A STUDY ON HANDLING SPARSENESS IN COLLABORATIVE FILTERING

Yegwende Vincent TIEMTORE

Eskişehir, 2017

A STUDY ON HANDLING SPARSENESS IN COLLABORATIVE FILTERING

Yegwende Vincent TIEMTORE

MS Dissertation
Graduate School of Sciences
Computer Engineering Program
Supervisor: Prof. Dr. Yaşar HOŞCAN

Eskişehir
Anadolu University
Graduate School of Sciences
May 2017

FINAL APPROVAL FOR THESIS

This thesis titled “**A Study on Handling Sparseness in Collaborative Filtering** ” has been prepared and submitted by **Yegwende Vincent TIEMTORE** in partial fulfillment of the requirements in “Anadolu University Directive on Graduate Education and Examination ” for the Degree of MSc in Computer Engineering Department has been examined and approved on 26/05/2017.

Committee Members		**Signature**
Member (Supervisor)	: Prof. Dr. Yaşar HOŞCAN	
Member	: Asst. Prof. Dr. Ahmet ARSLAN	
Member	: Asst. Prof. Dr. Mehmet KOÇ	

Director

Graduate School of Science

ABSTRACT

A STUDY ON HANDLING SPARSENESS IN COLLABORATIVE FILTERING

Yegwende Vincent TIEMTORE

Department of Computer Engineering

Anadolu University, Graduate School of Sciences, May 2017

Supervisor: Prof. Dr. Yaşar HOŞCAN

With the advent of the Internet, the number of choices that are opened to us online is constantly increasing. Movies, books, recipes, world news..., as many sets where we need to select without the possibility of considering all the necessary information. So how to choose? As we are not only faced with the same choice, if anyone has similar tastes to ours and he liked such a recent film, the chances that we also liked the film seem bigger. It is therefore possible to take advantage of available information on choice of other agents to induce preferences over our own choices. Now with the availability of Internet and major databases on user preferences, it becomes possible extending to large-scale, the concept of word of mouth. The formalization and operation of this intuition are the subject of collaborative filtering.

Collaborative Filtering (CF) has become one of the most used filtering technique used to cope with the" information overload" problem. However, CF suffers from important bottlenecks: privacy, cold-start, sparsity... Many researchers have proposed methods for handling latter problem but it remains a great and important research area.

Keywords: Collaborative Filtering, Cold Start, Sparsity Problem.

ÖZET

İŞBIRLİKÇİ FİLTRELEME SEYREKLİĞİ KULLANIMİ İLE İLGİLİ BİR ÇALIŞMA

Yegwende Vincent TIEMTORE

Bilgisayar Mühendisliği Anabilim Dalı

Anadolu Üniversitesi, Fen Bilimleri Enstitüsü, Mayıs 2017

Danışman: Prof. Dr. Yaşar HOŞCAN

İnternetin gelişiyle, sunulan çevrimiçi seçenek sayısı sürekli artıyor. Filmler, kitaplar, yemek tarifleri, dünya haberleri ... gibi birçoğu şey için, gerekli tüm bilgileri düşünme imkânı olmadan bir tercih yapmalıyız. Peki ya nasıl? Biz yalnız başımıza aynı problem ile yüzleşmediğimiz halde, eğer birisi bizim ile aynı zevklere sahip ise ve o son zamanlarda bir film sevdiyse, bizim o filmi sevme ¸sansımız da büyüyor. Başkalarına ait varolan bilgilerden, kendi kararlarımız üzerine iyileştirmeler yapmak için yararlanmak, işte bu nedenle mümkündür. Şimdi internet ve başkalarının tercihlerine ait büyük veriler sayesinde, ağızdan çıkan her bir kelimenin yayılmasına olanak sağlanıyor. Bu sezginin işleyişi ve resmileştirilmesi, İşbirlikçi Filtrelemeye (CF- Collaborative filtering) tabiidir.

İşbirlikçi Filtreleme, "Bilgi bombardımanı" sorunlarıyla başa ¸çıkmak için, dünyanın en˙ çok kullanılan filtreleme tekniği haline geldi. CF bazı tıkanmalardan dolayı zarar görüyor: gizlilik, soğuk başlangıç, kıt bilgi- kıtlık problemi ... Birçok araştırmacı, sonradan gelen problem için birçok yöntem ¨önerdiler ancak hala çok büyük ve önemli bir araştırma alanı olarak biliniyor.

Anahtar Sözcükler: İşbirlikçi Filtreleme, Soğuk Başlangıç, Kıtlık Problemi.

ACKNOWLEDGEMENTS

I want to thank in the first place, the entire teaching staff of the Computer Engineering Department, Anadolu University, for their valuable permanent help and support throughout my studies.

Special thanks to my supervisor Prof. Dr. Yaşar HOŞCAN for his accessibility and guidance during my thesis. I am very thankful to Asst. Prof. Dr. Ahmet ARSLAN for his encouragements and advices.

I cannot conclude without expressing my gratitude to my family: my parents, brothers and sisters, Sibdou Martine COMPAORE, Mariussia Eudoxie SAWADOGO for all the sacrifices they made for me during my stay in Turkey. Thank you!

Yegwende Vincent TIEMTORE

12/06/2017

STATEMENT OF COMPLIANCE WITH ETHICAL PRINCIPLES AND RULES

I hereby truthfully declare that this thesis is an original work prepared by me; that I have behaved in accordance with the scientific ethical principles and rules throughout the stages of preparation, data collection, analysis and presentation of my work; that I have cited the sources of all the data and information that could be obtained within the scope of this study, and included these sources in the references section; and that this study has been scanned for plagiarism with "scientific plagiarism detection program" used by Anadolu University, and that "it does not have any plagiarism" whatsoever. I also declare that, if a case contrary to my declaration is detected in my work at any time, I hereby express my consent to all the ethical and legal consequences that are involved.

Yegwende Vincent TIEMTORE

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

ABBREVIATIONS

CF	: Collaborative Filtering
CBRS	: Content-Based Recommender Systems
CFRS	: Collaborative Filtering Recommender Systems
LSI	: Latent Semantic Indexing
MAR	: Missing At Random
MBCF	: Memory-Based Collaborative Filtering
MCAR	: Missing Completely At Random
MNAR	: Missing Not At Random
MOBCF	: Model-Based Collaborative Filtering
PCA	: Principal Component Analysis
SNA	: Social Network Analysis
SVD	: Singular Value Decomposition
TL	: Transfer Learning

1. INTRODUCTION

Nowadays, large amount of information is available to everyone through the development of information technology, the Web is a perfect example. Therefore, the problem of information overload has quickly laid and constitutes a challenge to overcome. The problem leads by information overload is that, there is an exponential growing difficulty for people in finding the most relevant things they want when they need it in a way that best meets their requirements. Many researches have been done to cope with the information overload problem. One of the most important technology are Web search engines. It provides meaningful doorways to deal with the huge amount of available information.

A study leads by the Pew Internet and American Life Project, states that 84% of American adult Internet users have used a search engine to gathered web information. According to the same research study on any given day, more than 60 million American adults send over 200 million information requests to Web search engines, making the latter, second most popular online activity [16]. However, search engines have demonstrated their inability to provide customized and personalized results to user's queries. Indeed, the result returned to users are often most irrelevant and doesn't meet his expectations. The user must manually select what is relevant to him. That is a painful and a tedious task. Hence the introduction of recommender systems. In contrast to the information search engines (Google, Yahoo...), which require the user a systematic formulation of its need using keywords, recommender systems provide relevant resources to users according to their preferences. The user sees not only his search time reduced but also receives top relevant suggestions from the system to which he would not have spontaneously pay attention. More than just an indispensable information filtering technique, recommender systems have become a defining component of human condition allowing him to find his tastes over online 's endless supply of information in a reasonable time.

Recommender systems or recommendation systems use different technologies. They can be classified as content-based systems and collaborative filtering systems. Both techniques present advantages and drawbacks. Collaborative filtering has emerged as the most used technique for having more advantages over the content-based technique. It is important to jot down that there are also hybrid techniques that are combining multiple recommendation techniques to achieve a cooperation between them [3].

Many scholars have been worked on Recommender systems specifically on collaborative filtering to outperform user expectations but it still experiences some challenges, such as privacy, scalability, privacy, sparsity, cold start problem... This present thesis focuses on how to handle the sparsity problem in collaborative filtering. To start with, in this first chapter, we will present in more details recommender systems, as well as their limitations.

In chapter 2, we are going to review related work about dealing with the sparsity problem in collaborative filtering.

A new method to handle the sparsity problem in CF is presented in chapter 3. Conclusions and future work are presented in the last chapter.

1.1. Review of Recommender Systems

A recommender system or recommendation is the term used to describe a variety of process designed to provide information to people, information that is in line with the interests of these persons. Recommendation consists of finding a” prediction” as to the usefulness of the information for the user. This prediction is performed based on the” profile” of the user and results in decision making. Results are then classified as” recommended” or” not recommended” information.

Recommender systems are essential due to the exponential growth of information which made it too expensive for users to try all possible alternatives offered to them independently.

Users to achieve individualized recommendations (books, music, documents, television programs, web pages...) typically use such systems. An effective solution to reduce complexity when searching on the Internet was given by recommender systems.

According to [3], the roots of recommendation systems date back to the important work in cognitive science, the approximation theory, information retrieval, prediction theories, and have links to management science, and to modeling consumer choice in marketing. Recommender systems have also emerged in the areas of security, fraud detection...

Different approaches have been used to provide recommendations. They are known as collaborative filtering, content-based, hybrid etc. In Figure 1.1, we are showing a formal definition of a recommender system [3].

