Faculty of Computer Science and Information Technology
Sudan University of Science and Technology

An Enhancement of Markov Chain-based Recommendation Systems

تحسين نظم التوصية المبنية علي سلسلة ماركوف

A thesis submitted in fulfillment of the requirements
for the award of the degree of Doctor of Philosophy
(Computer Science)

Ahmed Adam Ahmed Abdelrehman

Supervisor
Prof. Dr. Naomie Binti Salim

Apr. 2016

بسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

أَلَمْ نَشْرَحْ لَكَ صَدْرَكَ {1} وَوَضَعْنَا عَنْكَ وِزْرَكَ {2} الَّذِي أَنْقَضَ ظَهْرَكَ {3}
وَرَفَعْنَا لَكَ ذِكْرَكَ {4} فَإِنَّ مَعَ الْعُسْرِ يُسْرًا {5} إِنَّ مَعَ الْعُسْرِ يُسْرًا {6} فَإِذَا
فَرَغْتَ فَانْصَبْ {7} وَإِلَى رَبِّكَ فَارْغَبْ {8}

صدق الله العظيم

سورة الشرح آية 1 − 8

To my parents, my wife and daughters, my brothers and sisters, my teachers, and all my friends, for them I have been enjoying my life.

Acknowledgement

First of all, we must thank Allah for every think in our live then I would like to thank my supervisor Professor Dr. Naomie Binti Salim, who shares with me a lot of researches and references. Her valuable advices serve me to keep the right track in my PhD research. Thanks for her important supports throughout this work.

I would like to thank Professor Dr Izzeldin Osman who encourages me to start my PhD research after a quite long stopping. And I would like to thank all my teachers for numerous discussions and lectures on related topics that helped me improve my knowledge in the area.

I am indebted of Faculty of Computer Science and Information Technology, Sudan University of Science and Technology for full support, co-operation, and help to complete this work.

Finally, I would like to thank my parents, my wife and daughters, my brothers and sisters, and all my friends, without their encouragements it would have been impossible for me to complete this work.

Abstract

This thesis aims to design a new recommendation system to be used by web application to recommend interesting items to users. Recently, many web applications are created and published in the internet to allow their users to access millions of items, and the number of users as well as the number of items is increasing exponentially. These web applications use recommendation techniques that are based on users' preferences for items to recommend a few interesting items to users.

Collaborative filtering (CF) is currently the most successfully used recommendation system; it is based on the similarities between users and the similarities between items. However, users' opinions and items' popularities vary with time. These variations decrease the recommendation accuracy. On the other hand, many researchers investigate ways of using Morkov model in recommendation systems; however, the time factor can be better used in new techniques.

We introduce a new recommendation system, then we enhance it using the time factor and friends feature. The contributions in this thesis are as follows:

- We propose a new technique entitled the basic Markov Chain Recommendation System MCRS that is based on Morkov model, and the time factor.
- We enhance the basic MCRS using items' popularities in general.
- In the second enhancement, we use items' popularities in the last period of time.
- In the last enhancement, the basic MCRS is weighted by friends weights of items.

We compare our new technique with the conventional CF recommendation system for the evaluation. We conduct the experiments using dataset from MovieLens and LastFm. The evaluation is done by using precision-recall, accuracy, area under ROC curve, and mean absolute error.

The result illustrates that the basic MCRS outperforms the conventional CF recommendation system, and the time factor affects positively or negatively in the recommendation. All the enhancements outperform the basic MCRS. The friend weights MCRS outperforms the basic MCRS and its enhancements.

المستخلص

يهدف هذا البحث الي تصميم نظام اقتراحات جديد ليستخدم بواسطة تطبيقات الوب التي صممت ونشرت لتسمح لمستخدميها أن تتعامل مع الملايين من البنود والخدمات. وقد تضاعف عدد مستخدمي الانترنت بمضاعفة عدد هذه التطبيقات التي تستعمل نظم الاقتراحات لتسهيل استخدامها. يتم تصميم نظم الاقتراحات بناءً علي تعاملات المستخدمين مع البنود والخدمات التي تقدمها هذه التطبيقات. وهي تستعمل لاقتراح ما يناسب المستخدم من البنود والخدمات وبطريقة آنية وآلية وبدون استخدام محركات البحث والاستعلامات. يعتبر نظام التصفية التعاونية من انجح النظم المستخدمة لاقتراح البنود المناسبة للمستخدم في لحظة استخدام التطبيق. يعتمد نظام التصفية التعاونية علي تشابه البنود التي ترغب مجموعة متشابهة من المستخدمين في التعامل معها. ولكن اختلاف اذواق المستخدمين وتغير شعبية البنود مع تقدم الزمن يؤثران في دقة نظم الاقتراحات. ومن ناحية اخري, فقد استخدم كثير من الباحثين نمازج ماركوف في تصميم نظم الاقتراحات وبعضهم استخدم عامل الزمن ولكن يمكن أن يستخدم عامل الزمن بطرق افضل من ذلك. في هذه الأطروحة تم اقتراح تقنية جديدة مبنية علي سلسلة ماركوف مع الوضع في الاعتبار عامل الزمن. تمت المقارنة بين نظامنا المقترح ونظام التصفية التعاونية التقليدي حيث استخدمت قواعد بيانات من موفي لينث ولاست إف إم. وتمت المقارنة باستخدام الدقة والتذكير ومتوسط الخطأ المطلق. واثبتت النتيجة تفوق نظامنا المقترح علي نظام التصفية التعاونية التقليدي. تم تحسين النظام المقترح باستخدام شعبية البنود بصورة عامة وشعبية البنود في فترة تقديم المقترحات. اثبتت التجارب تفوق النظم المحسنة علي نظام ماركوف المقترح. وكذلك اضفنا تحسينا آخر باستخدام شعبية البنود بالنسبة لأصدقاء المستخدم وقد تفوق التحسين الجديد علي النظام المقترح وتحسيناته.

Table of Contents

List of tables

List of figures

List of Abbreviations

RSs	Recommendation systems
CBRSs	Content-based Recommendation Systems
CFRSs	Collaborative Filtering Recommendation Systems
MDP	Markov Decision Process
KBRS	Knowledge based recommendation systems
CF	Collaborative filtering
MF	Matrix Factorization
MC	Markov Chain
FPMC	Factorized Personalized Markov Chain
MCRS	Markov Chain Recommendation System
PASS	Probabilities of accessing items by the same user at the same time
MAE	Mean Absolute Error
ROC	Receiving operating Characteristic
AUR	Area Under ROC curve

List Publications

No.	Article	Journal	date
1.	Using trend analysis and social media Features to enhance recommendation Systems: a systematic literature review.	Journal of Theoretical and Applied Information Technology.	30th September 2013. Vol. 55 No.3.
2.	Markov Chain Recommendation Systems MCRS.	ISSN 2394-7314 **International Journal of Novel Research in Computer Science and Software Engineering.**	Vol. 3, Issue 1, pp: (11-26), Month: January-April 2016, Available at: **www.noveltyjournals.com**
3.	The enhancement of Markov Chain Recommendation System (MCRS) using items' popularity.	ISSN 2394-7314 **International Journal of Novel Research in Computer Science and Software Engineering .**	Vol. 3, Issue 1, pp: (1-10), Month: January-April 2016, Available at: **www.noveltyjournals.com**

Chapter 1

INTRODUCTION

1.1. Overview

Nowadays, the internet is become widely used by many websites to provide many services and items to their users; services can be provided by e-marketing, e-government, online news, and etc.; items can be books, movies, photos, and games[1]. These websites are widely visited by millions of users to access millions of items e.g. users can view a movie, and they can mark a photo as like[2]. It has become an easy task to produce and upload new items to these websites, but it is very difficult to retrieve the actually needed items. Websites as well as their users face the problem of information overload [3]. For example, if the user U of the website W views a list of movies, then what is the next subset of movies he will view out of the other millions? Recommendation systems (RSs) are software that can be used to address this problem. RSs have been used by many websites, to recommend items to their users [4]. These RSs are based on items' descriptions and contents, and users' opinions and their preferences for items. According to this information about users and items, RSs techniques are divided into several categories e.g. Content-based recommendation systems CBRS, Collaborative filtering recommendation systems CFRS, Knowledge-base recommendation systems KBRS, Hybrid recommendation systems Hyb-RSs, and etc. [5] [6].

Content-based recommendation systems (CBRSs) can be used to generate lists of recommended items to users; using items' descriptions or properties, and users' profiles[7]. Items are normally stored in database tables that consist of fields i.e. attributes or columns, and listed in records that contain their properties. Users of websites access these items to generate users' profiles that contain users' preferences for items. Users' profiles are created and updated automatically, with respect to users' activities on their interesting items; they contain properties of items that have been used by CBRSs to generate lists of recommended items to users. These techniques have been used in different domains (e.g. online news websites [8] and e-commerce [9]). This, thesis does not consider descriptions of items in the recommendation processes, instead of that, it only looks at users' preferences for items and the time factor to design new recommendation systems, that can be used to recommend items to users.

Collaborative filtering recommendation systems (CFRSs) are widely used techniques in the last decades [10]–[14]. They have been used to recommend items to the active user, with respect to his similar users’ ratings on that item or similar items. Users’ ratings can be generated from their activities and the interactions with web applications that have been using CFRSs. Users' ratings for items can be explicit or implicit [15]–[17]. Explicit ratings are provided by users’ responses to queries that are given by the system. Their main drawback, of the explicit ratings, is the need of users’ responsibility, with users that do not respond to queries. Implicit ratings do not need users’ responses; they can be taken from users’ interactions on web applications such as, viewing movies, or clicking on links to choose their items of interest.

Collaborative Filtering algorithms can be either Memory-based or Model-based techniques.

Memory-based CF techniques are based on the history the active user’s preferences for items that can include the activities of his similar users in the past. Memory-based techniques use the information that stored in databases, such as ratings of users for items, to generate recommendations to users [18]. This thesis does not discuss or use memory based CF techniques.

Model-based CF technique uses training datasets of users' preferences to train a model that can be used for ratings prediction and items recommendation. Well-known model-based techniques include clustering and machine learning [19], [20]. We propose a new model based techniques; more details about them will be given later.

CFRSs have been studied by many researchers [21]–[24]. Ding et al. [25] and Lee et al. [26] present time weighted collaborative filtering. They introduced the time function that weights ratings of users for items according to time periods. Their solutions weight all items using the same value in the same period of time. In their solutions, old events have less values while most recent events have the highest values. However, items popularities have different variations. Many items may have the same popularities in all periods of time; another group of items may have low popularities at the beginning and they become more popular later and vice versa. Therefore, our new propose technique is based on users' preferences for items and the time factor that are used to calculate its main parts: (1) The probabilities of accessing items by users in the same period of time. (2) The popularities of items in general. (3) The popularities of items in a period of time. (4) The popularities of items that have been accessed by the active user's friends. Our proposed solution emerge the time factor and users' ratings for items to calculate the probabilities of accessing all items with any item.

Many researchers study context-aware recommendation systems [27], [28], [28]. The information about users, items, or services is called their contexts. This information contains descriptions of users that can be used to play more significant role in recommendation processes. It can be used to add the trust factor, instead of only using ratings of users for items. These descriptions can be used to cluster items e.g. the interesting items that can be accessed by users at night may be different from items of interest to him in the morning [29]. On holidays users may need to play online games, but on normal days they might follow their friends in social media. The recommender systems that utilize users environment information are called context-aware recommender systems [24], [27], [29]–[35]. Context aware recommendation systems can be pre-filtered or post-filtered. However, these techniques separate users' preferences from the times of users' preferences because they recommend items to users then they filter these recommendations using users' context, and vice versa, our proposed solutions contrast emerge users' preferences and their events time to generate users' ratings for items.

Markov chain model has been used in recommendations system. Shani et al. [36] proposed an MDP-Based Recommender System (Markov Decision Process). Their technique is based on frequents sequences of items that are followed by an item. They consider sequences of k items; however, many users can access less than k items and others can access more. The main drawback of MDP-based recommendation systems is the limitation of using k sequence of items while users can access more or less than k items. This limitation leads to an inaccurate result, because they use small sequence of items. Mainly, they use k=5; and the system is become inapplicable if they take a big sequence of items. Rendle et al. [37] present a method that bringing together both matrix factorization (MF) and Markov chains (MC). Their method is based on personalized transition graphs over underlying Markov chains. In their solution, a transition cube is consisting of transition matrixes that generated for any user. Empirically, the FPMC model outperforms both the matrix factorization and Markov chain MC model.

The mentioned Markov chain solutions are based on frequents sequences of items that are followed by an item. We do not use a sequence of items. Instead, our proposed techniques are based on sets of items that accessed by the same user in the same period of time, and they generate one transition matrix for all users; it is not a transition cube that consist of users' transition matrixes. More details will be discussed later in this thesis.

The rest of this chapter gives the problem background in Section 1.2. In Section 1.3, we discuss the research questions studied in the thesis. The thesis hypotheses are listed in Section 1.4. Section 1.5 shows the research philosophy. The research objectives are given in Section 1.6. Section 1.7 identifies the research scope. The thesis contributions are listed in Section 1.8. Then the thesis outlines are given in Section 1.9.

1.2. Problem background

Collaborative filtering recommendation systems are based on users' similarities [10], [15], [38], [39], and they are used to recommend items to any user according to his similar users' activities. If some users have accessed the same subset of items then they are similar to each other. Example: if some users have viewed the same subset of movies then they are similar and have the same opinions. On the other hand, social media feature can be used as trust factor to link group of users that refer to same opinions. The traditional users' ratings are immerged with their social activities [40]. However, users' activities are related on time since the events happen at specific points of time [41]. This section investigates users' similarities, Social media features, and the time factor; to identify the problem background of the thesis.

Users' similarities:

The problem of information overload is raised, as more websites are published on the internet. Thus, websites have been using recommendation systems to ease their usage and to introduce interesting services to their users. Many of these recommendation systems are based on users' similarities [42], [43]. These similarities between users or items can be calculated using different techniques e.g. vector cosine similarities and Pearson Correlation Coefficient techniques. Kaushik and Tomar [44] prove that vector cosine similarity function is the most accurate method for generating recommendation. In these techniques, however, the similarities between users' or the similarities between items are based on users' opinions that vary with time. Consider, we have two users that rate for movies in a long time interval e.g. two years. Then there are three situations, as follows:

- First: The two users have rated the same movies by the same rating.
- Second: They only share the ratings for a subset of movies.
- Third: They don't share any rating for any subset of movies.

According to the first situation, the users are 100% similar to each other. In the second situation, they are partially similar. In the third situation, they are not similar because the similarity calculation is based on users' preferences for items. However, users' opinions and items' popularities vary with time. In the first mentioned situation, one of the two users can rate movies in the first year; while the other user can rate for the same items in second year. This means:

- The two users are similar to each other in the long term, two years.
- They are not similar to each other in short terms (e.g. the first half of the first year).
- They are partially similar to each other, if we take their ratings in one year; the second half of the first year and the first half of the second year.

These situations lead to inaccurate similarities between users; Hence, the recommendation systems that are based on users' similarities might be inaccurate too. In addition, the following limitations apply [45]:

- Cold Start: CF systems needs initial data for any user to make exact recommendations to him.
- Scalability: The amount of information is huge, at the same time CF recommendation makes recommendation according to only few data.
- Sparsity: The number of items is huge and only a small subset of the entire database is rated by most of the users.

Thus, new ways are needed to use users' preferences for items in recommendation systems. The new techniques can use the variations of users' opinions and items' popularities with time.

Social media Features:

Recently, many social media websites are published on the internet, and the number of users of social media websites is exponentially increasing as well as the number of the provided items and services by these websites to their users. The amount of information is duplicated every day, but ways of using this information in recommendation systems remain a big challenge.

Boyd and Ellison [46] study ”Social Network Sites Definition, History and

Scholarship”, and they introduce a background about social media. Boyd and Ellison’s study has become an essential reference in the field. Yu and Kak [47] gave ”A Survey of Prediction Using Social Media”, and they introduce three main issues that must be taken in consideration if we need to predict trends from social media:

- The event must be done by a human
- The distribution of people in social media must be similar to the real world.
- The event must be easy for the public to access items and services, that are provided by their websites.

Many researchers study ways of using social media features in recommendation systems [40], [48]–[52]. Many of them use friends and tags features. Sun et al. [48] use friends feature in collaborative filtering on social networks; they use a novel approach in measuring the trust factors using data-mining techniques. Friends and tags features are used on social networks and collaborative recommendation [49] [50]. Tang et al. [40] introduce a social recommendation CF model and classify social recommender systems into two categories: memory based social recommender systems, and model based social recommender systems.

However, users' activities depend on time. Friends' preferences for items can be used with respect to the time factor. Also, the previous studies use friends and tags feature in collaborative filtering technique that based on users' similatities. We use friends feature to enhance **our new proposed Markov Chain recommendation systems**. More details about this issue will be discussed later in this thesis.

Go et al. [51] consider users’ online media sharing activities in recommendation system. They found that taking users’ preferences from file sharing enhance the recommendations. Lobzhanidze et al. [52] present an empirical study in mainstream and social media using real-world data. They help to enhance video recommendation and prove that mainstream media help the trend detection more than social media. The sharing, mainstream features, and video recommendation are not used in this work.

It is clear that, social media websites can be used to enhance recommendation systems. They provide a great opportunity to design new techniques to face the incremental increasing number of users and items.

The time factor:

Many researchers study ways of using the time factor in recommendation systems. Kupavskii et al. [53] study ways of prediction of retweet cascade size over time. Wang et al. [54] introduce 'Temporal Summaries' that order personal events according to variant periods of time. Campos et al. [32] prove that using an information near to the recommendation time improves recommendation results, and reduce the problem of information overload. Ding and Li [55] propose the time function f(t) that increasing with time t, and the values of the time weights lies in the range (0,1). While old ratings hold smaller weights, the recent users preferences have higher weights. All these solutions generate a time function that independent of events times.

Lee and Park [56] propose a recommender system to be used by mobile with respect to users' purchase time and launch time of the products to increase the recommendation accuracy. The old products are less interesting tocustomers, while the new purchases increase the popularities of items. By using these facilities, they recommend short list of the products that are suitable with mobile e-commerce. However, the launch time of a movie might not affect positively or negatively on users interesting for that movie.

It's clear that the time factor effects positively in recommendation processes and the trend analysis can be used to enhance the Recommendation Systems.

However, these studies do not emerge users' preferences and the events time. We propose a new solution that based on linking any event with its instant time i.e. it is based on accessing items by any user in specific period of time. We use the feature "accessing items by any user in specific period of time" to generate the probabilities of accessing all items with any item in specific period of time by the same user to generate the general weights of items, the period weights of items, and weights of items that accessed by friends. We use Markov chain model to recommend items to users. More details about our new proposed solution will be discussed later in this thesis.

Motivation of new recommendation systems:

We can summarise that, many websites are published in the internet to provide to their users the interaction among each others; and the amount of information is duplicated. The increasing number of users and items leads to the problem of information overload. Many recommendation systems are used to cope with this problem. The researchers study and enhance the conventional recommendation system techniques. However, there are a great opportunity and many factors that can be used to design a new recommendation system.

