based recommendation and collaborative filtering recommendation;
- Hybrid recommendation techniques: three strategies by combining variant recommendation techniques in the form of several ways;
- Other recommendation techniques: such as knowledge-based recommendation, association rule-based recommendation and utility-based recommendation.

2.1 Traditional Recommendation Techniques

Since the first collaborative filtering based recommender systems, which named Tapestry was born in 1990s, the research and application of the intelligent recommendation method has been developed rapidly. With the rise of e-commerce, the recommendation technology ushered in the climax of development. Today, almost every large e-commerce sites regard the recommender system as an important marketing strategy to meet the user's personalized needs. In 2000, Schafer, Konstan and Riedl published a paper entitled “E-commerce recommendation applications” in the journal “Data Mining and Knowledge Discovery”, which is the most basic literature in the field of recommender system research. It is a model of academic research and industrial practice. More and more researchers and engineers in the business community are involved in the research and practice of recommendation method.

Due to the huge demand, the intelligent recommendation method has been widely concerned by many scholars and research institutions in the world. The American Computer Association (ACM) has set up the recommender system as a research topic. The high order international journals in the field of data analysis, such as “IEEE Transactions on Knowledge and Data Engineering”, “ACM Transactions on Information System”, have published special issue on the recommender system. The Netflix competitions from 2006 to 2011, also take the recommender system research as a contest theme and set off a research boom of the recommendation technology.

At the present, recommender system has become an independent research field, which mainly focuses on the improvement and expansion of the recommendation technology to adapt to different recommendation occasions.

2.1.1 Content-based Filtering Recommendation Technique

As the name implies, content-based filtering exploits the content of items to predict their relevance by matching user profiles and item features. Content-based recommendation approach calculates the similarity between the users according to the user has chosen the property of items, and then makes the corresponding recommendation. The current content-based recommendation algorithm analyze the profile of user's preference model and descriptions of items based on the features of the items rated by the user, which is recommended the similar items with their interests for users [5]. The recommendation process is matchmaking between the attributes of the user profile and the attributes of item profile. The recommendation result is a relevance judgment that represents the level of user's interest for that item.

The content-based recommendation technique focuses on discovering the similarities between user's preferences and item characteristics to make recommendations. The recommender systems match user's preferences with similar characteristics in the items database. The content-based approach creates a search user profile via the associated features of items that the user had positively rated and recommends items based on their matching associated features [46]. A content-based recommender system recommends products or services that are similar to what has previously interested the user. The systems exploit supervised machine learning to discriminate which items are likely to be interest to users and which items are likely to be uninteresting [4, 6]. For example, in a movie recommendation application, to recommend movies for a target user, content-based recommender system tries to understand the user's preferences from the information of movies user has rated highly in the past (genres, main actors, directors, release date, countries, storyline, plot keywords, etc.). Then, the movies that have high degree of similarity with user's preferences would recommend to the target user.

The content-based recommender systems are used to recommend the resources containing text information, such as documents, web pages and online news, etc. The most typical application is Web Recommendation System FAB, which developed by the digital library project group in the Stanford University. In 2000, Billsus implement the content-based recommendation in the mobile news access system, to help users find similar news in recent news reports. In 2005, Adomavicius and Tuzhilin presented a theoretical description of the content based recommendation, which utilize machine learning method to get the user's interest data from the characteristics of the content, but it does not need to be based on the user's evaluation of the item. In the content-based recommendation method, the item is defined by the relevant features and attributes. The user profile depends on the learning method, such as the decision tree, neural network, and the representation method based on vector. The user data come from the historical data of the user, so the user profile may change with the user's preference change. In 2007, Pazzani and Billsus proposed a new description of the content-based recommendation method. This method is used to evaluate the similarity between the items and the items that have not been seen in the past. The core factor is to obtain the feature description and the important records of these characteristics and observe the matching degree between the user interests and to be predicted items based on the features of the user evaluation of items.

The content-based recommendation technique has several advantages:

- *User independence.* Content-based recommendation technique can recommend items to users who have a special interest without the data of other users. Instead, to find the "nearest neighbors" of the target user, collaborative filtering recommendation technique analyzes ratings from other users that have similar tastes since they rated the same items;
- *Transparency.* Content-based recommendation technique can explain how the system works to produce items and why those items are recommended by listing the content features of the recommended items. Conversely, collaborative filtering recommendation technique is a black box since the only explanation is recommending item rated by that unknown users with similar ratings;
- *New item.* Unlike collaborative filtering technique make recommendations which rely solely on users' ratings, content-based recommendation technique can recommend new or not very popular items not yet rated by any user.

Nonetheless, the content-based recommendation technique has several shortcomings:

- *Limited content analysis.* Content-based recommendation technique has a requirement for domain knowledge. For example, in a movie recommender system, the directors and main actors, domain ontologies need to be known by system. If the analyzed content does not contain enough information to distinguishing features of items, content-based recommender system cannot determine which items users are like, it also cannot provide suitable suggestions for the target user. In other word, some representations capture only certain aspects of the content, but for Web pages, feature extraction technique usually ignores additional information, it would directly influence the user's experience.

- *Over-specialization*. Content-based recommendation cannot find something unexpected items. When the items suggested by system are high matched the user profile, the recommended items are often similar to the items that have been rated by user. The content-based recommendation technique rarely finds the novel items, this drawback will lead to a lack of novelty and serendipity to the recommendation results.
- *New user*. Collecting enough ratings is essential work for really learn user preferences in a content-based recommendation. Therefore, for a new user, the recommender system will not be able to provide reliable or accurate recommendations when his ratings are sparse.

Balabanovic and Shoham pointed out the limitations of the content-based recommendation technique in 1997. For instance, the requirements of the content can be easily extracted into meaningful features, the requirements of the characteristics of a good structure, and the user's taste must be able to express in the form of content. As these limitations, the pure research on content-based recommendation receives less attention.

2.1.2 Collaborative Filtering Recommendation Technique

To address the issues of content-based recommendation, the collaborative filtering approach [2, 7] has been used in recommender systems. In 1992, Goldberg first proposed a collaborative filtering algorithm in the academic paper entitled "Using collaborative filtering to weave an information tapestry". The basic idea of this algorithm is to use the user's historical information to calculate the similarity between the users to the target users to recommend. For instance, in news group, calculate the similarity between them in the taste according to the news of the user clicks, and use this similarity to further recommend the relevant news for user. This is the early embryonic form of collaborative filtering recommender system.

Collaborative filtering recommendation algorithm is one of the most widely used algorithms for research and application. It generally uses the nearest neighbor technology, the user's historical preferences information to calculate the distance between the users, and then use the nearest neighbor of the target user to predict the value of the items evaluation of the target user's preferences. And the system can be recommend items for the target users based on their preference. The advantage of collaborative filtering is that no special requirements for the recommended object, collaborative filtering recommender system can handle unstructured complex objects, such as music, movies. In 1994, GroupLens first proposed a recommendation method based on user's ratings, which was widely used in CDNOW, Amazon and other electronic commerce websites. The application of Amazon book recommender system is the most successful, but this system ignore an important factor the dynamic transfer characteristics of user's consumption preference only based on the user's historical purchase record and other user's purchase history. To improve the accuracy and efficiency of the recommendation algorithms, the researchers have made a lot of improvements. In 2001, Sarwar proposed a collaborative filtering algorithm which calculates the similarity using correlation and cosine. To a certain extent, this approach improved the accuracy and efficiency for searching neighbor users.

In collaborative filtering recommendation, the user selects the information through mutual cooperation, that is, through the user's ratings to the information resources to produce a recommendation. The algorithms mainly include user-based collaborative filtering algorithm and item-based collaborative filtering algorithm. User-based collaborative filtering recommendation predicts the target user's ratings for the items via finding the most similar user with a target user and his ratings. Item-based collaborative filtering recommendation predicts the user's ratings for the target items based on the similarity of most users to different items, to find some items that are similar to the target item, and then the final recommendation is generated.

Collaborative filtering recommendation technology has been applied in many systems, such as mail recommender system Tapestry developed by the research center of famous Xerox company, the movie recommender system GroupLens developed by University of Minnesota [23, 28], and personalized music recommender system Ringo developed by MIT media lab [60], that used collaborative filtering algorithms to generate the recommendation that helps people find relevant information on the Web. Other instances of collaborative recommender systems include Amazon system that recommends books, IMDB that recommends movies on the Web, and the Jester system that recommends jokes [4, 25, 63].

The collaborative filtering recommendation technique has several advantages when compared to the content-based one:

- can be filter out information that is difficult to perform automatic content analysis via machine, such as artwork, music, movie, etc..
- sharing the experience of other users to avoid incomplete and inaccurate of content analysis, and can be filtering information based on a number of complex, difficult to represent the concept, such as information quality, personal taste.
- can be recommend new information. This is a big difference between collaborative filtering and content-based filtering, content-based recommendation finds a lot of content that users are already familiar, and collaborative filtering can find the user's potential but not found the interest of users.
- utilize effectively the feedback information of other similar users, and to accelerate the personalized learning.

Although the collaborative filtering recommendation technique overcomes some of the weaknesses of the content-based recommendation described earlier, such as limited content analysis, overspecialization and new user problem. However, collaborative filtering recommendation technique has their own limitations [46, 64], as described below.

- *Cold-start.* For new user, collaborative filtering systems cannot learn users' preferences. Therefore, for a new user that added to recommender system, collaborative filtering systems would not be able to make accurate recommendations until this new user has rated a substantial number of items. For New item, collaborative filtering systems rely solely on ratings that users give to make recommendations. Therefore, for a new item that added to recommender system, collaborative filtering systems would not be able to produce recommendations until this new item is rated by a substantial number of users. Cold-start problem can be addressed using hybrid recommendation approaches that described in the section 2.2.
- *Data sparsity.* The number of items rated by users is usually very small compared to the number of items in any recommender system. Then, how to predict effective ratings from this small number of examples is becoming very important. The success of the collaborative filtering recommendation depends on the availability of critical ratings of mass users. For instance, in a movie recommender system, there are many movies that have been rated by only few users, even if these users gave high ratings to them, these movies also would be recommended very rarely. Moreover, for the user who has distinctive tastes, system will produce poor recommendations due to there are not any other users who have particularly similar preferences [46].

To sum up, collaborative filtering is a typical application of the technology, but it has still a lot of problems to be solved. In addition to the new user problem, the collaborative filtering recommender systems frequently suffer from the new item problem, since they rely solely on user's ratings data. Also, the data sparsity of ratings is a ubiquitous problem in collaborative filtering recommender systems, since the number of available ratings that can be used to predict user preferences are very rare. An extreme situation is if there is no user to evaluate a certain item, the item cannot be recommended [65].

2.2 Hybrid Recommendation Techniques

To address the above problems, researchers attempt to combine the content-based and the collaborative filtering approaches into a hybrid approach [4, 8]. Hybrid recommender systems commonly combine two or more recommendation techniques to achieve better performance. The most of research and application is the combination of content-based and collaborative filtering recommendation, which integrating the feature of content filtering into collaborative filtering algorithm to build a unified recommendation model. In 2002, the Burke proposed classification criterion of hybrid recommendation method in a report, which entitled "Hybrid Recommender System: Survey and Experiments" [8]. The simplest approach is to use the content-based recommendation method and collaborative filtering recommendation method, to produce their own prediction results, and then generate a combination of their results. In 2005, Adomavicius and Tuzhilin presented a comprehensive survey of the latest research on the hybrid recommendation, and pointed out that the most important principle of the hybrid recommendation is to avoid or compensate for the weaknesses of the recommendation technology via the combination. Although collaborative filtering is the most prominent recommendation technique, a number of its hybrids remain unexplored. The existing hybrid approaches suggest some interesting types of recommendation as follow.

- Implementing collaborative filtering and content-based methods separately and combining their predictions to achieve the feature augmentation hybrid.
- Incorporating some content-based elements, e.g., characteristic-based sorting into the collaborative filtering approach.

- Incorporating some collaborative characteristics, e.g., latent semantic indexing into the content-based approach [47, 48].

This section introduces combining content-based and collaborative filtering approaches together into the hybrid recommendation approach in the form of seven different ways (Weight, Switch, Mixed, Feature combination, Feature augmentation, Cascade, Meta-level). From a comprehensive perspective, these seven ways can be abstracted into three recommendation strategies: monolithic, parallelized, and pipelined hybridization.

2.2.1 Monolithic Hybrid Recommendation

Monolithic hybrid recommendation incorporates several recommendation strategies into one algorithm implementation. The other two hybrid recommendation techniques require at least two different recommendation algorithms, their recommendation results are combined. However, monolithic hybridization contains only one recommended unit, integrated a variety of approaches through preprocessing and combining different types of input data. According to the classification method presented by Burke, feature combination and feature augmentation ways can be assigned to monolithic hybridization.

- *Feature combination.* It combines the characteristics from different data sources and used by another recommendation algorithm. For instance, in 1998, Basu et al. proposed a feature combination hybrid method that combines the collaborative features and content features of catalog items, and finds a new hybrid feature based on the data of groups and items. In 2009, Zanker et al. proposed another feature combination method based on accuracy and effectiveness of prediction from different types of ratings to find similar users.
- *Feature augmentation.* The additional feature information generates by a technique is embedded into the feature input of another recommendation technology. In 2002, Melville et al. proposed a

collaborative mechanism of content-based predictive result to predict the user's ratings. In 1999, Mooney and Roy proposed a discussion on the content-based book recommender system, as well as the research papers recommender system designed by Torres in 2004, these hybrid schemes have been employed the feature augmentation.

The superiority of monolithic hybrid recommendation technique is that little additional knowledge was needed in terms of the feature level. Normally, some additional preprocessing steps are required and made minor modifications on the crucial algorithm or its data structures.

2.2.2 Parallelized Hybrid Recommendation

Parallelized hybrid recommendation can use several recommender systems simultaneously. These systems work independent of each other and generate the recommendation list respectively. Then, integrate their output results in the final recommendation by using a special hybrid mechanism. As defined by Burke, weight, switch and mixed ways can be assigned to this category.

- *Weight.* This strategy is weighting the results from a variety of recommendations. In 1999, Claypool et al. proposed the combination of dynamic weighting in the field of Journalism and the output results based on the content and collaborative filtering technology. In 2009, Zanker et al, In the collaborative filtering and sensitivity analysis based on knowledge recommendation, hoping to find the most appropriate weighting method.
- *Switch.* The switch strategy can be adopting different recommendation technique through the background of problem and the actual conditions or requirements. In 2000, Billsus and Pazzani proposed two methods of content-based recommendation and a collaborative filtering strategy to recommend news orderly. In 2009, Zanker and Jessenitschnig proposed a strategy to switch between the

hybrid scheme that used two kinds of collaborative filtering methods and the knowledge-based recommender system.

- *Mixed.* The mixed strategy exploits a variety of recommended techniques and provides a recommendation for the target user. In 1999, Wasfi proposed an approach that combines the results from different recommender system. The items that have the highest score of each recommender system are presented to the user simultaneously. However, the conflict will be generated when the different results are combined. In 2000, Cotter and Smyth used the predefined priority rules between different recommendation functions to solve this conflict.

Parallelized hybrid recommendation technique is the least change for existing implementation methods since it only made an additional post processing step. Nevertheless, matching the variant parallelized algorithms to calculate the recommendation score would be essential work due to some additional running complexity was increased.

2.2.3 Pipelined Hybrid Recommendation

The pipelined hybrid recommendation links a plurality of recommender systems via pipelined architecture. The output of the previous recommender system becomes input of next recommender system. That is, the posterior recommender system can optimize the results produced by the previous recommendation. Following Burke's taxonomy, both cascade and meta-level ways are examples of pipelined hybrid recommendation.

- *Cascade.* In this strategy, one recommendation technique is used first to produce a rough recommendation candidates and another technique refines the more accurate recommendation from this candidate set. In 2007, Felferning et al. proposed an alternative ranking mechanism that can be optimized recommendation results in the algorithm design. In 2009, Zanker and Jessenitschnig proposed a hybrid method, which can be ranked the recommended results with a weighted method.

- *Meta-level.* A meta-level hybrid refers to the model generated by one recommendation technique is used as the input of another algorithm. In 1997, Balabanovic and Shoham employed a content-based recommendation approach that builds a user model based on a set of vectors of term categories and the users' interest. Based on this work, the recommendation results were gained by exploiting a collaborative filtering approach. In 1999, Pazzani applied the content-based collaborative filtering approach in restaurant recommender system. The experimental results showed that this hybrid recommendation method is better than the single use of content-based and collaborative filtering method, especially when the same records between the users is very rare. Zanker evaluated a meta-level hybridization that combines collaborative filtering and knowledge-based recommendation in 2008. The evaluation result showed that this approach produced more successful predictions than artificial repository.

Pipelined hybrid recommendation technique is one of the most sophisticated hybridization. It needs to comprehend the function of algorithm deeply to ensure high-performance running computation. However, it can be performed well when two antithetic recommendation algorithms are combined, such as, collaborative filtering recommendation and knowledge-based recommendation.

2.3 Other Recommendation Techniques

In addition to the aforementioned mainstream of recommendation techniques, the researchers also present knowledge-based recommendation, association rule-based recommendation and utility-based recommendation [4, 6, 13].

Knowledge-based recommendation was first proposed by Burke in 2000. In a way, it can be seen as a reasoning or inference technology, which is dependent on the property and the knowledge of the goods. In 2004, Thompson proposed a knowledge-based approach that integrates natural language interface, which helps users to reduce the overall cost of interaction. Knowledge-based recommendation methods are obviously different from the diverse functional knowledge they use, which can be divided into the knowledge recommendation based on case and the knowledge recommendation based on constraint. In 2005, Bridge proposed a knowledge recommendation method based on the instance, which retrieve similar items according to similarity measure. In 2006, Felferning proposed the knowledge recommendation method based on constraint, which search for the items to be recommended in the collection of all the items that are in conformity with the recommendation rules, according to a well-defined recommendation rule set. In 2008, Felferning and Burke defined knowledge-based recommendation as a method that can map the user needs to the items, it doesn't use in the content and collaborative filtering method.

Association rule-based recommendation is the basis of association rules, which have been purchased as a rule head, and the rule body is recommendation candidate. This method provides recommendation services based on the generated association rule model and the user's current purchasing behavior. In 1993, Agrawal first proposed the Aprior association rules recommendation algorithm, but the rule discovery is very time-consuming. So it becomes the bottleneck of this algorithm. With the increase of the number of rules, the management of the system becomes more and more complex. Association rule-based recommender system, such as: WebSpher developed by IBM, which allows the system administrator to set rules according to the user's static characteristics and dynamic attributes, to provide different services in different situations. The advantages of association rule-based recommender system are simple and direct. The disadvantages are that the rules quality is hard to be guaranteed, and can't be updated dynamically. Moreover, with the increase of the number of rules, the system will become more and more difficult to manage.

Utility-based recommendation is calculate the utility function based on the utility of the project. The key issue is how to determine the utility function for each user. Therefore, the user profile is largely determined by the utility function of the system. The advantage of utility-based recommendation is that the calculation of the utility takes into account the non commodity properties, such as supplier reliability and product availability.

Overall, the literature presented in this chapter suggests that there is a great need for research in this area.