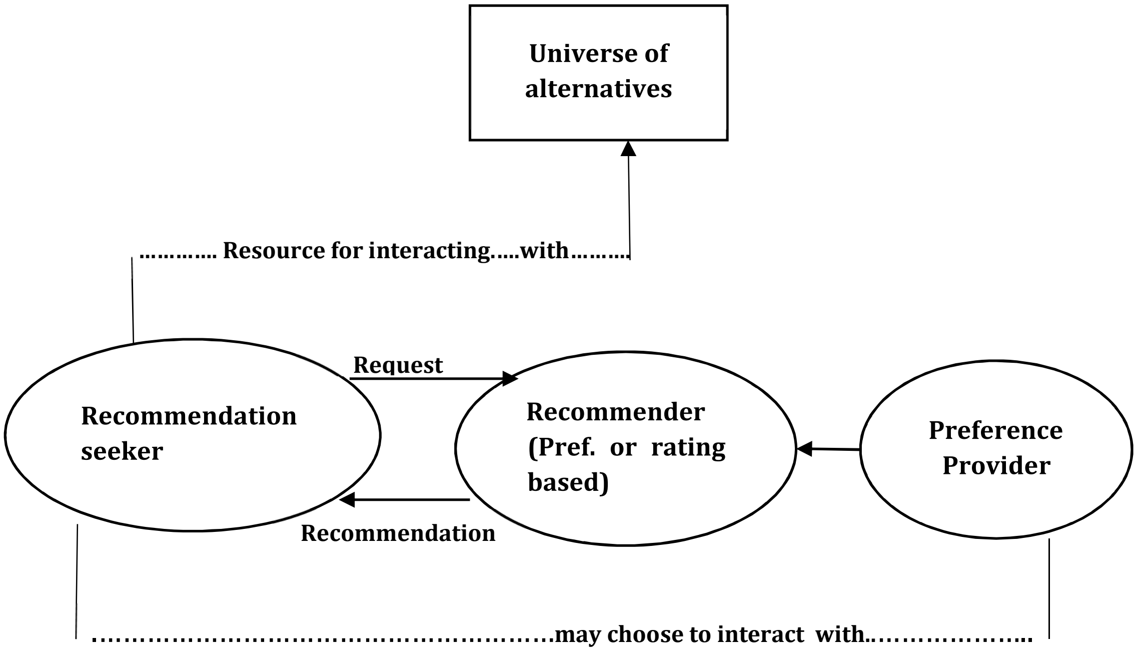

Universe of alternatives
………… Resource for interacting…..with………
Request
Recommendation seeker
Recommender (Pref. or rating based)
Preference Provider
Recommendation
………………………………………………may choose to interact with…………………..

Figure 1.1. *Formal definition of recommender*

1.1.1. Content-based filtering

Content-based filtering systems recommend similar items to those that the user has already appreciated. The similarity is calculated by comparing the user interest (introduced explicitly through a survey, for example, or implicitly through the monitoring of its behavior) with the metadata or document characteristics, without considering the views other users. Content-based recommenders are usually suited for recommending web pages, TV programs, articles etc., by using techniques such as tf-idf, vector-space queries [17],” intelligent” agents [18] and information visualization [19].

Content-based filtering process

In basic content-based recommender system we can identify two main components. These are the item profile and the user profile.

- item profile: items to be recommended and the corresponding features;

- user profile: users provide information about their preferences to the system. User information can be provided explicitly (questionnaire for example), or implicitly (by monitoring user behavior clicks, time and frequency of consulting an item...)

Then the recommender engine recommends items to the user according to the existing information. The recommendation is processed by combining items features information with the user preferences. The following Figure 1.2 is an explanation of content-based recommendation method.

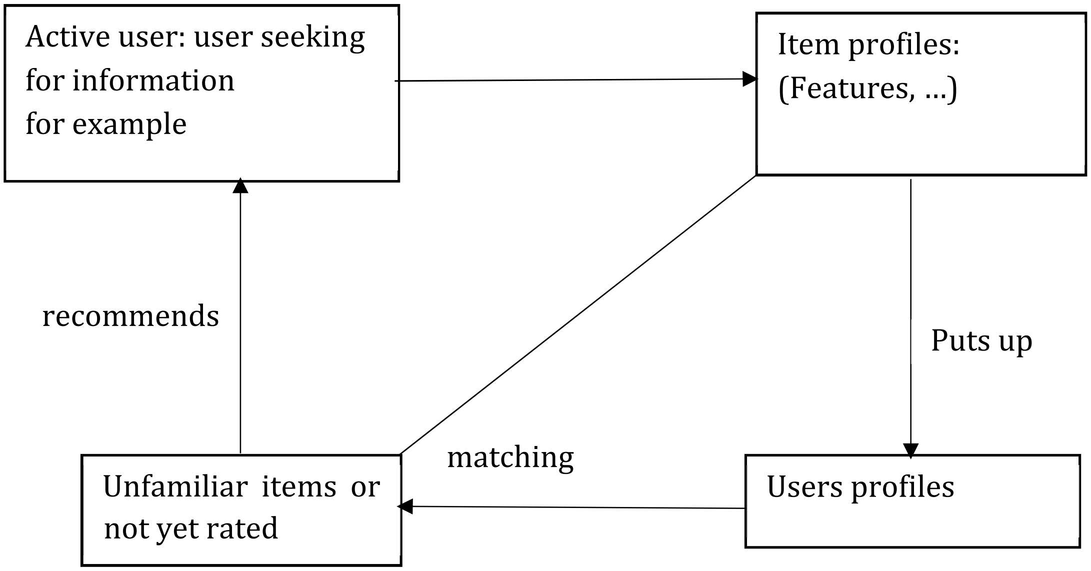

Active user: user seeking for information for example
Item profiles: (Features, ...)
recommends
Puts up
matching
Unfamiliar items or not yet rated
Users profiles

Figure 1.2. *Content-based recommender process*

Content-based filtering limitations

Content-based filtering suffers from several limitations including the inability to recommend multimedia documents that do not have information on their contents. In addition, in text documents retrieval this kind of filtering cannot provide suitable recommendations when polysemy, synonymy, multi-word concepts (homograph, homophony...) occur in the keywords.

The funnel effect restricts the users field of vision; this type of filtering is unable to recommend items that are different from those that the user has already seen and evaluated. For example, if a user has solely rated movies directed by James Cameron, he will be recommended just that type of movies. The user therefore never has the opportunity to see and try these new different items to which he can be interested in. This problem is known as over-specialization or serendipity problem; that is to state the tendency of CBRS to provide recommendations without degree of novelty.

The user must also evaluate enough items before the recommendation engine start being able recommending the relevant items to him, this is not the case for new users. This problem is known in the literature as one of the cold start problem specifically the new user problem.

Content-based techniques have a natural limit in the number and type of features that are associated, whether automatically or manually, with the objects they recommend [20]. Considering an example of recommending movies, CBRS needs to have some domain knowledge (actors, directors) or domain ontologies before being able to recommend relevant movies to the user. When the content available is not sufficient to discriminate items the users may like from the ones he doesn't, CBRS will not be able to provide suitable recommendations.

This problem is defined as limited content analysis.

Content-based filtering advantages

Content-based recommender systems present some advantages over collaborative filtering.

- User independence: content based recommenders only operate on the feedback provided by the active user to build his own profile while collaborative filtering methods need other users' preferences(ratings) in order to find the users that are closest (nearest neighbors) to the active user, i-e ., users that have similar tastes as they rated the same elements similarly; then solely the items that are most liked by the neighbors of the current active user will be recommended to him;

- Transparency: content-based recommender systems can provide explanations on how an item occur in the recommended item list; that is by explicitly returning the content features and descriptions on recommended items. So, this information can be used in deciding whether a recommendation is relevant. Conversely, collaborative systems can be seen as black funnels since the only description we have from a recommended item is that unknown users having the similar tastes (most relevant neighbors) liked the active user liked that items.

- New item: Content-based recommender systems (CBRS) are suited in recommending items that have not been yet rated by any user while new item problem affects Collaborative Filtering recommender systems (CFRS). In fact, in CFRS until the number of users having rated a new item reach a certain level, CFRS are unable to make recommendations. CBRS don't experience new item problem because the recommendation doesn't depend on other users' ratings information on items.

1.1.2. Collaborative filtering

To face up to the problem of information overload, collaborative filtering is a recommending approach using the ratings that users have made on certain items so as to recommend these same items to other users similar to them i-e having same preferences. The main rule of thumb behind Collaborative Filtering (CF) is that people who have same preferences in the past will also have similar tastes in the future. That is, based on surmise that people in search of information should be able to use what others have already found and evaluated.

CF refers to a class of techniques used in recommender systems, that recommend items to users that other users with similar tastes have liked in the past [21]. Collaborative filtering methods are divided generally into memory-based approaches and model-based approaches.

Collaborative filtering process

As shown in 1 Figure 1.3, there are three main processes in a collaborative filtering system:

1. A user squeezes out his tastes and preferences by evaluating items (for example by providing ratings to books, films, or CD) of the system. A user can express implicitly or explicitly,

    (a) implicitly: the system induces the user satisfaction through his actions (clicks, duration of consulting for example a page...),

    (b) explicitly: The user gives a numerical value on a given scale or a qualitative value of satisfaction, for example, poor, fair, good, and excellent. In this thesis, we are going to consider solely users given explicit numerical values.

2. Building a group of similar users; the collaborative filtering engine compare active user ratings against other users and then output the most relevant community to which he may be the closest. (a group of users having the same tastes and preferences). The similarity can be computed using Pearson or cosine similarity...,