This thesis **introduces a new recommendation systems** based on:

- Users' preferences for items (e.g. users' rating for items).
- The time factor (time of users' preferences).
- Social media feature (e.g. friends feature).

The new technique is designed using Markov Model. More details about the new proposed recommendation system are given in Chapter 3.

1.3. Questions Studied in The Thesis

- ### General Question:

How can the time factor and social media features be used to design a new recommendation system?

- **Sub-questions:**
    - How can we use the time of users' preferences for items in recommendation systems?
    - How can we embed users' preferences for items and the time before they be used in recommendation systems?
    - How can trends be predicted and used in recommendation systems?
    - How can Markov Model be used to design a new recommendation system?
    - How can friends feature be used to enhance the new Markov chain based recommendation system?

1.4. Research Hypothesis

- The time of users' preferences for items in web sites can be used to design a new recommendation system.
- First, users' preferences for items and the time can be embedded then later they can be used in recommendation systems.
- Trends can be predicted from users' preferences for items.
- Trend analysis can be used to design a new recommendation system.
- Markov Chain can be used to design a new recommendation system.
- Friends feature can be used to enhance the new Markov chain based recommendation system.

1.5. Research Philosophy

The internet has become an essential communication tool that links users among each others. Many websites are published in the internet to provide their users with a huge amount of information. Everyone is using the internet, or he is going to use it.

The problem of information overload is continuously increasing since the numbers of websites as well as the number of their users are increasing exponentially. Recommendation systems are used by many of these websites to cope with the problem of information overload. However, this area needs more studies; moreover, it needs new techniques to be used in recommendation systems.

Many recommendation systems are based on users' or items' similarities, and many techniques are used to calculate these similarities. Similarities in Collaborative Filtering techniques are based on users' preferences; however, users' opinions vary with time, and these variations can positively or negatively affect in the similarity calculations accuracy.

Collaborative filtering techniques are based on users' preferences for items at different points of time; the time factor can play an essential role in recommendation processes; therefore, many researchers study ways of using the time factor in recommendation systems, but the area needs more studies.

In 1907, A. A. Markov began the study of an important new type of chance process [57], [58], known as Markov chain. Markov Chain model is used in recommendation systems [15], [36], [59], [60], and it can be used to design a new technique using different ways to recommend items to users.

1.6. Research Objectives

1.6.1. The research aim

- To design a new recommendation technique based on users' preferences for items and the time factor.

1.6.2. The research objectives

- To find new ways that can be used to exploit the time factor in the recommendation processes.

- To use Markov model to design a new recommendation system.

- To use friends feature to enhance the newly designed recommendation technique.

1.7. Research Scope

The scope of the proposed solution can be identified as follows:

- Users' preferences for items will be used to design a new recommendation systems

- The time of users' preferences for items will be used in the newly designed recommendation system.

- The variation of users' opinions and items popularities will be used to enhance the new designed recommendation system.

- Friends feature will be used to enhance the new designed recommendation system.

- Datasets from Movielens and LastFm websites will be used to evaluate the new designed recommendation system and its enhancements.

1.8. Major Contributions

In this thesis, we introduce Markov Chain Recommendation system; and we enhance the new technique using items' popularities in general, items' popularities in a period of time, and friends feature.

We can group the thesis contributions as follows:

1.8.1. Designing a new recommendation system, based on Markov Chain model.

Shani et al. [36] use Markov Chain model in the recommendation processes; they use the feature of accessing an item that follows **a sequence** of items.

We propose a new recommendation system using users' preferences for items. Our proposed solution uses Markov model and the time factor in recommendation processes. The main feature that used in designing the new technique is ***“accessing items by the same user in the same session or period"***; accessing items **in sequences** is not considered in this work. This feature is used to design Markov transition matrix, and Markov initial vector is calculated using the previous accessed items by the active user. The result is the vector product of the initial vector and the transition matrix. Items of the highest probabilities of this result can be listed to the active user as recommendation. The thesis contributions in this subsection are:

- **Contribution one:** We propose a new way that can be used to calculate the similarities between users or items. The new proposed solution is 'the calculation of the probability of accessing items by the same user in the same session or period (PASS)'.

- **Contribution two:** Using PASS, we propose a new recommendation system based on users' preferences for items and Markov Chain model.

1.8.2. Enhancing the new recommendation system using the time factor.

Recommendation systems have been used to recommend items to users based on users' preferences for items; however, users' opinions and items popularities vary with the time; therefore, many researchers enhance the conventional recommendation systems using the time factor [15] [16]. This thesis use users' opinions and items' popularities that are calculated from users' preferences for items to enhance the new proposed technique. The enhancement of the new technique as follows:

- **Contribution three:** The new technique is enhanced using items' popularities in general.
- **Contribution four:** Another enhancement is done using items' popularities in the last period of time.

1.8.3. Enhancing the new technique using the friends feature.

Many researchers enhance the conventional recommendation systems using contextual information about users and items. There are two approaches of Context-aware techniques.

First: the post-filtering approaches that generate suggestion for items using the recommendation systems then they filter these suggestion to identify the final recommended list of items to the active user.

Second: the pre-filtering approaches that filter users' rating first then they calculate the recommendation to the user.

However, there are many variations of users' opinions that lead to many variations of items popularities, and the context-aware techniques separate between them; therefore, we use a new way to enhance our proposed technique using the friends feature.

- **Contribution five:** Enhancing the new proposed technique using the friends feature. Friends access the same websites and view the same movies. We use the weights of items that are accessed by the active user's friends to enhance our new proposed Markov chain based recommendation system.

All the mentioned contributions will be discussed in details later in this thesis.

1.9.The thesis outline

This chapter introduces the introduction of the thesis. Chapter two reviews the literature related to recommendation systems; it illustrates that many web applications are published in the internet, and millions of users access millions of items. This situation leads to the problem of information overload, and the chapter discusses the motivation of using recommendation systems by web application to cope with this problem. Then, it introduces the general concepts of recommendation. Moreover, it reviews the general concepts of Collaborative filtering (CFRSs), Content based CBRSs, and Context aware recommendation techniques. The chapter investigates ways of using the time factor, social media feature, and Markov model in recommendation system; and it finally gives some discussions.

In chapter three the research methodology is discussed. First, the research design is illustrated in details, and the chapter gives the research framework that contains four phases. Phase one is the literature review; phase two is data preparation; phase three gives the new technique designing and enhancements, and phase four represents ways of the evaluation of the new technique and its enhancements. The chapter indicates that datasets from MovieLens and LastFm websites are used for the evaluation purposes.

In chapter four, the thesis proposes a new recommendation system. Firstly, the limitation of the conventional recommendation systems are illustrated, and the motivation of designing new recommendation systems are listed. Secondly, the new recommendation system is designed; it based on users' preferences for items. The new recommendation system is Markov Chain Recommendation System (MCRS). A dataset from MovieLens websites is used to evaluate the new technique; the evaluation is done using mean absolute errors, and precision and recall.

Chapter five finds some limitation of the new proposed solutions, Markov Chain recommendation systems. It discusses the motivation of the enhancement of the new technique that based on the time factor. MCRS is enhanced twice. The first enhancement is done using the general weights of items while the second enhancement is done using the period weights of items. A dataset from MovieLens website is used for the evaluation that is done using the accuracy, and the precision and recall.

Chapter six finds more limitations of the new proposed solutions and its enhancements. The chapter discusses the motivation of the new enhancement that based on friends feature. It introduces a new enhancement of MCRS using the friends feature. A dataset from LastFm website is used for the evaluation.

The last chapter introduces the discussions and the future work.

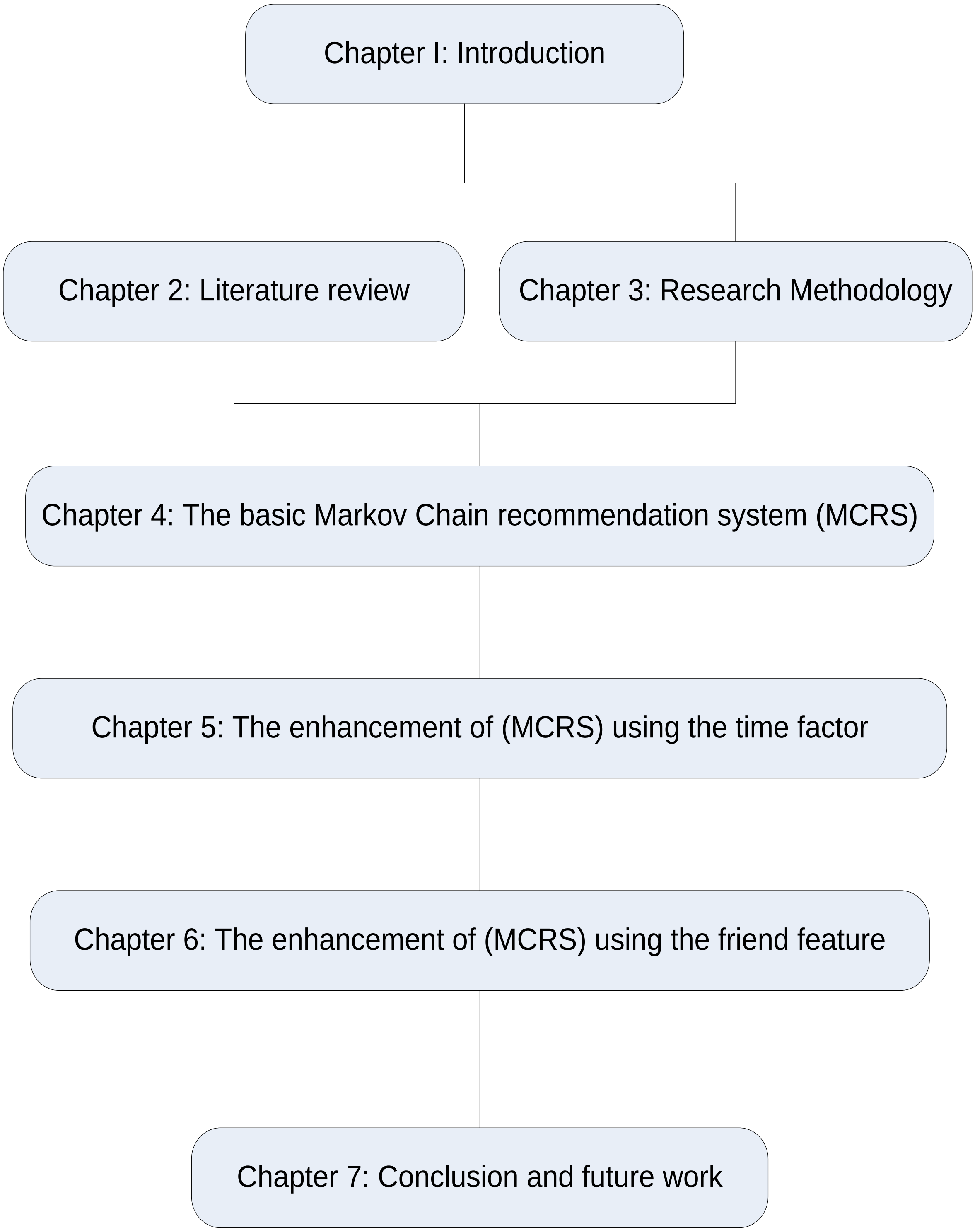


Figure 1-1 : The structure of the thesis

Chapter 2

LITERATURE REVIEW

2.1. Introduction

Recently, the number of users is exponentially increasing in the internet because many web applications are created and published to allow millions of users to access millions of items. Items are things that can be provided by web applications to their users e.g. they can be elements of movies, photos, devices, etc. In this thesis, “users access items” means they rate, buy, view, or upload etc. these items. The internet usage generates a huge amount of information in different domains e.g. movies, books, CDs, news; and diverse types e.g. rate, text, video, and sound. It is a difficult task to manage the internet; this situation leads to the problem of information overload [61]. Many researchers study ways and techniques to be applied by web applications to ease their usage and to solve this problem, and many of these techniques use the historical data about activities of users that can be divided into two classes: The first class contains the data about users' profiles, and descriptions and contents of their accessed items. The second class generates the data from users' preferences for items e.g. an opinion of any user about a specific item. The first class can be used to design Content-based recommendation systems CBRSs [7], [62]–[64]. The second one can be used to design collaborative filtering recommendation systems CFRSs [10]–[13], [15], [38], [63], [65]–[73]. This chapter will discuss these techniques and investigates their strengths and weaknesses.

Social media websites allow their users to interact among each other [74]–[76]. They add extra amount of information to the traditional ones and complicate the problem of information overload, as they provide to their users many features that can be used in the interaction between any user and others. This chapter studies ways, techniques, and social media features that have been used in recommendation systems. It discusses these topics and gives their advantages and drawbacks.

In the traditional and social media websites, users can access items at specific points of time [25], [26], [67]. Therefore, the time factor is another great opportunity that can be used to enhance recommendation systems. This chapter reviews ways of using the time factor in recommendation systems, and it gives their strengths and weaknesses.

Many recommendation techniques are designed using Markov chain model [36], [38], [77]. This chapter tries to find out ways of using Markov chain model in recommendation systems, and it identifies their weaknesses.

The rest of the chapter gives the motivation of recommendation systems in section 2.2. The general concepts of recommendation systems are discussed, and their strength and weakness are illustrated in Section 2.3. Ways of using social media features in recommendation systems are introduced and discussed in Section 2.4. Section 2.5 illustrates ways of using trend analysis to enhance recommendation systems. Ways of using Markov Chain model in recommendation systems are introduced in section 2.6. Section 0 discusses weaknesses of RS techniques that use the time factor and social media features in recommendation processes. Finally, the chapter is summarized in section 2.7.

2.2. Motivation for recommendation systems

Recently, many web applications have been published in the internet, and the number of users is duplicated because the internet providers receive people every-where at anytime [78]. For this reason, websites as well as their users suffer from the problem of information overload [3], [79]. Search engines are applied by many websites, such as Google and Yahoo, to allow their users to retrieve diverse information [80]. They have been used to search about key words, using queries submitted by the target user. Search engines are partially helping users to find information, despite the problem of finding the actually needed items. For example, when a user searches in Google about the phrase 'information technology'; the result returned contains 745 million items, as shown in Figure 2.1. It is impossible for him to access all of them, and he will access only the first tens of these items.

Figure 2-1 : the result, of searching Google about 'information technology’, contains 745 million items.

Moreover, there are many online activities that need new techniques to generate suggestions for interesting items and services to websites' users, as follows:

- E-business websites use the internet for the interaction between their customers; therefore, they need appropriate techniques to introduce the interesting items and services to users. New e-business websites need to introduce interesting items and easy services to observe their users [81].

- Shops and markets are opened online to introduce interesting items to their visitors that can purchase interesting items from any country across the globe, and the amount of online available items and services is very huge while different users need to purchase different items when they access these online shops [82], [83].

- Social media websites allow their users to interact with each other, exchange items and opinions, and create groups etc. Users are faced by a very complicated situation that can't be managed using search engines only [84].

The above examples of web applications have been using Recommendation systems to solve the problem of information overload.

Roles and goals of recommendation systems

There are two different roles of recommendation systems, that can be applied by websites:

- The first role is played on behalf of service providers or websites itself, as the smart websites need to discover users' interesting items. They use recommendation systems to ease their usage because users need best and quick services, and interesting items [5], [10], [35].

- The second role is played on behalf of users of these websites to generate recommendations for items to individuals or groups of users [16], [17], [85]. Sometimes, users do not know what they need; they just follow the recommendation systems suggestions for items.

There are many goals for using recommendation systems:

- Many Web applications use recommendation systems to personalize items to their users i.e. RSs are usually personalized, as different users and groups of web applications receive different recommendations [86].

- Recommendation systems can be non-personalized. This type of recommendation is used in newspapers and magazines e.g. top ten selections of books or CDs [12], [87].

- Online shops and service providers commercially use recommendation systems to increase the number of items and services sold. This goal is achieved since recommender systems give suggestions for interesting items and services to users [1].

- The non commercial web application can use RSs to increase the number of their users e.g. news readers [88].

- RSs can be used to enable users to select more diverse items and help Web applications to know more about their users to understand what they need [89]. This goal needs more studies, to be achieved.

- RSs have been used by web application to generate suggestions for interesting items to users. They have been also used to recommend some or all valuable and interesting items to the active user[33]. Instead of recommending single items, RSs can be sometimes used to generate a sequence of needed items to users.

In addition to the mentioned roles and goals, there are many facts indicating that the needs of recommendation systems are dramatically increasing. All the highly rated internet sites have been using recommendation systems, e.g. Facebook, Amanzon.com, YouTube, Netflix, and Yahoo, [42], [90], [91]. There are dedicated conferences for recommendation systems e.g. ACM Recommender Systems (RecSys) that established in 2007 [92]. These conferences encourage many researchers to study and investigate this area.

The next section gives the general concepts of Recommendation systems.

2.3. The Concept of Recommendation Systems RSs

Recommendation systems RSs are software tools and techniques that can be used by web applications to generate suggestions for interesting items to users [5], [10], [15], [39], [69], [93]–[97]. Items are things and services that can be recommended to users; they can be elements of CDs, books, movies, photos, or news. Suggesting some books to Amazon[1] visitors, and interesting movies to MovieLens[2] users are examples of operations of recommendation systems. The design of recommendation systems depends on the domains and the characteristics of the data available. There are different recommendation systems according to ways of analyzing the data. The most successfully used techniques are Collaborative filtering recommendation systems (CFRSs) [39], and Content-Based Recommendation Systems (CBRDs) [7].

More illustration about collaborative filtering recommendation systems CFRSs is given in subsection 2.3.1. In subsection 0, we introduce some details about content-based recommendation systems CBRDs.

2.3.1. Collaborative Filtering Recommendation systems (CFRSs)

Collaborative filtering recommendation system (CFRS) is the idea of using activities and opinions of the active user's and his similar users' to generate a list of interesting items to him e.g. CFRS can use the rating for movies that have been viewed by the active user and his similar users to recommend new interesting movies to him. CFRSs can be memory-based or model-based techniques [98] [99]:

Memory-based techniques use user-item entire database to predict users' preferences for items because everyone is a member of a group with similar interests [15], [100]. User's preferences for new items can be produced, using preferences of the nearest neighbours of the active user then the top N most frequent items are listed as recommendations.

[1] http://www.amazon.com/

[2] https://movielens.org/

Model-based techniques use history data which is collected from users' preferences for items, to formulate a model that can be used later to predict or recommend an interesting item[101], [102]. Model-based techniques calculate the similarities between users or items using their historical data e.g. ratings of users for items. These similarities can be used by an appropriate technique to predict the expected ratings for items then recommend a list of interesting items to the active user.

To design a collaborative filtering recommendation system, we respectively need to collect historical users' ratings for items then calculate the similarities between users or items followed by predicting the active user's ratings for unknown items and finally generate a list of recommendations to him.

Users' rating for items:

The first step of designing a collaborative filtering recommendation system is to collect users' preferences for items. Normally, users' preferences are their ratings, that can be collected implicitly or explicitly from their activities [56]. The implicit rating can be categorized into scalar ,binary or unary [15]. Scalar ratings can be numeric stars like (1, 2, 3, 4, or 5); it can be ordinary ratings like very good, good, normal, bad, and very bad. Binary ratings can be represented as like or unlike, yes or no, one or zero, and so on. Unary ratings can be taken explicitly from users' activities; for instance, if a user frequently purchases an item he most likely likes it.