Chapter 3
Problems Description in Personalized Recommender Systems

The goal of a recommender system is to model the users' preferences in order to recommend new items that the users are likely to find of interest. Capturing user preferences is a crucial task in recommender systems. Simply asking the users what they want is too intrusive and prone to error, and then finding meaningful patterns is difficult. The existence of a user model is central to every recommender system; the way in which this information is acquired and exploited depends on the particular recommendation technique. Capturing accurate user preferences is however, an essential task if the recommender systems are to respond dynamically to the changing needs of their users. In this chapter, we firstly explain the definition of personalization. Three problems in this research to be solved for personalized recommendation are described as follow:

- Cold-start recommendation: Cold-start is a prevalent problem in recommender systems which involve automated modeling of data [14]. Specifically, if system has not yet gathered sufficient information, the recommender system would be incapable of drawing any inferences for users or items;

- Context-aware recommendation: User preferences may differ with context, such as time of day, season, mood, device characteristics, location, and options offered by recommender system. A context-aware recommender system takes the context into account when computing suggestions and can provide recommendations for target user under user preferences are dynamically changed[6, 29];
- Diversity recommendation: the diversity of recommendation aims to avoid providing too similar recommendations for matching the individual user's crossover preferences [45].

3.1 Definition of Personalization

Personalization is defined as "the combined use of technology and user information to tailor electronic commerce interactions between a business and each individual user" or is described as being "about building user loyalty by building a meaningful one-to-one relationship; by understanding the needs of each individual and helping satisfy a goal that efficiently and knowledgeably addresses each individual's need in a given context" [9]. A typical personalization process includes three steps: understanding the users through profile-building, delivering personalized offerings based on knowledge about the items and the users, and measuring the impact of personalization [10]. The aims of personalization are to offer users what they want without asking them explicitly and to increase profit by retaining existing users and attracting new users. A successful recommendation process can boost marketing, which means that users are willing to spend more money and continually buy more products. In other words, a major vehicle that makes the personalized recommendation possible is matching the potential products with user preference to achieve effects of transforming browser on the web into customers.

3.2 Personalization in Recommender Systems Applications

The goal of raising recommendation quality in electronic commerce recommender systems require an emphasis on personalized services. To this end, it is essential to understand individual user preferences by some ways, including attract new user, take into consideration contextual multidimensional information, and provide diverse recommendation when systems recommend products to users.

3.2.1 Cold-start Recommendation

The cold-start problem is most prevalent in content-based and collaborative filtering recommender systems. Cold-start recommendation refers to systems generate the appropriate results, such as news, books, movies, or web pages, that are likely of interest to the user in the absence of sufficient data of users and items. Cold-start is mainly divided into three categories: user cold-start, item cold-start and system cold-start.

User cold-start is mainly to solve the new user problem to make personalized recommendation. New user in the system has not any historical data, so it is difficult to predict their preferences and make the appropriate recommendation. Item cold-start is mainly to solve the new item problem that is recommended to users might be interested in it. New item has no data of interaction with the user, so the common recommender system often ignores these new items. This has resulted in the recommendation imbalance and lack of diversity. System cold-start problem is to solve poor recommendation when users and items do not have enough data.

In the content-based recommendation, the system must be constructing a sufficiently-detailed user's preferences model to match the characteristics of item against relevant features. In the collaborative filtering recommendation, the system can identify the users who have the similar preferences with the target user via user ratings, and recommend the items which the like-minded users favored to the target user. In above both case, the cold-start problem would cause system propose nothing to new user. And new item are rarely considered in the recommendation process.

Typically, the cold-start problem is reduced via hybrid approach that combines content-based and collaborative filtering techniques. New items which have not yet received any ratings from the clusters would be assigned a rating automatically, based on the ratings assigned by the clusters to other similar items. Item similarity would be determined according to the characteristics of items. Furthermore, the user's profile may be constructed automatically by integrating information from other user activities, such as learn the user's potential preferences by observing the user's browsing behavior and predict the items that the user may be interested.

3.2.2 Context-aware Recommendation

The goal of a recommender system is to model the users' preferences in order to recommend new items that the users are likely to find of interest. However, user preferences are influenced by contextual conditions, such as the time of the day, mood, or current activity, but this type of information is not exploited by standard models.

Context is a multifaceted concept that has been studied across different research disciplines, including computer science (primarily in artificial intelligence and ubiquitous computing), cognitive science, linguistics, philosophy, psychology, and organizational sciences. A well-known business researcher and practitioner C.K. Prahalad has stated that "the ability to reach out and touch customers anywhere at anytime means that companies must deliver not just competitive products but also unique, real-time customer experiences shaped by customer context" and that this would be the next main issue for the practitioners [6].

Since we focus on recommender systems in this work and since the general concept of context is very broad, we try to focus on those fields that are directly related to recommender systems. Recommender systems are also related to e-commerce personalization, since personalized recommendations of various products and services are provided to the customers. The importance of including and using the contextual information in recommendation systems has been demonstrated in [29], where the authors presented a multidimensional approach that can provide recommendations based on contextual information in addition to the typical information on users and items used in many recommendation applications. It was also demonstrated by Adomavicius et al. that the contextual information does matter in recommender systems: it helps to increase the quality of recommendations in certain settings.

One methodology of deciding which contextual attributes should be used in a recommendation application and which should not is presented in [29]. In particular, Adomavicius et al. propose that a wide range of contextual attributes should be initially selected by the domain experts as possible candidates for the contextual attributes for the application. For example, in a movie recommendation application, we can initially consider such contextual attributes as Time, Theater, Companion, Weather, as well as a broad set of other contextual attributes that can possibly affect the movie watching experiences, as initially identified by the domain experts for the application. Then, after collecting the data, including the rating data and the contextual information, we may apply various types of statistical tests identifying which of the chosen contextual attributes are truly significant in the sense that they indeed affect movie watching experiences, as manifested by significant deviations in ratings across different values of a contextual attribute. For example, we may apply pairwise tests to see if good weather vs. bad weather or seeing a movie alone vs. with a companion significantly affect the movie watching experiences as indicated by statistically significant changes in rating distributions. This procedure provides an example of screening all the initially considered contextual attributes and filtering out those that do not matter for a particular recommendation application. For example, we may conclude that the Time, Theater and Companion contexts matter, while the Weather context does not in the considered movie recommendation application.

3.2.3 Diversity Recommendation

Diversity is generally defined as the opposite of similarity. In some cases suggesting a set of similar items may not be as useful for the user, because it may take longer to explore the range of items. Consider for example a recommendation for a vacation [42], where the system should recommend vacation packages. Presenting a list with five recommendations, all for the same location, varying only on the choice of hotel, or the selection of attraction, may not be as useful as suggesting five different locations. The user can view the various recommended locations and request more details on a subset of the locations that are appropriate to her.

Most of recent studies have focused on increasing the individual diversity, which can be calculated from each user's recommendation list (e.g., an average dissimilarity between all pairs of items recommended to a given user) [42, 43, 44, 61]. The diversity of recommendation aims to avoid providing too similar recommendations for the target user. On the other hand, except for some work that examined sales diversity across all users of the system by measuring a statistical dispersion of sales [62], there have been few studies that explore diversity in recommender systems, despite the potential importance of diverse recommendations from both user and business perspectives.

The most explored method for measuring diversity uses item-item similarity, typically based on item content. Several metrics can be used to measure aggregate diversity, including the percentage of items that the recommender system is able to make recommendations for often known as coverage [25]. Then, we could measure the diversity of a list based on the sum, average, min, or max distance between item pairs, or measure the value of adding each item to the recommendation list as the new item's diversity from the items already in the list. The item-item similarity measurement used in evaluation can be different from the similarity measurement used by the algorithm that computes the recommendation lists.

3.3 Summary

As this chapter clearly describes, three key problems to be solved in this research, including cold-start recommendation, which attempts to present items that are likely of interest to the new user; context-aware recommendation, which aims at improving user satisfaction to recommendations by tailoring these to each particular context; and diversity recommendation, which provides a variety of items when systems recommend products to users. In the next Chapter 4, Chapter 5 and Chapter 6, we will take all these definitions to discuss our presented approaches adapt the idiosyncratic needs of recommender systems.

Chapter 4

Reasoning New User preferences in Cold-start Recommendation

4.1 Overview

Recommender systems are popularly being employed in electronic commerce to provide product information for customers and help them select suitable products to meet their personal demand. The application of recommender systems is well known one way to achieve mass customization in the development of e-commerce. Customizing recommendation has recently become a buzzword technology in our real life, especially in online video service territory. The aim of customizing recommendation is to offer users what they want without asking them explicitly and to increase profit by retaining current users and attracting new users [1]. A good recommendation can boost the cross-sell and the up-sell by suggesting additional products for user to purchase. In other words, a major vehicle that makes the customizing recommendation possible is matching the potential products with user preference to achieve effects of transforming browsers on the web into buyers.

A typical customizing recommendation process includes three steps: understanding users through profile building delivering personalized offering based on knowledge about the items and the users, and measuring personalization impact [9]. The products can be recommended based on the top overall sellers on a website, on the demographics of consumer, or on an analysis of the past buying behavior of consumer as a future behavior prediction. Recommender systems identify user preference over time and automatically suggest products purchase that fit for user preferences. However, the common drawbacks of current recommender systems are frequently disregard some useful information and imply some unjustifiable inferences. Most research in the past with a view to calculate the similarity between user's interests and product characters, hoping to improve the prediction accuracy of recommender systems [3, 11, 12]. This study focuses on discovering the potential relationship among items by expansion and inference to provide effective recommendation for user [69, 70]. The key element in proposed model is the utilization of Formal Concept Analysis and tracking the current behavior of user on real time.

The existent recommender systems depended on three approaches: content-based filtering, collaborative filtering and hybrid filtering. Nevertheless, several limitations that implied unjustifiable inferences have been still unsolved [4, 14].

New user problem: For a new user, having very few information, would not be able to obtain the accurate recommendation. Specifically, it is difficult to make a quality recommendation for user who has no records.

Gray sheep problem: For a user whose tastes are unusual, there will not be any other users who are particularly similar, leading to poor recommendation.

In order to address the above problems, this study aims to design a heuristic recommendation approach based on Formal Concept Analysis. The goal of our recommender systems is to discover the preferences of individual users and provide personalized services such as help new users reach a decision to meet their demands. To this end, we propose a knowledge-based recommendation model to identify new user preference timely and automatically recommend items for making the new users feel instantly satisfaction. And exploit user behavior tracking strategy conquers gray sheep problem to guide a suitable recommendation [75].

The remainder of this chapter is organized as follows. Chapter 4.2 briefly introduces the related theory and their applications. In Chapter 4.3, we describe a heuristic, knowledge-based framework that incorporating the Formal Concept Analysis into customizing recommendation. An implementation with respect to movie recommender system is given in Chapter 4.4, followed by experimental studies that evaluate recommendation performance of our proposed approach in various settings. Finally, we conclude in Chapter 4.5 with further research issues.

4.2 Related Theory

4.2.1 Formal Concept Analysis

Formal concept analysis is a principled way introduced by Rudolf Wille in 1984, deriving a concept hierarchy or formal ontology from a collection of concepts, objects and their properties, and relations with formal expression. Each concept in the hierarchy represents the set of objects sharing the same values for a certain set of properties; and each sub-concept in the hierarchy contains a subset of the objects [16, 17, 18].

In formal concept analysis, extension of the concept is consisting of all objects that share the given attributes, and the intension is consisting of all attributes shared by the given objects. All concepts are formed as a concept lattice, together with the generalization / specialization relationship between them. Concept lattice is the core data structure of this theory. Each node of the concept lattice is a concept, which is composed of extension and intension. Extension is an instance of the concept, and intension is the description of the concept, it is the common feature of the examples covered by this concept. The structure model of concept lattice is the core data structure in formal concept analysis theory. It essentially describes the connection between objects and features, and represents binary relations between objects and attributes. The process of constructing the concept lattice is semi-automated.

4.2.2 Knowledge Representation Formalism

Knowledge formalization lends itself to dealing with the information of product and makes it possible to capture customer requirements more accurately. In an intelligent recommendation, acquisition of knowledge matched with the user preference is a crucial and complex problem. To increase the recommendation rapidity it is necessary not only to find the required sources but also to identify their relationship and usefulness for solving a particular problem. The formalization of item knowledge has been chosen for the ontology domain representation. Our tracking recommendation model is based on this conceptual hierarchy of knowledge formalization. This solution was mainly motivated by such factors as increase intelligibility of items knowledge representation, facilitate the knowledge domain sharing, and discovery agilely a candidate set of items that will be of interest to a certain user.

4.3 Proposed Knowledge-based Tracking Recommendation Approach

In this work, we present a new approach that incorporating the Formal Concept Analysis into recommendation process. Our recommendation approach infers user preference based on analysis and extension of current user behavior. We exploit deep domain knowledge in the form of mapping between user preference and required items characteristics. The proposed recommendation approach is consisting of three main components: knowledge representation formalism, description of knowledge repository, and tracking recommendation procedure. One of the major components of our recommendation framework is a repository that includes knowledge source model, user interest model and associative inference rule descriptions.

4.3.1 Description of Knowledge Repository

The main component of our proposed approach is a repository that includes knowledge of user preference, item properties and associative extension rules. We define this repository as three modules: (i) user profile model is an organized storage of information for a user behavior and interest; this module is used for faster capturing and analyzing user request and his preferences. (ii) item profile of knowledge resources model is used for knowledge representation; this module describes the synthetical information about item resources and their characteristics. (iii) FCA model is a formal description of items and their properties, relationships between them; this module provides a generic mapping domain that captures the key distinctions, facilitates extraction of the concepts among these domains, and speeds up the process of candidate recommendation set choice.

(1) **User profile model**

User profile is formalization description of user interests (12). For each user, the user profile is defined as follows:

$$\mathrm{UP} = \{up_x \in \mathrm{UP} \mid 1 \le x \le \mathrm{P}\} \cdots\cdots\cdots\cdots\cdots\cdots (4.1)$$

where P is the total number of user records. UP means the user profile set and up_x is element of this set that has quite different in the different domain. up_x can represent behavior record of user. User profile is a dynamic set; it will be update constantly with the changing of user behavior.

(2) **Item profile model**

Item profile is formalization description of item features for specific domain. We defined this item profile as a tripartite set containing object subset, classificatory attribute subset and descriptive attribute subset as follows:

$$\mathrm{IP} = \{ip_k = [\{O_k\}, \{ca_i^k\}, \{da_j^k\}] \mid ip_k \in \mathrm{IP}\} \qquad (4.2)$$

IP means the item profile set and ip_k represents one record of item profile. Here, we use OBJ as the finite set of items on the given domain and O_k represent the element of this object subset. This can be expressed as follows:

$$\text{Object set: OBJ} = \{O_k \in \mathrm{OBJ} \mid 1 \le k \le \mathrm{L}\} \cdots\cdots (4.3)$$

Likewise, we use two level attribute set to describe or classify the object subset, the prime level is called classificatory attribute set (CA) and the secondary level is called descriptive attribute set (DA).

Classificatory attribute set:

$$CA = \{ ca_i^k \in CA \mid 1 \leq i \leq M \} \cdots\cdots\cdots\cdots\cdots\cdots (4.4)$$

Descriptive attribute set:

$$DA = \{ da_j^k \in DA \mid 1 \leq j \leq N \} \cdots\cdots\cdots\cdots\cdots\cdots (4.5)$$

where L is the total number of items, M is the number of classificatory attribute, N is the number of descriptive attribute. Note that ca_i^k and da_j^k are respectively element of each attribute set.

(3) **FCA model**

The key element in our proposed model is the feature analysis of attributes, where items have the internal relation according to their two-level attributes. This tracking recommendation model can recommend to user related items with update of the user behavior records. We exploited Formal Concept Analysis (FCA) methodology generate a conceptual hierarchy of domain by finding all possible formal concepts that reflect relationships between attributes and objects. As well known, FCA is a mathematical theory that formulates understanding of concept as a unit of thought comprising its extension and intension as a way of modeling a domain [16].

In a formal context, the objects and attributes can be grouped into the equivalence classes based on equivalence relations. The order relations define hierarchical structures of the equivalence classes. The elements of one type are called "formal objects", the elements of the other type are called "formal attributes". There exist some basic sets of objects and attributes that are used to not only reflect equivalence relation and order relation, but also constitute all formal concepts derived from the formal context. If a formal concept has a formal attribute then its attributes are inherited by all its sub concepts [17].

Table 4.1. The principal formal context about movies.

Object		Classificatory attribute				Descriptive attribute		
No.	movies	a. comedy	b. adventure	c. romantic	d. story	year	nation	main actor
1	Mr. Bean's Holiday	√			√	2007	UK	Rowan Atkinson
2	Beyond Borders			√	√	2003	USA	Angelina Jolie
3	Runaway Vacation	√	√		√	2006	UK	Robin Williams
4	Transformers	√	√	√	√	2009	USA	Megan Fox
5	Angels and Demons		√		√	2009	USA	Tom Hanks
6	Napoleon with me	√	√	√		2006	Italy	Daniel Auteuil
7	Johnny English	√	√		√	2003	UK	Rowan Atkinson

Table 4.1 shows an instance that consist of objects set {1, 2, 3, 4, 5, 6, 7} and the classificatory attributes set {a, b, c, d}. There are four sorts that describe seven concepts as prime level attribute set. They are expressed respectively in row and column of matrix. In a formal context, a set of objects is referred to as extension and represents instances of a concept. A set of attributes is referred to as intension and characterizes the features or properties of a concept. In other words, an object set and an attribute set can uniquely associate with each other. We build a relation network model between object set and the prime level attribute set of items by FCA model with some specific nodes; the nodes are item set which have some common properties. Construct the FCA map that consists of nodes and relationships according to Table 4.1 as shown in Figure 4.1. This FCA map is a hierarchical integration of conceptions from the root node to a specific node. A node stands for several conceptions and what have common feature. A relationship between two nodes represents the order of two conceptions.

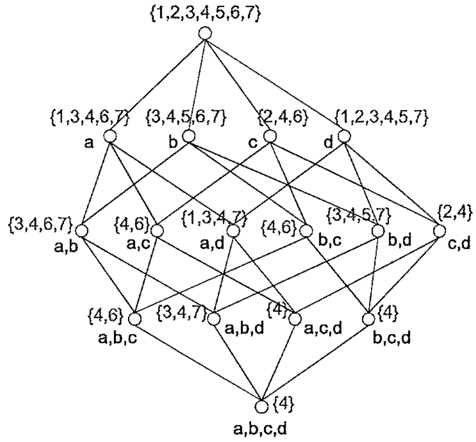


Figure 4.1. The FCA map of movies.

The layer of FCA map is determined by the number of attributes. As shown in Figure 4.1, the FCA map has five layers. The first layer includes one node; which is a set that includes all of the objects. And the last layer includes one node; which is a set that includes all of the attributes. Each node in this FCA map denotes some objects had some common attributes.

4.3.2 Tracking Recommendation Procedure

Once the FCA mapping domain was built, the recommendation set can be obtained efficiently by inferring user preference on second level properties. The tracking recommendation process is explained as Figure 4.2.

When a user clicks one item, our recommendation model can recommend timely related items to user. With the update of user's behavior records, the result of recommendation is changing constantly regardless of the user profile enough or not. The process of tracking recommendation is described in more detail below.

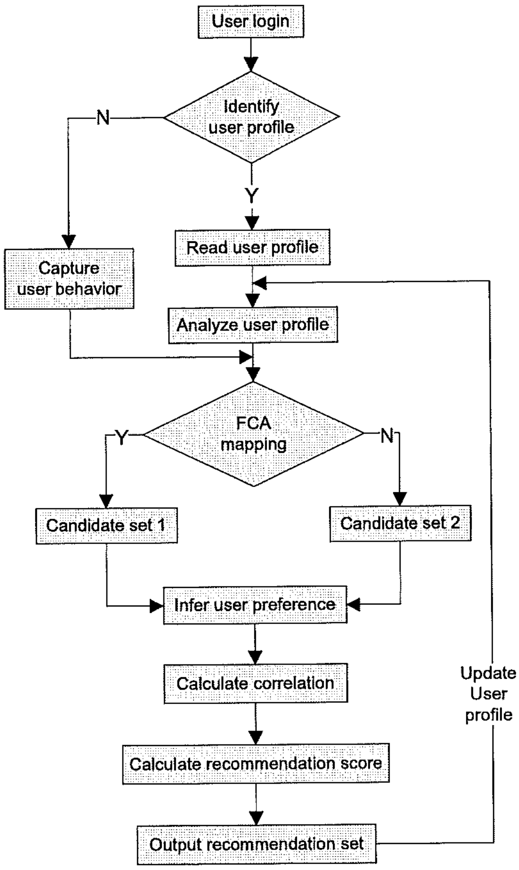


Figure 4.2. The process of tracking recommendation.