3. Providing recommendation or producing prediction.

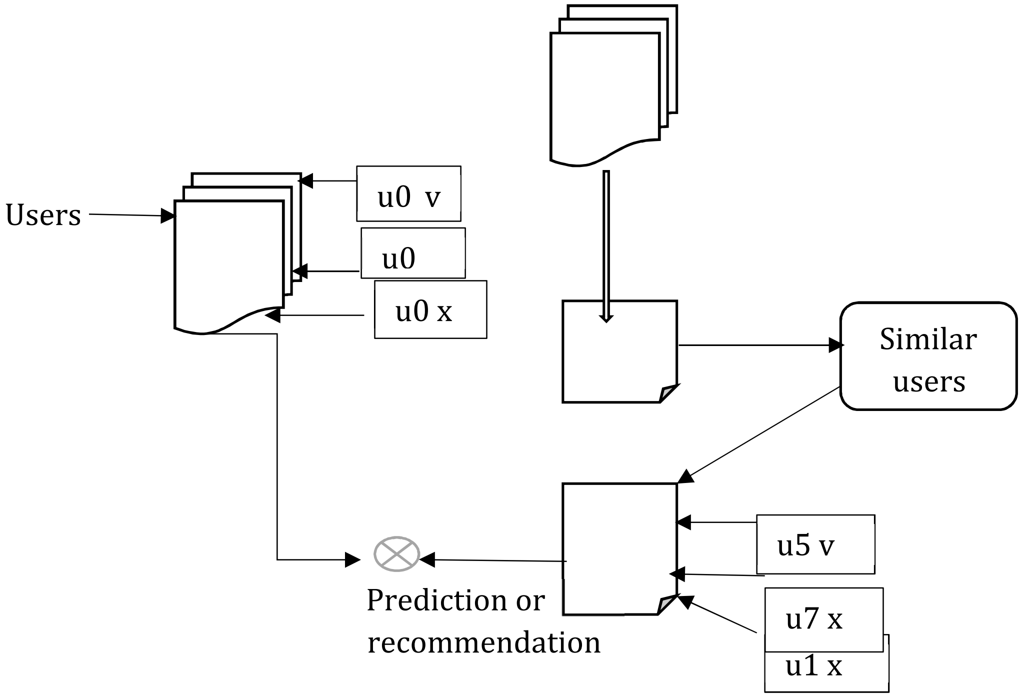
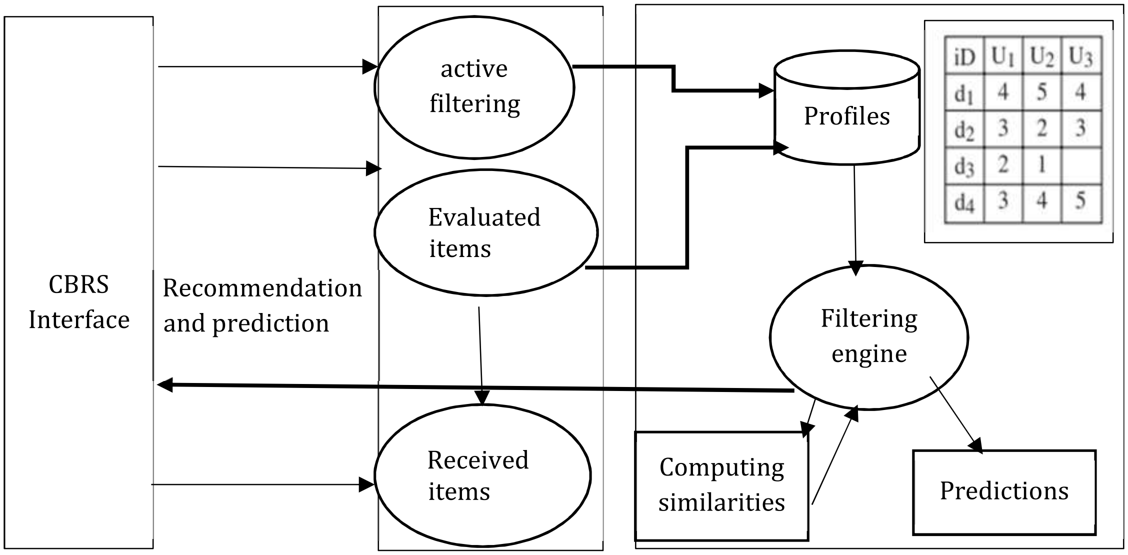


Figure 1.3. *Collaborative Filtering process*

uq e: means user’s q item e. In the figure above user u0 rated a certain items v, w, x. The system computes similarity and provide u0 similar users u5, u7, u1. Finally, prediction or recommendation are made for user u0.

In the following Figure 1.4, we are showing the main architecture of a CFRS. An active user evaluated items that are push up to constitute the user profiles (users*items matrix). The system then computes similarities and at the end produces recommendations or predictions that are showed to the interface.



iD	U_1	U_2	U_3
d_1	4	5	4
d_2	3	2	3
d_3	2	1	
d_4	3	4	5

Figure 1.4. *Main architecture of a collaborative filtering*

Collaborative filtering techniques

Collaborative filtering techniques are usually categorized into two main categories:

1. Memory-based approaches: MBCF methods use the entire or a sample of the user matrix data to produce a prediction. Each user belongs to a group of people with similar tastes or preferences. Memory-based techniques are mainly computed into two steps: based on users’ ratings compute similarities from partial information

about the active user, and a set of weights calculated from the users*items matrix; then provide prediction of the unknown ratings or a top n recommending items.

(a) user-based: in user based systems, the similarity between users are calculated by comparing their ratings on the same item, and then compute the predicted rating for item j by user i as a weighted average of the ratings of j by users similar to user i, where weights are the similarities of these users with i [3].

(b) item-based: in item-based systems, the similarity between two items is determined by comparing the rating made by same user i on the items. Then, the predicted rating of item j by user i is obtained as a weighted average of the ratings of i on items, weighted by the similarity between those items [3].

2. Model-based approaches: Model-Based Collaborative Filtering (MOBCF) implicate constructing a model based on the users*items matrix of ratings. that is, digging into data and pulling out some knowledge which is used as a model to provide recommendations and predict unseen ratings. In that way, we don’t have to use as like in MBCF the complete dataset every time. Model-based thus offer considerably benefits in terms of speed and scalability.

As shown in Table 1.1, we did a simple comparison of memory based against model-based collaborative filtering techniques.

Table 1.1. *Comparison of memory-based and model-based techniques*

	Some advantages	Some limitations
MBCF	More accurate predictions online simple algorithm to implement easy to update the database	very slow over-fitting may occur
MOBCF	scalability prediction speed over-fitting avoidance less sensitive to data sparsity	inflexibility poor quality of predictions offline time cost for building the model must learn a model for every new user

Some collaborative filtering advantages over content -based methods

CBRS are not sensible to the overspecialization problem stated in the content-based limitations section. In fact, these systems use other users' ratings recommendations, so they can process any type of content (multimedia stream not treated by CBRS) and make predictions or recommendations for any items, even those dissimilar to the ones the active user have already seen in the past.

Collaborative Filtering provides three key additional advantages to information filtering that are not provided by content-based filtering [1]:

- support for filtering items whose content is not easily analyzed by automated processes.
- the ability to filter items based on quality and taste.
- the ability to provide serendipitous recommendations.

Nonetheless, CFRS present some limitations.

- new user problem: it's the same problem as stated in content-based limitations section. Recommender systems cannot predict or recommends items to new users since these users suffered from lack of ratings and purchase history.
- new item problem: as long as the recommender systems work new items are reckon up. As CBRS are based only on user's tastes to make recommendations, until the number of users having rated a new item entering the system reaches a certain support, the recommender system would be unable to recommend that item (there is no substantial information to compute similarity).
- sparsity problem: sparsity is one of the main bottleneck of collaborative filtering systems. Data sparseness reduces considerably the recommendations quality. The main reason of sparsity problem is that generally the number of items is very huge and have been rated by few users. As long as the number of items increases so does the matrix dimension and sparsity gets greater. CFRS experience this problem since these systems are mainly based on ratings provided by users on items. Let U be the number of users in the matrix, I the number of items, and V be the number of rating values. Dataset sparsity S is calculated as following:

$$S = 1 - \frac{V}{U * I} * 100\% \qquad (1.1)$$

Both new user problem and new item problem are categorized as the cold-start problem. In some literature, a new system problem is also defined as in part of cold start problem. It is also important to spot out that cold start problem is known as a special case of the sparsity problem. The sparsity problem will be deeply presented in chapter 2 as our thesis is about. To cope with the limitations of CBRS and CFRS hybrid collaborative approaches have been introduced.

Table 1.2. *Content-based vs collaborative filtering*

	Some advantages	Some limitations
CBRS	user independence transparency not sensible to new item problem	cannot recommend streams overspecialization or serendipity limited content analysis poor quality of recommendations
CFRS	support for filtering items having complex content ability to filter items based on quality and taste ability to provide serendipitous recommendations not sensible to overspecialization Memory-based produce easily recommendation ability to add incrementally with easy new data predictions performance in MOBCF	new item problem new user sparsity scalability privacy

1.1.3. Hybrid approaches

Hybrid recommender systems are combination of multiple recommender systems. The main purpose of hybrid systems is creating a cooperation between different recommenders in order to tackle the hurdles they experienced.

Different ways to combine collaborative and content-based methods into a hybrid recommender system can be classified as follows [24]:

- implementing collaborative and content-based methods separately and combining their predictions;
- incorporating some content-based characteristics into a collaborative approach;

- incorporating some collaborative characteristics into a content-based approach, and
- constructing a general unifying model that incorporates both content-based and collaborative characteristics.

1.2. Some Recommender Systems Challenges

In the previous section 1.1, we have presented the different types of recommender systems, each with its advantages and limitations. In addition to these issues, here are some problems that should be addressed in recommender systems for better performance.

- Big-data
- Novelty and diversity of recommendations
- Erroneous and malicious data
- Conflict resolution while using ensemble/ hybrid approaches.
- Ranking of the recommendations
- Impact of context-awareness
- Impact of mobility and pervasiveness
- Privacy concerns

In this chapter, we have presented a brief review of recommender systems. We showed the type of existing systems, we then we spotted out some challenges these systems are experiencing. In the next chapter, we are going to discuss the sparsity problem in recommender systems; first we will introduce the missing data theory and secondly exhibit related work about sparseness in recommender systems.

2. RELATED WORK

A collaborative filtering data can be seen as a matrix array M where each row denotes a user and each column corresponds to an item. Let U be the number of users in the matrix, I the number of items, and V be the number of rating values. A matrix of rated items indicator R is introduced to indicate whether an item is rated. So: if R_{iu} is observed R_{iu}=1 else R_{iu}=0 to mean that the item has not been yet rated. This is missing rating that can affect performance of collaborative filtering.

In this chapter, we are going to present the sparsity problem in CFRS. But, first we will introduce the missing data theory to show how missing data occur and what impact they have in recommender systems processes.

2.1. Missing Data Theory

Missing data can be defined as the absence of data items that hide some information that may be important. Missing data is characterized by the pattern of missingness and the mechanism of missingness.

One important consideration when dealing with datasets containing a large amount of missing data is the question why data is missing? Answering to this question is known as the missing data mechanism.

2.1.1. Pattern of missingness

Pattern of missingness visually illustrates how values are missing in a dataset. Some of the patterns of missingness described by Schafer et al. (2002) [27] include univariate, monotone and arbitrary.

- univariate: refers to the situation in which the missing values occur in only one of the variables while the other variables are completely observed.

- monotone: is about the dropout pattern. That is, if a user has missing data in the i^{th} position, then all the subsequent values are also missing.
- arbitrary: it is the situation where the missing data occur in any of the variables at any position.

2.1.2. Mechanism of missingness

It deals with the probabilistic definition of the missing value. Little and Rubin [28][29] came up with the classification of missing data mechanism into different types: Missing At Random (MAR), Missing Completely At Random (MCAR), Missing Not At Random (MNAR).

Let us consider a complete dataset as D_{comp}. D_{obs} denotes observed data and D_{mis} is representing the unobserved data. So, we have D_{comp}= (D_{obs}, D_{mis}). Let's N be the probability of missingness.