Similarity calculation:

Similar users usually access similar sets of items e.g. similar users view similar movies. If two users access the same list of items, their similarity is one. We say that they are 100% similar to each other. If they do not share accessing any items, their similarity is zero. Otherwise, their similarity lays between one and zero. The similarities between users or items can be calculated using the data of users that access a set of items. There are many ways that can be used to calculate the similarity between users or items [103].

Consider the user U has rated the set of items $U_{(i)} = \{r_{ui}: i = (1,2,3, \dots, n)\}$,and the user V have rated the set of items $V_{(i)} = \{r_{vi}: i = (1,2,3, \dots, n)\}$, n is the number of rated items, r_{ui} and r_{vi} are the ratings of the users U and V for the item i. The similarity between U and V ($s_{(uv)}$) can be calculated using Pearson correlation coefficient [10]; the similarities between users lay in the interval (1,-1). The equation (2-1) is used to calculate $s_{(uv)}$. Where $s_{(uv)}$ is the similarity between users U and V, $\bar{r}$ is the average rating that can be given by users to items. The number of rated items by users is n. r_{ui} and r_{vi} are the rating of the users U and V for item i, respectively.

$$\boldsymbol{s_{(uv)}} = \frac{\sum_{i=1}^{n}(\boldsymbol{r_{ui}} - \bar{\boldsymbol{r}})(\boldsymbol{r_{vi}} - \bar{\boldsymbol{r}})}{\sqrt{\sum_{i=1}^{n}(\boldsymbol{r_{ui}} - \bar{\boldsymbol{r}})^2}\sqrt{\sum_{i=1}^{n}(\boldsymbol{r_{vi}} - \bar{\boldsymbol{r}})^2}} \qquad \textbf{2-1}$$

The similarities between users can also be calculated using Vector cosine similarities [104]. It lays between 0, to denote users are not similar, and 1 to denote users are 100% similar. The equation **(2-2**) is used to calculate $s_{(uv)}$. Where $s_{(uv)}$ is the similarity between the user U and the user V.

$U_{(i)}$ is the rating the users U for the i^th items.

$V_{(i)}$ is the rating the users V for the i^th items.

$$\boldsymbol{s_{(uv)}} = \frac{\overline{|\boldsymbol{U_{(\iota)}}|} \cdot \overline{|\boldsymbol{V_{(\iota)}}|}}{\|\boldsymbol{U_{(\iota)}}\| \cdot \|\boldsymbol{V_{(\iota)}}\|}. \qquad \textbf{(2-2)}$$

Most studies have been using these techniques to calculate the similarities between users or the similarities between items. However, the amount of items is increasing and users access only a very small subset of items e.g. users U and V may access only tens of items out of millions. This means these techniques suffer from the problem of sparsity.

Prediction techniques:

The k nearest neighbour algorithms is one of the most successfully used techniques to predict users' ratings for items and to recommend a list of items to the active user [105]. They can be user-based or item-based k nearest neighbours' algorithm. These techniques are based on the similarities between users and the similarities between items which can be calculated using vector cosine similarity, Pearson correlation measure or conditional probability-based similarity [39], [103], [106].

Ekstrand [107] gives an introduction to the important issues underlying recommenders and current best practices for addressing recommendation problems.

- **User-based CF:** Many recommender systems [10], [87], [108], [109] use average ratings of neighbours of user U_u on item I_i to predict his ratings as in equation (2-3) :

$$\text{Prediction}_{(u,i)} = \overline{r_u} + \frac{\sum_{(n similar users to u)} sim(u,n) * (r_{ni} - \overline{r_n})}{\sum_{(n similar users to u)} sim(u,n)} \qquad (2\text{-}3)$$

Here $\overline{r_u}$ is the average rating of user U_u on the neighbor of item I_i and r_{ni} is the rating of the n^{th} neighbour of user U_u on item I_i and $\overline{r_n}$ is the average rating of the n^{th} neighbour of user U_u on neighbours of item I_i. Then, the maximum rated items are recommended to the user.

User-based techniques have been used by many applications [10], [87], [108], [109]. However, there are many limitations in ways of calculating users similarities and items similarities that are calculated to be used later in the prediction stage.

- **Item–based CF:** Item-based recommendation algorithms can be used for items predictions to users [38], [73]. These approaches are based on the rated set of items, by the active user, to calculate their similarities to the target item i then select its k most similar items $\{i_1, i_2, \ldots, i_k\}$ with their corresponding similarities $\{s(i_1), s(i_2), \ldots, s(i_k)\}$. The prediction is computed by taking a weighted average of the active user's ratings on these similar items. The prediction of the i^{th} user's opinion (O_{ij}) on the j^{th} item that have I_c nearest neighbors, can be calculated using the equation **(2-4**) [38]:

$$\mathbf{O_{ij}} = \frac{\sum_{c=1}^{k} \boldsymbol{O_{ic}} * \boldsymbol{sim}(\boldsymbol{I_j}, \boldsymbol{I_c})}{\sum_{c=1}^{k} \boldsymbol{sim}(\boldsymbol{I_j}, \boldsymbol{I_c})} \qquad (2\text{-}4)$$

The mentioned prediction techniques are based on the similarities between users and the similarities between items; these similarities are calculated using users' rating for items.

However, these techniques suffer from sparsity problem as individual users rate for tens of items out of millions ones, in addition to the fact that says: users' opinions vary with the time and old users' rating for items may give different impression about them i.e. ways of calculating the similarities are not accurate.

Chapters three and four give more details about a new way that used to design a new recommendation system. The new technique merges users' preferences and the time of users' preferences for items. On the other hand, users' opinions and items' popularities vary with time; therefore, the new technique is enhanced using these variations.

Context-aware approaches:

The information about users and their environment are called context. Users' context contains the detailed data that can play a more significant role in the recommendation processes than only using the rating of users for items. Interesting items to a user at night may be different from interesting items to him in the morning. In holidays, users may need to watch movies' in TV channels; but on normal days, they may try to watch news. The techniques that utilize users environment information are called context-aware recommender systems [29], [24], [27], [30]–[35].

$$\text{Context-Aware CF} = \text{the traditional CF} + \text{Context analysis} \tag{2-5}$$

There are two approaches of Context-aware CF [78].

- The post-filtering approaches calculate the recommendation, using CF techniques, then filter the result using users' context.

- The pre-filtering approaches filter users' rating first then calculate the recommendation to the user, using CF techniques.

Users normally rate different items in different periods of time i.e. users' opinions vary with the time. The variation of users' opinions leads to the variation of items popularities. Users' context information and their ratings for items are complementing each other. However, the context-aware techniques separates between them since they pre/post-filter the rating of users for items before/after using the conventional recommendation systems to recommend items to users; Therefore, ways of using pre/post- filter context aware CF recommendation systems need to be enhanced. Moreover, social media provides many features that can be used to enhance recommendation systems.

Content-Based Recommendation Systems CBRSs

Content-based recommendation systems (CBRSs) are designed using the information about users' preferences for items, and items' descriptions [4], [5], [7], [35], [61], [63], [94], [95], [110], [111]. CBRSs are used by many web applications to recommend items to users by analyzing user's preferences, web contents, and items' descriptions to generate users' profiles. Users' preferences (e.g. links and rates) can be used to collect web contents and list of items respectively. Items are normally represented as sets of structured features or attributes (e.g. movies can be described by title, author, editor, etc). Sometimes features are generated from unstructured data such as text, web page etc. The active user's profile can be used later to recommend items that might be interesting by him. The process of Content-based recommendation system is performed in three steps [112].

- Firstly, the content analyser represents items content in a suitable form that can be used in the second step. Items' description is used by the content analyzer to extract features from texts contents or from information about texts to generate structured items representation (a suitable form) that stored in repository represented items.

- Secondly, it generates user's profile from the active user's feedback and items representation.

- Finally, it uses the active user's profile to generate recommendations for items to users.

Content-Based recommendation systems [33] use simple retrieval methods like the vector space model (VSM) with the basic TF-IDF weighting (TF Term Frequency - IDF Inverse Documents Frequency) [113]. Any document is represented as vector of terms weights. Consider, There are a set of D documents, and a set of T word dictionary that obtained by natural language processing operation. Then, for any $d_i \in D$ and $t_J \in T$ the TF-IDF can be calculated using the equation **(2-6)**. Where N is the number of all documents and n_k is the number documents that contain t_j.

$$\boldsymbol{TF - IDF(d_i, t_j) = TF(d_i, t_j).\log\frac{N}{n_k}} \quad (2\text{-}6)$$

$$\boldsymbol{TF(d_i, t_j) = \frac{f_{(i.j)}}{Max(f_{(i,j)})}} \qquad (2\text{-}7)$$

$TF(d_i, t_j)$ is calculated using the equation **(2-7)**. Here, $\boldsymbol{f_{(i,j)}}$ is the frequency of the term t_j in the document d_i.

The weights of the term t_j of the document d_i are calculated using the equation **(2-8)**.

$$\boldsymbol{w(d_i, t_j) = \frac{TF - IDF(d_i, t_j)}{\sqrt{\sum_{i=1}^{T}(TF - IDF(d_i, t_j)^2}}} \qquad (2\text{-}8)$$

The similarity between the document $\boldsymbol{d_i}$ and the document $\boldsymbol{d_j}$ is calculated using the equation **(2-9)**.

$$\boldsymbol{sim(d_i, d_j) = \frac{\sum_{i,j=1}^{k} w_{(k.i)} \cdot w_{(k.j)}}{\sqrt{\sum_{i=1}^{k}(w_{(k.i)})^2 \cdot \sum_{j=1}^{k}(w_{(k.j)})^2}}} \qquad (2\text{-}9)$$

Content-based recommendation systems are based on terms or key words $t_i \in T$ that contained in the collection of documents $d_i \in D$. However, any text may contain polysemy words that have multiple meanings, and it can contain synonymy or multiple words with the same meaning. These types of words decrease the recommendation accuracy; if the technique doesn't take them in account. The recommender systems that consider these kinds of terms are called semantic based approaches.

The adoption of the content-based recommendation paradigm has several advantages, compared with collaborative filtering techniques:

- **User independence:** Content based recommendation systems uses the user accessed items to generate his profile that is used later to generate the recommendation i.e. items recommendation don't need similar items to be accessed by similar users' to the active user.

- **Transparency:** all used items have a clear description this leads to the trust and transparency.

- **New term:** Content-based recommenders are capable of recommending items not yet rated by any user.

However, Content-based recommendation techniques generate suggestion for similar items to items that already have accessed by the active user. Normally, any user accesses only very limited subset of items. The user doesn't benefit from other users' opinions. Many researchers study hybrid recommendation systems, [63], [112], [114]–[116] that use users preferences for items (ratings) and items contents to generate recommendation using content-based Collaborative Filtering recommendation techniques. The Hybrid recommender systems outperform the conventional ones.

This thesis doesn't use Content based techniques. It uses users' preferences for items, social media features, and the time factor to generate a new technique to be used to recommend items to users.

2.3.2. Hybrid recommendation systems

Hybrid recommendation systems are consist of mixture of two or more recommendation techniques [33], [35], [64], [114]. They aim to cope with the problems that face the conventional recommendation systems such as cold start, sparsity.

Hybrid recommendation techniques are combination of the following techniques [114]:

- Collaborative filtering recommendation system (CFRS) [39].
- Content-based recommendation system (CBRS) [7].
- Demographic based recommendation system [1] .

- Utility-based recommendation system [65].
- Knowledge-based recommendation system (KBRS)[112] .

Hybrid recommendation systems outperform the single component systems by combining these multiple techniques. Table 2-1 gives a general background about recommendation techniques. It gives the input and the processes of the techniques.

Table 2-1 : Recommendation techniques [112]

Techniques	Background	Input	Process
Collaborative	Ratings of users on items.	Rating of Users on items.	Identify similar users by using their rating on items
Content-based	Descriptions of items	Ratings of users on items	classifications of items needed by users
Demographic	Demographic of users and their rating on items	Demographic information about users	Identify similar users by locations and items
Utility-based	Descriptions of Items.	A utility function over items that rated by users	Apply the function and recommend items to users
Knowledge -based	Features of items And Knowledge about items that meet a user's needs.	Description of items needed by users.	identify items to users.

2.4. Using Social Media in RSs

Social media are web applications that have been used by users and organizations to create, exchange, and publish contents e.g. text, audio and video, [47], [117]. Twitter and Facebook are examples of social media.

Twitter is a web application that allows users to read and send text messages, known as tweets [84]. It has the following features:

- **Tweet** refers to posting a message of up to 140 characters, known as tweets. The content of tweets may vary from users' daily activities to news. Some messages may also include URLs to web pages or hash tags to relate tweets of similar topics togeth-er. Each hash tag is a keyword prefixed by a # sym-bol. For example, #Egypt and #Jan25 have been used to group tweets related to Egypt's revolution in January 2011.

- **Retweet** refers to forwarding a tweet from another user to the followers. Such re-sharing of tweets is a prevailing mechanism in Twitter to diffuse infor-mation.

- **Follow** refers to linking to another user and receiving the linked user's tweets after that. The user creating such a link is called the follower and the linked user is known as the followee.

- **Mention:** One may mention one or more users in a tweet by including in the tweet the mentioned user name(s) prefixed by the @ sign. The mentioned us-er(s) will subsequently receive the tweet. This is a means for users to gain at-tention from the other users so as to start new conversations.

Facebook is an online social networking service founded in September 2006 by Zuckerberg [74]. Facebook changes the map of the world, specially, in fields of in-formation technology.

Table 2-2 summarize Facebook features.

Table 2-2: Facebook features

The feature	The possibility of using the features in Recommendation Systems	Possible ways of using the features in Recommendation Systems
Friend	Yes [118]	By identifying neighbors of users
Networks	Yes [119]	
Groups	Yes [74]	
Wall	No	
Profile	Yes	By predicting trends of users' preferences on items
Timeline		
Like	Yes [120]	Can be used to identify users' preferences on items
Comment		
Share		
Pages	No	
Photos	Yes [74]	items that users can use to interact and give their preferences
Videos	Yes [74]	
Chat	No	

Boyd and Ellison (2007) [46] define social media as websites that allow their users to construct their profiles, share items and connection, and view and traverse their items by others within the system. Boyd and Ellison's study has become an essential reference in the field. Yu and Kak (2012) [47] introduce three main issues that must be taken in consideration, to predict trends from social media:

- The event must be done by human.
- The distribution of people in social media must be similar to the real world.

- The event must be easy for public to collaborate with it.

These studies can be taken as a basis to discuss ways of using social media features in recommendation systems. The traditional recommendation systems consider individual users as each of them rates for a subset of items independently. However, social media add extra information about users' interaction that provide a great opportunity to enhance the recommendations[40].

2.4.1. Social media features

The most successfully used social media recommendation systems are based on the conventional collaborative filtering techniques in addition to the aggregated information of trusted users[40].

Tang et al. [40] classify social recommendation CF model into two categories:

- Memory based social recommender systems
- Model based social recommender systems.

They use the conventional collaborative filtering recommender systems in addition to the social information model. To design a recommender system based on information from social media, we need to identify the following issues:

- We need to identify feature (s) (e.g. groups, friends) that can be used to generate and aggregate trust ratings of users for items (e.g. links, movies, photos).
- We need to apply an appropriate technique(s) that can be used in the recommendation process.

Group feature: Many researchers use the group feature to generate its rating by aggregating preferences of individual users (e.g. rating of users for items) [76], [121]–[123]. The aggregation of preferences of groups’ members reduces a huge amount of information and eases the use of recommendation technique.

However, members of one group may have variance contexts e.g. it can contain users of different opinions from different locations and gender. These contexts have different weights (popularities) which can affect positively or negatively on recommendation accuracy. Users' opinions and items' popularities can be used to enhance recommendation systems.

We do not use group feature in this thesis.

Friends feature: Recommendation systems are used to recommend friends to users[124] [125]. The friends feature simulates users' real world with respect to the flexibility and endless locations in real time interaction. The feature is used in recommendation systems to develop a model-based recommendation system that are based on friends' preferences [40]. They recreate a new user’s profile from his profile and his friends’ profiles. Their solution increases the number of items that are accessed by the user and decreases the sparsity problem. However, normally social media friends are unknown as many friends’ requests received from people from different countries, age, and cultures. This situation violates the trust factor and decreases the accuracy of recommendation. Lyle et. al [126] solve this problem and group people into clusters based on the items they have purchased. Their model is attractive for many domains: books, CDs, movies, etc., but it does not always work well since it faced by sparsity problem. Konstas et al. [49] use friendship as well as social tags to capture explicit rates of users for items then they use the collected data in the recommendation processes.

The friends feature can be used to generate popularity of items according to preferences of the active user's friends for items. It can play an essential role in the recommendation processes.

Therefore, **we propose a new recommendation system**; we put the following ideas as a basis of our new solution:

- In general, items have variant popularities within the group membership i.e. within the group of users that have accessed a set of items.

Items popularities give users' opinions, and it can be calculated as given in equation **(2-10)**.

$$\boldsymbol{w}(\boldsymbol{i}) = \frac{(\boldsymbol{count\ of\ accessing\ item\ i\ by\ all\ users}\ )}{(\boldsymbol{count\ of\ all\ accessed\ items\ by\ \ all\ users}\ )} \qquad (2\text{-}10)$$

In this case we can calculate the general weights of items according to the group boundary.

- The popularity of items can be depending on locations of group members. For instance, the popularity of the items i that accessed by users in the location c can be calculated as follows in equation ***(2-11)***.

$$\boldsymbol{w}(\boldsymbol{i},\boldsymbol{c}) = \frac{(\boldsymbol{count\ of\ accessing\ item\ i\ by\ all\ users\ in\ location\ c})}{(\boldsymbol{count\ of\ all\ accessed\ items\ by\ \ all\ users\ in\ location\ c})} \qquad (2\text{-}11)$$

In this case we can calculate items weights using preferences of group members in location c.

- The popularity can be calculated using users' gender g of group members as given in equation **(2-12)**.

$$\boldsymbol{w}(\boldsymbol{i},\boldsymbol{g}) = \frac{(\boldsymbol{count\ of\ accessing\ item\ i\ by\ all\ users\ of\ gender\ g})}{(\boldsymbol{count\ of\ all\ accessed\ items\ by\ \ all\ users\ of\ gender\ g})} \qquad (2\text{-}12)$$

In this case we use preferences of male or female that can be taken from all data to calculate weights of items.

- We can restrict these popularities by time. In this case, we can represent popularities of items in time function as represented in equation **(2-13)**.

$$\boldsymbol{w}(\boldsymbol{i},\boldsymbol{t}) = \boldsymbol{F}\left(\frac{(\boldsymbol{count\ of\ accessing\ item\ i\ by\ all\ users\ at\ time\ t})}{(\boldsymbol{count\ of\ all\ accessed\ items\ by\ \ all\ users\ at\ time\ t})},t),t\right) \qquad (2\text{-}13)$$

The time interval of the collected data that can be used in recommendation systems can be divided into periods. The popularities of items change from any period of time t to other.