Step 1: Identify user profile.

For a new user that has no information about his preference, we capture his behavior and add current browsing item $O \in OBJ$ into UP. For the user who has historical record, we need to acquire his preference firstly. In our approach, user behavior is clicked or watched a movie. We use a set recording the result of user behavior. This result of user behavior is a set that includes the titles of movies that watched by user. For example, if a user watched movie "Transformers", "Johnny English" and "Mr. Bean's Holiday", then this user profile can be expressed as {Transformers, Johnny English, Mr. Bean's Holiday}.

One case is we can directly discover the minimal set which contains user previous behavior record set UP, and extend one layer upwards in FCA map to construct the initial candidate items set. The layer of FCA map is determined by the number of attribute. As shown in Figure 4.1, this FCA map has five layers. In order to obtain initial candidate set, we need to extend user behavior. For example, if user clicked items {3,4,7}, then we look for the minimal set which includes this set bottom-up in Figure 1. By extending one layer upwards, we can find the minimal items set {3,4,6,7}; {1,3,4,7}; {3,4,5,7} which include set {3,4,7}. Then, the union of these minimal items set and the initial candidate set {1,5,6} are easy to get.

Another case is after traversing the FCA map; we find the node that contains user previous record set UP simultaneously is inexistent. On this condition, search the minimal set which includes each element of this set, and extend one layer upwards in FCA map respectively is essential.

Step 2: Confirm the candidate set CS.

The primary candidate set is facilitated from FCA map. Uniting these set and removing the historical record from it, then we can get the candidate items set CS. This candidate set CS provides the candidate items and their attributes information for user. In our recommendation model, candidate set CS is consist of candidate item e_i, $i \in [1,t]$. e_i denotes the candidate item of candidate set CS; i is the number of candidate items.

Step 3: Infer individual interest in candidate set.

Intuitively, we discover user preference merely relies on FCA map is impractical. But it is facilitated to find the potential relations between user record and the candidate items based on secondary level attribute. Table 4.1 shows an illustration of descriptive attributes about movies.

Regard secondary level attributes set as the key criterions of weight. We define the user's interest is expressed as the ratio of times maximal frequency items for each secondary attribute j over the total number of same items. We denoted this weight w_{xj} as a weight to represent the important degree of user for a descriptive attribute value and defined as follows:

$$w_{xj} = \frac{N_{xj}}{\sum_{x=0}^{P} N_{xj}} \quad \cdots\cdots(4.6)$$

The annotations, N_{xj} represents the most frequency items for the secondary attribute j, $j \in$ DA. N_{xj} equals maximal number of same items for second level attribute when attribute j has the same items. Otherwise, N_{xj} equals 1.

Step 4: Calculate correlation C_{xi} between up_x and e_i.

The correlation C_{xi} represents pertinent degree between user former record up_x and candidate item e_i. C_{xi} is the sum of matching degrees f_{xi}^j between up_x and e_i for each secondary attribute j multiplied by the corresponding weight w_{xj}. C_{xi} is calculated by the following formula:

$$C_{xi} = \sum_{x=0}^{P} f_{xi}^{j} \times w_{xj} \quad \cdots\cdots(4.7)$$

Note that f_{xi}^j describes the matching degree between up_x and e_i for each secondary attribute j. It can be expressed as the vector $(f_{xi}^1, \cdots\cdots, f_{xi}^j)$. f_{xi}^j equals 1 when up_x match e_i for attribute j; otherwise f_{xi}^j equals 0. For each $e_i \in$ CS, we can get all correlation C_{xi} with up_x. Here, w_{xj} is satisfied with normalization:

$$\sum_{x=0}^{P} w_{xj} = 1, \ w_{xj} > 0 \quad \cdots\cdots(4.8)$$

Step 5: Calculate recommend score *rs*.

The formulation of recommender score *rs* for each e_i as follows:

$$rs = \frac{n_c}{n_{up} + n_c} \times \overline{S} + \frac{n_{up}}{n_{up} + n_c} \times \sum_{x=0}^{P} C_{xi} \times R_i \quad \cdots\cdots(4.9)$$

In this formula, $\overline{S}$ is the average score of mass rating for each item. R_i is user rating for each candidate items i. n_c represents the number of candidate item where $\sum_{x=0}^{P} C_{xi} > 0$. n_{up} represents the number of history record for each candidate item where $C_{xi} > 0$. From this formula, we can see the larger n_{up}, that is the more historical record of associated with candidate item, the larger probability that user is interested in this item.

Step 6: Output the recommendation result.

The items can be recommended by ranking score of candidate items on a website and on tracking analysis of user behavior as prediction of his future behavior. The recommender score for each candidate item by Formula (4.9). Finally, top-y items as recommendation results are returned to user with the highest recommender score. y is the number of items which recommended to user. In general, decision maker can determine this threshold value. Here, we set y is less than or equal to 10.

4.4 Experimental Study

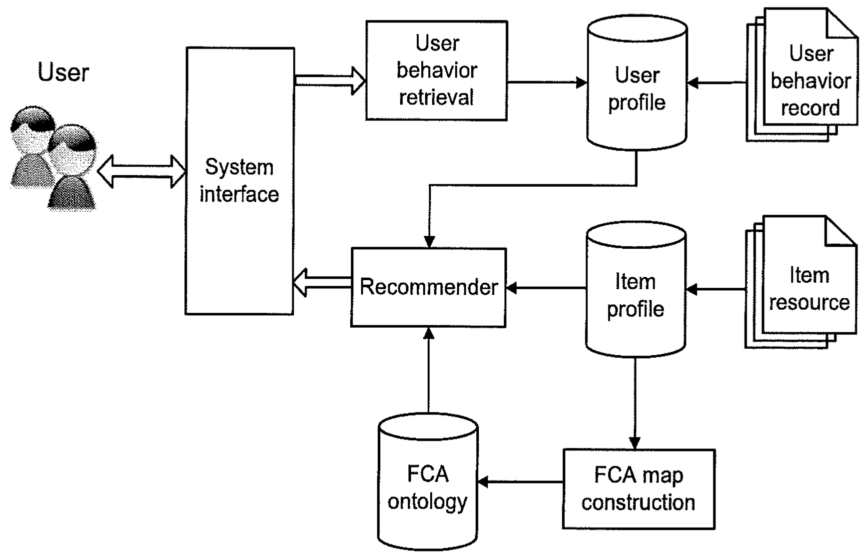


Figure 4.3. The architecture of movie recommender system.

In order to prove the approach presented in Chapter 4.3, several experiments were performed focusing on testing recommendation performance in new user conditions. To this end, we implement a recommender system on movie recommendation application. This prototype movie recommender system was performed on Dell workstation in Microsoft Windows environment and developed by JSP, JavaScript, and MySQL. The architecture of our movie recommender system is illustrated in Figure 4.3. The goal of our system is attempting to improve recommendation performance when the number of history records is small or user's preference is unique.

4.4.1 Implementation

In our experiment, we select the following two public datasets shown in Table 4.2 as benchmark:

- MovieLens is a famous recommender system and virtual community website that recommends films for its users to watch, based on their film preferences. This dataset provides 100,000 ratings for 1682 movies by 943 users. User ratings of movies were recorded on a discrete scale from 1 to 5. More than 100 movies were rated per user [19].
- EachMovie is one of the most widely used datasets in recommender systems research area. This dataset contains ratings from 72916 users on 1628 movies. User ratings were recorded on a discrete scale from 0 to 5. On average, each user rated about 30 movies [20].

Table 4.2. Summary of two datasets.

Dataset	Description	Profile	Ratings per user
Movielens	Ratings of movies in scale of 1-5	100,000 ratings for 1682 movies by 943 users	On more than 100 movies
Eachmovie	Ratings of movies in scale of 0-5	2,811,983 ratings for 1628 movies by 72916 users	On more than 30 movies

The experiment process of this study is shown as follows:

(1) In the first stage of experiment, according to the genres of movie in two movie datasets, we generate two FCA ontologies respectively. The one FCA ontology includes 18 layers and 1682 movies for MovieLens dataset. And the other FCA ontology includes 11 layers and 1628 movies for EachMovie dataset.

(2) In the second stage of experiment, we test the measures for new user condition where only from 1 to 25 ratings by 5. Our recommender system sequentially deletes user's newer records and makes each user's history records fewer than 25 movies. Using three recommendation approaches: our proposed approach, the standard collaborative filtering approach and hybrid approach to list the top 10 recommended movies to the same users respectively. Thus, our recommender system can respectively produce three groups of recommendation results for two movie datasets. We test the performance of three algorithms by calculating the average rating of users for these recommended movies.

(3) In the third stage of experiment, we test the user's history records with the percentage of user's history records were increased from 0 to 100% by 20. Then, each user only has fewer than 25 movies. We use the data described above test the performance. Our recommender system returns the list of recommendation to users based on their history records. We observe these results and make the comprehensive evaluation using metrics that introduced in Chapter 4.4.2.

(4) In the fourth stage of experiment, we use the collaborative filtering algorithm to test all of users in two movie datasets. We find approximate 15% of gray sheep users in datasets cannot get the recommendation results. In other word, the collaborative filtering recommender system doesn't work for these 15% of users. Next, we apply our proposed algorithm to produce the recommendation results for these 15% of users. We observe that our recommender system whether can work for these gray sheep users and discuss in Chapter 4.4.4.

4.4.2 Empirical Evaluation

Our evaluation was based on precision and recall, a standard benchmark that has been widely used in Information Retrieval [17]. In our experiment, precision is the percentage of actual relevant items that displayed to users, whereas recall is the percentage of active displayed items that system recommended. The confusion matrix is shown in Table 4.3.

Table 4.3. Confusion matrix for precision and recall.

	Relevant	Non-relevant
Displayed	TP	FP
Not displayed	FN	TN

TP(True Positive) is the number of relevant items that displayed to user in the test set; FP(False Positive) is the number of Non-relevant items that displayed to user; and similar for FN(False Negative) and TN(True Negative). Referring to Table 4.3, the precision and recall are calculated respectively by following formulas:

$$precision = \frac{TP}{TP+FP} \quad \cdots\cdots(4.10)$$

$$recall = \frac{TP}{TP+FN} \quad \cdots\cdots(4.11)$$

Note this metric is focus on measuring the capacity of the recommender system for making successful decisions and doesn't take TN into account [18].

4.4.3 Result Comparisons with Traditional Approaches

To demonstrate the advantages of our approach, we choose a standard collaborative filtering (CF) recommendation approach [19] and a representative hybrid recommendation approach [20] to compare based on two different datasets. To this end, we also simulate the above two algorithms on our movie recommender system and get different recommendation results for the same user with the same behavior records. We use the above evaluation metric to evaluate the performance of our proposed algorithm and other two recommendation algorithms. The testing results are shown in Figure 4.4, Figure 4.5, Figure 4.6 and Figure 4.7 respectively.

Firstly, we simply compared the average user ratings when user's records are not more than 25 based on two datasets. As shown in Figure 4.4 and Figure 4.5, X-axis represents user's records from 1 to 25. Y-axis represents the user average rating. The results in Figure 4.4 and Figure 4.5 show that the user average ratings of our approach are higher than the user average ratings of other two approaches. This result indicates better performance of the proposed approach.

The above experimental result shows the strength of our approach for new users. However, there is another condition: users with the few purchase records but sufficient history records. Accordingly, we do the next experiment to test the performance of our approach using the metrics described in Chapter 4.4.1.

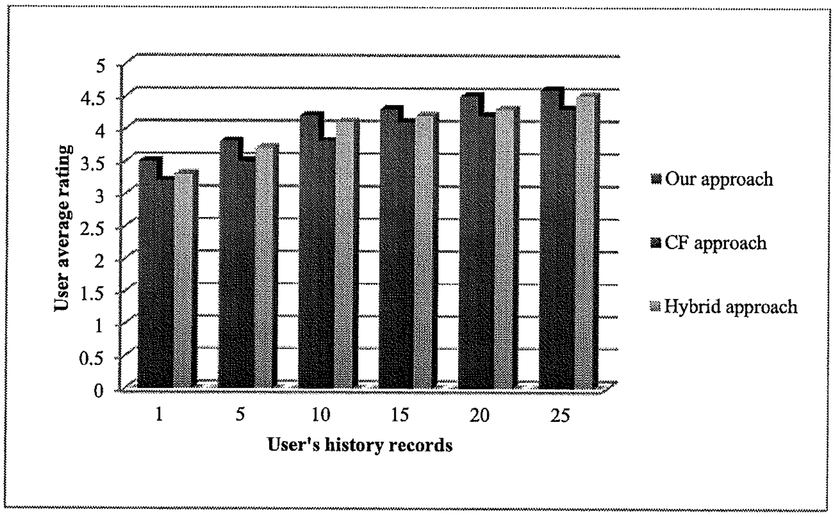


Figure 4.4. User average rating when user's records are not more than 25 for MovieLens dataset.

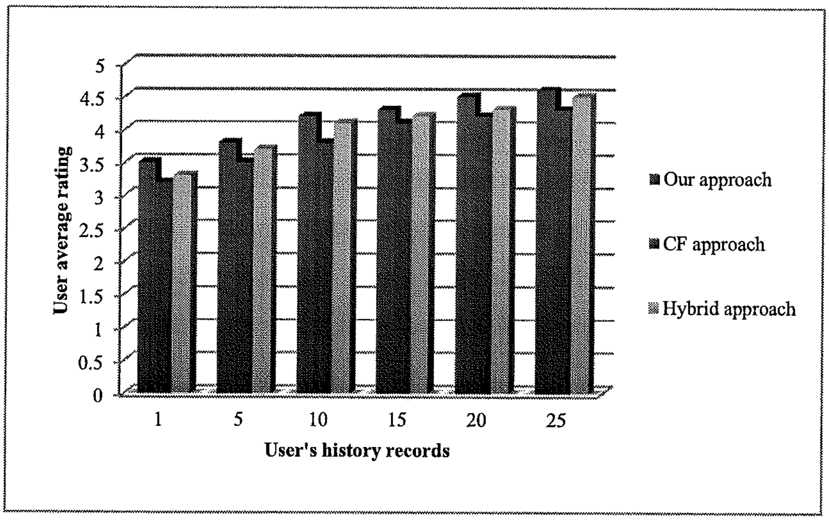
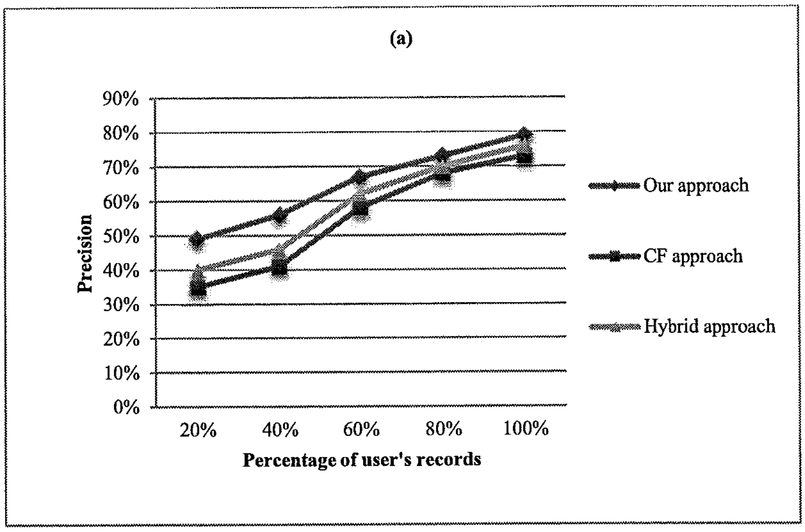


Figure 4.5. User average rating when user's records are not more than 25 for EachMovie dataset.



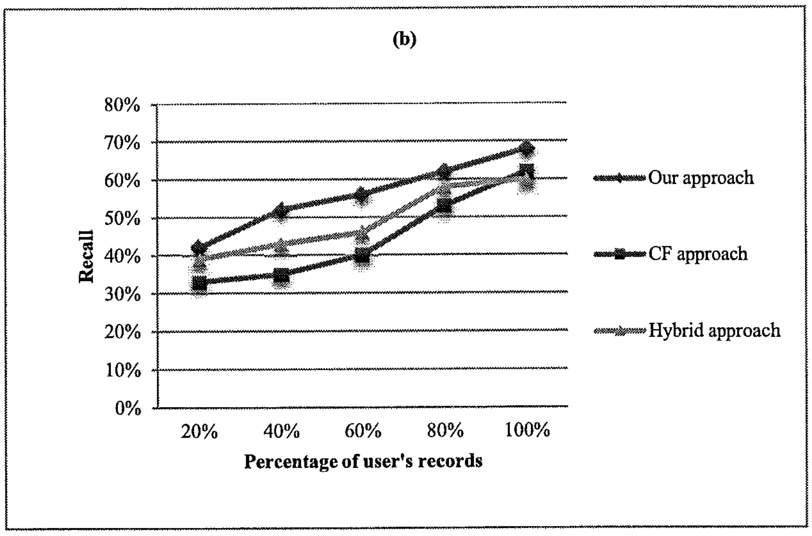
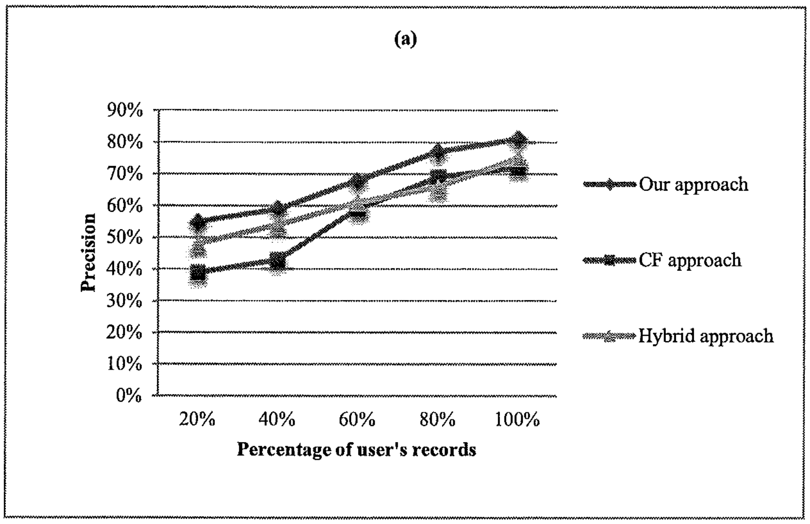


Figure 4.6. Different percentage of user's records (no more than 25) on MovieLens.