- MAR: the situation where the probability of missingness is only dependent on observed values and not on any unobserved data. That means the missingness is related to other dataset variables, but not to the underlying values of the incomplete variable itself.

$$P(N|D_{comp}) = P(N|D_{obs}) \tag{2.1}$$

  MAR is considered as ignorable missingness that is, there it is not worth to specify the missing data mechanism explicitly. Missing data is ignorable if these three conditions hold (Rubin 1976) [28]:

  - the missing data mechanism is MAR.
  - the complete data parameter D_{comp} can be decomposed as:

$$D_{comp} = (D_{obs}, D_{mis})$$

  - D_{obs} and D_{mis} are a priori independents.

- MCAR: the probability of missingness is not dependent on any observed or unobserved data. That is, the probability of missing data on a variable T is not related to other variables values in the dataset and to the variable T itself. Some reasons that can explained MCAR data are equipment hazards, or users not providing data correctly (e.g: entering a rating above the max defined scale). Another way to think of MCAR is to note that in that case any piece of data is just as likely to be missing as any other piece of data [37]. In this case the corresponding pattern of missingness is arbitrary.

$$P(N|D_{comp}) = P(N) \tag{2.2}$$

- MNAR: The probability of missingness is only dependent on unobserved data.

How missing data occur

These are some reasons that can explained why missing values may occur in datasets.

- Not relevant to a particular case
- could not be recorded when collecting data
- ignored due to privacy concerns
- unavailability of data
- corrupt data due to data inconsistency
- items not yet rated

2.2. Collaborative Filtering and the Sparsity Problem

CF Has been the most widely and successful recommendation system method to date. The main goal of CF is recommending items to a user based on the preferences and tastes of other users. In their natural form, CF systems are not considering the content of the items at all, they rely wholly on the judgement of users to provide recommendations whether items are relevant [38]. CF is applied in various applications.

Tapestry [39] is one of the first computerized CF systems. Tapestry was built for a small group of users. Thanks to Tapestry, users were able to winnow the information streams (emails and Usenet news articles). The evaluation of items by items were done by text annotation or by giving ratings (numeric or binary). Other users were then having the possibility to perform queries such as " show me the items that Bob annotated with Excellent" etc.

A similar approach is proposed by Maltz and Ehrlich's "active collaborative filtering" allowing users to direct recommendations to their friends and colleagues through a Lotus Notes database [38]. Another system based on the same analogy was the Grundy system which can build user's preference model to recommend relevant books to every user [41]. These systems as shown some limits (poor quality of recommendations as long as users number grows so precision get lowest). Then we assist to the apparition of automated collaborative filtering. These are systems using statistical approaches to compute neighborhood of users. Some of the automated collaborative filtering include the GroupLens Research [42][43] providing recommendations for Usenet items (news and movies). We can also quote Ringo [45] and Video Recommender [44] are respectively email and web systems generating recommendations on music and movies.

Many other typically collaborative recommendation systems exist: Jester [53], Amazon.com [46] etc.

As we showed in chapter 1 (section advantages of collaborative filtering over content-based) and in the present section, collaborative filtering has been a substantial success but there are several problems that CF systems are still suffering from. We report these hurdles in chapter 1. We are going next to be interested in specifically on the sparsity problem.

2.2.1. The Sparsity problem

Collaborative filtering has two main goals: providing recommendations to users or making ratings predictions based on other users' preferences and tastes. In typical CFRS users are represented by the items they have already evaluated, purchased, or rated. Therefore, for better performance of CBRS, the history of users' preferences is necessary. Notwithstanding, user's database is everlastingly scarce. Let us consider an online movie collaborative filtering system having 5 million items (movies). Every user will be represented by a Boolean matrix vector of 5 million movies. The Boolean matrix shows whether a user has evaluated a movie in the past. The evaluation is processed by providing ratings in a certain scale; let's say an interval [1-5], 1 being the least value if a user did not like a movie.

The user-item matrix |U|*|I| is represented as:

$$R_{iu} = \begin{cases} k, & \text{k=1...5, if user u rated item i} \\ 0, & \text{otherwise} \end{cases} \tag{2.3}$$

In many collaborative filtering large-scale systems, the couple number of users and number of items are very overwhelming. So, the user-item interaction matrix is intensely sparse; that is, in the interaction matrix R there are very scanty items whose value is not 0. This hurdle is known as the sparsity problem. The sparsity issue includes two aspects [54]:

- the number of users rating is very small compared to the number of items.

- the overlapping number of two users' ratings is very small.

Data scarceness is an inescapable problem in collaborative filtering and it has a great negative impact on collaborative filtering effectiveness. CF in sparse data results in loss of recommendation accuracy and performance. That is, because of the sparsity issue, it is tremendously plausible that the similarity between two users is zero, rendering collaborative filtering useless [55].

The cold-start problem supplementary, spotlights why it is necessary to solve the data scarcity issue. The cold-start problem in CF emphases the issue when a new user or a new item just enter the CF system [56]. CF is unable to provide suitable recommendations for a new user since this user does not have a purchase or ratings history. Comparatively, when a new item enters the system, CF is unlikely recommending it to many users since negligible users have been rated that item. The cold-start can be viewed as a special case of the sparsity problem, where much columns or rows of the user-item interaction matrix R are 0 [26].

Data sparseness is one of the major bottleneck in CF. Sparsity affects highly CF recommendations quality and system performance. Many researchers have focused on the sparsity problem proposing methods to address the issue. Some solutions have been proposed. This is what we are going to discuss in the following section.

2.2.2. Previous work

Collaborative Filtering is a personalized recommendation system that has been extensively used in many areas [57][58][59] etc. However, the sparsity problem dramatically reduces user experience in CF systems. In order to improve recommendations quality and performance, many solutions have been proposed by researchers.

Dimensionality reduction

Essentially, dimensionality reduction approaches to alleviate the scarcity problem in CF, proceed by generating a substantial user-interaction matrix. Indeed, only users having rated many items and items that have been highly rated by important number of users are considered. Predictions are then made by using the resulting reduced matrix. Basically, dimensionality reduction is known as complete case analysis.

Breese et al. proposed a simple strategy of dimensionality reduction that consists of clustering items and users and then use the resulting clusters as basic unit in the prediction

[60]. Better advanced techniques use statistical techniques such as Principal Component Analysis (PCA) [53], Latent Semantic Indexing (LSI) [55] and Singular Value Decomposition (SVD) [61].

Gong proposes another method et al., using combination of SVD and item-based recommendation system in CF. In this approach, the results of SVD are used to fill the missing values and then utilized item-based method to recommend items [73].

B. Sarwar et al. (2000) showed that dimensionality reduction can greatly improve recommendation quality in certain applications but also performs badly in others [61]. Some of the disadvantages of dimensionality reduction are:

- Loss of information that leads to unreliable predictions.
- Loss of efficiency;
- Loss of money. It is costly to obtain data.

Using hybrid approaches

A different approach proposed by researchers to cope with the sparsity problem is by using hybrid recommender systems. Hybrid systems combine content-based and collaborative filtering to take advantages of both systems [75] [38] [76] [77] [74]. Indeed, with user-item interactions, hybrid techniques take advantages also of the similarities between items. These similarities are calculated based on items contents and lead to more accurate predictions. However, hybrid approaches also have some disadvantages. In fact, item content information must be available otherwise they cannot be used. Furthermore, a meaningful metric for computing similarities between items should be used.

In practice, it is costly to acquire item content information, or information may be unavailable in many well-known datasets; the similarity metric also sometimes is not immediately available.

Content-boosted CF

Content-boosted [78] [79] [18] like hybrid approaches require item content information and a similarity metric to compute similarities between items. A. Popescul et al. proposed a CF approach based on a unified probabilistic model to integrate content information to alleviate data scarcity [80].

Content-boosted CF prove great improvements in terms of recommendation quality.

Nonetheless, they present the same disadvantages as like in hybrid systems.

Implicit ratings

Many CF systems attempt to fill up user ratings by observing and monitoring user behavior. The GroupLens Research system found that time spent reading an article on Usenet news articles system can be an effective rating measure [81]. Terveen L. et al. determined that URLs mentioned in Usenet postings after been filtered could be used to provide recommendations [84]. Further systems that explored user behavior or user history are Siteseer [83] and [82].

Imputation methods

Imputation techniques are class of procedures that consist of filling the missing ratings with estimated ones. Imputation methods are divided into two categories: single imputation and multiple imputation.

- single imputation: filling missing data by a kind of predicted values. Most common single imputation methods are: mean imputation, cold deck imputation, hot deck imputation, and regression imputation [29]. Single imputation provides greater consistency, however leads to underestimation of standard errors (variance for example). Multiple imputation deal with this issue.
- multiple imputation: instead of filling each missing data by a singular value, replace the missing value with a set of probable values representing the uncertainty about the correct value.

Weiwei Xia et al. proposed an imputation method for dealing with the sparsity problem [5]. This study took advantage of user demographic information to fill the missing ratings. They assumed for example that users in the same age range may have similar preferences and then used available info to impute missing ratings.

Transfer learning techniques

Transfer Learning (TL) can be seen as the ability of a system to recognize and apply knowledge and skills learned from previous tasks to novel tasks [9]. The domain from where knowledge is taken is called auxiliary domain or auxiliary task; and the domain to which knowledge is transferred the target domain or target task. TL methods are collective [15][13] or adaptive [12][7] in collaborative filtering. In TL, one should ask three fundamental questions [11]:

- what to transfer: we are seeking for the part of knowledge that can be transferred across domains or tasks.
- how to transfer: which algorithm is most suitable.
- when to transfer: in which conditions and situations, transferring knowledge should be done.

TL has been widely applied to collaborative filtering to reduce the sparsity problem. Bin Li et al. reduce sparseness in a book (target task) collaborative filtering system by transferring knowledge learned from a movie CF system (auxiliary task) [7]. The main drawback was that both target domain and auxiliary domain should have the same ratings scale. This problem is solved in the approach introduced by Wan et al. [8]. Missing ratings are filled up by applying TL via features tags. However, this method needed target domain and auxiliary domain having common tags which are used to describe the features of users.

Graph based methods

Graph-based approaches are category of methods that consider the user-item interactions matrix as a bipartite graph were each node represents a user and an edge (*u*, *i*) exists between a user *u* and an item *i*, if *u* has evaluated *i*. Furthermore, an edge (*u*, *i*) might have a weight *w* corresponding to the rating given by user *u* to item *i*. These methods are based on graph theoretic measures and demonstrated their capability of deriving global similarities between users or items.