We conclude that the best recommender systems consider the consistency of users' preferences that can be achieved by combining their rating in all situations.

2.4.2. The trust factor

The trust factor of users' preference indicates the reality of users' opinions that can be created from their connections and interactions [127] [128]. Mican et. al [129] consider the relation between two users is represented by a vector that contains their interactions. The same formal approach is used with regard to the relationships between users and content. The vectors that contain the interactions among users are stored in tables of the form: user-user, while the vectors that contain the interactions among users and pages will be stored in a table of the form: page-user. Mican system aggregate the collective intelligences to calculate the trust scores between users (the number of existing links among them) revealed by the information of users interactions and connections with each other. The system uses these scores to enhance the recommendation. Their approach outperforms CF-based approaches and content-based recommendations. It provides cold start to users, scalability and serendipitous recommendations. However, the trust factor can be lost if we do not represent the data according to their chronological order i.e. old preferences of users for items violate the trust factor because at any point of time users can access different items and change their opinions.

Example: Consider two users have linked five pages in three periods of time as represented in Table **2-3** below (pages-users table):

Table 2-3 Two users have linked five pages in three different periods of time.

	page 1	page 2	page 3	page 4	page 5	period
User 1	1	1	0	0	0	P1
User 2	0	0	1	1	0	P1
User 1	0	0	1	1	0	P2
User 2	0	0	1	1	0	P2
User 1	0	0	0	1	1	P3
User 2	1	1	0	0	1	P3

The trust factor that ties the two users can be considered as the reality of linking these five pages. When the user links a page, we put one in the corresponding cell (page-use) otherwise we put zero. According to this, there are four different similarities between the two users as follows:

- In period P1, the similarity between them is zero.
- In period P2, the similarity between them is one.
- In period P3, the similarity between them is greater than zero and less than one.
- In general, the similarity between them is one when we do not consider the time factor, and we just count the number of pages that have been linked by all users.

Therefore, the time factor affects positively or negatively on recommendation systems accuracy. Here, we use the linking feature to calculate users' rating, and we use the time as a trust factor. The trust factors are indispensable to design recommendation technique because the information is exponentially growing. The effective solution for reducing the complexity of the problem of information overload cannot be done without considering the trust factor.

Ways of creating the trust factor remain a big challenge and need more studies.

2.4.3. Data sources:

Data sources in the traditional recommendation systems can be considered as users' ratings for items that can be created implicitly or explicitly. Web applications allow to their uses to sign items as like or unlike thus we can generate lists of liked or un-liked items by users. These lists can be used to calculate the similarities between users or items then they can be used to generate the recommended lists to the active user. Social media websites provide to their users plenty of features, e.g. friendship and groups that can be used as data sources.

Many studies consider friendships feature as data source. Ioannis et al. [92] investigate the role of using relationships in recommendation systems. They use social annotations and friendships in social graphs of users, as data source to enhance the collaborative recommendation system.

We use friends feature to design our new proposed Markov Chain based recommendation systems. There are two cases of using friends feature:

- Case one: the relations between friends can be considered as data source. In this case, users' friends feature can be used to calculate the similarities between users or the similarities between items. This case is used by Ioannis al. [92], to enhance collaborative filtering recommendation systems.

- Case two: friends feature can be used as trust factor. In this case, a list of accessed items by friends can be used to calculate the popularities of items. We use this case to enhance our new proposed solution, Markov chain based recommendation system.

However, friends' opinions in the same item can be different. While some of friends like to view specific movie, some of them may like to view another one. If all friends share viewing movies, they create trust opinions. The friendship feature can add extra effect if we use their sharing activities as data source. Guy et al. [130] and Geonhyeok et al. [131] use sharing feature to enhances the efficiency of recommendation systems. Also, data sources stream users' preferences for items according to events time.

Therefore, the time factor must be considered when we retrieve the data from any data source. Because, data sources are used to calculate the similarities between users and the similarities between items that can be used to generate recommended lists of items to users.

Recommendation systems are based on ratings of users for items. The traditional recommender systems have been using the direct users' rating that can be collected implicitly or explicitly from users' activities. Social media networks flood a huge amount of information as the reaction of using social media feature(s) by users. Social media feature can be considered as data sources, but we need to transfer the collected data from these sources to ratings that can be used later to calculate the similarities between items or users. The information representation is a big challenge faces the researchers these days.

Many studies investigate ways of information representation. Yu, Asur, and Huberman (2011) [132] study the biggest Chinese social network website (Sina Weibo) and compare it with the international ones. They found that there are big differences between them. While Chinese users retweet media contents like jokes, Twitter's users throw in new events. The feature "retweet" is used to generate users' ratings that represent users' opinions of liking the tweets. However, users can retweet a message to their friends to identify that they like or dislike it. Therefore, the rating may be more trusted if we involve tweets similarities before creating users' rating. Lobzhanidze et al [52] present an empirical study in mainstream and social media using the real-world data. They use trending in mainstream as information source. They collect news articles from mainstream media using Really Simple Syndication (RSS) feeds. The document of RSS is formatted XML file that contains a short summary of the article, publication date, title, and the link to the original article. Twitter data do not need this technique to collect users' tweets. They help to enhance video recommendation and then prove that the mainstream media help the trend detection more than the social media. However, the collected data using RRS does not represent human ratings. It can be more usable if we identify items (e.g. movies and news), then we combine the collected data to be represented as one data source.

In general, social media features can be combined to build a data source that can be used to generate more consistent information. Table **2-4** presents social media features and ways of using them to enhance recommendation systems.

Table 2-4 : Features and techniques that used to enhance the efficiency of recommendation systems

Feature& Social Media	**How it used**	**Evaluation**	**Efficiency**
Publishing [133] interaction [14] Intel-epciune.ro You Tube	In Content recommendation, using Users publishing and interaction with in social network to calculate the similarity between users then recommend items to a user. Using sharing behavior as Implicit Feedback for Collaborative Filtering.	Comparing Top-N popular and Item-based algorithms	It improve the efficiency
Tagging [134] [135] Delicious[3],	In Collaborative and, Content-based Filtering, extending the nearest neighbor and Pearson	Comparing CF approach with Collaborative	It improve the efficiency

[3] *https://**delicious**.com/*

Feature& Social Media	How it used	Evaluation	Efficiency
CiteULike[4], BibSonomy	Correlation algorithms by adding information about tagging behavior to enhance recommendation.	-Content-based Filtering	
Semantic [136] Word Net[5]	Semantics of content-base profile were used to Group users, then the recommendation calculated within the group.	Compare using Mean Absolute Error method	It improve the efficiency
Like [14]	In Collaborative Filtering, like and unlike counted to calculate the similarity between users.	Comparing precision and recall	It improve the efficiency
Rating [137] [50] Facebook	In collaborative filtering, the ratings of users were used to clustering users to groups, to construct a social community. Then the similarity of users calculated within the group. To be used for prediction and recommendations	Comparing precision and recall	It improve the efficiency
Friendship and tagging, LastFm . [92]	On social network collaborative filtering recommendation system.	Comparing precision and recall	It improve the efficiency

It's clear that, users' events on social media websites effect positively in recommendation systems. Social media features can be used to enhance the Recommendation Systems. We use the friends feature from LastFm dataset to enhance our new proposed solution, Markov chain based recommendation system.

[4] http://www.citeulike.org/
[5] http://wordnet.princeton.edu/

2.5. Using trend analysis in RS

Trend analysis is a method of using historical data to generate suggestion for things that will be happen in the future. In recommendation systems, trend analysis is also based on historical data about the users' preferences that give the overall trends of the users' interactions with each others. It takes into account the time of users' activities to recommend items to the active user. Recommender systems can involve the time factor in the recommendation processes.

This section investigates about ways of trends prediction and ways of using trend analysis to enhance the efficiency and effectiveness of RSs.

Trends predictions:

The first step of trends predictions is splitting the time into periods. Marinho et al. (2012) [138] propose clustering method based on location and time. Their method put users' activities in classes and does not give trends of items. Wang et al. (2009) [54], Lee and Park (2008) [26], and Ding and Li (2005a) [25] order all events according to the time. They split the time into old, middle, and recent. In their solution, recent events are given greater rating value. Koren (2009) [11] goes farther more and uses a time period in two methods (1) factorization model and (2) item-item neighbourhood model. Splitting users' events into periods of time is the main advantage that can be used to learn the behaviour of items' popularities to predict their trends.

However, the mentioned techniques use time functions that do not merge events and events' time. While, these solutions only use CF techniques and the time factor; contrary, we propose a new Markov chain based recommendation system, that based on users preferences and the time factor; our solution merges users' preferences for items and the time factor.

The second step of trends prediction is identifying trends of items (popularities) that vary with time. After splitting the time into period; the goal is usually to predict trends of users' preferences for items because the time positively or negatively affects on this prediction . Many researchers study the time affecting on users' preferences for items. Ding and Li (2005a) [25] propose the time function f(t) that decrease with time t and the values of the time weights lies in the range (0,1) i.e. While old ratings hold small weights, the recent users preferences have the high weights. Lee and Park (2008) [26] also propose a recommender system to be used in mobile application with respect to users' purchase time and launch time of the products to increase the recommendation accuracy. The old products are less interesting to customers; while the new purchase increases popularities of items. By using these facilities, they recommend short lists of products that are suitable with mobile e-commerce.

All these studies use the same time affects for all items. However, items popularities can be vary with the time in three situations:

- First: items popularities can increase with time.
- Second: items' popularities can decrease with time.
- Last: items popularities do not change with time.

In the first situation, the solution can, therefore, be used positively because items popularities increase while the function value increase. In the second situation the solution cannot be used positively because items popularities decrease while the function value increase. The main drawback in their solution is the lack of calculating values that can be given for any item in different points of time. These studies use the time in the recommendation process, but they do not benefit from the time of users' preferences for items to calculate the popularities of items. They use the same popularity value for different items.

Table **2-5** presents ways of using trends predictions and calculations in recommendation systems.

Table 2-5 : Ways of using trend analysis in recommendation system

Topic and reference	What Trend	Trends values	Time splitting to periods
RS use rating of users for items within time t [139]	RS is updated at every seven days. m = 7.	Trends' values are calculated using ratings of users for items.	The time is split into weeks
RS collect events by year, month, week, day, [32]	RS aggregate events in (year, month, week, day, hour)	Trends' values are calculated using ratings of users for items.	The time is split into years, months, days.
RS use location and time stamp [138]	RS cluster users per year and month.	The time is used for clustering purpose.	Time is not split.
Users' rating and time stamp [56]	CF invest the state of old items and new sails	The same trends values are used for all items.	The time is split into old, middle and recent.

These studies prove that using the time factor can improve the efficiency of recommendation systems, but the area needs more studies.

2.6. Using Markov Chain In Recommendation Systems

Markov proposed that the outcome of a given experiment can affect on the outcome of the next experiment[57], [58]. This type of process is called Markov chain.

Markov chain contains states, the process function of moving from one state to another, and the start state(s) [140]–[143]. If we have a set of states S={s1,s2,s3,...,sn} then the process starts from one of these states and moves successively to another and each move is called a step. If the chain is currently in state si then it can move to state sj at the next step with a probability that denoted by p(i,j), and this probability does not depend on which states the chain was in before the current state. All probabilities are called transition probabilities, and the process can remain in the same state i with the probability p(i,i). Initial probabilities are given to the starting states; this is usually done by specifying particular states as the starting states.

Markov chain model is used in recommendations system. Shani et al. [36] propose an MDP-Based (Markov Decision Process) Recommender System. The states in their model represent the relevant information about the active user, and their technique considers only the most frequents sequences of 5 items. The transition matrix in their technique can be formulated by the probabilities of accessing a set of items that followed by an item. The initial vector is estimated using user's data, and users with no data are considered that they access a missing items. MDP-Based Recommender System has these weaknesses:

- It does not consider the time factor in the recommendation process while items popularities normally vary with time.
- It face the sparsity problem because web applications provide to their users millions of items while individual users access only tens from them.
- Shani solution considers sequences of only five items, with no consideration to the time factor, while users can access more than five items in one session. Therefore, the time factor must be considered.
- They use the feature of viewing items in sequential order i.e. viewing the item A leads to viewing the item B. However, They miss that viewing the item B can also leads to viewing the item A.

The mentioned drawbacks are faced when applying MDP-Based Recommender System in web applications. Moreover, social media websites add new features that flood more information from different domains at the same time in the same application. For example, Facebook network contains users, companies, news agents, TV channels, schools, universities, market shops etc. The information coming from all these sources can be used in the recommendation process. Therefore, Shani model cannot be used to handle all this amount of information. Wu et al. [144] proposed Personalized Markov Embedding (PME) to recommend the next song for the active user, and they embed users and songs into Euclidean space. The Euclidean distance between songs and users represent their relationship that is used to generate the next song. They evaluate their new technique on real dataset from ihou.com, and the results clearly demonstrate the effectiveness of PME.

Rendle et al. [37] presented the method that bring together both matrix factorization (MF) and Markov chains (MC). Their method is based on personalized transition graphs over underlying Markov chains. Every user has a transition matrix i.e. all users generate a transition cube. Empirically, the FPMC model outperforms both the matrix factorization and the non-personalized MC model; however, millions of items are available to users, and individual users access only tens of these items. Steffen et. al solution generates a transition cube that used in the recommendation i.e. this solution suffers from sparsity.

Eirinaki et al. [145] present a hybrid probabilistic predictive model based on Markov models using link analysis methods. They propose the use of a PageRank-style algorithm to generate suggestions for websites according to their importance. Empirically, the results show that this approach outperforms the pure usage-based approaches.

Huang et. al. [59] propose the use of the learning sequence recommendation system (LSRS) using recommendations based on group-learning paths to recommend web pages to users. They use Markov chain model, which is a probability transition model, and employ an entropy-based approach to assist this model in discovering one or more recommended learning paths through the course material. Their study identified benefits for teachers, providing them with ideas and tools needed to design better online courses.

Fouss et. al. [146] propose Multi Agent System (MAS) that consider each agent as a node and the interaction between any two agents as a link. They use Markov chain model to suggest movies people should watch based upon what they watched in the past. Experimental results show that their approach outperforms all the other methods.

Many studies use Markov Chain model in recommendation systems, and they consider the sequence events of users' interaction as states of Markov chain. However, these studies faced by sparsity. Recently the amount of information increases exponentially; web applications provide millions of items to their users, and millions of users introduce their opinions that can be used in recommendation systems. Moreover, social media provide a great opportunity to enhance RSs. All these events and activities of users, when used at the Web, depend on time. The time factor can then be used to predict trends in social media. Social media features and trends analysis can be used to enhance recommendation systems.

We use Markov model to design a new recommendation systems that combine social media information and the time factor, and we use them as one data source to recommend items to users. The mentioned techniques based on accessing frequencies sequences of items that followed by an item, while our proposed solution is based on the feature: *"accessing items by the same users in the same periods of time"*.

Table **2-6** summarizes ways of using Markov chain model in recommendation systems; it gives the used techniques, way of generating the transition matrix, ways of evaluating the models, and comments.

Table 2-6 Ways of using Markov chain in recommendation systems

The Reference	**The technique**	**The Transition matrix**	**Evaluation**	**Comments**
Shani et. al (2005) [36]	An MDP-Based Recommender System.	The probability of accessing an item by the user after accessing a se-	Suffer from the sparsity problem. The sequence of	Million of items are available. Users access of

The Reference	The technique	The Transition matrix	Evaluation	Comments
		quence of items.	5 items only is used.	sequence of hundreds items
(Wu et al. (2013) [144]	Personalized Markov Embedding (PME)	The Euclidean distance between songs and users represent their relationship that used to generate the next song	the PME effects positively in RSs.	No consideration of the time factor.
Rendle et al. (2010) [37]	Both matrix factorization (MF) and Markov chains (MC).	Any user has a transition matrix i.e. all users generate a transition cube.	The FPMC effects positively in RSs.	No consideration of the time factor.
Eirinaki et al. (2005) [145]	Page Rank-style algorithm	Markov models using link analysis methods.	outperforms the pure usage-based approaches	No consideration of the time factor.
Huang et al. (2009) [59]	The learning sequence recommendation system (LSRS).	probability transition model	This study identified benefits for teachers.	No consideration of the time factor.
Fouss et al. (2001) [146]	Multi Agent System (MAS).	The suggestion based upon what people watched in the past.	It outperforms all the other methods.	No consideration of the time factor.

Discussion

Recommendation systems RSs are software tools that have been used by many web applications to ease their usage and to recommend items to their users. There are many categories of RSs such as Content based, Collaborative filtering, and hybrid recommendation techniques. Many researchers study ways of using users’ context to enhance the accuracy of recommendation systems. However, social media provide new features that add extra amount of information to users and complicate the problem of information overload.

The conventional recommendation systems have been used to cope with these problems. These techniques have some limitations which need more studies and investigations. Many techniques are based on users' or items' similarities e.g. collaborative filtering techniques. Many studies investigate ways of using the time factor in the recommendation processes, and many researchers study ways of using social media features in recommendation systems, also Markov Chain model is used in recommendation systems.

Many of the conventional recommendation techniques are based on users' similarities. If users access the same subset of items then they are similar to each other [44]. However, Users' opinions vary with the time. This variation violates the similarities accuracy because any two users can be similar in short term while they are not similar in the long term. If we learn these two users in long term then we can discover that they change their opinions. It is clear that the similarity calculations have some limitations.

Many researchers study ways of using the time factor in recommendation systems. Ding and Li (2005a) [25] propose the time function f(t) that decreases with the time t and the values of the time weights lies in the range (0,1). The oldest events are weighted by lowest weights and the resent events are weighted by the highest weights. However, Items popularities variations are irregular, and many items' popularities have been increasing while others are decreasing; at the same time, the popularities of many of them stay at the same level. Therefore, ways of using the time factor in recommendation systems have some limitations.

Social media features have been used in recommendation systems. Chen and Fong (2010) [118][147] develop Collaborative filtering recommendation system that based on the trust data that taken from friends preferences. They recreate a new profile using his friends' profiles. However, users' preferences are depend on the time. Therefore, friends' preferences can be used to generate the trends of their accessed items. Separating users' preferences from the time factor is a limitation in recommendation systems.

Many recommendation systems use Markov model to make suggestions of items to users. Shani et al. [36] propose An MDP-Based Recommender System (Markov Decision process); their model can predict the item that follows a sequence of items, and they use the optimum sequences order of k items which is unknown. Low orders Markov Models violate the accuracy while high order ones complicate the system. Wu et al. (2013) [144] propose Personalized Markov Embedding (PME) to recommend the next song to the active user. They embed users and songs into Euclidean space; however, users may access items in different sequences e.g. any user can access item A first then he accesses items B next, but the same user can access the item B first then accesses the item A i.e. $A \longrightarrow B$ means $B \longrightarrow A$. The area needs new ways of using Markov Chain Model in recommendation systems.

To design a new technique, we define the following expressions: (1) Items' popularities, (2) Users' opinions (3) Context factors.

- **Items popularities**: Items can be considered as anything that can be recommended to users. In books domain, any book can be described by (title, author, editor, publisher, etc), and any book version can be considered as item. The same consideration can be done for any domain (e.g. CDs, movies, songs, etc). The most accessed items by users are the most popular ones i.e. items' popularities are measured by the number of users that have accessed them.