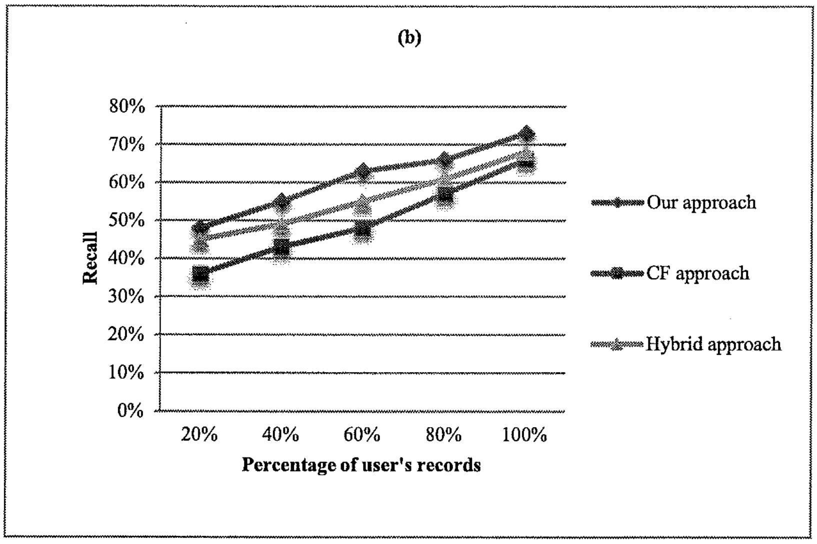


Figure 4.7. Different percentage of user's records (no more than 25) on EachMovie.

As shown in Figure 4.6 and Figure 4.7, X-axis represents the percentage of user's records. Y-axis represents the precision in Figure 4.6 (a) and Figure 4.7 (a); Y-axis represents the recall in Figure 4.6 (b) and Figure 4.7 (b).

From the results shown in Figure 4.6 and Figure 4.7, it can be concluded that our proposed approach showed advantage over the standard collaborative filtering recommendation approach and hybrid recommendation approach in the cold-start conditions, especially when the percentage of user's history records is small. In other word, despite the precision rates and the recall rates are still less than 60% when user's history records fewer than 40%. But our approach can improve the precision rates and the recall rates compared to the standard collaborative filtering approach by up to 15.3% and 12.5% respectively in new user cold-start conditions.

4.4.4 Summary of the Findings and Discussion

The summary of the experimental results is given in Table 4.4. Although recommendation result that returned to new user is still unsatisfied, the precision and recall of our presented approach are respectively better than that of standard collaborative filtering recommendation approach and hybrid recommendation approach. The evaluation results indicate that our proposed approach could improve the recommendation performance and dominate previous recommendation approaches [21, 22]. This improvement also demonstrates our approach had advantage in dealing with the new user cold-start problem.

Table 4.4. Summary of the experimental results.

Experiments	Results
New users (not more than 25 records)	Our proposed approach is slightly better than the hybrid approach and dominates the standard collaborative filtering approach
The different percentage of user's records (from 0 to 100% by 20)	Our proposed approach outperforms the traditional recommendation approaches. Despite the precision rates and the recall rates are still less than 60% when user's records fewer than 40%. But our approach can improve the precision rates and the recall rates compared to standard collaborative filtering approach by up to 15.3% and 12.5% respectively in new user cold-start conditions
Gray sheep (approximate 15% of users)	Our proposed approach successfully avoids the occurrence of gray sheep problem

For the users that are new to a recommender system, no information about their preference is initial known. The collaborative filtering recommender systems typically request them to rate a set of query items. Gained the rating on these query items, recommender systems can start working. We have three reasons to disapprove this way: (i) users do not like to rate a long list of items; (ii) it is difficult to rating unknown items for users; (iii) the rating results of some items might be informative for determining user's profile whereas rating results for other items might not provide useful new information. To avoid these occurrence, we inference user preference based on analysis and association of user behavior, and make a prediction for the items which user might be interested in the future.

For gray sheep problem, finding a similar group of user is impracticable in collaborative filtering recommender system when user has a larruping taste. Our proposed approach takes into more consideration the individual interesting with update of his behavior records. Tracking his recent behavior, we can capture the individual preference on real-time and produce a powerful recommendation by analyzing the attribute of current preferred items and extending on FCA map without finding the users who have similar preference. Therefore, our approach can avoid completely the occurrence of gray sheep problem under applying the superiority of FCA map.

4.5 Concluding Remarks

This study introduces a new heuristic approach that utilized rationally tracking analysis of user behavior and the two-level properties of items set as key factor of an associated extension. One advantage of this approach is that no additional information is required for the shortcomings of current recommender systems. Our presented algorithm is more flexible to solve the new user problem through integrated utilization of FCA ontology. To deal with new users, our approach can discover user preference without user's additional information by analysis and extension of user behavior in FCA map. We simulate a movie recommender system to demonstrate our proposed method was indeed more robust against drawbacks and obtained better recommendation results than the conventional approaches in some cold-start conditions. And our recommendation strategy successfully avoids the occurrence of gray sheep problem that is ubiquitous limitation in collaborative filtering recommender systems.

In this work, we have sketched how FCA model can be used to help solve the cold-start problem. However, our study in this area was very small, and a lot more work is required to validate and optimize this approach.

Chapter 5
Tracking User Transitional Preferences in Context-aware Recommendation

5.1 Overview

From a commercial perspective, recommender systems not only boost sales for enterprises but also resolve information overload problems [1, 27, 28]. However, despite these advances, the extent of improvements necessary to create more effective, accurate and individualized recommendation methods that represent user preferences and behaviors and are applicable to a broader range of real-life applications is still considerable. The question is not whether intelligent recommender systems can mimic emotions, but whether recommender systems can mimic intelligent behavior without emotions. This approach is necessary to make the recommendation process more transparent to the users. To achieve peak performance, a recent and notable question in this research area focuses on how best to learn about user preferences with relevant contextual information from their surroundings and to provide real-time and vivid recommendations. Therefore, we must take into account not only information about the users and the products but also must incorporate information about the situational context into the decision-making environment; such user information as time, location and emotional state is crucial to the quality of recommendation results. In other words, if the systems are disconnected from the recommendation environment or situational context, it is quite difficult to make valid decisions for recommendations.

The goal of improving the recommendation quality in electronic commerce recommender systems requires an emphasis on personalized services that must recommend the right product in the right environment to the right user. For example, recommending a suspenseful movie to a user who has had a busy day is unadvisable. To this end, it is essential to understand the individual user preferences and emotions and to take into consideration the contextual multidimensional information required for systems that recommend products to users [29]. With this motivation, our study focuses on integrating psychological factors and chromatics into the decision-making process. Our profile model not only relies on the traditional two-dimensional factors (i.e., user and item) but also utilizes contextual factors via mood and color to solve the movie recommendation problem. Furthermore, understanding how best to incorporate these multidimensional factors and exploit them in the recommendation model is the major purpose of our study.

This study presents a novel affect-based approach for timely retrieval of information of individual interest from a repository and subsequent suggestion of items to assist users in reaching a decision that meets their preferences [71, 72]. The rest of this Chapter is organized as follows. In Chapter 5.2, we briefly introduce the recent recommendation techniques and their limitations. Chapter 5.3 describes an affect-based multidimensional framework incorporating cognitive psychology into individualized recommendations. An implementation prototype is presented in Chapter 5.4 to demonstrate the proposed approach with respect to a case study. Chapter 5.5 concludes this Chapter with possible extensions for future work.

5.2 Related Theory

5.2.1 Spreading Activation

The spreading activation network is similar to the semantic network. This network structure typically consists of nodes connected by weighted links. A node represents a concept and a link represents the semantic relationship between two concepts. Concepts are expanded based on the semantics in the process of identifying user preference and matching items [35].

The spreading activation algorithm operates as a concept explorer. Given an initial set of activated concepts and certain restrictions, the activation flows through the network to reach other concepts that are closely related to the initial concepts. The spreading activation algorithm includes several steps: (a) adjusting inputs, (b) spreading concepts, (c) calculating outputs, and (d) spreading termination [39]. Spreading represents a diffusion process among the nodes. Two actions control the spreading handling procedures, namely the pulse spreading and the termination check. A pulse continuously spreads the data to the surrounding nodes and includes three handling actions, namely input value, message spreading, and output value adjustment. Input and output value adjustments control the range and weight of the spreading. The termination check is used to determine whether the termination condition is met.

5.2.2 Color Psychology

Color is ubiquitous and is a source of information. The prudent use of colors can contribute not only to differentiation of products but also to influence of moods and feelings. Given that a user's mood and feelings are unstable and that colors play a crucial role in forming user preferences, it is necessary to understand the importance of colors for decision-makers in personalized recommendation [30].

Color psychology indicates that colors may seem simple and unimportant, but color affects our daily lives more than we may know. Researchers have tested the associations between colors and emotions and have designated yellow, orange, and red as happy colors, and blue, black and gray as sad colors [31, 32]. The results show that colors are often associated with affect, and in certain cases, one color can produce multiple affective responses. These multiple affect responses for basic colors are summarized in Table 5.1 and indicate that colors may assist in gaining attention, conveying messages, and creating feelings that might increase the probability of purchase. The effects of such atmospherics have been demonstrated to influence one's emotional responses and behavioral intentions. For example, green makes people feel relaxed because it relaxes their muscles and makes them breathe deeper and more slowly. Yellow is mentally stimulating and activates memory, whereas red increases confidence. Additionally, blue lowers blood pressure, which makes one feel calm [33].

Table 5.1. The association of affective responses for basic colors.

Basic color	Affective responses
Red	anger, love, excitement
Purple	stately, dignified, fantasy,
Green	nature, leisurely, pleasant
Blue	secure, calm, tender
White	calm, lightness, neutral
Gray	depressed, hostile, yearning
Black	fear, sadness, anxiety
Yellow	joyful, cheerful, jovial

5.3 Proposed Affect-based Recommendation Approach

In our survey, we find that many of the recommender systems lead to poor recommendations that ignore the affect of user when recommending a product. We assume that the acquisition of user emotion information can be gained relatively easily via social network. Accordingly, the key research problem is how to elicit the real-time preferences of the user in a contextual recommendation space. To address this problem, we present a new approach that incorporates the user emotion information into the personalized recommendation process. Our reasoning approach infers the current preference of the given user based on cognitive psychology and adapts personal dynamic interests that evolve over time in a flexible manner. Specifically, we explore the both user's mood set and history record set joined by a color sequence set with the goal of uncovering meaningful semantic associations hidden in the knowledge base.

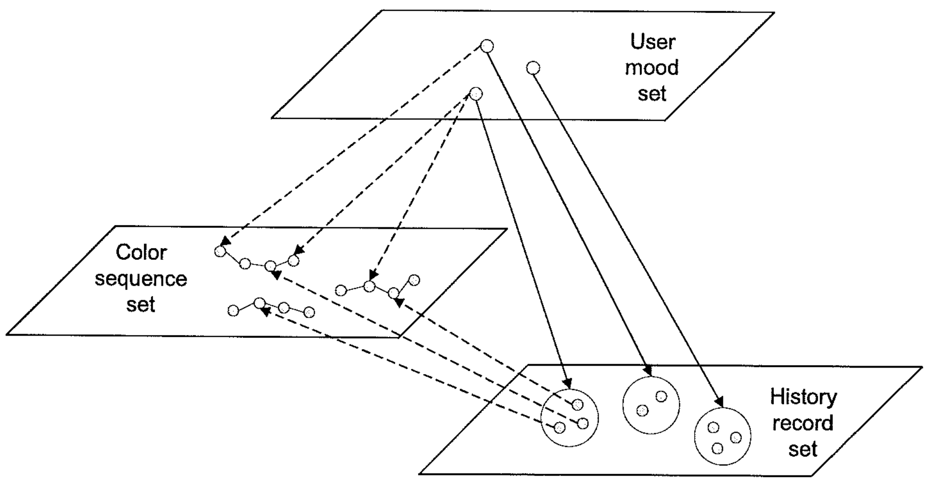


Figure 5.1. Conceptual mapping schema in the profile model.

We apply a knowledge-based conceptual mapping schema for the context model description. This context model description refers to the formal and explicit representation of concepts, which are conceived as a set of entities, relations, functions and instances. To acquire as much knowledge about the current user preference as possible, we use a color set mapping that represents the relationships of the user mood with items in a personalization environment. As shown in Figure 5.1, we classify the history records of the user by the user's past emotions. There exists a corresponding subset of history records for each user mood and there is a relationship between the user mood and the node of the color sequence; similarly, there exists a relationship between the record and the color sequence. Our proposal takes into account the potential relationships between the user states and the item features. To discover this potential relationship, it is necessary to formalize the domain of the user current state in a normalized knowledge base. Along with the user preferences and item features, this knowledge base represents specific instances referred to both the available color sequence in a recommendation that uses our strategy and to their semantic descriptions. The use of the knowledge base provides a basis for fully automatic recommendations through a domain-specific mapping mechanism. Our recommendation approach uses a unity-based notion of a multi-module conceptual hierarchy of knowledge formalization. We exploit deep domain knowledge in the form of mappings among the semantic descriptions of the item properties, a dynamic analysis of individual preferences, and knowledge from color psychology. The proposed solution was motivated primarily by these factors because increased intelligibility of item knowledge representation facilitates the ontological knowledge domain sharing and discovers the candidate items in which users would be interested.

5.3.1 Definition of Repository

Although the importance of building the knowledge base in personalized services is widely recognized, creating the specific domain is still a laborious process. We define a repository that includes the knowledge of user preferences, the item properties, and associative extension rules. The repository has three main components:

(i) The user profile model is an organized storage of information for a user's behavior and interest; this component is used for faster capture and analysis of user requests and preferences.

(ii) The item profile model is a formal description of items and their properties as well as the relationships among them. This component describes the synthetic information regarding the item resources and their features.

(iii) The color profile model is an accumulation of knowledge from life experiences, and provides a generic mapping domain that captures the key characteristics, facilitates extraction of the concepts among these domains, and speeds up the process of candidate recommendation set choice.

1) User Profile model

The user profile is a formalized description of the user interests and is a dynamic set that will be updated constantly with changes in user behavior. We define this User Profile as a triple UP = (U, M, H). U is a set of users, M is a set of user mood, and H is a set of history records. These sets can be expressed as follows:

- User set: $U = \{u_x \in U \mid 1 \leq x \leq P\}$
- Mood set: $M = \{m_x^k \in M \mid 1 \leq k \leq E\}$
- History record set: $H = \{h_x^{k,i} \in H \mid 1 \leq i \leq O\}$

where P is the total number of users, E is the number of user past states, and O is the maximal number of user history records.

$$up_x = [\{u_x\}, \{m_x^k\}, \{h_x^{k,i}\}],\ up_x \in UP \qquad (5.1)$$

Note that up_x is an instance of the user profile, and u_x, m_x^k and $h_x^{k,i}$ are the respective instances of each set.

2) Item Profile model

The item profile is a formalized description of the item features for a specific domain. We define the Item Profile as a triple IP = (I, D, S). This profile contains tripartite sets as follows:

- Item set: $\mathrm{I} = \{i_y \in \mathrm{I} \mid 1 \leq y \leq \mathrm{O}\}$
- Description of the item set: $\mathrm{D} = \{d_y^l \in \mathrm{D} \mid 1 \leq l \leq \mathrm{L}\}$
- Color Sequence set: $\mathrm{S} = \{s_y^j \in \mathrm{S} \mid 1 \leq j \leq \mathrm{N}\}$

where O is the total number of items, L is the number of descriptive attributes for each item, and N is the number of elements in the color sequence.

$$ip_y = [\{i_y\}, \{d_y^l\}, \{s_y^j\}],\ ip_y \in \mathrm{IP} \qquad (5.2)$$

Note that ip_y represents an instance of the item profile, and i_y, d_y^l and s_y^j are the respective elements of each set.

3) Color Profile model

The color profile is a formalized description of the color features. Similarly, we define the Color Profile as a two-tuples CP = (C, T). In this work, we use C as the finite set of colors on the given domain and T as a set of relevant descriptions for each color.

- Color set: $\mathrm{C} = \{c_z \in \mathrm{C} \mid 1 \leq z \leq \mathrm{N}\}$
- Description of the color set: $\mathrm{T} = \{t_z^q \in \mathrm{T} \mid 1 \leq q \leq \mathrm{G}\}$

where N is the number of colors, and G is the number of descriptive attributes for each color.

$$cp_z = [\{c_z\}, \{t_z^q\}],\ cp_z \in \mathrm{CP} \qquad (5.3)$$

Note that cp_z represents an instance of the color profile, and c_z and t_z^q are th respective elements of each set.

5.3.2 Color Library

To acquire the color sequence though an analysis of items feature, we construct a color library consisting of eight basic colors (red, yellow, blue, green, purple, white, gray, and black) and based on selected words related to the colors. We draw on three major semantic-expansion strategies: generalization, symbolization and specialization. Generalization is the representation of a concept derived from our intuitive experience, such as ivory and snow. Symbolization is the representation of a concept derived from color theory, such as truth, purity, and brightness. Specialization is the associated rational extension of a concept derived from human comprehension of an abstract object, such as angel, vacuity, and humility. This color library includes all corresponding words regarding the description of each color. Figure 5.2 illustrates an instance that the establishment of white by three major semantic-expansion strategies.

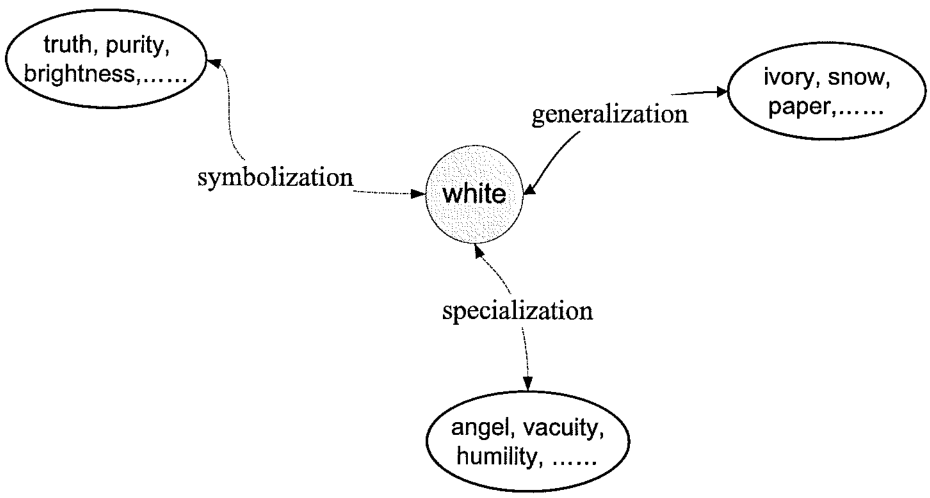


Figure 5.2 An instance of establishment of color library.

5.3.3 Characteristic Representation of Items

The key element in our proposed model is the analysis of feature attributes in which items have internal relationships according to their description attributes. We exploit the spreading activation model for retrieving this hidden information and generate a conceptual expansion of the domain by finding a color sequence that reflects the common features between the attributes and the context. The spreading activation model was developed in cognitive psychology for interpreting how semantic networks function in human brains and was inspired by the human brain cognitive models in which neurons fire activations to adjacent neurons [34]. This model includes two major components: the spreading activation network and the spreading activation algorithm.

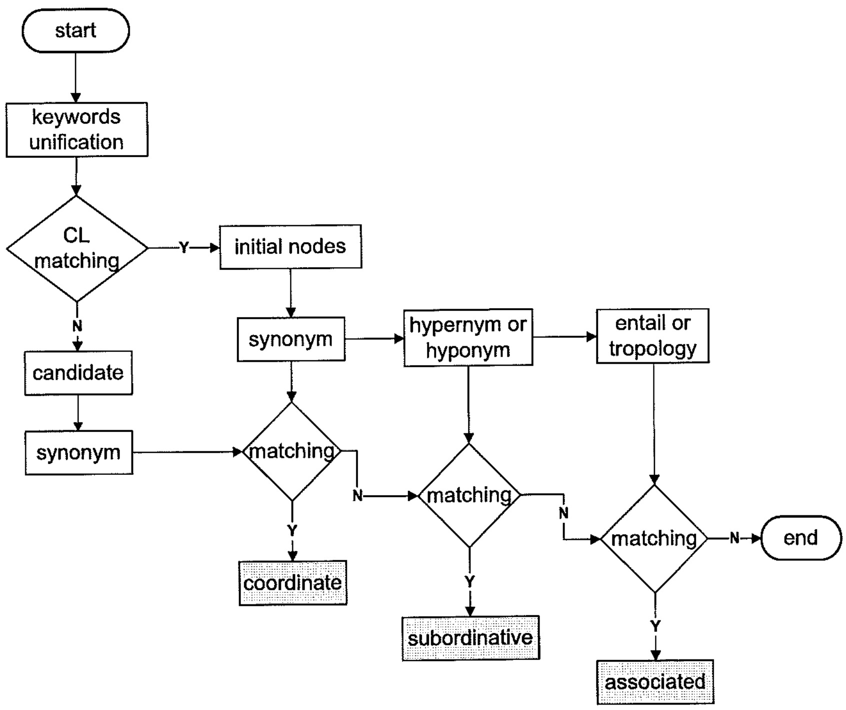


Figure 5.3 Generation of three hierarchical relationships among keywords via WordNet.