To alleviate the data scarcity in CF, F. Fouss et al. proposed such a method, where similarities between two users are computed as the average commute time between these users commute among their respective nodes in a random walk graph [30]. Another method using random-walk of the graph to compute similarities is proposed by M. Gori and A. Pucci [31]. The minimal hop distance of a graph and the spread activation of the nodes in the graph are also respectively used as a similarity measure in [32] and [26].

Graph-based methods have demonstrated great impact on solving the sparsity problem, howbeit the main drawback of these methods is that in the prediction process there is often no good interpretation of the similarity measures [6].

Association retrieval

The origin of association retrieval is in statistical studies of relationships between terms and documents in a text collection [4].

In CF, association retrieval techniques consist of exploring transitive relationships among users to address the sparsity problem. The basic rule of thumb behind association retrieval is, based on user-item interactions matrix build a graph model of items and users and then using this graph to find relationship between users and items in order to improve recommendation quality. In our daily life, this idea is also reflected. For example, if Bob is Alice's friend and Mat is Bob's friend, Alice can recommend a recipe to Mat since Bob is the transitive relationship between them.

Yibo Chen et al. make use of this assumption to propose a method improving recommendation precision in CF [4]. Another approach has been investigated by Zan Huang, Hsinchum Chen et al. [26]. The effectiveness of these approaches in solving the sparsity problem was evaluated and proved; and they demonstrated better recommendation quality, but they suffered from the scalability problem as long as users or items enter the system. In addition, they fail in expressing formally the subjective notion of the associations.

Trust approaches based on user social network

To deal with the sparsity problem some researchers have applied Social Network Analysis (SNA) to collaborative filtering. Kaya H. and Alpaslan proposed a one-class CF based on SNA to address ratings scarcity. In this CF system, a comparison of social networks belonging to specific domains against those belonging to more generic domain is done in terms of their usability [33]. Mingjuan Zhou presented a book recommendation system based on web social network. The problem of social trust has been analyzed and a model of social trust based recommender was been built [25].

Trust is defined as one's belief toward others in providing accurate ratings relative to the active user [2]. Social trust takes advantage with the development of social network in addition to transitive associations between users to build CF systems that are not sensible to data scarcity problem (users sharing the same social community seem to have similar preferences). Trust information can be explicitly collected directly from users (u_1 specify u_2 and u_3 as his trust neighbors) or implicit that is, inferred from users' ratings information.

Guibing Guo et al. proposed a merge trust CF system to deal with data sparsity and cold start problem. In this system, the ratings of trusted neighbors of an active user are merged by taking the average rating on the commonly rated items based on the extent to which trust neighbors are similar to the active user [2]. Three steps to make recommendations:

- identification and aggregation of the active user 's trusted neighbors.
- merging trusted neighbors' ratings into a single value for each item.
- probing similar users based on the merged ratings profile and recommendations generation.

Manos Papagelis, Dimitris Plexousakis and Themistoklis Kutsuras work on a system based on trust inferences. Instead of reducing the user-item interactions matrix, they proposed a method that made use of users' additional information to define transitive properties between users in the context of a social network. Then, they develop a computational model that allow the analysis of users' transitive similarities based on trust inferences for alleviating the sparsity problem [72]. However, most of the existing trust-based are based on explicit trust specify by users; this information may be unavailable because of privacy concerns for example.

Exploring novel similarity metrics

Most CF approaches are based on similarity metrics such as cosine, Pearson correlation coefficient and mean squared difference [62]. These measures have demonstrated their ineffectiveness in the situations where users-items interactions matrix is very sparse. Indeed, it is not possible to compute neighborhood when the available data is not sufficient. Methods exploring new similarities metrics have been proposed attempting to solve matrix sparseness and improve accuracy.

H. J Ahn introduced a new similarity measure called Proximity-Impact-Popularity (PIP) where 3 main aspects are considered: proximity, impact, and the popularity of the user users ratings. This measure does not consider the global preference of the users' ratings, it considers only local similarity computation [47].

Jamali and Ester proposed a similarity metric able to weaken the similarity of small commonly items between users. This measure is based on the sigmoid function [48].

Bobadilla et al. [50] proposed a similarity measure which is the combination of the Jaccard metric [49] and mean squared difference [51], by assuming that these measures are complementary. Another measure called Mean-Jaccard-Difference (MJD) based on the same assumption has been developed to solve the sparsity problem.

F. Ortega presented a similarity metric based on singularities emphasizes that traditional similarity measures can be improved by using contextual information. Similarity is calculated in three steps [52]:

- categorizing users' ratings as positive and negative.
- computing the singularity values of each user and each item.
- replace the similarity with the singularity value.

This approach demonstrated its effectiveness, furthermore has been improved: a significance based similarity metric was revealed. This measure was combined with the traditional Pearson correlation or with the cosine similarity. Significance metric consists first of calculating three types of significance:

- the significance of an item.
- the significance of a user to recommend to other users.
- the significance of an item for a user.

Then, traditional Pearson correlation or cosine similarity is used to compute the similarities between users accordingly to the significance.

Christian Desrosiers and George Karypis proposed CF system based on indirect similarities to address the sparsity problem. They proposed a new way of computing global similarities based on a system of equations relating user similarities to item similarities [6]. This system's metric is similar to the graph based CF systems using graph theoretical measures, however in contrast to these methods, easily take into consideration content-based similarities.

Data smoothing

Data smoothing is another method proposed to alleviate the sparsity problem and improve recommendation performance. Cluster-based smoothing method is proposed by X. Gui Rong et al. to handle data sparsity. This CF framework is a hybrid system combining memory-based and model-based methods that aim to produce recommendations by group of closely related users [34] and supporting Support Vector Machine [85]. Another cluster smoothed based methods was presented by Aulia Rahmawati et al. to cope with the data sparsity hurdle using random neighbor selection mechanism [40].

BP neural networks [35] and zero-sum reward and punishment mechanism [36] are also smoothed methods used to smooth missing ratings in CF for better accuracy.

In this chapter, we introduced first the missing data theory that is subdivided into mechanism of missingness and pattern of missingness. The latter just emphasizes how missing values are missing in a dataset, while the former is the probabilistic definition of the missingness. We show that mechanism of missingness is MAR, MCAR or MNAR. In collaborative filtering, most of the proposed methods to handle data sparseness are assuming that missing ratings are missing at random (MAR) [14].

Then, we presented more in details CF and the sparsity problem. It appears that sparsity is a major bottleneck that may affects recommendations accuracy and performance.

Finally, we discussed some of the methods proposed by researchers to address data scarceness in CF systems. We did not present all the existing methods, we looked for papers cited at least three times in different work. We showed also that these methods still have some limitations. Many works have been done to address the sparsity but it still remains an emerging research area. In the next chapter, we will present a novel algorithm for alleviating sparsity in CF.

3. PROPOSED ALGORITHM

In the previous chapter 2/section 2.2, we focus on the sparsity problem and presented some existing methods to handle this issue in CF. We have seen that several methods have been presented by different researchers, each with its advantages and disadvantages.

In this chapter, we will be proposing a new way to deal with data sparseness and cold start problems in CF. The method we are going to present takes advantages of Social Network Analysis (SNA). Indeed, with the development of social network, additional information can be incorporated from diverse sources to cope with data scarcity in CF. This additional data include friendship [66], membership [70] [65] and social trust [71] [63]. Among these three sources, social trust is seen as the most reliable and the less ambiguously [2].

Trust can be seen as the belief of the active user toward others in providing accurate ratings relative to his own preferences [2]. Implicit trust [68] [69] and explicit trust [67] [64] have been explored in the literature. The latter trust is inferred for example from users' ratings while the former is directly defined by the users.

3.1. The Graph Based Trust Method

The graph based trust method is decomposed in three main processes. Firstly, by using active user's explicit trust ratings matrix, an oriented graph is built. Each node representing active user's trust neighbors and their respective neighbors. Every edge represents the trust relationship binding two users. Edges are also weighted, each weight representing the corresponding explicit rating between two users.

Secondly, active user's new rating matrix profile is built based on the weighted oriented graph matrix. Thirdly, we compute similarities and provide item rating predictions for active user.

3.1.1. Used notations

The following notations will be used throughout this chapter. All items, users and all given ratings are specified as *I*, *U,* and *R* respectively. The symbols u_m and i_p are used to denote a specific user and item respectively; m ⊂ U, p ⊂ I.

In addition, r_{u_m,i_p} is used to denote the fact that user u_m has rated item i_p, accordingly to a specified rating scale for example from 1 to 5. One of the goals of collaborative filtering can be defined as given a user-item interaction matrix (u_m , i_p , r_{u_m,i_p}), produce a reliable prediction (u_m , i_p , ?) for user u_m on item i_p . The produced prediction rating is noted as, $\hat{r}_{u_m,i_p}$.

The set of trust neighbors that might have been identified by the active user in trust-aware CF is denoted as TN_{u_m} . A trust value $t_{u_m,u_t} \in [1,5]$ is also specified by an active user u_m to denote the level of trust he has toward his trust neighbor u_t . The trust rating value is used as the graph edges weight while building active user's trust network graph.

With the aim to build the active user new profile, based on the graph, we are taking into account the min weight to reach a trust neighbor node and the number of edges of distance. The new profile trust rating is denoted as $\overline{r}_{t_{u_m,u_t}}$ and we denote by d the number of edges we go through before reaching a trust user node.

3.1.2. Building active user trust based new rating profile

To build the active user new rating profile, first his trust network is implemented. This network is built through an oriented graph where each node represents active user direct trust neighbor or the trust neighbors of the ones of the active user. Weighted edges are linking two nodes and the weight represents the extent to which a user believes in his neighbor. That is the weight is equal to t_{u_m,u_t} . To compute the new rating, we took into consideration the distance *d* representing the number of edges and the cost to reach a node w_{min} , where w_{min} is calculated by using Dijkstra's algorithm or any algorithm that help computing shortest path. The active user new profile trust rating $\overline{r}_{t_{u_m,u_t}}$ is calculated as:

$$\overline{r}_{t_{u_m,u_t}} = \frac{\sum (r_{TN_{u_m},i_p} * \frac{w_{min}}{d})}{\sum (\frac{w_{min}}{d})} \quad (3.1)$$

In equation 3.1, $r_{TN_{u_m},i_p}$ defined the set of items rated by active user's trust neighbors composing his trust network. The process of calculating continues until all the items rated by at least a trust neighbor have been covered. Then, the active user new trust dense ratings profile is created. In addition, we also consider two conditions:

$$\overline{r}_{t_{u_m,u_t}} = \begin{cases} min_r, & if, \overline{r}_{t_{u_m,u_t}} < min_r \\ max_r & if, \overline{r}_{t_{u_m,u_t}} > max_r \end{cases} \quad (3.2)$$

Where in Equation 3.2 min_r and max_r are respectively the lower bound and the upper bound of the rating scale interval.