  Consider a set of books "Books" that have been accessed by readers.

If Books = {book1, book2, book3, book4, book5}, and the probabilities of reading these books by readers P (Books) that can be considered for example as follows:

P (Books) = {0.4, 0.1, 0.3, 0.2, 0}.

Then the most popular books are book1 and book3 respectively.

- **User's opinion:** Individual user of a web application rates for items implicitly or explicitly, and these ratings give users' opinion about items. Any user has independent rating for a subset of items. The ratings of all users generate their general opinions about all items i.e. the aggregation of users' opinions gives items popularities.

- **Context factors:** Context factor can be any feature that depends on the user when he has accessed an item. Consider, a girl user U that allocated in the place p, and she has accessed an item i at the time t then we can list the context as follows:

    - The gender of the user which can be male or female.

    - The location of the event where the user have accessed the item that can be country, city, home, or work.

    - The time when the user has accessed the item that can be specified by hour, day, week, month or year.

    - Other context can be identified for any web application user.

According to the traditional data sources, recommendation systems are categorized into content based and collaborative filtering techniques then these techniques are enhanced using the context of users. The best items that can be recommended to users are the most popular ones then many factors affect positively or negatively in the recommendation processes. Content-based RSs provide a list of items to users according to the descriptions of items and users' profiles [7], and they generate recommendation for items that similar to the accessed items by the user. This observation means very limited subset of items, from the most popular ones, can be recommended to users. In this case, the factor that affect in the recommendation process is closed to the active user's opinions i.e. user's opinions can considered as his preferences for small subset of items. The other factor is the description of the accessed items by the active user. This observation means Content based recommendation systems is not the best choice when we need to recommend the most popular items to users.

On the other hand, Collaborative filtering CF [39] techniques are widely used process in the last decades. CF techniques use users, items, and preferences of users for items to recommend items to users. Many studies investigate ways of using CF recommendation systems to cope with the problem of information. As mentioned, the best items to be recommended to users are the most popular ones. CF algorithms predict the average rating for items that can be interested by the active user. They add to the recommendation processes only the factor of similar users opinions. However, many other factors can be combined to enhance the recommendation systems. Many researchers study context-aware recommendation systems that add new factors, e.g. locations; they pre/post filter users' preferences before/after the recommendation processes. The context aware recommendation systems outperform the conventional ones, but the area needs more studies.

Social media provide a huge amount of information that can be used in recommendation systems. There are many features (e.g. friends, group, etc) that can be used to collect explicit or implicit ratings of users for items (movies, photos, etc). Social media features can be considered as factors that can be used to enhance recommendation systems. However, the mentioned recommendation techniques recommend items to users with no consideration of the moment of the recommendation processes while users rate for items at specific points of time. The time interval of any data set can be divided into periods of time; then, we can use the mentioned factors to predict trends of items popularities in general. They can be predicted according the location, gender, etc. Trends analysis and social media feature can be used to enhance recommendation systems i.e.

$$(The\ recommendation\ results) \propto (The\ trends\ of\ items\ popularities)$$

This thesis proposes Markov Chain based Recommendation Systems (MCRS) that use users' preferences for items, social media features, and the time factor to predict trends of items' popularities (items weights) that used to design a new recommendation technique to generate recommendation to users.

Figure 2-2 gives the general concept of Markov Chain Recommendation System (MCRS) that illustrated in details in chapter 3.

Figure 2-2 (A) illustrates the general weights of items. The most accessed items by users are the most popular ones.

Figure 2-2 (B) represents our new recommendation process.

Figure 2-2 (C) represents users' context. In this stage we can use the conjunction operations and the probability rules to combine these factors.

Figure 2-2 (D) gives items popularities in period of time when the recommendation is done. The result of all these processes is given in

Figure 2-2 (E).

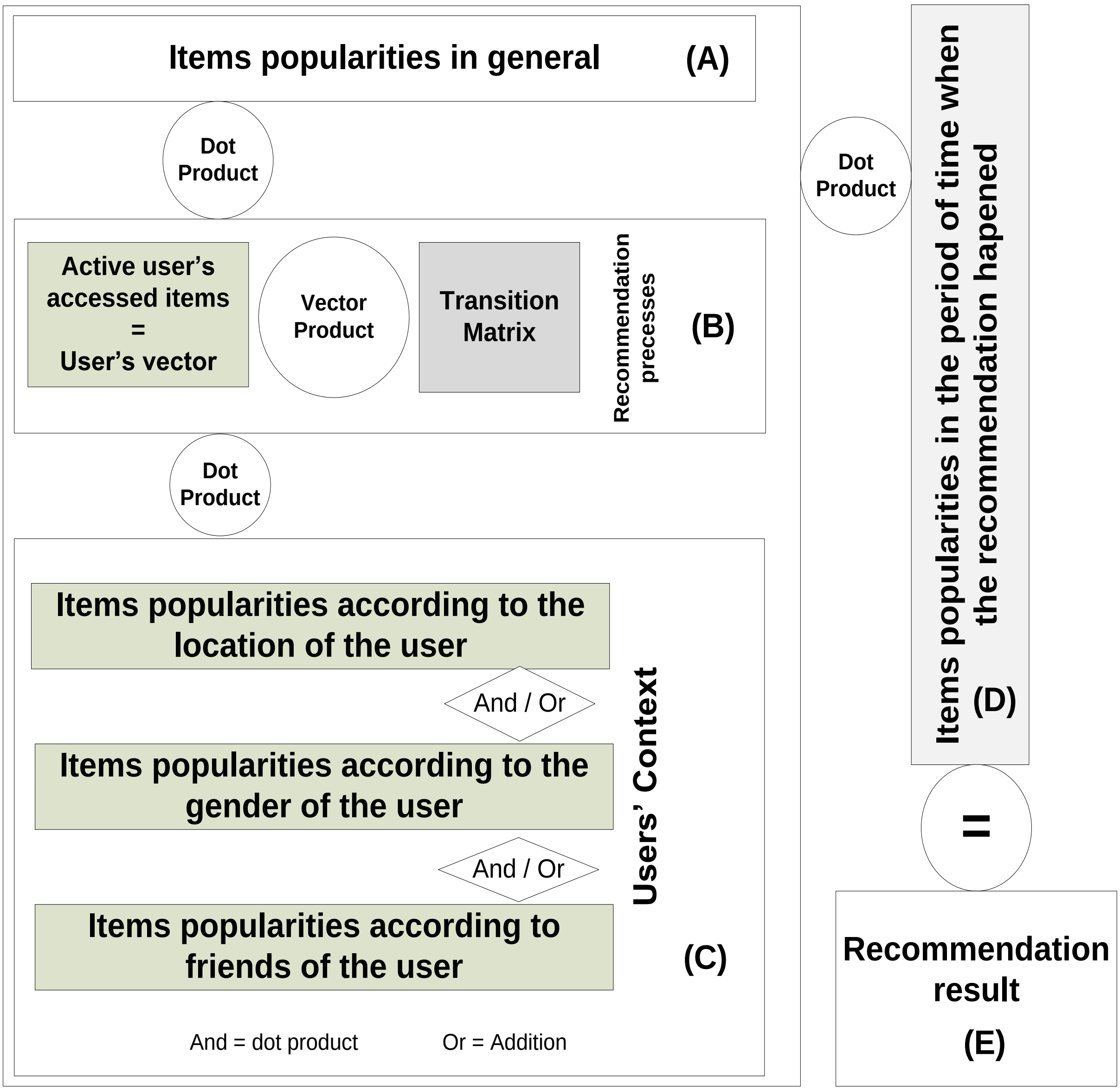


Figure 2-2: The recommendation processes with respect to general items popularities and users' context weights

2.7. Chapter summary

This chapter introduces the motivation for recommendation systems. Web applications as well as their users are exponentially increasing, and they face the problem of information overload that solved by applying recommendation systems. The chapter defines recommendation systems and gives the general concepts of collaborative filtering (CF) and content based techniques. CF is the most successfully used one. Many researchers study ways of enhancing the conventional techniques. Context aware recommendation system is an example of recommendation systems enhancements. However, social media networks add more and diverse data from different sources. The big challenges can be summarized in these questions. How to generate a trust data that can be used in recommendation systems? How to identify the suitable data sources? And how to represent the collected data before it be used in recommendation systems? On the other hand, users' interaction events are depending with the time i.e. the time factor can be used to enhance recommendation systems.

This chapter reviews way of using social media features and the time factor in recommendation systems. We conclude that the area needs more studies. Many researchers study ways of using Markov chain model in recommendation systems. However, their solutions suffer from sparsity problem, but they improve that Markov chain model can be used to design a new technique of recommendation systems.

The next chapter introduces the research methodology which describes the thesis proposed technique that based on users, items and preferences of users for items. The solution is enhanced using the time factor and the friends feature.

Chapter 3

RESEARCH METHODOLOGY

3.1. Introduction

The previous chapter introduces the literature review. It illustrates that many web applications have been published in the internet, and they provide a large number of items and services in different domains to their users. The chapter introduces the motivation of using recommendation systems and their general concepts. Moreover, Collaborative filtering CF, Content based CB, Context aware recommendation systems are discussed. These techniques have been used by many web applications to cope with the problem of information overload. However, social media websites complicate the problem since they provide to their users many features that increase users' interaction. Hence, the amount of information is exponentially increasing. This information depends on the time factor that links users' activities with time. Many techniques use the time of events in recommendation systems, but they separate users' preferences for items from the time factor. Many researchers study ways of using Markov model in recommendation systems, but their solutions suffer from the sparsity problem. In their solutions, they consider sequences of accessing items while users do not follow these sequences, and using sequences of accessing items in recommendation systems violate the accuracy. Therefore, we introduce a new technique based on users' preferences for items in the same period of time, using Markov Chain model, to recommend items to users.

This chapter represents the research methodology of the thesis. It introduces the application phases of Markov Chain Recommendation Systems (MCRS) and its enhancements, and the evaluation processes. The dataset contains information about users, items, users' preferences for items, and the time of accessing items by users. Users' ratings for items provide the data that can be used to design the new MCRS.

The rest of this chapter is organized as follows: In section 3.2, we represent the research design. In section 3.3, the research framework is illustrated in details. The data set introduced in section 3.4. In section 3.5 the data preparation. The evaluation is given in section 3.6. Then, we summarize the chapter in section 3.7.

3.2. Research design

Markov Chain Recommendation System (MCRS), that based on users' preferences for items, is represented to recommend items to users. Markov chain consists of an initial vector and a transition matrix. The active user accesses a subset of items, from all available items, in different period in the dataset time interval. The technique divides all items into two subsets. The first subset contains items that accessed by the active users; the other one contains items that do not accessed by him. The initial vector contains all items. If the active user has accessed any item then it is signed by one otherwise it is signed by zero. We then normalize the active user's vector to formulate the initial vector. If the active user has accessed n items then the probability of any accessed item is 1/n and the probability of the remained item is zero.

Example:

*Consider, the accessed items by all users are (**A**, B, C, **D**, E, **F**, and **G**).*

*If the active user accessed a subset of items (**A**, D, F, G) then the number of accessed items by the active user is **four**.*

*The active user's vector is (**1** ,0 , 0 , **1** ,0 , **1** , **1**).*

*And the initial vector is (**0.25** ,0 ,0 , **0.25** ,0 , **0.25** , **0.25**).*

The transition matrix contains rows of items, and any row represents one item and the probability of accessing all items with it. The transition matrix contains n rows, and any row contains n items. The result is the vector product of the initial vector and the transition matrix. The system suggests items of the highest probabilities to the active user as recommendations; however, items popularities vary with time. As more users access an item, its popularity increases and vice versa. Items popularities increase or decrease at any period. Therefore, we discuss the limitation of the basic MCRS. Then we introduce the motivation of enhancing the new techniques. We enhance the new proposed recommendation system using the general weights of items, which means the popularities of accessing these items. Before recommending items to users, we multiply the basic MCRS by the vector of items' popularities then we recommend items with the highest probabilities to the active user. In addition, we do an enhancement to the proposed system using the popularities of items in the last period. Finally, we enhance the system using the weights of items that accessed by friends of the active user.

The thesis contributions are:

- The basic MCRS).
- The general weights MCRS.
- The period's weights MCRS.
- The friends' weights MCRS.

In the evaluation, we compare between CF recommendation system and MCRS, and we compare between the enhancements of MCRS and the basic MCRS. The evaluation matrix consists of:

- Mean absolute error (MAE).
- Precision and recall
- Area under ROC (Receiver operating characteristic) curve.

To conduct the experiments for MCRS evaluation a datasets from LastFm dataset [148] and MovieLens [148] are used.

3.3. Research framework

This thesis proposes Markov Chain recommendation systems MCRS) then the proposed technique is enhanced using the general weights of items. We use the weights of items in the last period to do more enhancements of MCRS. For extra enhancements, we use the weights of items that accessed by friends of the active user. MCRS and its enhancements are evaluated using dataset from LastFm dataset [148] and MovieLens [148]. The research framework has four phases as follows:

Phase one: Literature review

This phase reviews the related work to the recommendation systems. Web applications provide a huge amount of information to their users, causing the problem of information overload. Many recommendation techniques have been used to address the problem. Recommendation techniques generate suggestions for items that might be interested by the active users. The most successfully used ones are Collaborative filtering and content-based techniques. These techniques ease the use of web application. Collaborative filtering techniques are based on users' rating for items. Many web site provide to their users to rate for items. These ratings are used by CF techniques to calculate the similarities between users and the similarities between items. However, users' opinions vary with the time. These variation violates the similarities calculation between users and the between items. Also, many researchers study the time factor in the recommendation process. The CF techniques suffer from many problems. Therefore, many researchers investigate ways of using users' context to enhance the conventional techniques. These enhancements outperform the conventional recommendation techniques, but they need to be more accurate. On the other hand, social media features provide a great opportunity to enhance recommendation systems. We review studies that investigate ways of using social media and time factor in the recommendation systems. We introduce in details the related works and identify their strengths and weaknesses. Then, a new recommendation system, Markov Chain Recommendation system MCRS, is proposed. It is based on users' preferences for items. The new techniques are enhanced using the general weights of items. More enhancements are done using the period weights of items and the weights of accessed items by the active user's friends.

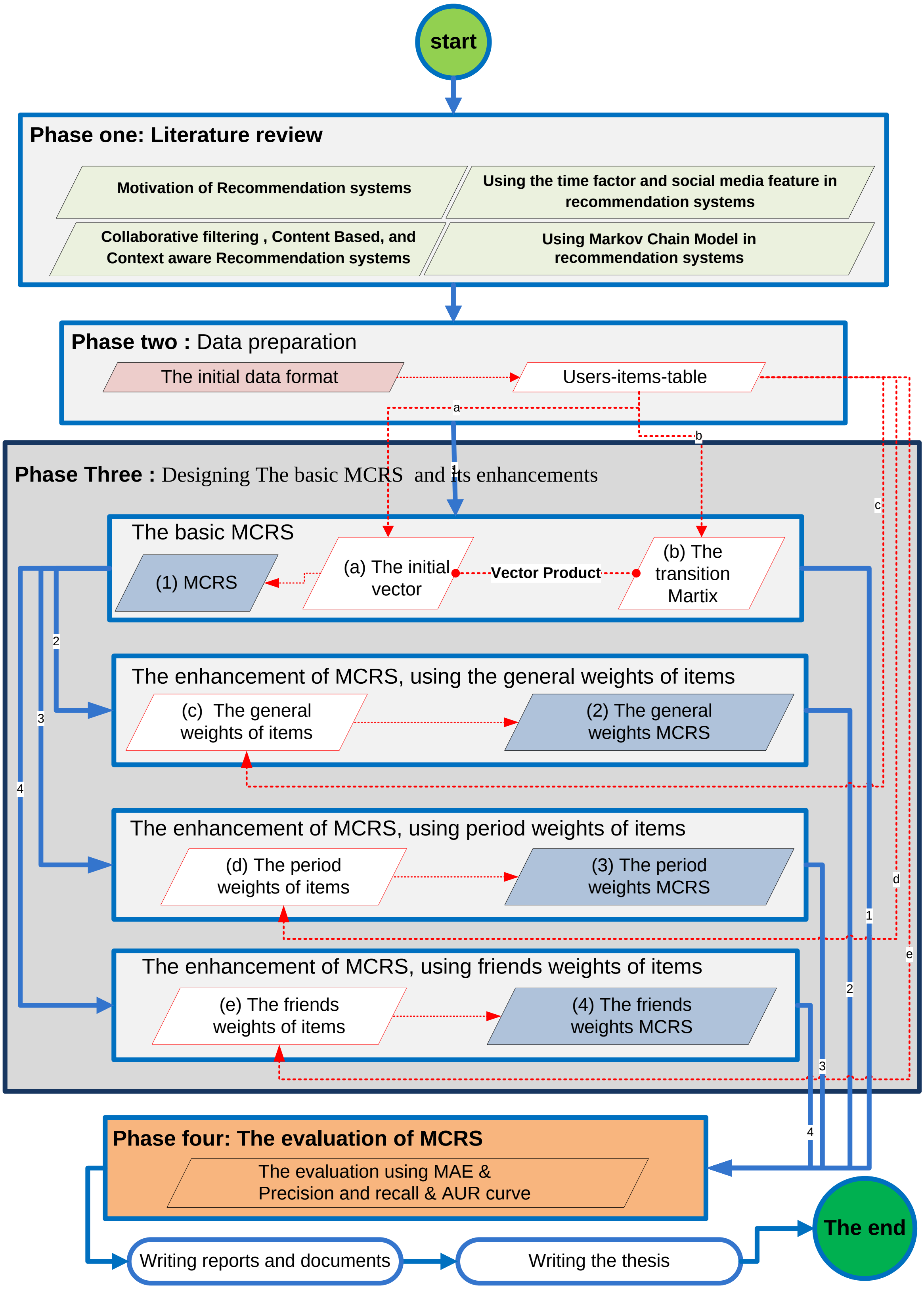


Figure 3-1:The research framework of MCRS and its enhancements

Phase two: The data preparation

The proposed techniques based on users' preferences for items. Thus, it is important to understand the data that is used to conduct the experiments for the evaluation. The datasets are released in the framework of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec 2011) [148]. There are two datasets. We take the first one from MovieLens web application. Initially, it is found in the form of the "initial-table" that contains, user Id, item Id, users' ratting for items, and the time (day, month, and year). The time interval of users' ratings on movies is divided into months. The second dataset are taken from LastFm web application. It has similar initial table and the same time format. We can use the initial-tables to formulate a new table 'user-item-table' that contains rows of users and columns of items. We can then use the new table to design MCRS and its enhancements. LastFm dataset contains a table of users' friends that used to evaluate the enhanced MCRS using the friends feature.

Phase three: Designing the basic MCRS and its enhancements

New models to recommend items to users will be designed with high accuracy. The conventional recommendation systems suffer from the problems of sparsity and information overload. Recommendation should be based on users opinion and items popularities that vary with time. On the other hand, social media features provide a huge amount of information that should be incorporated in the recommendation processes. To solve these problems, using the time factor and social media features, four models are designed as follows:

- Firstly: We design Markov Chain Recommendation System (MCRS). It solves the sparsity problem since it uses the accessed items by all users in periods of time, more details are given in Chapter 4.