To generate the spreading activation networks from descriptions of items, we create a semantic network that consists of nodes and the relationships between them using the keywords from movies according to specific analysis based on the WordNet, a semantic vocabulary database [36, 37, 38].

The spreading activation subsequently processes this semantic network to find the most available concepts for inclusion in an extended domain. As shown in Figure 5.3, identification of three hierarchical relationships among the keywords (e.g., coordinate relation, subordinate relation and associated relation) is used to confirm and classify the found relationships. In this concept hierarchy, a node stands for one or several concepts that have a common feature. A relationship between two nodes represents the order of two concepts. If a node includes multiple concepts, the relation of these concepts is coordinate. This conceptual network structure consists of nodes connected by weighted links. Figure 5.4 shows an example of a concept hierarchy for the movie "Avatar".

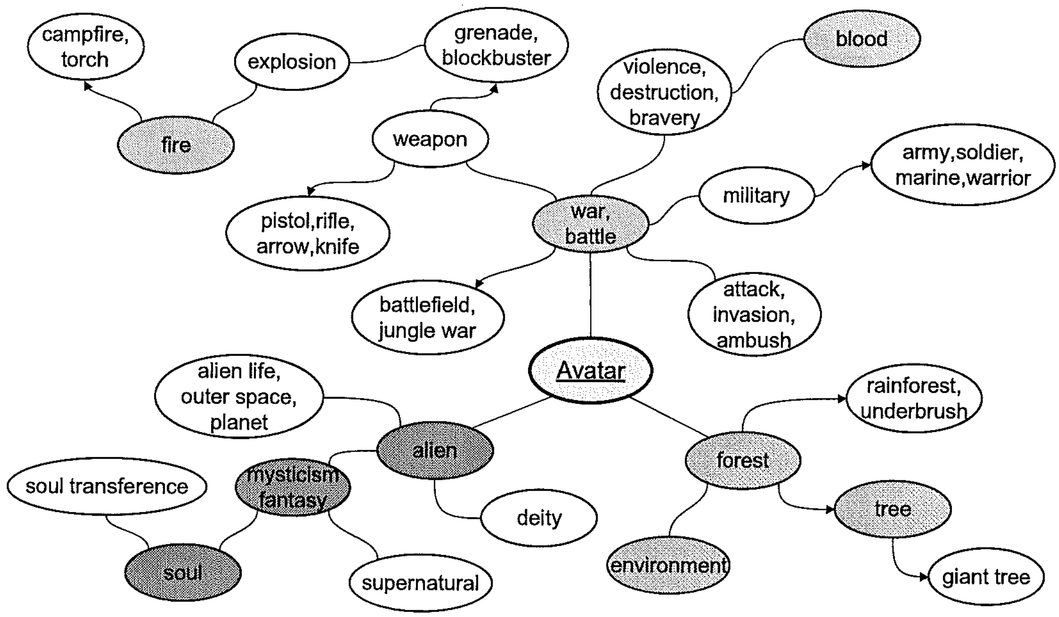


Figure 5.4 An example of a concept hierarchy network for the movie "Avatar".

We analyze the plot keywords of the movie "Avatar" to generate a characteristic sequence consisting of eight color nodes. We define this sequence with eight distinct colors: red, yellow, green, blue, purple, white, black and gray. Traveling along the concept network of the movie "Avatar", we can find the initial nodes. These initial nodes are the concepts that the plot keywords of movie "Avatar" match with the corresponding concept in the Color Library. The initial nodes for each color for the movie "Avatar" before spreading activation are expressed in Table 5.2.

Table 5.2. The initial nodes for each color for the movie "Avatar".

Color	**Initial nodes**
Red	blood, war, battle, fire, energy
Purple	mysterious, fantasy, soul
Green	environment, forest, tree
Blue	sky, calm
White	calm, purity
Gray	depression, yearning
Black	death
Yellow	moon

We use the spreading activation over the weighted graphs to calculate the activation value of each node. The initial activation node is the concept that exists in the Color Library during the spreading activation process. The spreading distance is the maximum number of levels that the system plans to spread. In our implementation, the weight represents a relevancy value between 0.0 (lowest) and 1.0 (highest) for each type of analysis. If a node has been activated, then its final activation value is the sum of the activation values of all concepts in this node.

In its simplest form, the activation value of activated node in the spreading activation is determined as follows:

$$L_j = \sum_{i=1}^{k} o_i \omega_{ij} \qquad (5.4)$$

where i denotes all nodes connected to node j, L_j is the activation value of node j calculated as the total input to node j, o_i is the previous level node i connected to node j, and ω_{ij} is a weight associated to the link connecting i and j.

The parameters are set as follows:

- The initial activation value of a node which exists in the Color Library is 1;
- The spreading distance is 1;

The weight of an associated relationship is 0.8, and the weight of a subordinate relation is 0.6.

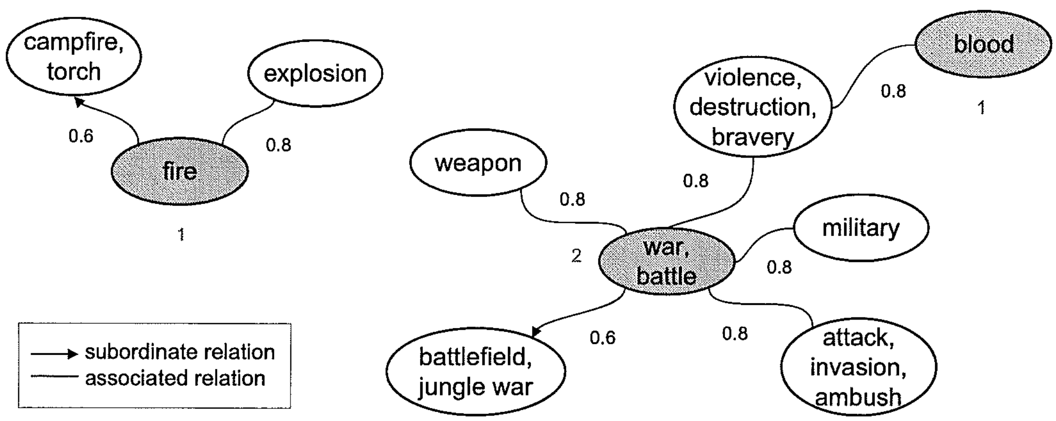


Figure 5.5 An example of the concept network after spreading activation for the movie "Avatar".

The resulting activation values of the initial concepts and the activated concepts for red after the spreading activation are shown in Figure 5.5. The activation value of the activated nodes is the activation value of the initial nodes multiplied by the corresponding weight of the links. If a node is activated by multiple initial nodes, its activation value is the sum of the activation values of each initial node, respectively multiplied by the corresponding weight that links it. For instance, the final activation value of red for the movie "Avatar" is the sum of the activation values of each node (1+1+2 + 0.6×2 + 0.8×1 + 0.6×2×2 + 0.8×2×5 + 0.8×2×3 + 0.8×1×3 = 23.6). Similarly, the final activation value of the other colors can be obtained. Thus, a color sequence for the movie "Avatar" ranked by the final activation value of each color is generated, and this color sequence expressed as a vector [red, purple, green, blue, white, gray, black, yellow].

5.3.4 Dynamic Extraction of User Preferences

To find the items that are semantically related to the user preference, we locate the user preference via a color property sequence. Once this color sequence that represents the current user interest has been inferred from the semantic associations, the irrelevant items can be filtered out, and the final recommendations are elaborated. Regarding the calculation of the user interest value, it is necessary to first compute two parameters: the color intensity of the user mood and color deviation of the user interest. The reasoning processes for the user preference are explained in detail as follows:

1) F_1: Color intensity of user mood.

Givn the user current mood k_1 and the user mood set $\{m_x^k\}$, we define the color intensity of the user mood to take account the different situations.

a) If the user current mood $k_1 \in \{m_x^k\}$, we analyze the records that satisfy this term.

$$F_1 = \frac{N_i(h_x^{k_1,i}, c_1)}{N_i(h_x^{k_1,i})} \cdot \sum_{i=1}^{a} s_i^j \qquad (5.5)$$

Note that $N_i(h_x^{k_1,i})$ denotes the number of the records that satisfy $k_1 \in \{m_x^k\}$ for user x. $N_i(h_x^{k_1,i}, c_1)$ is denoted as a to represent the number of the records that the color c_1 of the first node in the color sequence that matches with k_1 and simultaneously satisfies $k_1 \in \{m_x^k\}$. s_i^j is the color value of the property sequence for record i where j represents the node of this color sequence.

b) If the user current mood $k_1 \notin \{m_x^k\}$, we analyze the records of the first node of the color sequence matching with k_1.

$$F_1 = \frac{N_i(h_x^i, c_1)}{N_i(h_x^i)} \cdot \sum_{i=1}^{b} s_i^j \tag{5.6}$$

Note that $N_i(h_x^i)$ denotes the number of all history records for user x. $N_i(h_x^i, c_1)$ is denoted as b to represent the number of records that the color c_1 of the first node in color sequence matches with k_1 when $k_1 \notin \{m_x^k\}$.

2) F_2: Color deviation of user interest.

We define the color centrality of user interest to measure all of the relevant history records. This color centrality is the average value $\overline{s_i^j}$ with respect to the color intensity of each node in the color sequence for all history records. We find the records for which the color intensity value of the first node of the color sequence is greater than this average value. The color deviation of the user interest is determined using the following formula:

$$F_2 = \frac{N_i(h_x^i, c_1, s_i^1)}{N_i(h_x^i)} \cdot \sum_{i=1}^{d} s_i^j \tag{5.7}$$

The annotations, $N_i(h_x^i, c_1, s_i^1)$ are denoted as d to represent the number of corresponding records that satisfy $s_i^1 > \overline{s_i^j}$, where the color c_1 of the first node matches with k_1 and s_i^1 represents the color value of the first node in the property sequence for the record i.

The user preference degree is denoted by R_x and is defined as follows:

$$R_x = w_1 \cdot F_1 + w_2 \cdot F_2 \tag{5.8}$$

Once all color components have been identified, the final expression of the user preference is shown in (5.8); in this work, we assume the weights w_1=0.6 and w_2=0.4. Considering the ultimate recommendation set, the decision-maker can define a threshold value for the color sequence as the selection criterion.

5.4 Case Study

To demonstrate the approach presented in Section 5.3, we implement a system for a movie recommendation application, design the experimental process and report the results of the recommendation accuracy. Finally, we discuss selected initial findings based on this case study.

5.4.1 Implementation

To test our proposed approach, we developed a recommender system for a movie recommendation application. However, unlike the traditional movie recommender systems that provide recommendations based only on user preference, we take into consideration the affect information from the users when the system recommends a movie. This prototype movie recommender system was built on a Dell workstation in a Microsoft Windows environment and implemented via Java and XML. The architecture of this movie recommender system is illustrated in Figure 5.6.

Because none of the existing recommender systems were able to provide available data regarding the user's mood, we were unable to use existing public databases in this experiment. Thus, we needed to build our own database to address this problem. The main objectives of our experiment are to compare our approach with two traditional recommendation approaches and to show that the affect of the users does matter in a good recommendation.

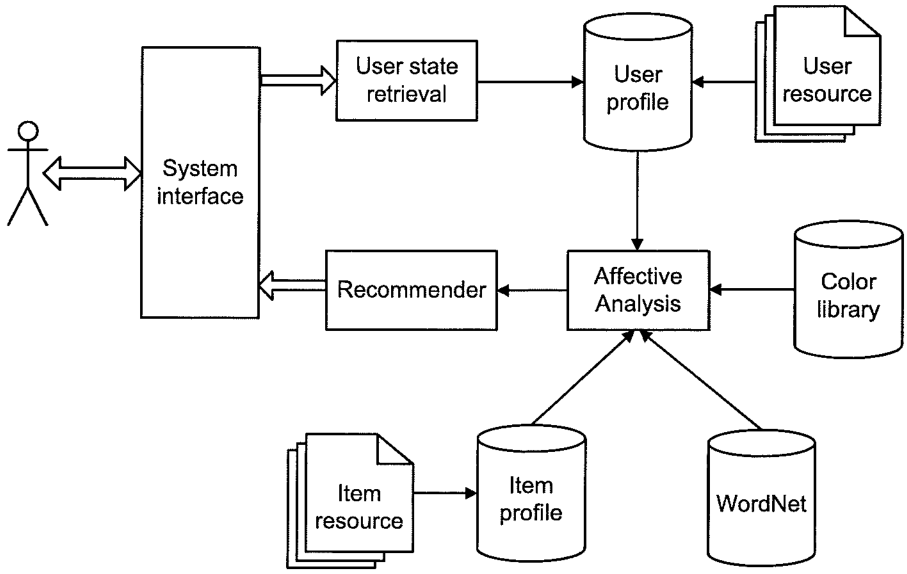


Figure 5.6 Architecture of the movie recommender system.

Table 5.3. Summary of dataset.

Dataset	Description	Profile	Mood
IMDB& Movielens	200 movies that include 45,169 plot keywords	200 movies by 60 users, per user provide at least 25 movies that he had seen under five moods	happiness, surprise, calmness, anger and sadness

The experimental process of this study is as follows:

(1) In the first stage of this experiment, we collect data of the user records from Movielens. Sixty people participated in our data collection, and the people who watched movies are used as the experiment samples. Every person provided information on at least 25 movies that they had seen under five moods, such as happiness, surprise, calmness, anger and sadness, as shown in Table 5.3.

(2) In the second stage of this experiment, we select 200 movies that include 45169 plot keywords from the Internet Movie Database (IMDB) [40]. Our movie database includes the movies that all of the above mentioned persons had seen. We use the algorithm introduced in Chapter 5.3 to generate a color sequence for each movie.

(3) In the third stage of this experiment, we implement the approach described in Section 3 and use the data described above to test the approach. We designate one of five moods as the user's current mood and the remaining four moods as the user's previous moods. Next, the corresponding movies that the user had seen under the previous mood are regarded as the history records. To do this, we used a 20%-80% ratio and performed the split five times. Our recommender system returns the list of recommendation to the user based on this user's current mood and his or her history records. We observe these recommendation results and use them to make the following evaluation.

5.4.2 Empirical Evaluation

To evaluate the recommendation performance of the proposed approach, we use two accuracy metrics to measure the recommendation results of the movie recommender system. Accuracy is a well-known performance metric in the field of Artificial Intelligence. Generally, accuracy is the most frequently used metric for estimation of the quality of nearness to the truth or the true value achieved by a recommender system [22, 25, 26]. In our evaluation, the accuracy can be formulated as in (5.9).

$$accuracy = \frac{times\ of\ successful\ recommendations}{times\ of\ recommendations} \qquad (5.9)$$

Next, we assume that a "successful recommendation" is equivalent to "the usefulness of the recommended items is close to the user's real preferences". In this work, we define two datasets: the current mood set (M_c) and the previous mood set (M_p), to compute the accuracy of the recommendation. Based on this definition, we assume that $M_c \cap M_p = \emptyset$ and $M_c \cup M_p = M$. If any movies that include the current mood set appear in the list returned to the user from the recommender system, the recommendation is deemed successful.

Precision and recall are the most popular metrics for evaluation information retrieval systems [22, 25]. Precision is a measure of exactness, and recall is a measure of completeness. In this work, we use these metrics to measure the quality of the recommendation. Precision is defined as the ratio of relevant items selected to the number of items selected, as shown in (5.10). Precision represents the probability that a selected item is relevant. As shown in (5.11), recall is defined as the ratio of relevant items selected to the total number of relevant items available. Recall represents the probability that a relevant item will be selected.

$$precision = \frac{number\ of\ correctly\ recommended\ items}{number\ of\ recommended\ items} \tag{5.10}$$

$$recall = \frac{number\ of\ correctly\ recommended\ items}{number\ of\ relevant\ items} \tag{5.11}$$

5.4.3 Result Analysis and Discussion

To compare our approach with the two traditional recommendation approaches, we also simulate content-based and collaborative filtering algorithms in our movie recommender system. This test focused on the 60 user samples and the 200 movies that they viewed. We use the evaluation metric described in Chapter 5.4.2 to evaluate the performance of our proposed algorithm and other two recommendation algorithms. The test results are shown in Figure 5.7 and Figure 5.8, respectively.

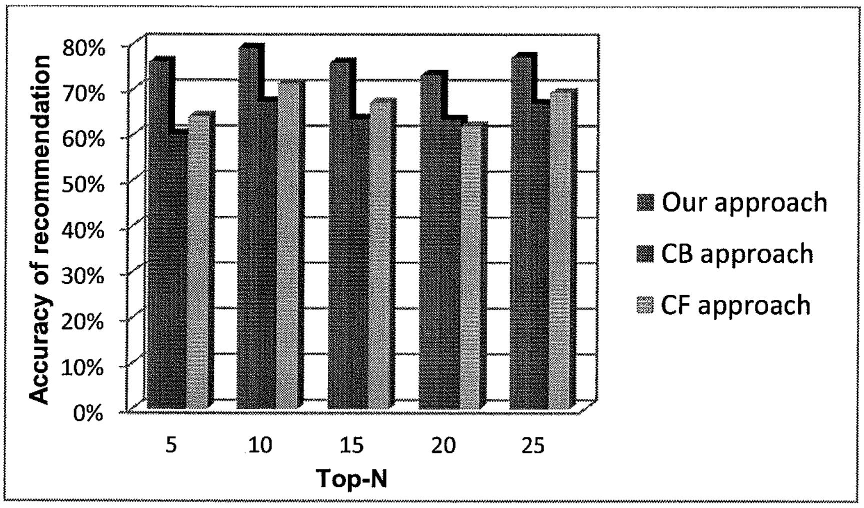


Figure 5.7 Comparison of three approaches for accuracy of recommendation.

Figure 5.7 shows the comparison of our proposed approach, the traditional content-based approach and the collaborative filtering approach in terms of the recommendation accuracy. These results indicate that our approach dominates the traditional recommendation approaches.