After establishing the active user trust network and producing his new trust rating matrix, we now have a denser matrix to calculate the similarities between the users and produce predictions on a common rated item.

3.1.3. Providing predictions

Based on the formed active user rating profile, we compute the similarity between users.

Several similarity measures exist in the literature, cosine similarity [24] Bayesian similarity [23]... In this work, we are going to use the Pearson Correlation Coefficient (PCC). PCC similarity formulae is indicated as following:

$$S_{u,v} = \frac{\sum_{i \in I_{u,v}} (r_{u,i} - \overline{r_u})(r_{v,i} - \overline{r_v})}{\sqrt{\sum_{i \in I_{u,v}} (r_{u,i} - \overline{r_u})^2} \sqrt{\sum_{i \in I_{u,v}} (r_{v,i} - \overline{r_v})^2}} \qquad (3.3)$$

where $S_{u,v}$ denotes the similarity between two users u and v. Generally, $S_{u,v} \in [-1,1]$;

$I_{u,v} = I_u \bigcap I_v$ is the set of the commonly rated items by user u and user v.

We can have three similarity correlation types based on the value of $S_{u,v}$ between two users u and v.

$$if\ S_{u,v} = \begin{cases} 0, & \text{then there is no correlation between u and v} \\ > 0, & \text{then u and v have positive correlation} \\ < 0, & \text{then there is opposite correlation between u and v} \end{cases} \qquad (3.4)$$

After computing similarities between users, a set of nearest neighbors to the active user is established based on the values of similarity and is noted *NN*$_{ua}$, where u_a indicates the active user. Then, all the ratings of *NN*$_{ua}$ are aggregated to provide a prediction on an item i_p for the active user u_a. Many predictions formula are available in the literature, but here we will use the weighted average prediction formulae because in the previous work related to ours [2] [67], that formulae have been used. Weight average is calculated as:

$$\hat{r}_{u_a,i_p} = \frac{\sum_{v \in NN_{u_a}} (S_{u_a,v} \cdot r_{v,i_p})}{\sum_{v \in NN_{u_a}} | S_{u_a,v} |} \tag{3.5}$$

3.2. Example Using Graph Based Trust

Our aim, is to present in this section an example of the graph based trust method in action. We are going systematically to produce prediction for a selected item. We suppose that, we have nine users and nine items. We have also the user-item matrix table 3.1 which is very sparse. Our goal is to generate a prediction on item i_5 specified by a question mark, for active user u_1. We denoted by u_k and i_j, respectively the set of users and the set of items; where $k,j \in [1,9]$. As showed in the table 3.1, users have rated few items by providing
explicit ratings on the scale [1-5].

Users have also expressed explicitly as in table 3.2 their trust neighbors. Trust values are specified in the range of [1-5], 1 meaning the lowest trust and 5 the highest trust value. The active user u_1 has reported u_2 and u_3 has his trust neighbors. The active user's trust neighbors have also reported their trust neighbors and so on.

The first step of the graph trust method is to establish the active user trust new rating profile. To achieve this goal, we have first to produce the active user trust network. This is done by linking the active user trust neighbors and the trust neighbors of his trust neighbors together. The trust network is represented by an oriented graph, where each node represents a trust neighbor and the weighted edges represent the trust links between two users. Weights denote the extent to which a user believes in his trust neighbor in providing accurate ratings. The trust network of active user u_1 is shown if figure 3.1. We are using a directed graph which implies that the trust information is asymmetric; that is, u_1 trusting u_2 does not imply u_2 trusts u_1.

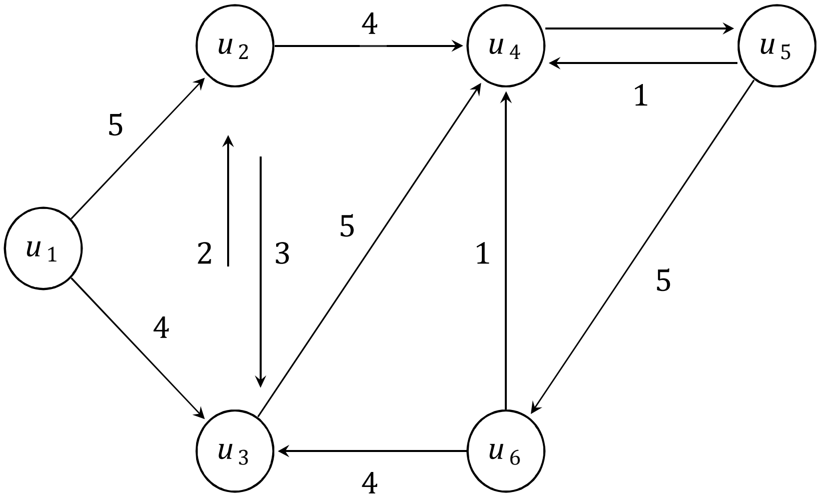


Figure 3.1. *User u_1 trust network*

Table 3.1. *User-item matrix*

User-item matrix	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9
u_1	-	-	5	-	?	-	-	-	-
u_2	5	-	4	-	3	-	-	2	-
u_3	-	4	-	3	-	-	-	1	-
u_4	3	-	5	-	2	-	-	-	-
u_5	-	4	4	-	3	-	-	3	-
u_6	-	3	3	5	5	-	-	-	-
u_7	-	-	-	-	-	-	5	-	4
u_8	-	-	4	-	2	-	-	1	-
u_9	-	-	4	-	5	-	-	5	-

Table 3.2. *User-user trust matrix*

User-user trust matrix	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9
u_1	-	5	4	-	-	-	-	-	-
u_2	-	-	3	4	-	-	-	-	-
u_3	-	2	-	5	-	-	-	-	-
u_4	-	-	-	-	3	-	-	-	-
u_5	-	-	-	1	-	5	-	-	-
u_6	-	-	4	1	-	-	-	-	-
u_7	-	-	-	-	-	-	-	-	-
u_8	-	-	-	-	-	-	-	-	-
u_9	-	-	-	-	-	-	-	-	-

The second step consists of producing active user's new trust ratings profile. The aim of this step is to produce a denser matrix that will help computing similarity between the active user u_1 and the other users composing his trust network. Before calculating similarity between users, we have first calculated the distance between them. As we said before, in section 3.1.1 two distance metrics are considered: the number of edges to reach a trust node *d* and the cost w_{min}. We summarize in table 3.3 *d* and w_{min}. The distance w_{min} is the shortest path from starting node u_1 to other nodes and is calculated using Dijkstra's algorithm.

Table 3.3. d, w_{min} and u_1 new trust rating profile

set of users	u_1	u_2	u_3	u_4	u_5	u_6
d	0	1	1	2	3	4
w_{min}	0	5	4	9	12	17

The active user new trust rating profile can now be generated by using equation 3.1. Let us calculāte $\boldsymbol{r_{u1, i1}}$. Item i_1 has not been rated by user u_1, but have been rated by u_1 trust neighbors composing his trust network. We can see that, on item i_1, u_2 gave 5 and for u_4 3 is given. The number of edges d from u_1 to reach u_2 and u_4 are respectively 1 and 2.

Concerning the shortest path from u_1 to u_2 and u_4, w_{min} is respectively 5 and 9.

$$\text{So, we have: } \overline{r_{u_1, i_1}} = \frac{(5 * 5 + 3 * \frac{9}{2})}{(5 + \frac{9}{2})} = 4.05$$

We proceed like we did for $\overline{r_{u_1, i_1}}$ to build u_1 new rating trust profile by covering all the items rated by at least one of u_1 trust neighbors. The complete profile is shown in table 3.4. In this example, we have considered the trust neighbor who's the distance *d,* they are far from the root node u_1 is less or equal to 3. This is the reason explaining why we did not consider u_6 rating on item i_5 while calculating $\overline{r_{u_1, i_5}}$.

Table 3.4. *u_1 new trust rating profile*

set of items	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9
$\bar{r}_{u_1,i_j}$	4.05	2.0	4.86	3	2.66	-	-	2	-

We are now able to compute similarity between u_1 and other users. This calculation is done by using Pearson Correlation Coefficient (PCC), like shown in equation 3.3. We have reported in table 3.5 similarity results.

Table 3.5. *Similarity between u_1 and other users*

set of users	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9
u_1	0.83	0.98	0.94	0.94	-0.92	-	0.99	-0.97

Prediction on item i_5 for active user u_1 can then pre-generated based on table 3.5 results and equation 3.5.

$$\hat{r}_{u_1,i_5} = \frac{3*0.83+2*0.94+3*0.94+2*0.99}{0.83+0.94+0.94+0.99} = 2.48$$

The predicted value on item i_5**,** $\hat{r}_{u_1,i_5} = 2.48$ based on solely trust neighbors and $\bar{r}_{u_1,i_5} = 2.66$ based on similarity between users are less different. This is due somehow on the conditions we observed in this experiment; that is holding distance $d \leq 3$, implying not considering $\bar{r}_{u_6,i_5}$

3.3. Known Issues and Limitations

The major drawback of trust based graph algorithm is the unavailability of data. Like hybrid approaches proposed in [7] [75] [74] [38] [77] [76] the proposed algorithm is not effective when trust data is not available. In our work, we used random data to calculate the prediction for a given item (in our case prediction on item i_5 for user u_1).

The second issue raises from using random data. In fact, we did not drive experiments to test the effectiveness of our proposed algorithm. A future work will be acquiring state of the art well-known dataset having trust data or infer implicitly trust data from users' ratings. Then split this dataset in 3 parts (train, test, validation), conduct experiments and evaluate the effectiveness of graph based trust algorithm.

Third, an issue raised on how to acquire trust data. It should be noted that user-trust data presented on Table 3.2 is completely theoretical. Explicit trust data acquiring can be sometimes impossible to achieve due to for example privacy concerns (Users not wanting to share their information). A solution to this problem is to prioritize the using of implicit trust ratings. Trust data can then be acquired from the users' behavior (clicks, followed links, time spent on viewing items….). A second implicit trust data acquiring can be by inferring directly trust ratings from users given ratings on items. Research has been made in this way, O' Donavan et al. [68], Lathia N et al. [86], Hwang C. et al [87], Papagelis M. et al. [72], Shambour and Lu [88] proposed different functions shown in the following table 3.6.