- Secondly: We enhance the new proposed model using the general weights of items; because items popularities vary with the time. This enhancement improves the model more details are given in chapter 5.

- Thirdly: The model is enhanced using the period weights of items because the popularity of items in the first period is different from that in the last period. This enhancement improve the accuracy of recommending items to users. More details are given in Chapter 5.

- Finally: The model is enhanced using weights of items that accessed by friends of the active user because items popularity is affected positively or negatively by the activities of the active users friends. More details are given in Chapter 6.

- Collaborative filtering recommendation system is designed for the evaluation purposes.

Phase four: The evaluation of MCRS

In this step we evaluate the new technique.

There are four situations for the final result depending on the general weights, the period weights of items, and friends' weights of items:

- The basic MCRS: The vector product of (the initial vector) and (the transition matrix).
- General weights MCRS: The vector product of (the initial vector) and (the transition matrix) weighted by the weights of items in a period of time.
- Period weights MCRS: The vector product of (the user' active vector) and (the transition matrix) is weighted by the weights of items in a period of time.
- Friends weights MCRS: The vector product of (the user' active vector) and (the transition matrix) is weighted by friends' weights of items.

In these four situations, the highest probability items are recommended to the active user. The evaluation is done using the average precision, mean absolute error MAE, accuracy, and area under ROC curve.

- First, we compare the basic MCRS with CF recommendation system.
- Second, we compare the general weights MCRS with the basic MCRS.
- Third, we compare the period weights MCRS with the basic MCRS.
- Finally, we compare the friends weights MCRS with the basic MCRS.

3.4. Data sets

To conduct the experiments for MCRS evaluation, a dataset from MovieLens is used [148]. It contains 855598 anonymous ratings of approximately 10,197 movies made by 2,113 users. It has avg. 404.921 ratings per user avg. 84.637 ratings per movie. The dataset is released in the framework of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec 2011). The time of users' ratings on movies is divided into months. The dataset is split into periods of time. Any period contains several months. The number of all periods is 137.

LastFm dataset [148] is, also, used to conduct the experiments to evaluate the MCRS. Data statistics are as follows: 1,892 users, 17,632 artists, 12,717 bidirectional user–friend relations, i.e. 25,434 (user_i, user_j) pairs, avg. 13.443 friend relations per user, 92,834 user–listened-to artist relations, i.e. tuples [user, artist, listening count], avg. 49.067 artists most listened to by each user, avg. 5.265 users who listened to each artist 11,946 tags, 186,479 tag assignments (tas), i.e. tuples [user, tag, artist], avg. 98.562 tas per user, avg. 14.891 tas per artist, avg. 18.930 distinct tags used by each user, avg. 8.764 distinct tags used for each artist. The dataset is released in the framework of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec 2011).

Table 3-1 the initial data format of users' preferences for items (Sample data from MovieLens dataset).

User ID	Item ID	Rates	Days	Months	Years	Hours	Minutes	Seconds
75	3	1	29	10	2006	23	17	16
75	32	4.5	29	10	2006	23	23	44
75	110	4	29	10	2006	23	30	8
75	160	2	29	10	2006	23	16	52
75	163	4	29	10	2006	23	29	30
75	165	4.5	29	10	2006	23	25	15
75	173	3.5	29	10	2006	23	17	37
75	296	5	29	10	2006	23	24	49
75	353	3.5	29	10	2006	23	17	0

Table 3-2 the initial data format of users' friends table (Sample data from LastFm dataset).

User ID	Friend ID
2	275
2	428
2	515
2	761
2	831
2	909
2	1585
2	1625
2	1869
3	78

Table 3-3 the initial data format of users' tags table (Sample data from LastFm dataset).

User Id	Artist Id	Tag Id	Day	Month	Year
2	52	13	1	4	2009
2	52	15	1	4	2009
2	52	18	1	4	2009
2	52	21	1	4	2009
2	52	41	1	4	2009
2	63	13	1	4	2009
2	63	14	1	4	2009
2	63	23	1	4	2009
2	63	40	1	4	2009
2	73	13	1	4	2009
2	73	14	1	4	2009
2	73	15	1	4	2009

The dataset is used to conduct twelve experiments to evaluate the MCRS models. The time of users' activities is divided into months. Any month represents a period of time. In any experiment the periods from 1 to p are used for training; and the next two periods are used for testing and the last two periods in the training data are used to calculate the period weights of items. In the first experiment we identify p=70, then we increment p by 3 in the next experiment.

3.5. Data preparation

Initially, the data is in the form of the "initial-table" that contains, user Id, item

Id, ratting of users' for items, and the time (day, month, and year). Users can give the rating of users for items implicitly. It is an explicit rating. The time represents the moment when a user rates for an item. It is in the form (day, month, and year). Normally, users’ access (e.g. view movies) items that provided by web application in sessions. In one session, the active user can access a subset of items; and we consider these items as accessed by the same user in the same period.

We can divide the time interval, when users have accessed items, into periods. Instead of using sessions, we can use these periods. We use the initial-table to formulate a new table 'user-item-table' that contains rows of users. Any row represents items accessed by a user in specific period. The number of rows of this table equals to the number of all sessions of all users. Columns of the table are user Id, items Ids, period Id. The number of column equals to the number of items plus two. The user-item-table is the base of Markov Chain Recommendation System. We use it to create the transition matrix, the initial vector, the general weights of items, the period weights of items and the weights of the active user friends.

Markov Chain Recommendation Systems MCRS is based on users preferences for items. Items can be movies, photos, or service. We can take Users' preferences for items from their activities. By "user accesses an item" we mean user views an item or marks it as like. Users' preferences for items are saved in "the initial data" that formatted as represented in **Table 3-1**:

- User Id field: It identifies the user's number. This field is used to link users' preferences table and users' table.

- Item Id filed: It identifies the item's number. This field is used to link users' preferences table and items table.

- Users' preferences field(s): Here we sign the field by "one" if the user accesses (e.g. view, rate, or mark as like) the item otherwise we sign the field by zero. Items are rated to imply that they are viewed. Then, we need the number of items that viewed and interested at the same time. Therefore, we can exclude all low rated items to retrieve the interested items.

- The time field (s): It contains the time of users' preferences for items that can be divided into periods (Days, Months, Years)

Table **3-1** can be used to generate "users-items" table using algorithm one. Users-items table contains m records, the number of users per periods i.e. any user can access in at least one period at least one item.

```
The dataset contains users, items, rating of users for items, and the time.
Consider
"Data"      : the table that contains users' rating for items.
 "Items"    : the list of all items in the given dataset.
"Users"     : the list of all users in the training data.
"Periods"   : the number of all periods of the time.
For all p ∈ Periods in data
  retrieve records of the period p from data.
  for all u ∈ Users in  data
          index-u= the index of the user u.
          for all i ∈ Items in data
                  index-i= the index of the item i.
                  If the user u rates for the item i then
                          users-items(index-u, index-i)=1;
                  else
                          users-items(index-u, index-i)=0;
                  end if
          end for
          users-items(index-u,(the number of all items)+1)=Users-Id;
          users-items(index-u,(the number of all items)+2)=Period-Id;
  end for
end for
```

Figure 3-2 : The creation of the table of rows of users and columns of items that contains users Id column and period’s id column.

The user can access a list of items. The fact of user accesses an item can be true (1) or false (0). Algorithm one and **Table 3-1** can be used to calculate Table **3-4** which contains (n+2) columns. The first n columns represent items, the $^{(n+1)}$ column is (user id), and the $^{(n+2)}$ is (period id) which can be filled byp_t.

Table 3-4: Rows of users' sessions and columns of items

$item_1$	$item_2$	$item_3$	. . .	$item_n$	user Id	Period id
1	0	1	. . .	0	$user_1$	p_1
1	0	1	. . .	0	$user_2$	p_1
0	1	1	. . .	1	$user_3$	p_2
...	...	...	. . .	...	...	...
1	0	1	. . .	1	$user_m$	p_P

The first record represents $user_1$. When, he accesses $item_1$; we put one in the cell $_{(user1,item1)}$. When, he doesn't access $item_2$; we put zero in the cell$_{(user1,item2)}$. This table will be used later to calculate the general weights of items W, the transition matrix $T_{n,n}$, and the weights of items in the last period of time p_t. MCRS is a special case of Markov Chain. It consists of the tuples (V, $T_{(n,n)}$, W, w_p) where:

V is the active user's vector (V is used to formulate the initial vector I of Markov Chain). $T_{(n,n)}$ is the transition matrix of MCRS, the result of the vector product of I and $T_{(n,n)}$ is denoted by R in this thesis. The highest probability items of R are recommended to the active user. W is the general weight of items. w_p is weights of items in the last period of time.

Users-items table is used to design MCRS. It used to generate the following:

- The initial vector of Markov Chain model.
- The transition matrix.
- The general weights of items.
- The period weights of items.
- The friends' weights of items.

3.6. Evaluation

In this section we introduce ways of the evaluation of the new technique (Markov Chain-based Recommendation System MCRS) that can be used to recommend items to users. Consider MCRS is used to recommend the list of items R and we have the really accessed items X. Then, the evaluation can be done by comparing between R and X.

There are different ways that are used in the evaluation processes, the mean absolute error (MAE), the precision/recall and the area under ROC curve (receiver operating characteristic) AUR.

The mean absolute error (MAE)

Consider, the set (X) of the most (k) really accessed items, X={x_i: i=1,3,3...,k} where x_i is the ith item and k is the number of elements of X. And the set (R) is corresponding items of X in R, that recommended using MCRS, R={r_i: i=1,3,3...,k} where r_i is the ith item and k is the number of elements of R.

The probabilities of accessing elements of X are: $P(X)=\sum_{i=1}^{k} p(x_i) =1$.

The probabilities of accessing elements of R are: $P(R)=\sum_{i=1}^{k} p(r_i) =1$.

The mean absolute error:

$$\text{MAE (P(X), P(R))}=\frac{1}{k}\sum_{i=1}^{k}\left|\mathbf{p}\left(\mathbf{x}_{(\mathrm{i})}\right)-\mathbf{p}\left(\mathbf{r}_{(\mathrm{i})}\right)\right| \qquad 3\text{-}1$$

The best result has the less MAE.

Precision/recall and AUR:

Consider the recommended set of items is R, and it recommended to the active users using MCRS. And the set of the really accessed items X, that taken from the test data.

For any r ∈ R there are four situations (

Table **3-5**.)

A: r ∈X where r is recommended to the user and he is actually interested inr.

B: r ∈X where r is recommended to the user but he is not interested inr.

C: r ∉X where r isn't recommended to the user but he is actually interested inr.

D: r ∉X where r isn't recommended to the user and he is not interested inr.

The evaluation can be done as follows:

The first step is recommending a set of items (R) to the user u.

The second step is identifying the set X i.e. the really accessed items.

The final step is comparing X with R using

Table 3-5.

Table 3-5 the result of recommending an item to user

<table>
<tr><td rowspan="4">Interested</td><td></td><td colspan="2">Recommended</td></tr>
<tr><td></td><td>(Positive)</td><td>(Negative)</td></tr>
<tr><td>True</td><td>TP: True IN& True RD
r ∈ R& r ∈X</td><td>TN: True IN& False RD $r \notin$
R & r ∈X</td></tr>
<tr><td>False</td><td>FP :True RD& False IN
r ∈ R& r ∉X</td><td>FN : False IN& False RD $r \notin$
R & r ∉X</td></tr>
</table>

$$\textbf{Precision} = \frac{TP}{TP+FP}*100\% = \frac{Compared\ (interested\ and\ recommended)}{Compared}*100\% \qquad \textbf{3-2}$$

$$\textbf{Recall} = \frac{TP}{TP+FN}*100\% = \frac{Compared\ (interested\ and\ recommended)}{interested(recommended\ or\ not\ recommended)}*100\% \qquad \textbf{3-3}$$

False positive rates:

$$\textbf{FPR= (1-spesificity)=} \frac{FP}{FP+TN}*100\% = \frac{false(recommended\ )}{false(recommended\ or\ not\ recommended)}*100\% \qquad \textbf{3-4}$$

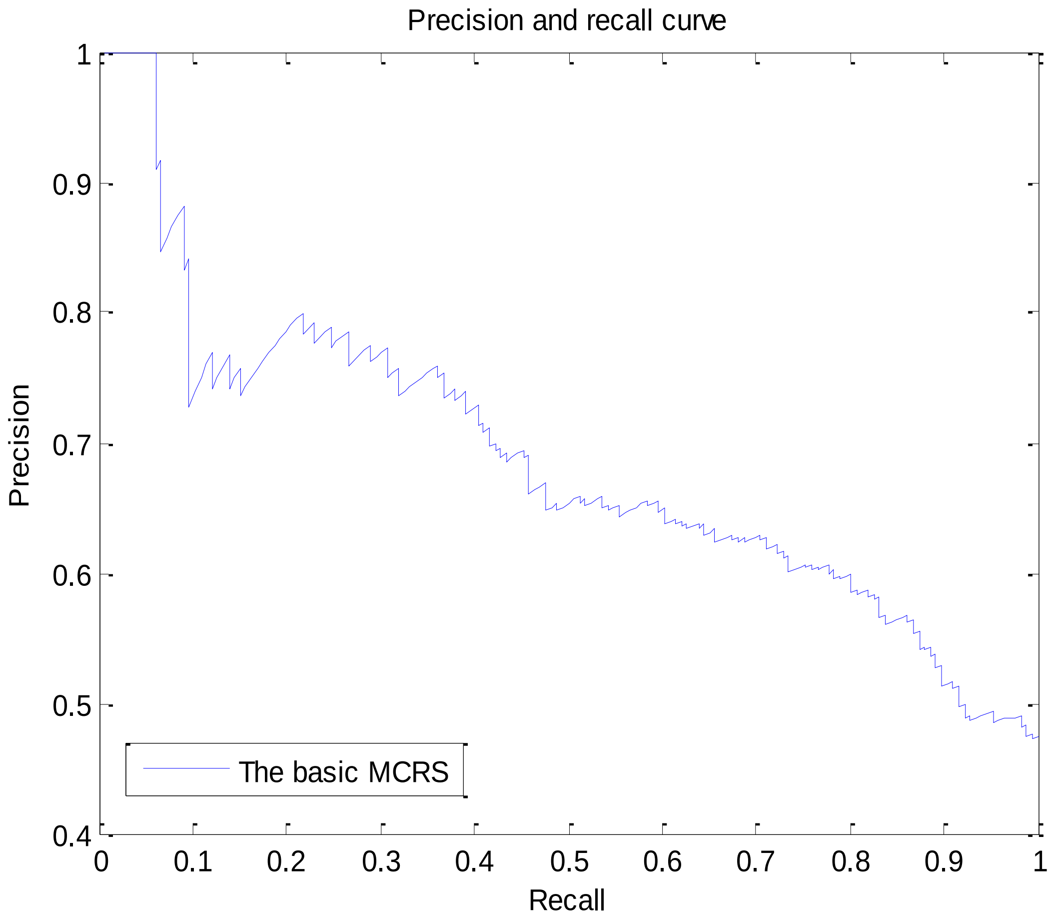


Figure 3-3 Figure 3.3: Example of the average precision

(The area under the curve)

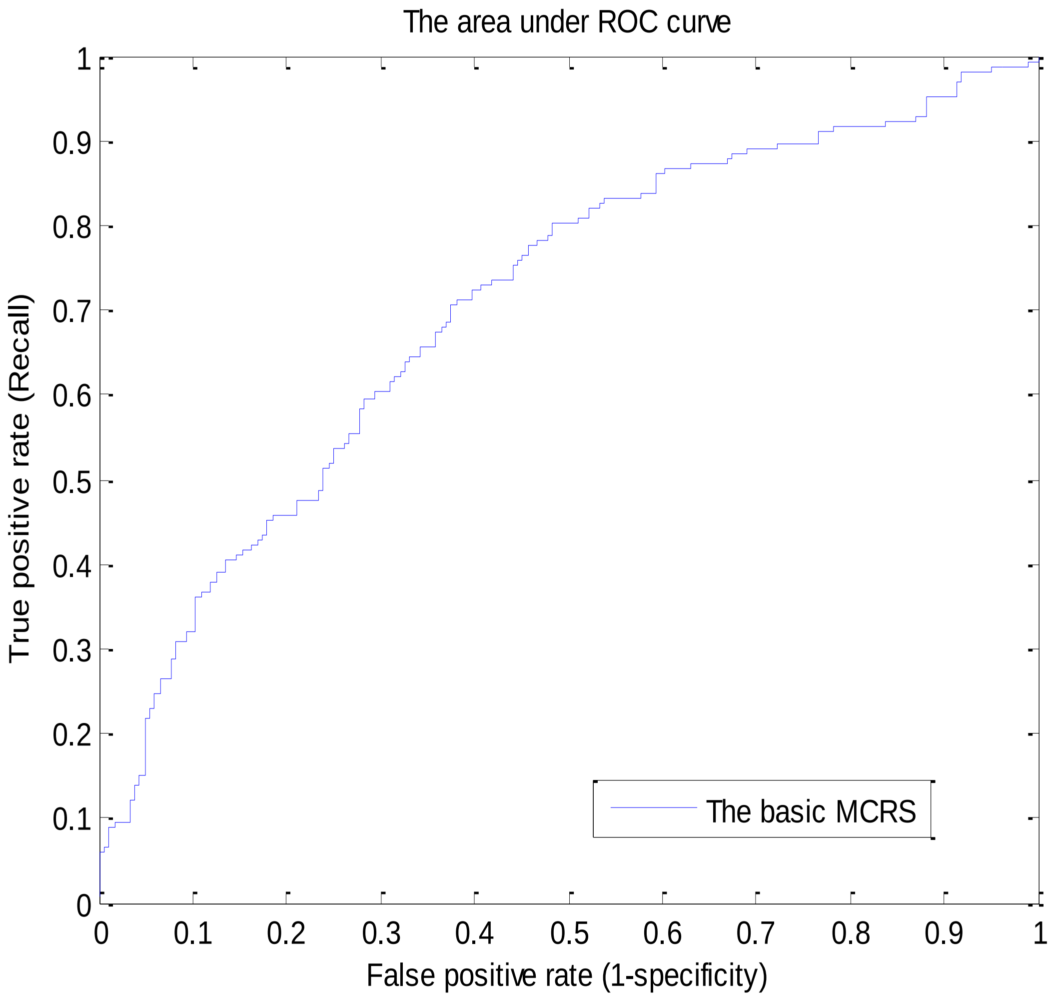


Figure 3-4: Example of the area under ROC (Receiving Operating Characteristics) curve.