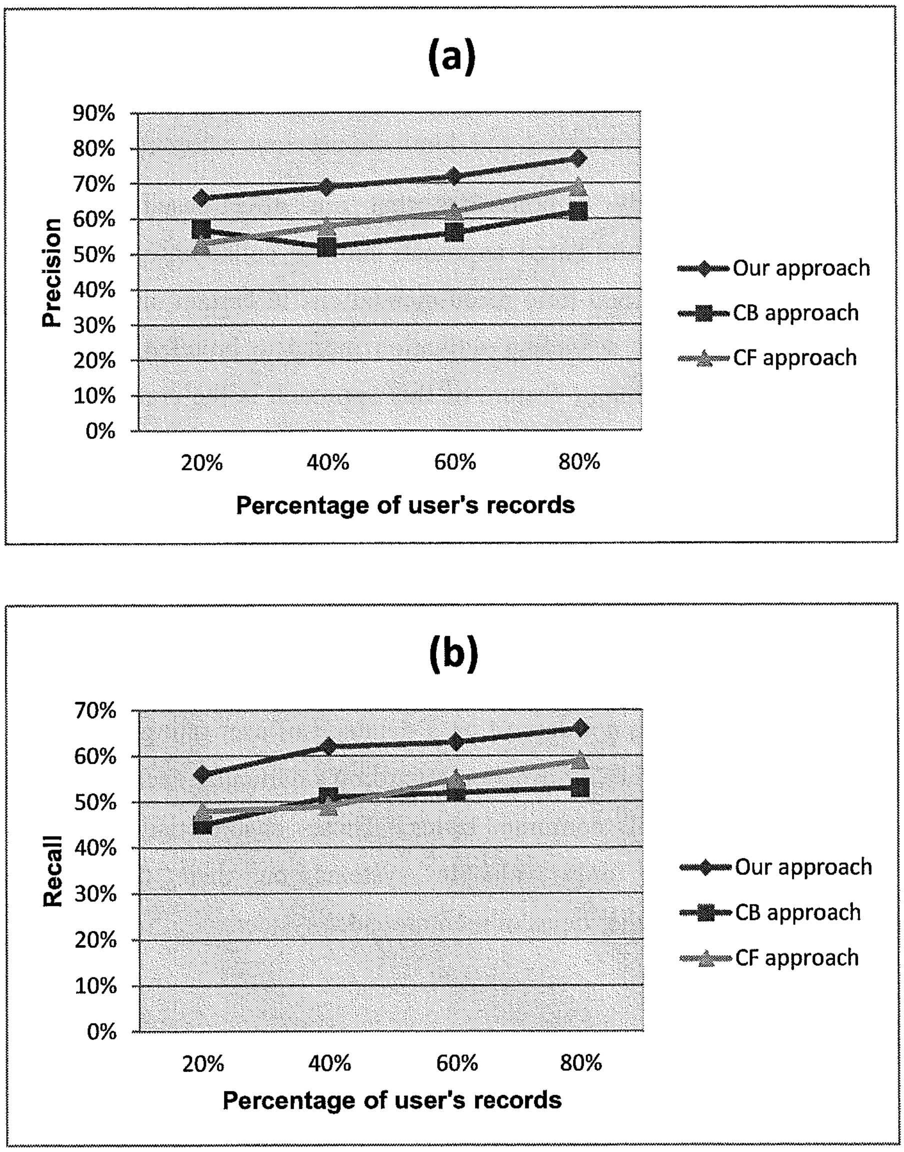


Figure 5.8 Precision and recall under percentage of user' records.

As shown in Figure 5.8, the X-axis represents the percentage of the user's records, and the Y-axis represents the precision in Fig.5.8-(a) and the recall in Fig.5.8-(b). Although the recommendation results returned to the user are unsatisfactory, the results in Fig.5.8 show that the precision and recall of our presented approach are better than those from the two traditional recommendation approaches. In summary, it can be concluded that our proposed approach showed distinct advantages over the two traditional approaches. The evaluation results presented above indicate that our proposed approach can improve the recommendation performance and dominate previous recommendation approaches.

This case study demonstrates that our affect-based approach is able to outperform the content-based approach and the collaborative filtering approach in accuracy testing of real-time recommendations in certain situations. Our presented approach adopts the spreading activation model to broaden the scope for the user profile analysis. A major feature of this approach is the construction of a semantic network that uses three relationships to connect the concepts. Apparently, building such a comprehensive and useful semantic network to cover major concepts with their relationships requires a great deal of work from professionals in the knowledge domain. Our proposal does not depend on large bodies of statistical data regarding particular rated items or particular users. This affect-based strategy avoids a subset of these drawbacks; for instance, it does not encounter the ramp-up problem because the recommendations do not depend on a database of user ratings, and thus this method does not need to gather information regarding a particular user because the judgments are independent of common tastes. These characteristics make affect-based recommenders not only valuable systems on their own but also highly complementary to other types of recommender systems.

5.5 Concluding Remarks

In this work, we describe a novel approach that exploits domain knowledge regarding product features and achieves a more rapid start-up to guide optimal choices for users by combining cognitive psychology into a personalized recommendation process. We present an intellectualized recommendation framework and create a color library via semantic description and expansion. A prototype system for movie recommendations was developed for implementing our approach and testing its performance. The evaluation results show that our approach performs better than the traditional two-dimensional approaches in terms of recommendation accuracy.

The specific novelty of our presented approach is the integration of cognitive psychology and the spreading activation network into the recommender application, which utilizes color psychology as a key factor for analysis of user preference. Additionally, we visualize the color-key of each movie to find a characteristic sequence by calculating and matching the plot keywords. Such participative architectures will enable the development of integrated knowledge domain bases and user applications. Consideration of the user affect could potentially result in more efficient recommendations in interactive environments and is strongly complementary to other types of recommendation methods. We believe that the integration of theories from cognitive psychology and decision-making into recommendation applications will provide new insights into the major factors that influence user behavior.

Context-aware recommendation is a research hot topic since it bridges the gap between recommender systems and other areas of research such as ubiquitous computing. The work presented in this chapter is only the starting point. In particular, large scale evaluations are required, as are investigations on the affect of group size. The challenge of modeling user preferences gathered from critiquing feedback is still a topic that a lot of current research continues to explore. The investigation of further approaches for the aggregation of user multi-preferences is another open challenge, as very little work has been carried out in this area to date.

Chapter 6
Detecting User Crossover Preferences for Recommendation Diversity

6.1 Overview

With the rapid growth of the Internet, information overload is becoming a daily and common problem for finding relevant content from the Internet. Recommender systems are a new and promising approach to help users handle vast amounts of information more quickly. They have made a great impacted on the way of users accessing to information. Recommender systems have become an enabling technology for e-commerce to provide interesting products and individualized services to provide the target user a personalized way to interesting or useful products among many possible options [41]. And they have been used in many applications and e-commerce web sites, including Amazon.com and Netflix.com [3]. With recommender systems, the possibilities are endless.

Most algorithms proposed in recommender systems literature have focused on improving recommendation accuracy and important aspects of recommendation quality. However, they have often overlooked the importance of recommendation diversity [42, 43]. High accuracy is often obtained by recommending the most popular items to a user, which reduces diversity. More diverse recommendations would be helpful for users and could be beneficial for some businesses [44, 45]. With this motivation, we focus on developing a new recommendation method that can increase the diversity of recommendation sets for a given individual user, often measured by an average dissimilarity between all pairs of recommended items, while maintaining an acceptable accuracy level.

This work aims to design a hybrid recommendation approach based on multidimensional clustering and collaborative filtering algorithms [73]. Our approach aims to discover users' potential preferences and provide a target user with highly idiosyncratic or more diverse recommendations, thus helping users reach a decision to meet their diverse demands. We thus propose a hybrid approach that incorporates multidimensional clustering into a collaborative filtering recommendation model to provide a quality recommendation [74, 76]. Also, this proposed approach provides a flexible solution that can improve recommendation diversity while maintaining an acceptable level of recommendation accuracy, as illustrated in this Chapter.

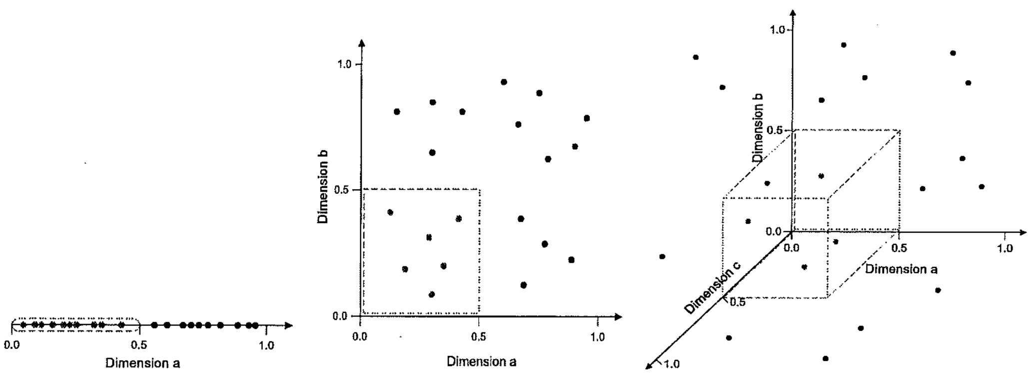
The remainder of this chapter is organized as follows. Chapter 6.2 reviews the collaborative filtering recommendation technologies and their limitations. We also briefly introduce state-of-the-art in clustering algorithms and advantage of multidimensional clustering. In Chapter 6.3, we describe a heuristic approach that incorporates multidimensional clustering algorithm into collaborative filtering recommendation. Chapter 6.4 gives an experimental study that evaluates the recommendation performance of our proposed approach. Finally, we summarize our results in Chapter 6.5.

6.2 Related Techniques

A cluster analysis seeks to discover groups with similar features. From a practical perspective, clustering plays an integral role in data mining applications, including scientific data exploration, information retrieval and text mining, spatial database applications, Web analysis and computational biology. Clustering approaches aim to detect clustered objects using all attributes in the full data space. Researchers have presented different clustering algorithms, e.g., partitioning, grid-based and density-based clustering [48].

6.2.1 Multidimensional Data Clustering

The objects in data mining could have hundreds of attributes. A challenge is how to detect clusters for the given multi-source data in one high-dimensional data space. However, independent of the underlying clustering model, full space clustering approaches do not scale to high-dimensional data spaces that cover multiple attributes. Figure 6.1 illustrates how the same data spreads out points with additional dimensions. This sample dataset contains 20 points randomly placed between 0 and 1. We define the size of a unit bin as 0.5. Figure 6.1(a), (b) and (c) show these data plotted in each of three dimensions. We observe these data's distribution, where data in one dimension are tightly packed. As the dimensions increase, data with similar features are easier to gather into a group.



(a) Cluster in one unit in one dimension (b) Cluster in one unit in two dimensions (c) Cluster in one unit in three dimensions

Figure 6.1. Same data plotted in each of the three dimensions.

To address this problem, recent research for clustering in high-dimensional data has introduced several approaches [49, 50]. Pioneers in this field called the underlying mining task subspace clustering or projected clustering. The common goal is to detect clusters in arbitrary subspace data projections. Each cluster is associated with a set of relevant dimensions in which this pattern has been discovered. Unlike traditional clustering algorithms, subspace clustering allows objects to be part of multiple clusters in arbitrary subspaces. Because traditional clustering techniques usually miss some concepts during the clustering procedure, other extensions of clustering techniques try to iteratively detect further orthogonal clusters [51, 52]. In each step, these approaches transform the data space and force the traditional clustering algorithm to find novel clusters in orthogonalized spaces. Unlike subspace clustering algorithms, they search for clusters in full space or space transformations. By transforming the data, the approaches impose constraints on the cluster detection. Orthogonal clustering typically detects some already detected clusters multiple times during the iterative procedure, it can provide knowledge about the hidden concepts in the transformed space.

6.2.2 Collaborative Filtering

The collaborative filtering approach has been presented and widely used in recommender systems to complement the content-based approach. The collaborative filtering recommender systems focus on aggregating data about users' preferences and make recommendations to other users based on similarities in overall purchasing patterns. The collaborative filtering approach aggregates users' products ratings, recognizes correlations among users' ratings and employs the product ratings of similar users to recommend new items of interest to individual users based on other customers with similar interests [2, 7]. The success of collaborative filtering recommender systems strongly depends on the classification algorithm and selection algorithm, which selects the Top-*N* product in that group. The collaborative filtering algorithms create similar groups of users who have previously purchased similar products. Though many studies provide rich evidence on their performance, there are still some shortcomings, including data sparsity and cold-start problems [47]. Data sparsity refers to the problem of insufficient data in the user-item matrix. The cold-start problem refers to the difficulty of recommending new items or new users with insufficient available records.

6.3 Proposed Multidimensional Clustering-based Collaborative Filtering Approach

In this work, we present a new approach that incorporates the multidimensional clustering algorithm into a collaborative filtering recommendation process. The procedure of our hybrid approach has been partitioned into three main phases.

Phase I includes data preprocessing and multidimensional clustering. In this phase, background data in the form of user and item profiles are collected and clustered using the proposed algorithm. After the clusters are generated in the attribute space of dimensional data, they can provide diverse sets for the future recommendations.

Phase II involves filtering the obtained clusters. In this phase, the poor clusters with similar features are deleted. The appropriate clusters are further selected based on cluster pruning. The optimal cluster sets are then made by computing the convergence and coverage rates in the selected clusters.

Phase III is the collaborative filtering recommendation process for the target user. Here, similarity and prediction ratings are calculated to choose candidate clusters. An item prediction is made by performing a weighted average of deviations from the neighbour's mean.

The proposed hybrid recommendation approach is described below, using the MovieLens dataset provided by GroupLens Research at the University of Minnesota.

6.3.1 Data Preprocessing and Multidimensional Clustering

Traditional clustering techniques like the *K*-means clustering algorithm aim to detect clustered objects using all attributes in the full data space. Full space clustering approaches do not scale to high dimensional data spaces that cover multiple attributes. They cannot detect multiple concepts in a single data space or more data spaces. They also cannot detect concepts spreading across different data spaces or compute new subsets of relevant dimensions. For each descriptive data attribute considered one dimensional, suppose there is a hyperplane with multiple dimensions. These dimensions represent different attributes that are selected from an attribute dataset. Consider that all data can be plotted in the hyperplane, we then formalize each unit according to the data distribution.

In most cases, data spaces contain objects specified by many attributes. To discover the different features of data in different data spaces, each object can participate in various clusters, reflected in different of attribute subsets in one high-dimensional data space. Objects can be described by multiple attributes, thus specifying their profile. In the movie recommendation applications, data are collected for analysis tasks using multiple attributes. As Figure 6.2 shows, a user might be grouped using the attributes "issuing year of movie" and "issuing area of movie" with other users who are interested in the western modern culture specified by these attributes. The same user might be a "comedy fan", which could be specified by high values in the attribute "watch comic film" and low values in "watch other genres film". We observe multiple possible interests for each user, which should be detected as clusters. Clusters may thus overlap in their clustered objects, i.e., each object may be represented in multiple clusters. Furthermore, each user behavior is described by specific attributes. Meaningful clusters appear only in these specific data subspace projections. While the attribute "watch fantasy film" is useful for to distinguish favorite, the attribute "issuing year of movie" is irrelevant for grouping style tastes of users.

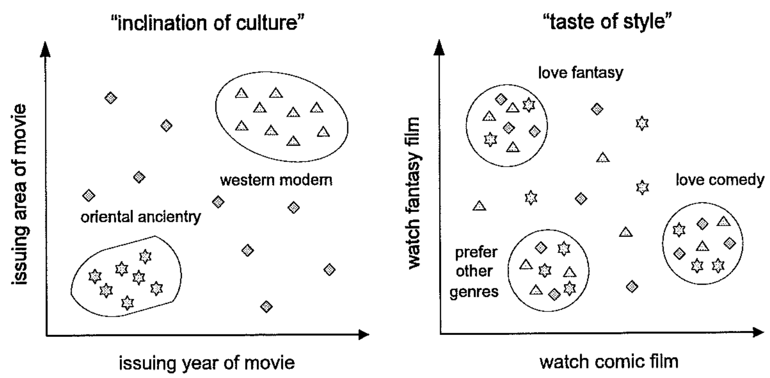


Figure 6.2. Example of two preference spaces.

We use the following multidimensional clustering algorithm to generate the initial clusters with user preference. Let $U = \{U_1, U_2, \ldots\ldots, U_m\}$ be a set of users and $I = \{I_1, I_2, \ldots\ldots, I_n\}$ be a set of items. The input attributes set $I_A = \{A_1, A_2, \ldots\ldots, A_d\}$ and $S = A_1 \times A_2 \times \ldots\ldots \times A_d$ are in the d-dimensional numerical space. We refer to A_1, A_2, $\ldots\ldots$, A_d as the dimensions attributes of S. We first need input two parameters, which decide the number of clusters. One is unit bin size δ, and the other is the minimum number of users τ in one unit bin. The detail of multidimensional clustering as shown in Figure 6.3 and is described below.

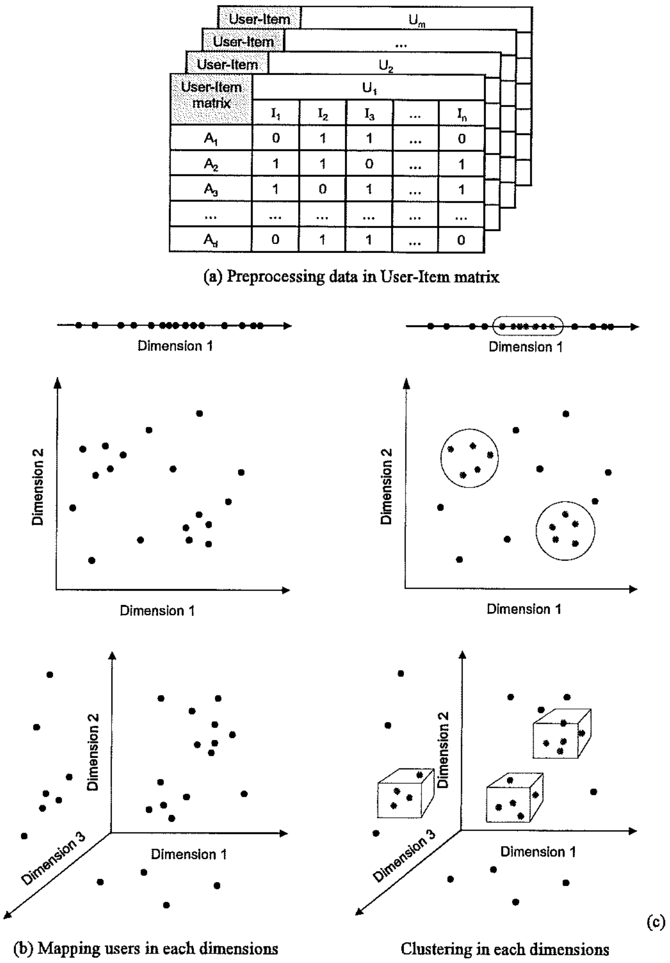


Figure 6.3. Procedure of multidimensional clustering.

Step 1: Select attributes $A_i \in I_A$ from the 1-dimensional subspace to d-dimensional subspace to generate the attribute space S.

Step 2: Formalize the width δ of the unit bin for each attribute $A_i \in I_A$. The units are obtained by partitioning every dimension into δ intervals of equal length, which is an input parameter.

Step 3: Detect all clusters $C = \{c_1, c_2, \ldots\ldots, c_n\}$ in each attribute subspace s, $s_i \subset S$. Given the threshold τ, if the number of users in unit is greater than τ, we consider the unit as a candidate.

Step 4: Join the connected candidate a cluster $c_i \subset C$.

Step 5: Collect full clusters in the current subspace $S_{current}$.

Step 6: Repeat Step 2 to Step 5, and output the set M that contains clusters.

We thus achieve clustering for all users according to the attribute item space. The obtained user cluster set M must not judge the effectiveness of M, which can cause similar clusters to exist. Therefore, we must prune this cluster set.

6.3.2 Choosing the Appropriate Clusters

In this phase, we present an algorithm to prune the cluster set M. To prove the validity of multidimensional clustering, we define two factors as pruning criteria for multidimensional clustering. One factor is user convergence, which describes the congregate degree of users in same dimension; the other factor is the coverage between clusters c_i and c_j, which describes the coverage degree of clusters for different dimensions. The details are described below.