Table 3.6. *Inferring implicit trust metrics*

Trust Metric	Computation function
O' Donovan and Smyth [68]	$t_{u,v} = \frac{\mid CorrectSet(v) \mid}{\mid \mathrm{Re}cSet(v) \mid}$ $Correct(r_{u,i}, r_{v,i}) \leftrightarrow P_{u,i} - r_{u,i}$
Lathia N. et al. [86]	$t_{u,v} = \frac{1}{\mid I_{u,v} \mid} \sum_{i \in I_{u,v}} (1 - \frac{\mid r_{u,i}, r_{v,i} \mid}{r_{max}})$
Hwang Chen et al. [87]	$t_{u,v} = \frac{1}{\mid I_{u,v} \mid} \sum_{i \in I_{u,v}} (1 - \frac{\mid p_{u,i}, r_{u,i} \mid}{r_{max}})$ $p_{u,i} = r_u + (\overline{r}_{v,i} - \overline{r}_v)$
Shambour and Lu [88]	$t_{u,v} = \frac{\mid I_{u,v} \mid}{\mid I_u \cup I_v \mid}(1 - \frac{1}{\mid I_{u,n} \mid} \sum_{i \in I_{u,v}} (\frac{p_{u,i} - r_{u,i}}{r_{max}})^2$
Papagelis M. et al. [72]	$t_{u,v} = \begin{cases} S_{u,v}, & \text{if } S_{u,v} \succ \theta_s \text{ , } \mid \mathrm{I}_{u,v} \mid \succ \theta_i \\ 0, & otherwise \end{cases}$ $S_{u,v} = \frac{\sum_i (\overline{r}_{u,i} - \overline{r}_u)(\overline{r}_{v,i} - \overline{r}_v)}{\sqrt{\sum_i (\overline{r}_{u,i} - \overline{r}_u)^2} \sqrt{\sum_i (\overline{r}_{v,i} - \overline{r}_v)^2}}$

Finally, we must define d (the shortest path in terms of number of edges between an active user and its given trust neighbor). Here we use the threshold $d \leq 3$, and we did not consider the active user 's trust neighbor composing its trust network having greater than 3.

4. CONCLUSION

We organized this thesis in four chapters. In the first chapter, a brief preview of recommender systems was presented. We presented the existing types of systems, their advantages, and the challenges these systems are experiencing.

In chapter 2, CF and the sparsity problem were presented. We first introduced the missing data theory and emphasize how missingness occur in datasets; then we presented in more details the sparsity problem in CF. We showed that data scarceness is one of the major bottlenecks for the effectiveness of CF in terms of accuracy of predictions, recommendations, and performance. Finally, we proposed a substantial number of methods that have been proposed by researchers to handle data sparsity in CF. These methods have shown their effectiveness but also have some drawbacks.

In chapter 3, we proposed a new method named graph based trust for coping with sparsity and cold users in CF. This method is essentially based on trust recommender systems; and have been inspired by previous work proposed by [67] [2]. One of this method major drawbacks is that, it cannot be used in the situations where additional information is unavailable.

Our objectives for future work is applying the graph based method on state of the art well-known datasets and taking into account implicit ratings as explicit trust information can be missing for example due to privacy concerns.

BIBLIOGRAPHY

[1] Jonathan L. Herlocker, Joseph A. Konstan,Al Borchers and Jonh Rield. An Algorithm Framework for Performing Collaborative Filtering. *Dept. of Computer Science and Engineering.University of Minnesota* *www.cs.umn.edu/Research/GroupLens*

[2] Guibing Guo, Jie Zhang, Daniel Thalmann. (2013). Merging trust in collaborative Filtering to alleviate dta sparsity and cold start. *Schoool of Computer Engineering, Nanyang Technological University, Singapore. Elsevier*.

[3] Meenakshi Sharma, Sandeep Mann. A Survey of Recommender Systems: Approaches and Limitations*. International Journal of Innovation in Engineering and Technology Special Issue-ICAECE-2013 ISSN:23-19-1058.*

[4] Yibo Chen, Chanle Wu,Ming Xie, and Xiaojun Guo. Solving the Sparsity Problem in Recommender Systems Using Association Retrieval. *Computer School of Wuhan University, Wuhan, Hubei, China.*

[5] Weiwei Xia, Liang He, Junzhong Gu, Keqin He. Effective Collaborative Filtering Approaches Based on Missing Data Imputation. *Department of Computer Science and Technology East China Normal University. Shanghai 200241, China.*

[6] Christian Desrosiers and George Karypis. (2008). Solving the Sparsity Problem: Collaborative Filtering via Indirect Similarities. *Department of Computer Science and Engineering. University of Minnesota USA. December 10*.

[7] Bin Li, Qiang Yang and Xiangyan Xue. (2013). Can Movies and Books Collaborate? Cross-domain Collaborative Filtering for Sparsity Reduction. *International Joint Conferences on Artificial Intelligence.*

[8] Jan Wan, Xin Wang, Yuyu Yin and Renjie Zhou. (2015). Transfer Learning in Collaborative Filtering for Sparsity Reduction Via Features Tags Learning Model. *School of Computer Science, Hangzhou Dianzi University, Hangzou China. Advanced Science and Technology Letters Vol. 81 (CST 2015), pp. 56-60. ISSN:2287-1233 ASTL SERCSC.*

[9] Weike Pan, Evan W. and Qiang Yang. (2012). Transfer Learning in Collaborative Filtering with Uncertain Ratings. *In Proceedings of the 26th AAAI Conference on Artificial Intelligence, 662-668.*

[10] Jan Wan, Xin Wang, Yuyu Yin and Renjie Zhou. Transfer learning to predict missing ratings via heterogeneous user feedbacks. *In Proceedings of the 22nd International Joint Conference on Artificial Intelligence, 23182323*.

[11] Weike Pan, S.J. and Qiang Yang. (2010). A survey on transfer learning*. IEEE Transactions on Knowledge and Data Engineering 22(10): 1345-1359.*

[12] Weike Pan, Xiang E., Liu N. and Qiang Yang. Transfer learning in collaborative filtering for sparsity reduction. *In Proceedings of the 24th AAAI Conference on Artificial Intelligence, 230-235.*

[13] Singh A. P. and Gordon G. J. Relational learning via collective matrix factorization*. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'08, 650-658. New York, NY, USA: ACM.*

[14] Benjamin M. Marlin, Richard S. Zemel, Sam Roweis and Malcolm Slaney (2007). Collaborative Filtering and the Missing at Random Assumption, UAI.

[15] Bin Li, Qiang Yang and Xiangyang Xue. (2009). Transfer Learning for Collaborative Filtering via a Rating-Matrix Generative Model. *Appearing in Proceedings of the 26th International Conference on Machine Learning, 617-624, Montreal, Canada*.

[16] Amanda Spink, Michael Zimmer(Eds). (2008). Web Search Multidisciplinary Perspectives. *Springer ISBN:978-3-540-75828-0.*

[17] Gerard Salton, Christian Buckley. (1988). Terms Weighting approaches in Automatic Text Retrieval. *Information Processing and Management, 24(5): 513-523.*

[18] Pattie Maes. (1994). Agents that Reduce Work and Information Overload. *Communications of the ACM, 37(7): 30-40 July 1994.*

[19] James A. Wise, James J. Thomas, Kelly Pennock, David Lantrip, Marc, Anne Schur and Vern Crowin. Pottie. (1995). Visualizing the non-visual: Spatial analysis and

interaction with information from text documents. *In IEEE information visualization '95, pages 51-58. IEEE Computer Soc. Press, 30-31 October 1995.*

[20] Pasquale Lops, Marco de Gemmis and Giovanni Semeraro. (2011). Recommender Systems Handbook Chapter 3: Content-based Recommender Systems: State of the Art and Trends. F. Ricci et al. *Editions. DOI 10.1007/978-0-387-85820-3 3, Springer Science+Business Media, LLC 2011*.

[21] Prem Melville and Vikas Sindhwani. Recommenders systems. *Watson Research Center, Yorktown Heights, NY 10598.*

[22] Zhou and T. Luo. A Novel approach to solve the sparsity Problem in Collaborative Filtering. Watson Research Center, Yorktown Heights, NY 10598.

[23] G. Guo, J.Zhang and N. Yorke-Smith. (2013). A novel Bayesian similarity measure for recommender systems. *In Proceedings of the 23th International Joint Conference on Artificial Intelligence.*

[24] Gediminas Adomavicius and Alexander Tuzhilin. (2005). Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering, VOL. 17, No. 6, June 2005.*

[25] Mingjuan Zhou. (2010). Book Recommendation Based on Web Social Network. *International Conference on Artificial Intelligence and Education. Hangzhou. 136139.*

[26] Zan Huang, Hsinchun Chen, et al. (2004). Applying Association Retrieval Techniques to Alleviate the sparsity problem in Collaborative Filtering. *ACM Transactions on Information Systems. Vol. 22, No. 1, January 2004, 116-142.*

[27] Joseph L. Schafer and John W. Graham. (2002*). Missing Data: Our View of the State of the Art. Psychological Methods. Vol. 7, No. 2, 147–177. Copyright 2002 by the American Psychological Association, Inc 1082-989X/02/$5.00 DOI: 10.1037//1082-989X.7.2.147.*

[28] Rubin. (1976). Inference and missing data. *Biometrika 63(3):581–592. DOI:10.1093/biomet/63.3.581.*

[29] Rubin. (2002). Statistical analysis with missing data*. 2nd edition. Wiley, New York, September 2002.*

[30] F. Fouss, J.M. Renders, A. Pirotte and M. Saerens. (2007). Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Transactions on Knowledge and Data Engineering, 19(3): 355-369.*

[31] M. Gori and A. Pucci. (2007). Itemrank: A random-walk based scoring algorithm for recommender engines. *In Proceedings of the 2007 IJCAI conference, pages 2766-2771.*

[32] H. Luo, C. Niu, R. Shen and C. Ulrich. (2008). A collaborative filtering framework based on both local user similarity and global user similarity*. Machine Learning, 72(3): 231-245*.