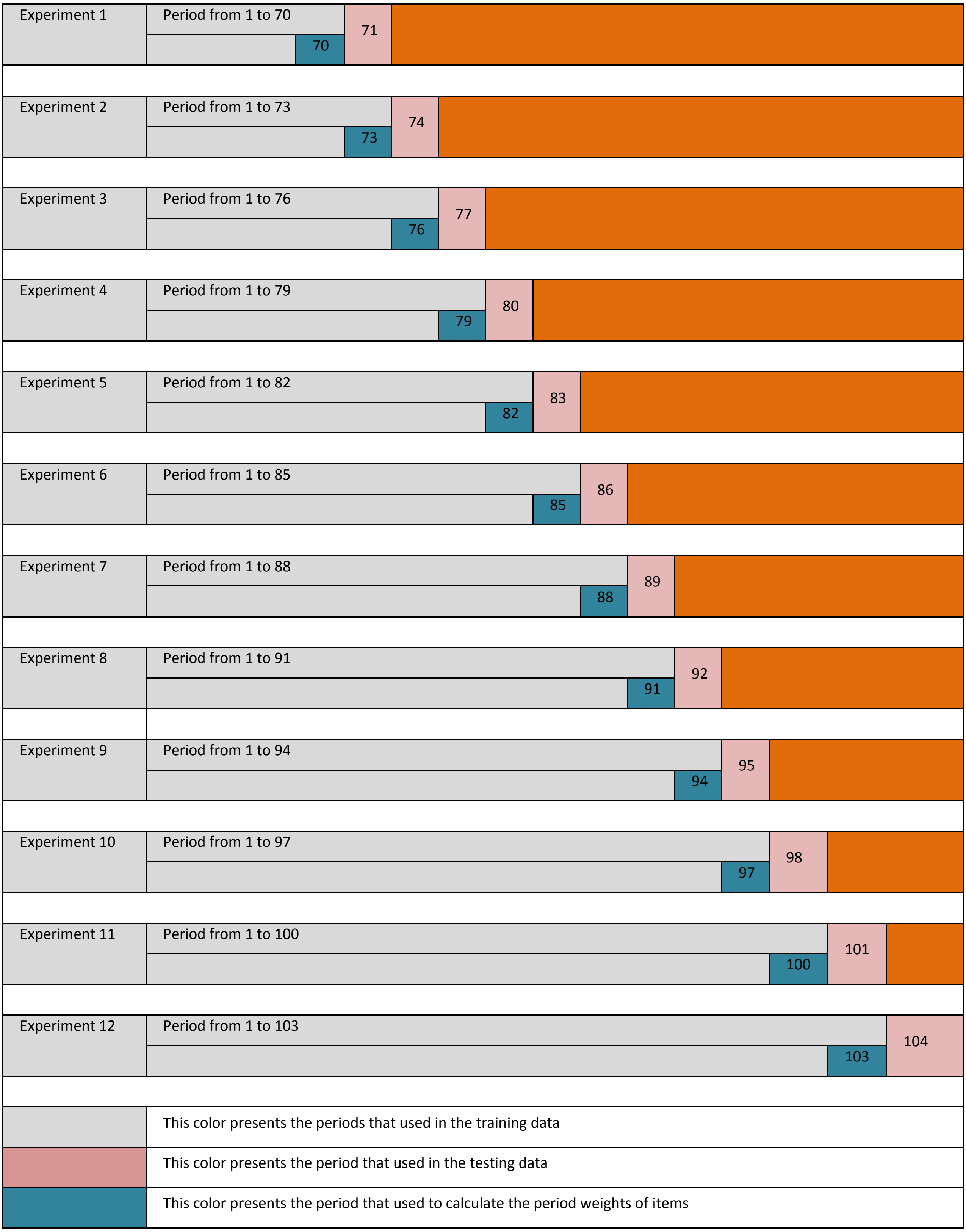


Figure 3-5 : The figure explain that how the dataset is split into training data and testing data for the twelve conducted experiments.

Experiments:

Two datasets, LastFm and MovieLens, are used to conduct twelve experiments to evaluate the new proposed model. In any one of them, the time interval of users' activities is divided into months. Any month represents a period of time. Figure 3-5 illustrates the way of dividing the time interval to M months. In any experiment the periods from 1 to p (here $1< p < M$) are used for training, the next x periods can be used for testing, the last x periods can be used to calculated the period weights and friends weights of items. In the next experiment p is incremented by y. Table **3-6** illustrates the interval division for the experimental purposes, using MovieLens and LastFm datasets.

Table 3-6 the interval division for the experimental purposes, using MovieLens and LastFm datasets.

The dataset	Months or periods	Initial value of p	Values x and y	No. of experiments
MovieLens	137	70	2, 3	12
LastFm	69	40	2 ,2	12

Example for MovieLens dataset:

The time interval is 137 months. In the first experiment we have to start by a suitable data, p months: (1) The training data is p moths. (2) The testing data is the next x months after p. (3) The last period is the last x months in p. The value of p=70, the value of x=2, and the value of y=3.

In the first experiment p=70. The training data set contains users' preferences in period from the first month to the70th month. the months (71 and 72) are used for test. The months (69 and 70) are used to calculated the period weights and friends weights of items.

In the second experiment p is incremented by 3, so p=73. The training data set contains users' preferences in period from the first month to the 73^{th} month. the months (74 and 75) are used for test. The months (72 and 73) are used to calculated the period weights and friends weights of items. The number of the experiments is 12.

3.7. Chapter summary

This chapter introduces the research methodology. It gives the literature review of recommendation systems in details to identify the research background to find limitations of recommendation systems. We propose a new recommendation system, respect to the motivations of; and we enhance It. The chapter gives the design of the research and describes the research framework that contains four phases. The used datasets are described; then, we introduce in details ways of using these datasets in the evaluation processes.

Chapter 4

THE BASIC MARKOV CHAIN RECOMMENDATION SYSTEM (MCRS)

4.1. Introduction

Recently, millions of users of web applications can access millions of items, and the amount of information is exponentially duplicated[17], [78]. Hence, websites as well as their users suffer from the problem of information overload[3], [79]; therefore, websites have been using recommendation systems to generate suggestions for items that might be interested by their users. In chapter two, we review the general concept of Collaborative filtering recommendation systems, ways of using the time factor, and ways of using Markov model in recommendation techniques.

Collaborative filtering techniques are based on users ratings for items [67]. These ratings are used to calculate the similarities between users and the similarities between items, and many techniques are used to calculate these similarities [106] e.g. Vector Cosine similarities and Pearson correlation coefficient, and many techniques are used to predict users' rating for unknown items. The prediction is computed by taking a weighted average of items that rated by similar users to the active user. The k nearest neighbour algorithms is one of the most successfully used techniques to predict users' ratings for unknown items and to recommend a list of interesting items to the active user [105]. These techniques have some limitation in ways of calculating the similarities between users or items.

Shani et. al (2005) [36] use Markov Chain model in the recommendation processes; they propose An MDP-Based (Markov Decision Process) Recommender System. Their technique is based on the most frequents sequences of k items to calculate the probability of accessing an item that follows the sequence of items. Their technique has some limitations.

The rest of the chapter is organised as follows. Section 4.2 finds the limitations of the conventional recommendation techniques and the limitation of ways of using Markov model in recommendation systems. The motivation of the new techniques is given in section 4.3. The basic MCRS is introduced in section 4.4. In section 4.5 and section 4.6 we have the experimental design and the results. Then, we summarise the chapter in section 4.7.

4.2. The limitation of the conventional CFRSs

Collaborative filtering techniques are based on the similarities between users or items [67] that are calculated using the similarity algorithms, then the K nearest neighbour algorithms are used to generate the suggestions for items to users, using these similarities. On the other hand, the conventional Markov model techniques are based on the sequences of accessing items by users while many users do not work to access items in sequences [36]. The limitation of these techniques can be considered as follows:

- **The first limitation:** The similarity between two users depends on their accessed items [15], [100]. If two users access the same subset of items then they are similar, and if they access different items and share the accessing of others then they are partially similar; otherwise, they are not similar. However, users' opinions vary with the time because any user can be interested to access items in the earlier periods of time then he may change his opinions, and he later accesses different kind of items. He might be no longer interested in the first ones. If we look to users' accessed items in the long term then we clearly find that no relationship can be used to link between items, and users' opinions cannot be used in this relationship.

Example:

Consider, the user u1 has viewed and interested in movies (A,B,C,D); the user u2 has viewed and interested in movies (A,B,C,D), and the user u3 has viewed and interested in movies (F,G,D,J) in the first month for one year. Then, users' opinions might be changed in the same year. Consider, the user u1 has viewed and interested by movies (F,G,D,J); the user u2 has viewed and interested by movies (V,N,M,K), and the user u3 has viewed and interested by movies (Q,R,T,Y) in the tenth month in the same mentioned year. Then, it clear that users (u1 and u2) are (100%) similar to each other in the first month, and they are not similar in the tenth. In general, if we look to users in long term, u1 and u2 are partially similar, the same as u1 and u3.

This means users' similarities are positively or negatively affected by the time factor. The limitation of Collaborative filtering techniques is the lack of using the time factor in users' similarities calculation.

- **The second limitation:** The similarities between items depend on users rating for these items [103] [106] [39]. Any item might be accessed by users of different opinions because new items might be interesting all users that can access some of them. Few users can also access these new items in short term. However, items similarities is based on users' preferences for items while new items are only accessed by few users. This means, the approaches that are based on the similarity between items can produce inaccurate lists of recommendations, and new items might not be recommended.

- **The third limitation:** Limitations in ways of using the time factor in recommendation processes. Yi Ding and Xue Li [25] propose the time function f(t). The time interval is divided to t periods and the values of the function lies in the range (0,1) i.e. while old ratings are weighted by smaller weights, the recent users' preferences are weighted by the higher weights. Wang et al. [54] introduce 'Temporal Summaries' to invest the time factor in the recommendation process. Kostas et al. [149] propose a framework for time-aware recommendations that improve the recommender accuracy. However, their solutions separate between events of accessing items and the time. The value of their function f(t) lays between 0 and 1, and it intend to zero in the first period and intend to one in the recent periods; moreover, the values of f(t) is equal for all items at any point of time. But, these values must be randomly distributed according to items popularities, and they must be different from any item to others because item popularities vary with time and depend on users' preferences for it.

- **The last limitation:** Markov chain recommendation techniques are based on sequences of accessing items [36], [140]–[143]. They aim to predict the item that follows a sequence of items. Shani et al. [36] propose an MDP-Based (Markov Decision Process) Recommender System. The states in their model represent the relevant information about the active user. Their technique considers only the most frequents sequences of k items , and they have k=5. But, the law order needs user to access less number of items. The law orders violate the accuracy as users can access more items. The high order result in better accuracy, but they increase the complexity of the application.

These limitations can be fixed by using new technique that considers the variation of users' opinions and items popularities with time.

4.3. The motivation of new recommendation techniques

Recommendation systems have been used by many web applications to ease their usage and to generate suggestion for items to users. The most successfully used techniques are collaborative filtering recommendation systems. However, these techniques have some limitations as mentioned in Section 4.3. New techniques are needed to address these limitations as follows:

- The new technique can be used to solve the first limitation and consider users' accessed items per session or period of time. These techniques look to users' preferences in short terms. This means the new techniques consider the variation of users' opinions with time.

- The new technique can be used to solve the second limitation using the same idea. Here, we consider items that have been accessed by the same user in the same period of time.

- The new techniques can also be used to solve the third limitation. They consider the time when users have accessed items and the variation of users' opinions and item popularities.

- The new techniques can be used to solve the last mentioned limitation; since, it don't considers the sequence of accessing items and only look to items that accessed by the same user in the same session or period of time.

4.4. Markov Chain Recommendation System (MCRS)

We propose **Markov chain recommendation system (MCRS)** that consists of four main components [59], represented as follows :

- **State:** we consider that any item represents state. When, a user accesses an item then he will access with it another item(s). We say the target user moves from an item to another item(s).

- **The relation between states:** the relation between items is given by the feature: accessing items by the same user in the same period of time e.g. if the user U has accessed the set of item (A,B,C) in the period of time p, then the relation between A, B, and C is ' The user U has accessed items A, B, and C in the period p'.

- The initial vector that contains the starting states i.e. the set of items that the active user has been accessed at the recommendation moment.

- The transition matrix that represents the probabilities of moving from any state to others e.g. the probability of accessing all items with any item. If the set of all items is S=(A,B,C,D), then the transition matrix has four rows and four columns. Any row represents the probabilities of accessing any item with all items.

The result of the proposed recommendation system is the vector product of the initial vector and the transition matrix. Then, items of the highest probabilities are recommended to the active user.

Users' access items when they visit web applications. Recommendation systems generate suggestion for items that might be interested to users. Initially, any user might choose a random item to access. In this case, the item can be interested by the user or not. Instead of that, he can access the most popular item. So, the website in this case needs to use a suitable tool to suggest the most popular item. Another solution, users can use search engine to find the interesting one. If the active user have accessed his first interesting item then our new proposed technique starts working to recommend items to him according to his previous accessed item. The proposed technique aims to recommend items that have been accessed by all users with the first item.

For example: if the first accessed item by the active user is A then the basic MCRS suggests the most accessed items by all users with A. Hence, we retrieve sessions of all users that access A to calculate the probability of accessing items with it.

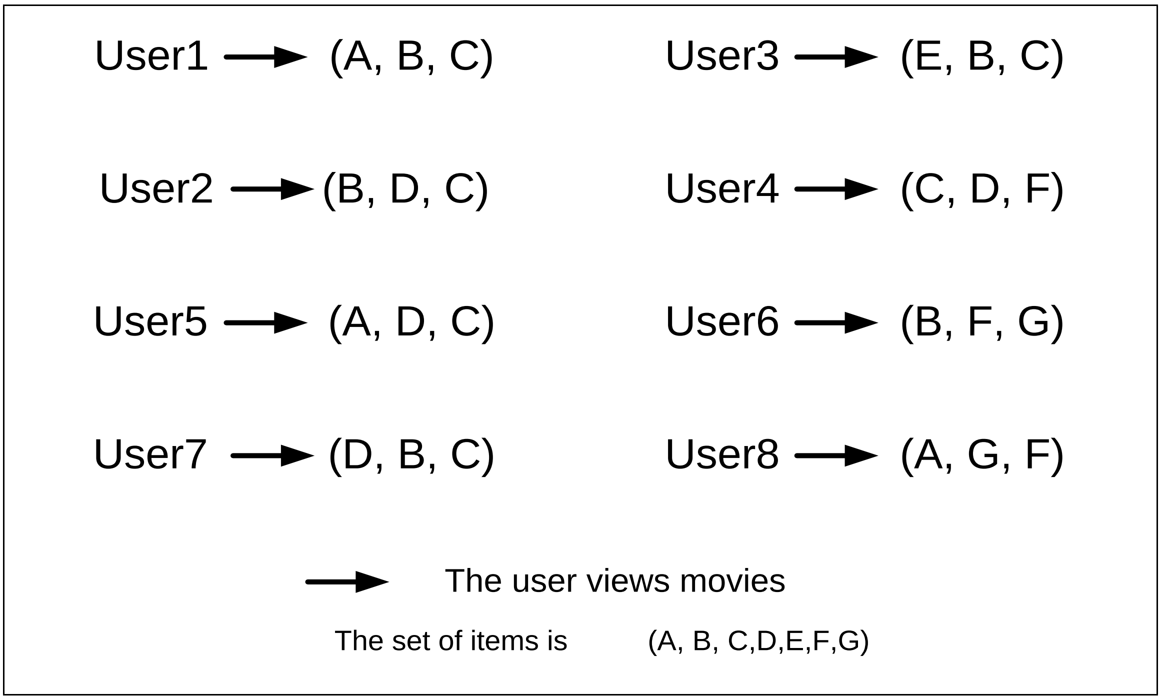


Figure 4-1 : The viewed movies by all users with the movie A.

Figure 4-1 illustrates that three users have accessed item A; and the accessed items with item A are (B,C,D,F,G).

The statistic of accessing items with item A is given as follows:

A = 3, B =1, C=2, D=1, E=0, F=1,G=1 .

All items are accessed 9 times.

The probabilities of accessing items with item A can be given as follows:

p=(3/9, 1/9, 2/9, 1/9, 0, 1/9, 1/9).

The most accessed item with item A is item C.

If the user access two items then the proposed recommendation system suggests items to him according to items A and C.

4.4.1. Probability of accessing items by the same user in the same session (PASS)

In this thesis, we introduce a new technique to generate new relations that can link between items. We use all users' accessed items to calculate the (Probability of accessing items by the same user in the same session (PASS)). Normally the user U accesses a list of items in one session. We consider all sessions of all users to calculate the vectors of accessing all items with item j, $S_{(ji)} = \{r_{ji}: i, j = (1,2,3, \dots, n)\}$ where n is the number of all items r_{ji} is the explicit rating of users for item i that is accessed with item j in the same session. If the user U accesses items (i and j) in the same session, then the rating $r_{ji} = 1$ otherwise it's $r_{ji} = 0$. Our goal is to calculate the vector of accessing all items with item j in the same session, where j=1,2,3,...,n .

$$\boldsymbol{SS}_{(\boldsymbol{ji})} = \{\sum_{(\boldsymbol{All\ usrs'\ sessions\ that\ contains\ item\ j})}^{\boldsymbol{All\ users'\ sessions}} (\boldsymbol{r_{ji}}) : \text{i=1,2,3,...,n}\} \tag{4-1}$$

The probabilities $P_{(ji)}$ of accessing all items with item j can be given as follows:

$$\boldsymbol{P}_{(\boldsymbol{ji})} = \{\boldsymbol{p}_{(\boldsymbol{ji})} = \frac{\boldsymbol{SS}_{(\boldsymbol{ji})}}{\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{SS}_{(\boldsymbol{ji})}} : \mathbf{j}, \mathbf{i} = \mathbf{1}, \mathbf{2}, \mathbf{3}, \dots, \mathbf{n}\} \tag{4-2}$$

$$P_{(ji)} = \begin{pmatrix} p_{(11)} & p_{(12)} & p_{(13)} & & p_{(1n)} \\ p_{(21)} & p_{(22)} & p_{(23)} & \cdots v & p_{(2n)} \\ p_{(31)} & p_{(32)} & p_{(33)} & & p_{(3n)} \\ & \vdots & & \ddots & \vdots \\ p_{(n1)} & p_{(n2)} & p_{(n3)} & \cdots & p_{(nn)} \end{pmatrix} \qquad (4\text{-}3)$$

$$\sum_{i=1}^{n} p_{(ji)} = 1 \qquad (4\text{-}4)$$

Our new technique solves the problem of the sparsity since it considers all sessions of all users i.e. it guarantees the calculation of the probabilities of accessing all items with any item. Any user can sit for several sessions in different periods of time i.e. the new technique ties events of user's activities to the time of the session.

4.4.2. The basic MCRS

The main components of Markov Chain Recommendation System are the initial vector and the transition matrix. To generate the initial vector, we need to understand the active user's vector.

The active's user:

The active user's vector is the target of the recommendation system. It represents all items that have been accessed by all users, which can be divided into two subsets. The first subset contains items that accessed by the active user which can be used to recommend items from the other subset that are not accessed by him. The first set "set-A" contains s items.

set-A={i_j=1 : j=1,2,3 ... s for 1<=s<=n} **(4-5)**

here s is the number of accessed items by the active user and n is the number of all items. The second set set-B contains (n-s) items set-B={i_z: z=1,2,3, ..., (n-s)}. The active user vector is V=(set-A)∪(set-B). Normally, items are distributed and items of the active user's vector are not sorted.