For cluster $c_i = \{U_1, U_2, \ldots\ldots, U_m\}$ in a subspace, $s_i = \{c_1, c_2, \ldots\ldots, c_n\}$, s_i is a set of clusters in a subspace. The convergence of users in a subspace is determined as follows:

$$convergence(U) = \frac{number\ of\ users\ plotted\ in\ cluster}{number\ of\ all\ users\ in\ hyperplane} \cdots(6.1)$$

As 2-dimensions in Figure 6.4 shows, the convergence of users in the 2-dimensions is 27/33. For a given threshold α, if α < 27/33, this suggests that clustering in this dimension is successful because most users of this subspace are classified into the clusters. Thus, we need not prune the clusters in this subspace.

We define the clusters in lower dimensional subspace are removed from M only if $coverage(c_i, s_j)$ is larger than a given threshold β. Thus, the pruning in set M is valid. The coverage of c_i with respect to s_j is determined as follows:

$$coverage(c_i, s_j) = \frac{Num_{(c_i \cap s_j)}(U)}{Num_{(c_i)}(U)} \quad \cdots\cdots\cdots\cdots (6.2)$$

When coverage degree →1, the cluster in lower dimensional cannot be replaced by other clusters in higher dimensional. Consequently the cluster with high overlap in lower dimensional must be delete. On the contrary, when coverage degree →0, this cluster should be selected because the cluster in lower dimensional cannot be replaced by other clusters.

Let us assume that M contains clusters c_1 to c_5 and possible further clusters in other subspaces. As Figure 6.4 shows, if we choose c_4 the user group corresponds to $s_2 = \{c_1, c_2, c_3\}$. However, the clusters in s_2 already cover most users (i.e., 23 of 25) from c_4, and c_4 obtains little information about the user ($coverage(c_4, s_2)$ = 23/27). For a threshold β < 23/27, cluster c_4 is considered redundant with respect to M. In other word, if most users from cluster c_4 are overlap with the users from clusters set $\{c_1, c_2, c_3\}$ in the higher dimensional subspace, cluster c_4 can be deleted. The operation of c_5 is the same as c_4.

Finally, we can obtain the new cluster set M' after pruning. M' is an optimal set, which includes clusters with diverse characters.

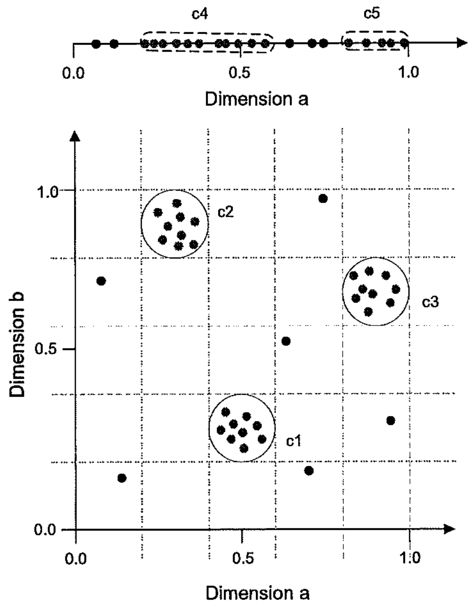


Figure 6.4. Example for pruning clusters based on coverage.

6.3.3 Recommending for Target User

In this phase, we utilize a collaborative filtering algorithm to provide recommendations for a target user. We first analyze the users' preferences and infer the attribute subspace in which he may be interested. Then, we can find the user clusters in *M'* according to this attribute subspace. The details are provided below.

Step 1: For a given user u with an initial history record, construct a set $T = \{i_1, i_2, \ldots\ldots, i_n\}$, $T \subset M'$, which includes possible clusters, and get all items from the clusters set T. We regard the items that contain T as candidate items.

Step 2: For item i from history record of u, search in the set of items I in the T to which item i belongs. Calculate the similarity between the items from target user u and all items from T.

The similarity measure of the target user profile is calculated using each cluster to find the clusters that contain users with similar preferences. There are several possible measures for computing the similarity, including the Euclidean distance metric, cosine similarity and Pearson correlations metric. Cosine similarity has once been used to calculate user similarity, but it has one shortcoming: the difference in rating scales between different users result in different similarities. For instance, user A may only rate movies with scores of 4 for his liked movies and never rate any movies a 5, or he may rate his disliked movies using scores of 1, instead of the standard level score 2. Conversely, user B may always rate according to the standard levels. He gives scores of 5 to his liked movies and 2 to his disliked movies. If we use the traditional cosine similarity, both ratings are quite different. The adjusted cosine similarity is provided to offset this drawback.

Step 3: We use only the adjusted cosine similarity because it has gotten good accuracy in measuring the item similarity in the *m-dimensional* user space. The similarity *sim (i, j)* between item i and item j in T is measured as follows:

$$sim(i,j)=\frac{\sum_{u\in U}\left(R_{u,i}-\bar{R}_u\right)\left(R_{u,j}-\bar{R}_u\right)}{\sqrt{\sum_{u\in U}\left(R_{u,i}-\bar{R}_u\right)^2}\sqrt{\sum_{u\in U}\left(R_{u,j}-\bar{R}_u\right)^2}} \quad \text{..........(6.3)}$$

where *sim (i, j)* is the value of similarity function to measure the similarity between item i and j; $R_{u,i}$ is the rating from user u for item i, and similar to $R_{u,j}$; Ru is the average ratings of the user u.

Step 4: According to the similarity calculated in Step 3 and input parameter k, where k is the number of items from the clusters set T, construct the similarity set *sim* = $\{sim(i,j_1), sim(i,j_2),\ldots\ldots, sim(i,j_k)\}$. List the elements in this similarity set *sim* in descending order, and define a threshold value γ. If the similarity between item i from target user u and item j in T is above this threshold γ, then we select the items from the T as the optional candidates in set T'.

A predicted rating for an item is then computed by calculating a weighted average of deviations from the neighbour's mean. We use the Top-N rule to select the highest N items in the optional candidate set T' based on item similarities.

Step 5: Based on the candidate set *T'* obtained in Step 4 and the input user-item rating matrix *R*, predict the ratings by user *u* for item *i*. The prediction formula for item *i* from target user *u* as follows:

$$P_{u,i} = \bar{R}_i + \frac{\sum_{l \in T'} \left(R_{u,l} - \bar{R}_l\right) \times sim(i,l)}{\sum_{l \in T'} \left|sim(i,l)\right|} \quad \text{........................(6.4)}$$

where $P_{u,i}$ represents the rating predication for the target user *u* on candidate item *i*; $R_{u,l}$ indicates the rating from *u* for item *l* in *T'*; R_i is the average ratings for item *i;* *sim(i, l)* denotes the similarity between item *i* and its neighbouring item *l* in *T'*; R_l denotes the average ratings for item *l*.

Step 6: Repeat Step 4 and Step 5, predict the ratings for all items in the optional candidate set *T'* for which target user *u* has no ranking and list these rankings in descending order these ratings. We can then select the Top-*N* candidate and recommend the *N* items with the top ratings to the target user.

6.4 Experimental Study

To demonstrate the advantages of our approach, we evaluate the quality and efficiency of the proposed recommendation approach and compare it with two variants of collaborative filtering techniques. We implement the Item-based collaborative filtering [54] and *K*-means cluster based collaborative filtering algorithms [55] to compare on Movielens datasets [56].

Our experimental study includes two works. One is a performance evaluation of multidimensional clustering [53]. We use the users' coverage degrees for different numbers of clusters to verify the validity and quality of the proposed multidimensional clustering algorithm. The other is a recommendation performance evaluation, including recommendation accuracy and diversity. We use the deviation of predicted recommendation ratings and the precision in the Top-10 of recommendations to measure the quality of our recommendation results.

6.4.1 Implementation

In our experiment, we select a public dataset MovieLens as a benchmark. MovieLens is a famous recommender system and virtual community website that recommends films for its users to watch, based on their film preferences. This dataset provides 100,000 ratings for 1,682 movies from 943 users, as shown in Table 6.1. The movies' user ratings were recorded on a discrete scale from 1 to 5, and each user rated over 100 movies [56]. We select 17 genres of movies as dimensions for multidimensional clustering. These 17 dimensions include Action, Adventure, Animation, Children's, Comedy, Crime, Documentary, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War and Western.

Table 6.1. Summary of dataset.

Dataset	Description	Profile	Ratings per user	Dimension
Movielens	Ratings of movies in scale of 1-5	100,000 ratings for 1682 movies by 943 users	On more than 100 movies	Action, Adventure, Animation, Children's, Comedy, Crime, Documentary, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War and Western

6.4.2 Empirical Evaluation

We first use the global coverage of users to evaluate our multidimensional clustering algorithm. For the generated cluster set $C = (c_1, c_2, \ldots\ldots, c_n)$, $c_i = (U_1, U_2, \ldots\ldots, U_m)$, the users' global coverage is determined below:

$$Global\ coverage(C) = \frac{Num(\bigcup_{i=1}^{n} c_i)}{number\ of\ all\ users} \quad \ldots\ldots\ldots\ldots(6.5)$$

To evaluate recommendation quality, we use Precision [25, 26]. Precision measures exactness and it is the most popular metrics for evaluating information retrieval systems. We use it to measure recommendation accuracy. Precision is defined as the ratio of the selected relevant items to the total number of items selected, as in (6.6). It represents the probability that a selected item is relevant.

$$precision = \frac{number\ of\ correctly\ recommended\ items}{number\ of\ recommended\ items} \quad (6.6)$$

And we use the total number of distinct items R_N recommended across users as a diversity measure, which we call the diversity-in-top-N. The diversity-in-top-N metric indicates the personalization level provided by a recommender system [45].

$$diversity\text{-}in\text{-}top\text{-}N = \frac{R_N(u)}{N} \quad \cdots\cdots\cdots\cdots\cdots\cdots (6.7)$$

where $R_N(u)$ represents the number of recommendations that have distinct features in recommended N items for user u.

6.4.3 Result Analysis and Discussion

We use the above evaluation metric to evaluate the performance of our proposed algorithm and other two recommendation algorithms. We first measure the users' global coverage for different cluster scales. Table 6.2 shows that the global user coverage of our approach is better than the Item-based CF approach and our approach can obtain more candidate recommendations as the number of clusters increases. This indicates that our multidimensional clustering approach can help as many users as possible to find their own interest group.

Secondly, we empirically analyze how precision evolves with different numbers of clusters. Figure 6.5 shows the obtained testing results. Likewise, we have selected representatively 5, 10, 15, 20, 25, 30, 35, 40, 45 and 50 clusters to represent different density degrees for the rating matrix. The number of clusters also does affect the recommendation quality. As Figure 6.5 shows, the precision curve of our algorithm is slightly higher than that of the other two algorithms. This means that maintaining recommendation accuracy while increasing recommendation diversity has less impact in our proposed algorithm.

Table 6.2. Comparison of global coverage and candidate recommendations for different cluster scales.

Number of clusters		15	25	35	45	55
Global coverage of users	Item-based CF	0.73	0.69	0.78	0.76	0.75
	Our approach	0.82	0.87	0.86	0.85	0.88
Number of candidate recommendations	Item-based CF	49	81	107	113	126
	K-means CF	55	72	99	127	133
	Our approach	68	95	136	152	171

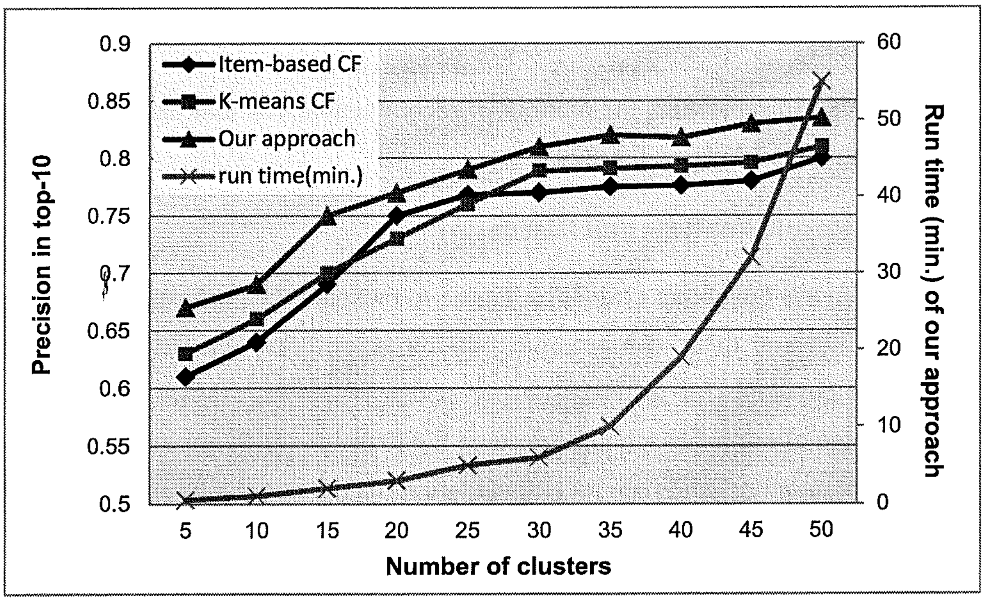


Figure 6.5. Precision comparison with Item-based CF and *K*-means based CF.

From Figure 6.5, we can see the precision rate is almost no increased when number of clusters is from 30 to 50. In other word, the precision rate did not improve markedly with the increase of cluster numbers. But we find running time increase rapidly with the cluster scale is from 30 to 50. Consider the trade-off between running time and cluster scale, then we evaluate the diversity-in-top-*N* of our algorithm under cluster scale is 35.

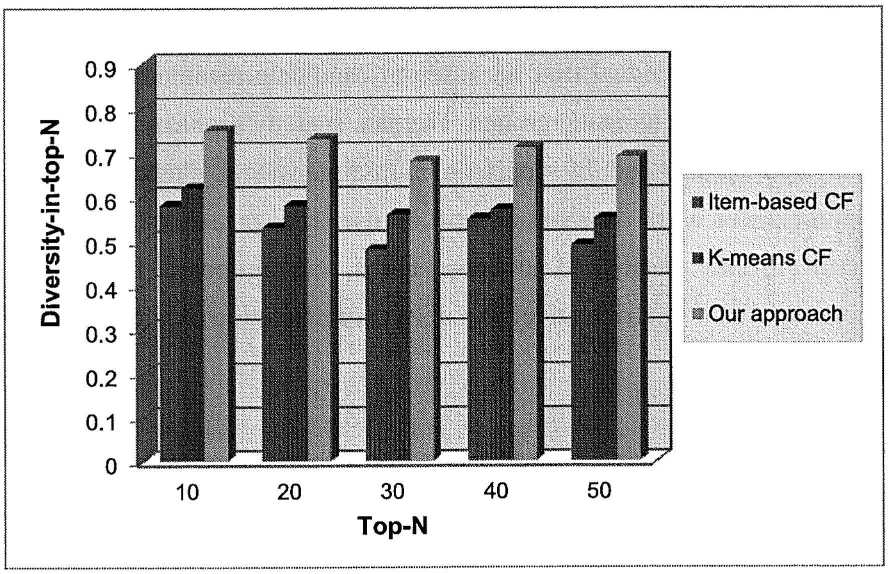


Figure 6.6. Diversity comparison with Item-based CF and *K*-means based CF.

Thirdly, we measure the recommender systems performance based on the top-*N* recommended item lists that the system provides to users. Low diversity-in-top-*N* indicates that users are being recommended the similar top-*N* items, whereas a high diversity-in-top-*N* indicates that users receive unique top-*N* items. As Figure 6.6 shows, the diversity of our algorithm is significantly better than the other two algorithms for different top-*N* scales.

We discuss the merit of our approach based on two aspects. We use multidimensional clustering to group users by the features of their history records. The traditional recommendation approach strongly depends on users' ratings to classify the user group. A collaborative filtering recommendation usually requires a user to rate a set of query items. After gaining ratings for these query items, recommender systems can start working. We have two reasons to disapprove this. On the one hand, users often do not like to rate a long list of items. And on the other hand, a user's rating of some items might not provide useful information for the recommender system. This leads to generating poor user group and a lack of diverse recommendations.

Conversely, our proposed hybrid approach avoids providing too many similar items for an individual user because our candidate recommendation set comes from multiple users' interesting groups. The data sparsity problem in collaborative filtering can also be avoided by applying multidimensional clustering, as our clustering approach for users does not depend on user ratings. Although the recommendation result is still unsatisfied in some terms, our proposed hybrid recommendation approach can improve diversity of recommendation while maintaining adequate accuracy.

6.5 Concluding Remarks

The above experimental study shows that it is possible to obtain higher recommendation quality by recommending diverse items. In this work, we explore a hybrid recommendation approach that can increase recommendation diversity using multidimensional clustering and collaborative filtering algorithms. Our proposed approaches consider multidimensional factors when ranking the recommendation list to substantially increase recommendation diversity while maintaining comparable accuracy levels. In traditional recommendation, similarity is normally the only heuristic used in the recommendation process, whereas similarity is combined with multidimensional data clusters in our proposed hybrid recommendation approach. This helps explore other clusters that have similarities closer to the target user and can provide him with good recommendation sets. This study also provides a comprehensive empirical evaluation of the proposed approach, where we test it with MovieLens datasets in several different settings. The evaluation results show that our approach performs better than traditional two-dimensional approaches in terms of improving recommendation quality.

Chapter 7

Conclusions and Future Study

7.1 Conclusions

In this thesis, we have firstly presented a non-technical overview of the evolution of critiquing research in recommender systems over the past decade. We presented three original ideas can be used in recommender systems application in this area. Key issues and contributions have been discussed in detail as follows.

Chapter 4 presents a new tracking recommendation approach to provide the relevant items for new user who has a small number of records. A new recommendation model based on the synergistic usage of information from repository, which includes user's limited history records and items' two-level property, was constructed. The hierarchical structure map based on the classificatory attribute of items generating by Formal Concept Analysis (FCA) technique is used to discover the candidate items that the user may be interested. The appropriate recommendation returned to a new user utilizes the filtering techniques based on the corresponding descriptive attribute ranking of these candidates. The experimental study includes simulating a prototype recommender system for implementing the proposed approach and testing its performance. Experiments using two datasets demonstrated under new user cold-start conditions, our approach has improved the precision and the recall rates by up to 15.3% and 12.5% respectively while compared to traditional collaborative filtering approach.

Chapter 7

Chapter 5 focuses on exploring an new approach for catching users' transitional preferences and guide timely recommendations based on the transformation of user preferences. Firstly, the conceptual network of each items based on their descriptive keywords are created using WordNet and color theory. Secondly, color orders of each item are generated via the spreading activation. Thirdly, a characteristic sequence consisting of color nodes was extracted from color orders of users' record for inferring the contextual information; this information is subsequently used to perceive the changing of user preference. The performance of the proposed approach was illustrated using the example of a movie recommendation application. In terms of accuracy, the experiment analysis results showed that the proposed approach outperformed the traditional collaborative filtering approach by approximately 15%. This work also presents a novel insight into the exploitation of a rich repository of domain-specific knowledge to better understand user preferences.

Chapter 6 describes a new hybrid recommendation approach that incorporates multidimensional clustering technique into a collaborative filtering recommendation process for detecting the individual users' crossover preferences from diversification view. The presented approach aims to help users reach a decision to meet their potential demands and provide the target user with highly idiosyncratic or more diversified recommendations. The proposed hybrid recommendation procedure has been partitioned into three phases: i) data preprocessing and multidimensional clustering; ii) choosing the appropriate clusters; iii) collaborative filtering recommendation process for the target user. Convergence and Coverage are defined as two pruning criterion respectively to prune the initial clusters set. The performance of the proposed approach is evaluated using a public movie dataset and it was compared with two representative recommendation algorithms. The comprehensive empirical evaluation demonstrated that the proposed approach has increased 15%-20% in diversity when compared to the traditional recommendation approaches, and further the recommendation accuracy is maintained at 75%.