[33] H. Kaya and Alpaslan FN. (2010). Using Social Networks to Solve Data Sparsity Problem in One-Class Collaborative Filtering. *In Proceedings of the 7th International Conference on Information Technology: New Generations. Las Vegas. 249-252.*

[34] X. Gui Rong, L. Chenxi, Y. Qiang, X. Wensi, Z. Hua Jun, Y. Yong, C. Zheng. (2005) Scalable collaborative filtering using cluster-based smoothing. *In Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 114–121.*

[35] Chen. DanEr. (2009). The collaborative filtering recommendation algorithm based on BP neural networks*. In Proceedings of the International Symposium on Intelligent Ubiquitous Computing and Education, pp. 234–236.*

[36] L. Nan, L. Chunping. (2009). Zero-sum reward, and punishment collaborative filtering recommendation algorithm. *In Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence Agent Technology, pp.548–55.*

[37] Lyle D. Broemeling. Bayesian Methods for Repeated Measures. *A Chapman & Hall Book CRC Press Taylor & Francis Group. International Standard Book Number: 13:978-1-4822-4820-3 (ebook PDF).*

[38] Badrul M.Sarwar, Joseph A.Konstan, Al Borchers,Jon Herlocker, Brad Miller, and John Rieldl. (1998). Using Filtering Agents to Improve Prediction Quality in the

GroupLens Research Collaborative Filtering System. *GroupLens Research Project, University of Minnesota Copyright ACM 1998 1-58113-009-0/98/11.*

[39] Goldberg, D.Nichols, D.Oki et al. (1992). Using Collaborative Filtering to Weave an Information Tapestry. *ACM - Special issue on information filtering Volume 35 Issue 12, Dec. 1992 Pages 61-70.*

[40] Aulia Rahmawati, A. T. Wibowo, G. S. Wulandari. (2015). Cluster-Smoothed with Random Neighbor Selection for Collaborative Filtering. *IEEE International Conference on Computer, Control, Informatics, and Its Applications*.

[41] Rich E. (1979). User modeling via stereotypes. *Cognitive Science*, *3(4):329-354.*

[42] Konstan J. A, Miller, B. N, Maltz, J. L Gordon et al. (1997). GroupLens: Applying Collaborative Filtering to Usenet News. *CACM. 40(3), March 1997.*

[43] Resnick, P.Iacovou, N. Suchak, M. Bergstrom and Rield. (1994). GroupLens: An Open Architecture for Collaborative Filtering of Netnews. *In Proceedings of CSCW '94. Chapel Hill, NC.*

[44] Hill, W. Stead, L. Rosenstein, M. Furnas. Recommending and Evaluating Choices in a Virtual Community of Use. *Proceedings of CHI '95.*

[45] Shardanand, U. and Maes, P.(1995). Social Information Filtering: Algorithms for Automating" Word of Mouth". *In Proceedings of the CHI '95. Denver, CO. May 1995.*

[46] Linden G, Smith B, et al. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing. 7(1): 76-80.*

[47] H.J. Ahn. (2008). A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem*. Inform. Sci. 178 (1) 37–51.*

[48] M. Jamali, M. Ester. (2009). TrustWalker: a random walk model for combining trust-based and item-based recommendation. *In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 397–406.*

[49] G. Koutrica, B. Bercovitz, H. Garcia. (2009). FlexRecs: expressing and combining flexible recommendations. *In Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 745–758.*

[50] J. Bobadilla, F. Ortega, A. Hernando, J. Bernal. (2011). A collaborative filtering approach to mitigate the new user cold start problem. *Knowledge-Based Syst. 26-225–238.*

[51] F. Cacheda, V. Carneiro, D. Fern´andez, V. Formoso. (2011). Comparison of collaborative filtering algorithms: limitations of current techniques and proposals for scalable, high-performance recommender system. *ACM Trans. Web 5 (1) -1–33.*

[52] J. Bobadilla, F. Ortega, A. Hernando. (2012). A collaborative filtering similarity measure based on singularities Inform. *Process. Manage. 48-204–217.*

[53] Goldberg K, Roceder T, et al. (2001). Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval. 4(2):133-151.*

[54] Yang Yuije, Zhang Zhijun and Duan Xintao. (2014). Cliques-based Data Smoothing Approach for Solving Data Sparsity in Collaborative Filtering. *Telkomnika Indonesia Journal of Electrical Engineering. Vol. 12, No. 8, August 2014, pp. 6324 6331 DOI: 10.11591/telkomnika. v12i8.4617.*

[55] Billsus, D. Pazzani, M. J. (1998). Learning Collaborative Information Filters. *In Proceedings of the 15th International Conference on Machine Learning, 46-54.*

[56] Schein, A. I., Popescul et al. (2002). Methods and metrics for coldstart recommendations. *In Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval SIGIR. (Tampere, Finland), 253-260.*

[57] N. Zheng, L. Qiudan, L. Shengcai, Z. Leiming. (2010). Which photo groups should I choose? A comparative study of recommendation algorithms in Flickr. *J. Inform. Sci. 36 (6) (2010) 732–750.*

[58] E. Brynjolfsson, Y.J. Hu, M.D. Smith. (2003). Consumer surplus in the digital economy: estimating the value of increased product variety at online booksellers Manage. *Sci. 49 (11) (2003) 1580–1596.*

[59] B. Shumeet, R. Seth, D. Sivakumar, Y. Jing, J. Yagnik, S. Kumar, D. Ravichandran, M. Aly. (2008). Video suggestion and discovery for YouTube: taking random walks through the view graph. *International Conference on World Wide Web, pp. 895–904.*

[60] J. S. Breese, D. Heckerman, and C. Kadie. (1998). Empirical analysis of predictive algorithms for collaborative filtering. *In Proceedings of the 14th annual conference on uncertainty in artificial intelligence. Pages 43–52. Morgan Kaufmann.*

[61] Sarwar BM, Karypis G, Konstan J.A., Riedl J. (2000). Application of Dimensionality Reduction in Recommender System - A Case Study. *ACM WebKDD Workshop. Report number: 0704-0188.*

[62] Haifeng Liu, Zheng Hu, Ahmad Mian, Hui Tian and Xuzhen Zhu. (2014). A new user similarity model to improve the accuracy of collaborative filtering*. Knowledge-Based Systems 56-156-166.*

[63] Josang Audun, D.K. Walter Quattrochicchi. (2011). *Taste and trust, in: Trust Management V, pp. 312-322.*

[64] J. Golbeck (2005). Computing and Applying Trust in Web-Based Social Networks*. PhD. Thesis.*

[65] I.Guy,I.Ronen, E.Wilcox. (2009). Do you know? : recommending people to invite into your social network. *In Proceedings of the 14th International Conference on Intelligent User Interfaces, pp. 77-86.*

[66] I. konstas, V.Stathopoulos, J.Jose. (2009). On social networks and collaborative recommendation. *In Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information retrieval, pp. 195-202.*

[67] P. Massa, P. Avesani. (2007). Trust-aware recommender systems. *In Proceedings of the 2007 ACM Conference on Recommender Systems, pp. 203-210.*

[68] J. O'Donovan, B. Smyth. (2005). *Trust in recommender systems*. *In Proceedings of the 10th International Conference on Intelligent User Interfaces, pp. 167-174.*

[69] A. Seth, J. Zhang, R. Cohen. (2010). Bayesian credibility modeling for personalized recommendation in participatory media. *In Proceedings of the International Conference on User Modeling Adaptation and Personalization, pp. 279-290.*

[70] Q. Yuan, S. Zhao, L.Chen, Y. Liu et al. (2009). Augmenting collaborative recommender by fusing explicit social relationships. *In Proceedings of Workshop on Recommender Systems and the Social Web, pp. 49-56.*

[71] C. Ziegler, G. Lausen. (2004). Analyzing correlation between trust and user similarity in online communities. *In Trust Management, pp. 251-265.*

[72] Manos Papagelis, Dimitris Plexousakis and Themistoklis Kutsuras. (2005). Alleviating the Sparsity Problem of Collaborative Filtering Using Trust Inferences. *(Eds.): iTrust 2005, LNCS 3477, pp. 224 – 239, 2005. Springer Verlag Berlin Heidelberg.*

[73] Song Jie Gong, Hong Wu Ye, YaE Dai. (2009). Combining Singular Value Decomposition and Item-based Recommender in Collaborative Filtering. *International Workshop on Knowledge Discovery and Data Mining. Moscow. 769772.*

[74] A. Szwabe, M. Ciesielczyk, T. Janasiewicz. (2011). Semantically enhanced collaborative filtering based on RSVD. *In Proceedings of the International Conference on Computational Collective Intelligence, pp. 10–1.*

[75] M. J. Pazzani. (1999). A framework for collaborative filtering, content-based and demographic filtering. *Artificial Intelligence Review, 13(5-6): 393-408.*

[76] Good, N. Schafer, J. et al. (1999). Combining collaborative filtering with personal agents for better recommendations*. In Proceedings of the 16th National Conference on Artificial Intelligence, 439-446.*

[77] Huang Z., Chung W. et al. (2002). A graph-based recommender system for digital library. *In Proceedings of the 2nd ACM/IEEE-CS Joint Conference on Digital Libraries (Portland, Ore). ACM, New York, 65-73.*

[78] M. Balabanovic, Y. Shoham. (1997). Fab: Content-based, Collaborative Recommendation. *Communication of the ACM, Mar. 40(3): 66-72*.

[79] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes and M. Sartin. (1999). Combining Content-based and Collaborative Filters in an Online Newspaper. *In Proceedings of ACM SIGIR Workshop on Recommender, August 1999.*

[80] A. Popescul, L. H Ungar, D. M. Pennock and S. Lawrence. (2001). Probabilistic Models for Unified Collaborative and Content-based Recommendation in Sparse-data Environments. *In 17th Conference on Uncertainty in Artificial Intelligence.*

[81] Miller B., Riedl. J and Konstan J. (1997). Experiences with GroupLens: Making Usenet Useful Again. *In Proceedings of the 1997 Usenix Technical Conference.*

[82] Resnick P. and Varian H. R. (1997). Recommender systems. *CACM. 40(3), pp. 56-58, March 1997.*

[83] Rucker J. and Polano M. J. (1997). Siteseer: Personalized Navigation for the Web. *CACM. 40(3), March 1997.*

[84] Terveen L., Hill W., Amento B., McDonald D. and Creter J. (1997). PHOAKS: A System for Sharing Recommendations. *CACM. 40(3), pp. 59-62, March 1997.*

[85] M. Grcar, D. Mladenic, B. Fortuna, M. Grobelnik. (2006). Data sparsity issues in the collaborative filtering framework. *Advan. Web Min. Web Usage Anal.*

[86] Lathia N., Hailes, S and Capra, L. (2008). Trust-Based Collaborative Filtering. *Trust Management II, IFIP Advances in Information and Communication Technology. 263/2008.*

[87] Hwang C. et al. (2007). Using trust in collaborative filtering recommendation. *In New trends in Applied Artificial Intelligence.*

[88] Shambour, Q. and Lu, A.J. (2012). Trust-semantic fusion based recommendation approach for e-business applications. *Decision Support Systems 54. (768-780).*