7.2 Directions for Future Research

Recommender systems represent a vibrant and constantly changing research area. Among the important recent developments, recommender systems have recently started adopting multi-criteria ratings provided by users, and in this thesis we investigated algorithms and techniques for recommender systems. In future study, we will attempt combine the three proposed approaches into a new system, or integrate the proposed approaches and innovative techniques into a new system. These new systems have not yet been studied extensively, and in this chapter we present a number of challenges and the future research directions for this category of recommender systems.

Reusing existing single-rating recommendation techniques. A huge number of recommendation techniques have been developed for single-rating recommender systems over the last 10-15 years, and some of them could potentially be extended to multi-criteria rating systems. For example, neighborhood-based collaborative filtering techniques may possibly take into account multi-criteria ratings using the huge number of design options. As another example, there has been a number of sophisticated hybrid recommendation approaches developed in recent years, and some of them could potentially be adopted for multi-criteria rating recommenders. Finally, more sophisticated techniques, e.g., based on data envelopment analysis (DEA) or multi-criteria optimization, could be adopted and extended for choosing best items in the multi-criteria rating settings.

Predicting relative preferences. An alternative way to define the multi-criteria recommendation problem could be formulated as predicting the relative preferences of users, as opposed to the *absolute* rating values. There has been some work on constructing the correct relative order of items using ordering-based techniques. For example, the RankBoost algorithm based on the wellknown AdaBoost method in multi-criteria settings, such algorithms could be adopted to aggregate different relative orders obtained from different rating criteria for a particular user.

Investigating group recommendation techniques for multi-criteria settings. Some techniques for generating recommendations to groups can be adopted in built by aggregating the diverse preferences of several users. Similarly, a user's preference for an item in multi-criteria rating settings can be predicted by aggregating the preferences based on different rating criteria. More specifically, there can be many different goals for aggregating individual preferences, such as maximizing average user satisfaction, minimizing misery (i.e., high user dissatisfaction), and providing a certain level of fairness (e.g., low variance with the same average user satisfaction). Multi-criteria rating recommenders could investigate the adoption of some of these approaches for aggregating preferences from multiple criteria.

Collecting large-scale multi-criteria rating data. Multi-criteria rating datasets that can be used for algorithm testing and parameterization are rare. For this new area of recommender systems to be successful, it is crucial to have a number of standardized real-world multi-criteria rating datasets available to the research community. Some initial steps towards a more standardized representation, reusability, and interoperability of multi-criteria rating datasets have been taken in other application domains, such as e-learning.

Constructing the item evaluation criteria. More research needs to be done on choosing or constructing the best set of criteria for evaluating an item. For example, most of current multi-criteria rating recommenders require users to rate an item on multiple criteria at a single level (e.g., story and special effects of a movie). This single level of criteria could be further broken down into sub-criteria, and there could be multiple levels depending on the given problem. For example, in a movie recommender system, special effects could be again divided into sound and graphic effects. More information with multiple levels of criteria could potentially help to better understand user preferences, and various techniques. As we consider more criteria for each item, we may also need to carefully examine the correlation among criteria because the choice of criteria may significantly affect the recommendation quality. Furthermore, as mentioned earlier, it is important to have a consistent family of criteria for a given recommender system application because then the criteria are monotonic, exhaustive, and non-redundant. In summary, constructing a set of criteria for a given recommendation problem is an interesting and important topic for future research.

In this chapter, we discussed several potential future research directions for multi-criteria recommenders that should be interesting to recommender systems community. This work is not meant to be exhaustive; we believe that research in this area is only in its preliminary stages, and there are a number of possible additional topics that could be explored to future recommender systems.

Bibliography

[1] J.B. Schafer, J.A. Konstan and J. Riedl, "E-Commerce Recommendation Applications", *Data Mining and Knowledge Discovery*, vol.5, pp.115–153, 2001.

[2] D. Jannach, M. Zanker, A. Felfernig and G. Friedrich, *Recommender Systems: An Introduction*, Cambridge University Press, 2011.

[3] M. David, "Explanation in Recommender Systems", *Artificial Intelligence Review*, vol.24, pp. 179-197, 2005.

[4] G. Adomavicius and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions", *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no.6, pp.734-749, 2005.

[5] P. Melville, R.J. Mooney and R. Nagaragan, "Content Boosted Collaborative Filtering for Improved Recommendations", *In Proceedings of the National Conference of the American Association Artificial Intelligence*, pp.187–192, 2002.

[6] F. Ricci, L. Rokach, B. Shapira and P.B. Kantor, *Recommender Systems Handbook*, Springer, 2011.

[7] J.L. Herlocker, J.A. Konstan and J. Riedl, "Explaining Collaborative Filtering Recommendations", *In Proceedings of ACM Conference on Computer Supported Cooperative Work*, pp.241–250, 2000.

[8] R. Burke, "Hybrid Recommender Systems: Survey and Experiments", *User Modeling and User-Adapted Interaction*, vol.12, pp.331-370, 2002.

[9] G. Adomavicius and A. Tuzhilin, "Personalization Techniques: A Process-oriented Perspective", *Communications of the ACM 48*, vol.10, pp.83-90, 2005.

[10] D. Riecken, "Personalized Views of Personalization", *Communications of the*

ACM, vol.43, no.8, 2000.

[11] J.M. Yang, K.F. Li and D.F. Zhang, "Recommendation Based on Rational Inferences in Collaborative Filtering", *Knowledge-Based Systems*, vol.22, no.1, pp.105-114, 2009.

[12] J.S. Lee and O. Sigurdur, "Two-way Cooperative Prediction for Collaborative Filtering recommendations", *Expert Systems with Applications*, pp.5353-5361, 2009.

[13] Z. Huang, H. Chen, and D. Zeng, "Applying Associative Retrieval Techniques to Alleviate the Sparsity Problem in Collaborative Filtering", *ACM Transactions on Information Systems*, 2004.

[14] A.I. Schein, A. Popescul and L.H Ungar, "Methods and Metrics for Cold-start Recommendations", *In Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.253–260, 2002.

[15] Y.J. Park and K.N. Chang, "Individual and Group Behavior-based Customer Profile Model for Personalized Product Recommendation", *Expert Systems with Applications*, pp.1932-1939, 2009.

[16] B. Ganter and R. Wille, *Formal Concept Analysis: Mathematical Foundations*, Springer, Berlin, Germany, 1999.

[17] B. Ganter, G. Stumme and R. Wille, *Formal Concept Analysis: Foundations and Applications*, Springer, Berlin, Germany, 2005.

[18] C. Carpineto, G. Romano, *Concept Data Analysis: Theory and Applications*, Wiley, 2004.

[19] MovieLens, http://www.grouplens.org/, (Accessed on May 2010).

[20] EachMovie, http://www.research.digital.com/SRC/EachMovie/, (Accessed on June 2010).

[21] J. Davis and M. Goadrich, "The Relationship between Precision-Recall and ROC Curves", *In Proceedings of the 23th International Conference on Machine Learning*, pp.233-240, 2006.

[22] F. Hernandez del Olmo and E. Gaudioso, "Evaluation of Recommender Systems: A New Approach", *Expert Systems with Applications*, vol.35, no.3,

pp.790-804, 2008.

[23] P. Resnick, N. Iacovou, M. Sushak, P. Bergstrom and J. Riedl, "GroupLens: An Open Architecture for Collaborative Filtering of Netnews", *In Proceedings of ACM Conference on Computer Supported Cooperative Work*, pp.175-186, 1994.

[24] C.W. Leung, S.C. Chan and F. Chung, "Applying Cross-Level Association Rule Mining to Cold-Start Recommendations", *In proceedings of IEEE/WIC/ACM International Conference on Web Intelligent and Intelligent Agent Technology*, pp.133-136, 2007.

[25] J.L. Herlocker, J.A. Konstan, L. Terveen and J. Riedl, "Evaluating Collaborative Filtering Recommender Systems", *ACM Transactions on Information Systems*, vol.22, no.1, pp.5–53, 2004.

[26] A. Gunawardana and G. Shani, "A Survey of Accuracy Evaluation Metrics of Recommendation Tasks", *Journal of Machine Learning Research*, vol.10, pp.2935-2962, 2009.

[27] G. Linden, B. Smith, and J. York, "Amazon.com Recommendations: Item-to-Item Collaborative Filtering", *IEEE Internet Computing*, 2003.

[28] B.N. Miller, I. Albert, S.K. Lam, J.A. Konstan, and J. Riedl, "MovieLens Unplugged: Experiences with an Occasionally Connected Recommender System", *In Proceedings of ACM Conference of Intelligent User Interfaces*, pp.263-266, 2003.

[29] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin, "Incorporating Contextual Information in Recommender Systems Using a Multidimensional Approach", *ACM Transactions on Information Systems*, vol.23, pp.103-145, 2005.

[30] R.W. Picard, E. Vyzas, and J. Healey, "Toward Machine Emotional Intelligence: Analysis of Affective Physiological State", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.23, no.10, pp.1175-1191, 2001.

[31] M. Argyle, *Bodily Communication*, London, U.K. Methuen, 1975.

[32] A. Ortony, G.L. Clore, and A. Collins, *The Cognitive Structure of Emotions*, Cambridge University Press, U.K. 1988.

[33] N. Kaya, and H.H. Epps, "Relationship between Color and Emotion: A Study of College Students", *College Student Journal*, pp.396–405, 2004.

[34] A.M. Collins, and E.F. Loftus, "A Spreading-activation Theory of Semantic Processing", *Psychological Review*, vol.82, no.6, pp.407-428, 1975.

[35] F. Crestani, "Application of Spreading Activation Techniques in Information Retrieval", *Artificial Intelligence Review*, vol.11, no.6, pp.453-482, 1997.

[36] C. Rocha, D. Schawabe, and M. Poggi, "A Hybrid Approach for Searching in the Semantic Web", *In Proceedings of 13th International Conference on World Wide Web*, pp.374–383, 2004.

[37] G. Semeraro, P. Lops, and M. Degemmis, "WordNet-based User Profiles for Neighborhood Formation in Hybrid Recommender Systems", *In Proceedings of the 15th International Conference on Hybrid Intelligent Systems*, pp.291-296, 2005.

[38] M.G. Hwang, C. Choi, and P.K. Kim, "Automatic Enrichment of Semantic Relation Network and its Application to Word Sense Disambiguation", *IEEE Transactions on Knowledge and Data Engineering*, vol.23, no.6, pp.845-857, 2011.

[39] T.P. Liang, Y.F. Yang, D.N. Chen, and Y.C. Ku, "A Semantic-expansion Approach to Personalized Knowledge Recommendation", *Decision Support Systems*, vol.45, pp.401-412, 2008.

[40] IMDB, http:// www. imdb.com/, (Accessed on June 2010).

[41] P. Resnick and H.R. Varian, "Recommender systems", *Communications of the ACM*, vol.40, no.3, pp.56-58, 1997.

[42] K. Bradley and B. Smyth, "Improving recommendation diversity", *In Proceedings of the 12th International Conference on Artificial Intelligence and Cognitive Science*, 2001.

[43] M. Zhang and N. Hurley, "Avoiding monotony: improving the diversity of recommendation lists", *In Proceedings of ACM Conference on Recommender systems*, pp.123-130, 2008.

[44] D. Fleder and K. Hosanagar, "Blockbuster culture's next rise or fall: The impact of recommender systems on sales diversity", *Management Science*, vol.55, no.5, pp.697-712, 2009.

[45] G. Adomavicius and Y. Kwon, "Improving recommendation diversity using ranking-based techniques", *IEEE Transactions on Knowledge and Data*

Engineering, vol.24, no.5, pp.896-911, 2012.

[46] M. Balbanovic and Y. Shoham, "Fab: content-based, collaborative recommendation", *Communications of the ACM*, vol.40, no.3, pp.66-72, 1997.

[47] X. Su and T. Khoshgoftaar, "A survey of collaborative filtering techniques", *Advances in Artificial Intelligence*, 2009.

[48] A. K. Jain, M. N. Murty and P. J. Flynn, "Data clustering: A review", *ACM Computing Surveys*, vol.31, no.3, pp.264-323, 1999.

[49] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, "Automatic subspace clustering of high dimensional data for data mining applications", *In Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp.94-105, 1998.

[50] L. Parsons, E. Haque and H. Liu, "Subspace clustering for high dimensional data: A review", *SIGKDD Explorations Newsletter*, vol.6, no.1, pp.9-105, 2004.

[51] C.C. Aggarwal and P.S. Yu, "Redefining clustering for high-dimensional applications", *IEEE Transactions on Knowledge and Data Engineering*, vol.14, no.2, pp.210-225, 2002.

[52] H.P. Kriegel, P. Kroger and A. Zimek, "Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering", *ACM Transactions on Knowledge Discovery from Data*, vol.3, no.1, 2009.

[53] E. Muller, S. Gunnemann, I. Assent and T. Seidl, "Evaluating clustering in subspace projections of high dimensional data", *In VLDB*, pp.1270-1281, 2009.

[54] B. M. Sarwar, G. Karypis, J. A. Konstan and J. Riedl, "Item-based collaborative filtering recommendation algorithms", *In Proceedings of the 10th ACM WWW Conference*, pp.285-295, 2001.

[55] M. Deshpandel and G. Karypis, "Item-based Top-N Recommendation Algorithms", *ACM Transactions on Information Systems*, vol.22, no.1, pp.143-177, 2004.

[56] MovieLens, http://www.grouplens.org/, (Accessed on June 2011)

[57] S.S. Anand, B. Mobasher, "Intelligent techniques for web personalization", Intelligent

Techniques for Web Personalization, Springer, pp.1–36, 2005.

[58] D. Billsus, C.A. Brunk, C. Evans, B. Gladish, and M. Pazzani, "Adaptive Interfaces for Ubiquitous Web Access", *Communications of the ACM*, vol.45, no.5, pp.34-38, 2002.

[59] W. Hill, L. Stead, M. Rosenstein, and G. Furnas, "Recommending and Evaluating Choices in a Virtual Community of Use", *In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1995.

[60] U. Shardanand and P. Maes, "Social Information Filtering: Algorithms for Automating 'Word of Mouth' ", *In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1995.

[61] B. Smyth and P. McClave, "Similarity vs. Diversity", *In Proceedings of the Fourth Intrenational Conference on Case-Based Reasoning: Case-Based Reasoning Research and Development*, 2001.

[62] D. Fleder and K. Hosanagar, "Blockbuster Culture's Next Rise or Fall: The Impact of Recommender Systems on Sales Diversity", *Management Science*, vol.55, no.5, pp.697-712, 2009.

[63] K. Goldberg, T. Roeder, D. Guptra, and C. Perkins, 2001. "Eigentaste: A constant-time collaborative filtering algorithm", *Information Retrieval*, vol.4, pp.133–151, 2001.

[64] SW.S. Lee, "Collaborative Learning for Recommender Systems", *In Proceedings of Intrenational Conference on Machine Learning*, 2001.

[65] K. Yu, A. Schwaighofer, V. Tresp, X. Xu, and H.-P. Kriegel, "Probabilistic Memory-Based Collaborative Filtering", *IEEE Transactions on Knowledge and Data Engineering*, vol.16, no.1, pp.56-69, 2004.

[66] M. Pazzani, "A Framework for Collaborative, Content-Based, and Demographic Filtering", *Artificial Intelligence Review*, pp.393-408, 1999.

[67] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Application of Dimensionality Reduction in Recommender Systems—A Case Study", *In Proceedings of ACM WebKDD Workshop*, 2000.

[68] D. Billsus and M. Pazzani, "User Modeling for Adaptive News Access", *User Modeling and User-Adapted Interaction*, vol.10, pp.147-180, 2000.

[69] X. Li and T. Murata, "A Knowledge-based Recommendation Model Utilizing Formal Concept Analysis and Association", *In Proceeding of the 2nd IEEE International Conference on Computer and Automation Engineering*, vol.4, pp.221-226, 2010.

[70] X. Li and T. Murata, "Customizing Knowledge-based Recommender Systems by Tracking Analysis of User Behavior", *In Proceeding of the 17th IEEE International Conference on Industrial Engineering and Engineering Management*, pp.65-69, 2010.

[71] X. Li and T. Murata, "Incorporating Affectivity into Preference Elicitation for Personalized Recommendation via Spreading Activation", *In Proceeding of the 3rd IEEE International Conference on Computer Research and Development*, vol.4, pp.268-273, 2011.

[72] X. Li and T. Murata, "Toward Affective Recommendation: A Contextual Association Approach for Eliciting User Preference", *In Proceeding of the IEEE International Conference on Information Society*, pp.47-54, 2011.

[73] X. Li and T. Murata, "Multidimensional Clustering Based Collaborative Filtering Approach for Diversified Recommendation", *In Proceeding of the 7th IEEE International Conference on Computer Science & Education*, pp.905-910, Melbourne, Australia, 2012.

[74] X. Li and T. Murata, "Using Multidimensional Clustering Based Collaborative Filtering Approach Improving Recommendation Diversity", *In Proceeding of the IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, pp.169-174, 2012.

[75] X. Li and T. Murata, "Exploiting Formal Concept Analysis in a Customizing Recommendation for New User and Gray Sheep Problems", *IEEJ Transactions on Electronics, Information and Systems*, Vol.132, No.5, pp.782-789, 2012.

[76] X. Li and T. Murata, "A Hybrid Method Using Multidimensional Clustering-based Collaborative Filtering to Improve Recommendation Diversity", *IEEJ Transactions on Electronics, Information and Systems*, Vol.133, No.4, pp.749-755, 2013.

List of Publications

Journal papers

- Xiaohui Li and Tomohiro Murata, "A Hybrid Method Using Multidimensional Clustering-based Collaborative Filtering to Improve Recommendation Diversity", *IEEJ Transactions on Electronics, Information and Systems*, Vol.133, No.4, pp.749-755, 2013.
- Xiaohui Li and Tomohiro Murata, "Exploiting Formal Concept Analysis in a Customizing Recommendation for New User and Gray Sheep Problems", *IEEJ Transactions on Electronics, Information and Systems*, Vol.132, No.5, pp.782-789, 2012.

International conference papers

- Xiaohui Li and Tomohiro Murata, "Using Multidimensional Clustering Based Collaborative Filtering Approach Improving Recommendation Diversity", *Proceeding of WIC 2012* (*The 2012 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*), pp.169-174, Macau, China, 2012.
- Xiaohui Li and Tomohiro Murata, "Multidimensional Clustering Based Collaborative Filtering Approach for Diversified Recommendation", *Proceeding of ICCSE 2012* (*The 7th IEEE International Conference on Computer Science & Education*), pp.905-910, Melbourne, Australia, 2012.
- Xiaohui Li and Tomohiro Murata, "Toward Affective Recommendation: A Contextual Association Approach for Eliciting User Preference", *Proceeding*

of i-society 2011 (*The 2011 IEEE International Conference on Information Society*), pp.47-54, London, United Kingdom, 2011.

- Xiaohui Li and Tomohiro Murata, "Incorporating Affectivity into Preference Elicitation for Personalized Recommendation via Spreading Activation", *Proceeding of ICCRD 2011* (*The 3rd IEEE International Conference on Computer Research and Development*), Vol.4, pp.268-273, Shanghai, China, 2011.
- Xiaohui Li and Tomohiro Murata, "Customizing Knowledge-based Recommender Systems by Tracking Analysis of User Behavior", *Proceeding of IE&EM 2010* (*The 17th IEEE International Conference on Industrial Engineering and Engineering Management*), pp.65-69, Amoy, China, 2010.
- Xiaohui Li and Tomohiro Murata, "A Knowledge-based Recommendation Model Utilizing Formal Concept Analysis and Association", *Proceeding of ICCAE 2010* (*The 2nd IEEE International Conference on Computer and Automation Engineering*), Vol.4, pp.221-226, Singapore, 2010.