Example:

$$\mathbf{V=\{1,1,0,0,1,0,0,0,0,0,1,1,0,0,1,0,1,0,0,0\}} \qquad \textbf{(4-6)}$$

In this case n=20 and s=7.

In the above example seven items are accessed by the active user and signed to 1. They can be used to recommend items from the other thirteen items which are signed to 0. The active user's vector can be represented as follows:

$$\textbf{V= \{}i_e\textbf{: z=1,2,3,...,n where n is the number of all items\}} \qquad \textbf{(4-7)}$$

Procedure two can be used to generate the active user vector.

Procedure two:

 # Users-items table (Table **3-4***) is used to generate the user' vector that used to generate the initial vector).*

Consider

"User" : the active user Id.

"Items" : the list of all items in the given dataset.

retrieve "User" from the users-items table.

user-vector = null;

for all records that contains "User"

    user-vector = user-vector + (fields of Items in the record);

end for

Figure 4-2 : Figure 4.2: The creation of the active user vector.

The initial vector:

Markov chain initial vector 'I' (Figure **3-1** (a), section 3.3, page 63), equals to the active user's vector V divided by the number of times of accessing all items by the active user i.e. the summation of items accessed (ones) in the active user's vector for every item divided by the number times of accessing all items by the active user.

$$I=\frac{\boldsymbol{V}}{(\textbf{\textit{The number of times of accessing all items by the active user}})} \qquad (4\text{-}8).$$

'I' is the initial vector that represents the probabilities of accessing items by the active user.

The transition matrix of MCRS

Table **3-4**, (section 3.5 page 68), can be used to formulate Markov Chain transition matrix $T_{(n,n)}$ (Table **4-1** page 91) where n is the number of all items. Any row in $T_{(n,n)}$ represents an item and items that has been accessed with it by the same user in the same period of time. $Row_{(i,j)}$ is the row of $item_i$ where i = 1,2,3,...,n rows and j=1,2,3,..., n columns, of items that accessed with $item_i$ i.e. any item has row and column. The value $p_{(i,j)}$ is the probability of accessing $item_j$ with $item_i$ in the same period of time. It can be calculated from the retrieved rows that have the value 1 in the column of the item. The probability vector of that item is the summation of the retrieved rows divided by the summation of these rows cells, see equation 11. This vector gives the row of $item_i$ in the transition matrix.

$$\mathbf{T}_{(\mathbf{i},\mathbf{j})}=\mathbf{p}_{(\mathbf{i},\mathbf{j})}=\frac{\sum_{\mathbf{i}=1}^{\mathbf{n}}(\mathbf{row}_{(\mathbf{i},\mathbf{j})} \textbf{ where the column of item}_{(\mathbf{i})}=\mathbf{1})}{\sum_{\mathbf{j}=1}^{\mathbf{n}}\sum_{\mathbf{i}=1}^{\mathbf{n}}(\mathbf{row}_{(\mathbf{i},\mathbf{j})} \textbf{ where the column of item}_{(\mathbf{i})}=\mathbf{1})} \qquad (4\text{-}9)$$

Table 4-1: Markov Chain Transition Matrix

	$item_1$	$item_2$	$item_{13}$	. . .	$item_n$
$item_1$	$p_{(1,1)}$	$p_{(1,2)}$	$p_{(1,3)}$	. . .	$p_{(1,n)}$
$item_2$	$p_{(2,1)}$	$p_{(2,2)}$	$p_{(2,3)}$	. . .	$p_{(2,n)}$
$item_3$	$p_{(3,1)}$	$p_{(3,2)}$	$p_{(3,3)}$	. . .	$p_{(3,n)}$
…	…	…	…	. . .	…
$item_n$	$p_{(n,1)}$	$p_{(n,2)}$	$p_{(n,3)}$	. . .	$p_{(n,n)}$

The basic MCRS is the vector product of (the initial vector) I and (the transition matrix) $T_{(i,j)}$.

$$R=I*T_{(i,j)} \qquad (4\text{-}10).$$

The result the equation (23) is the vector R that contains the probabilities of accessing items by the active user. We sort these probabilities descending.

Then, items of the highest probabilities are recommended to the active user.

4.5. Experimental design

MovieLens dataset is used to conduct twelve experiments to evaluate the MCRS model. The time interval of users' activities is divided into months. Any month represents a period of time. The number of all periods is 137. In any experiment the periods from 1 to p (here 1< p <137) are used for training; and the next two periods are used for testing. In the first experiment we identify p=70, then we increment p by 3 in the next experiment.

CF user-based algorithm and MCRS technique can be used to recommend items to an active user. The active user can access a set "A" of |A| items. Then, there is a set "B" of (n-|A|) items are not accessed by the active user. Items of A are known; but items of B are hidden, and needed to be recommended.

To evaluate MCRS, we can use the set A that accessed by the active user to recommend items to the active user, from the set B which is hidden. The dataset in any experiment split into training data and testing data. The training data can be used to recommend items to the active user as follows:

- In any experiment we identify the active user. His accessed items (elements of A) are used by all models to recommend items; and they used to identify the really accessed items from the testing data.

- The set A can be used by CF user-based algorithms to generate "CF result".

- It can be used by the basic MCRS to generate "The basic MCRS result".

The testing data can be used to retrieve the really accessed items as follows:

- The set A is used to retrieve "The really accessed items" from the testing data i.e. we retrieve all records that contains any item accessed by the active user. Then, we can calculate the accessibility of items by the summation of all retrieved records. Then, we can normalize the vector; such that the summation of the accessibility of all items equal to one.

The mean absolute error MAE:

The evaluation can be done using the mean absolute error MAE. CF and the basic MCRS results can be compared with the actually accessed items using MAE. From the actually accessed items we can identify the set X of the k highest probability items. The probabilities of accessing items of X can be normalized such that sum(P(X))=1. From the CF and the basic MCRS results we can list the corresponding items of X, "CF-result" and "MCRS-result" respectively. Then we can normalize the probabilities of accessing items of CF-result and items of MCRS-result such that sum(P(CF-result))=1 and sum(P(MCRS- result))=1. We can find the accessibility mean absolute error of CF and X, and of the basic MCRS and X. The best result has the less MAE.

The mean absolute error $\mathrm{mae}(e)$ can be calculated for the twelve tests, for e=1,2,...,12. Then, the average mean absolute error can be calculated as follows:

$$\mathbf{Average\_mae} = \frac{1}{12}\sum_{e=1}^{12} \mathbf{mae}(\mathbf{e}) \qquad 4\text{-}11$$

The mean average precision:

The evaluation can be done using the mean average precision. In this case, the K highest probability items can be taken from the really accessed items and the different results. The best result has the highest mean average precision.

Precision and recall are single-value metrics based on number of the recommended and interested items by the system to users. The recommendation is a sequence of items, and it is better to consider the order in which the recommended items are presented. Then, we compute precision and recall at every position in the ranked sequence of the recommendation; we can plot a precision-recall curve, the precision is x-axis and recall is y-axis. Precision p is a function of recall r. Average precision computes the average value of p(r) over the interval of r= (0,1).

$$\mathbf{Average\_precision} = \int \mathbf{p}(\mathbf{r})\mathbf{dr} \qquad 4\text{-}12$$

This means we calculate the area under the precision-recall curve. The same value of average precision can be calculated using the following equation:

$$\mathbf{Average\_precision} = \sum_{\mathbf{k=1}}^{\mathbf{K}} \mathbf{p(k)\Delta r(k)} \qquad 4\text{-}13$$

where small k is the index in the list of recommended items, capital K is the number of recommended items, $p(k)$ is the precision at the index k, and $\Delta r(k)$ is the change in recall from items number (k-1) to k.

The average precision, $\text{Average\_precision}(e)$, can be calculated for the twelve tests for e=1,2,...,12. Then, the mean average precision can be calculated as follows:

$$\mathbf{mean\_Average\_precision} = \frac{\mathbf{1}}{\mathbf{12}} \sum_{e=1}^{\mathbf{12}} \mathbf{Average\_precision}(\mathbf{e}) \qquad 4\text{-}14$$

4.6. Experimental results

A dataset from MovieLens is used in the evaluation processes. It split into months and each month represents a period of time. Twelve experiments are conducted and in each experiment the dataset split into training data and testing data. The actually accessed data is retrieved from the testing data, using the active user's accessed items. The same active user's data is used by CF and MCRS techniques to recommend items to him. The basic MCRS result is compared with standard RS (CF based on vector cosine similarity). Finally, we analyze and discuss the results.

4.6.1. The dataset

To conduct the experiments for MCRS evaluation, a dataset from MovieLens is used [148]. It contains 855598 anonymous ratings of approximately 10,197 movies made by 2,113 users. It has avg. 404.921 ratings per user avg. 84.637 ratings per movie. The dataset is released in the framework of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec 2011). The time of users' ratings on movies is divided into months. The dataset is split into periods of time. Any period contains several months. The number of all periods is 137.

4.6.2. The basic MCRS VS CF user-based results

To evaluate the basic MCRS twelve tests are conducted. In the first test the periods from 1 to p=70 are used for training. And the periods 71 and 72 are used for testing. Then for test from (2 to 12) we increment p by 3 periods. In the last test, periods from 1 to 103 are used for training and the periods 104 and 105 are used for test. Twelve users are used in the twelve tests. In any test the same user data is used by the basic MCRS and CF user-based algorithm in the training data to formulate the models. And the same user accessed items are used to identify the really accessed items from the testing data.

There are two cases with respect to the result of the recommendation results. The first case includes the active user's accessed items in the recommendation results. The second case excludes these items from the results.

Including the active user's accessed items in the results

In this case, the active user's accessed items are included in the list of the actually accessed items that are retrieved from the testing data i.e. the recommendation lists, which are generated by the basic MCRS and CF user-based algorithms. It can include items that are accessed by the user in the training data. In the second case we exclude these items from the recommendations lists and the actually accessed items i.e. we recommend novel items.

In this case the basic MCRS and CF user-based algorithm have similar result with small variation as shown in Figure 4-3. The result ,the average precision, of CF is **0.887218177** and the result the basic MCRS is **0.874561479**. CF outperforms the basic MCRS by **0.012656698**.

Test of Significance

Table 4-2 The average precision of the basic MCRS VS the CF user-based

The basic MCRS	**The user-based CF**
0.8368	0.8512
0.861	0.8404
0.8861	0.8591
0.8407	0.8646
0.863	0.8821
0.8167	0.8587
0.8133	0.8684
0.9084	0.9255
0.9091	0.927
0.9225	0.9314
0.9163	0.9203
0.9263	0.9175

Unpaired t test

Mean of the basic MCRS = 0.875017 (n = 12)

Mean of the CF user-based = 0.887183 (n = 12)

Assuming equal variances

Combined standard error = 0.015612

The degree of freedom df = 22

t = 0.779311

One sided P = 0.222

Two sided P = 0.4441

95% confidence interval for difference between means = -0.044544 to 0.020211

Power (for 5% significance) = 18.18%

<u>Assuming unequal variances</u>

Combined standard error = 0.015612

df = 21.217126

t(d) = 0.779311

One sided P = 0.2222

Two sided P = 0.4444

95% confidence interval for difference between means = -0.044544 to 0.020211

Power (for 5% significance) = 10.39%

<u>Comparison of variances</u>

Two sided F test is not significant

No need to assume unequal variances

However, our target is the suggestion of novel items to the active user; therefore, Items that have been accessed by the active user must be excluded from the recommendation results before the evaluation.

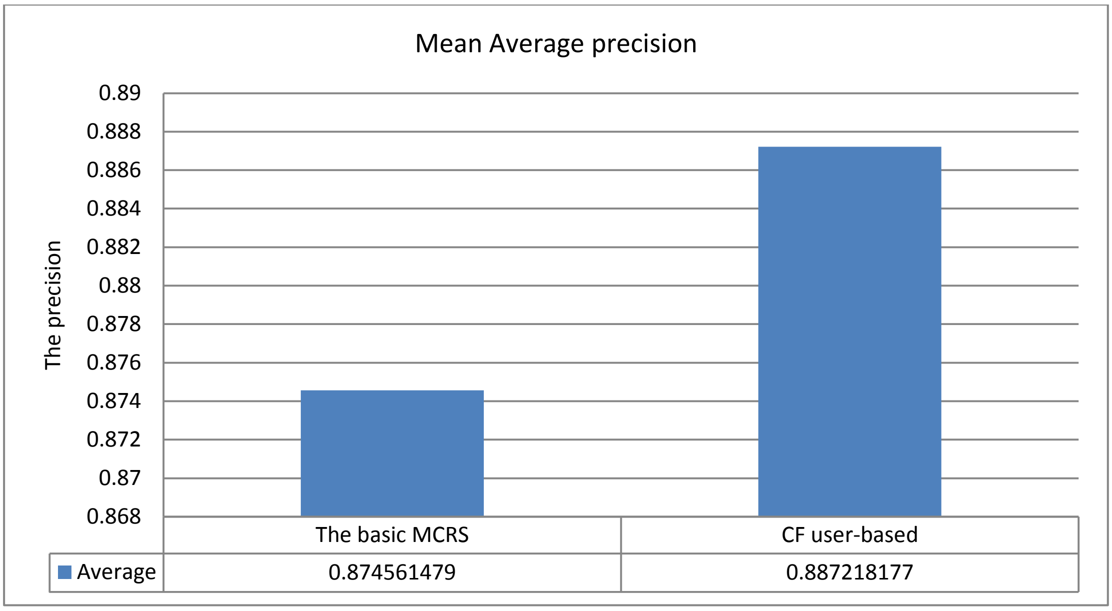


Figure 4-3 : Mean Average precision of the basic MCRS VS CF user-based algorithm

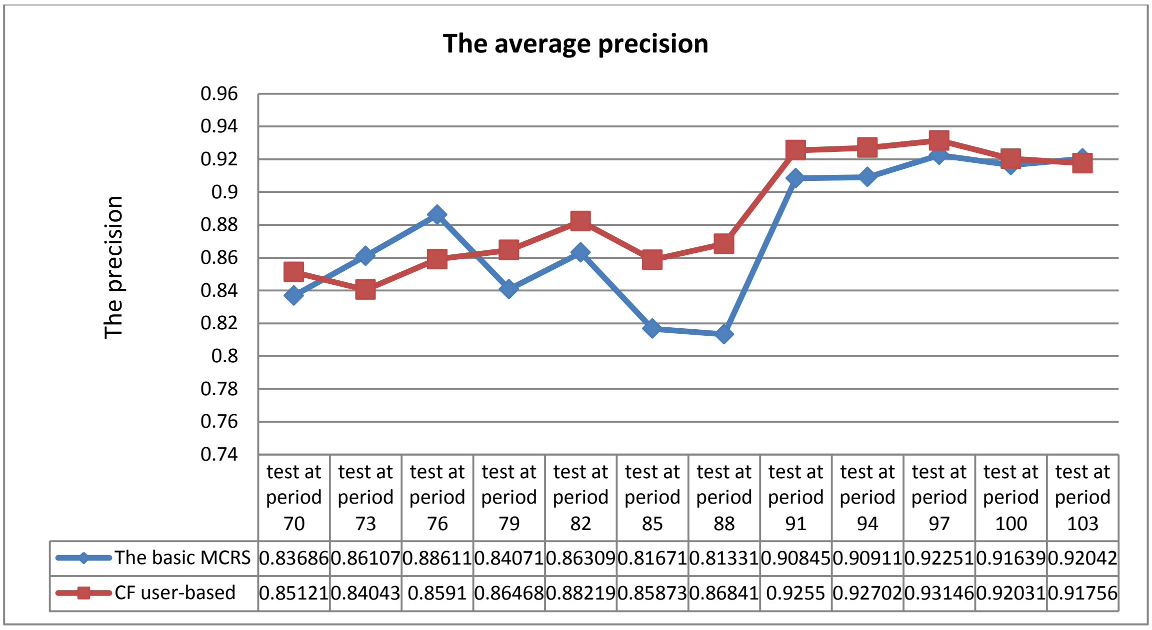


Figure 4-4 : The average precision of the basic MCRS VS the CF user-based algorithm in the case of including the active user's accessed items in the recommendation results

Excluding the active user's accessed items from the results

In the second case items that have been accessed by the active user are excluded from the results. The evaluation is done using the mean average precision, and the mean absolute error (MAE).

The mean average precision:

In the second case, the active user's accessed items are excluded from the recommendation lists and the really accessed item. In this case the basic MCRS has the mean average precision (**0.826424068**); it is better than the CF user-based algorithm by (**0.539729201**) as represented in Figure 4-5.

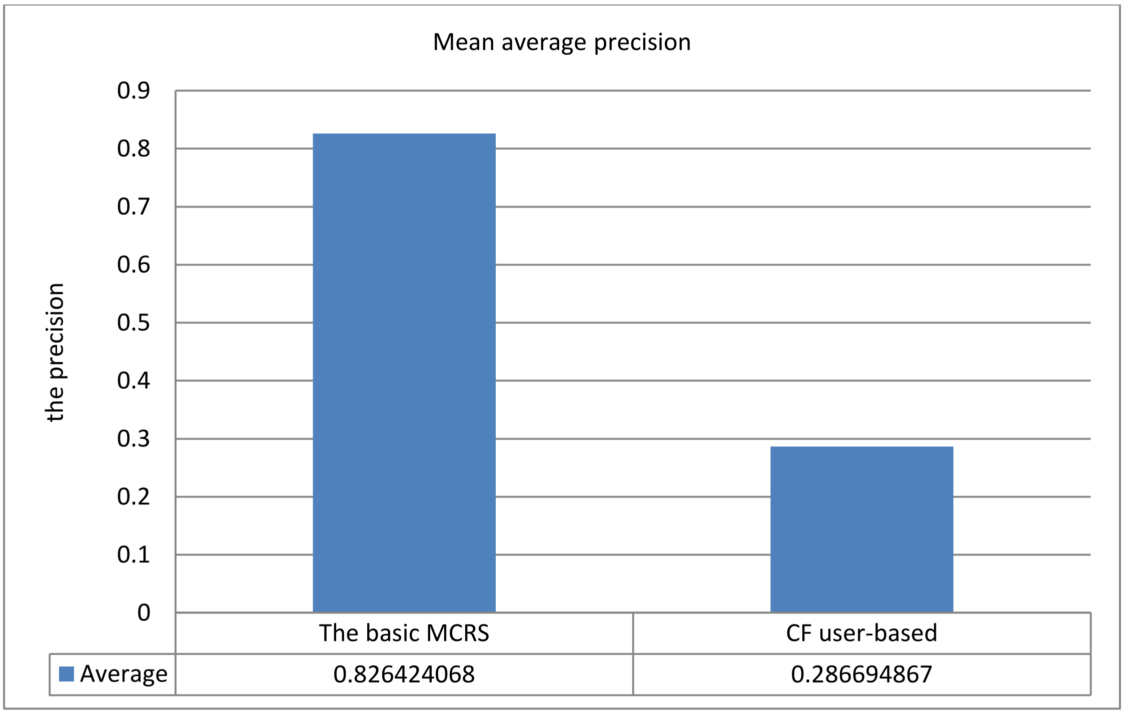


Figure 4-5 : The mean average precision of the basic MCRS VS the CF user-based algorithm in the case of excluding the active user's accessed items in the recommendation results

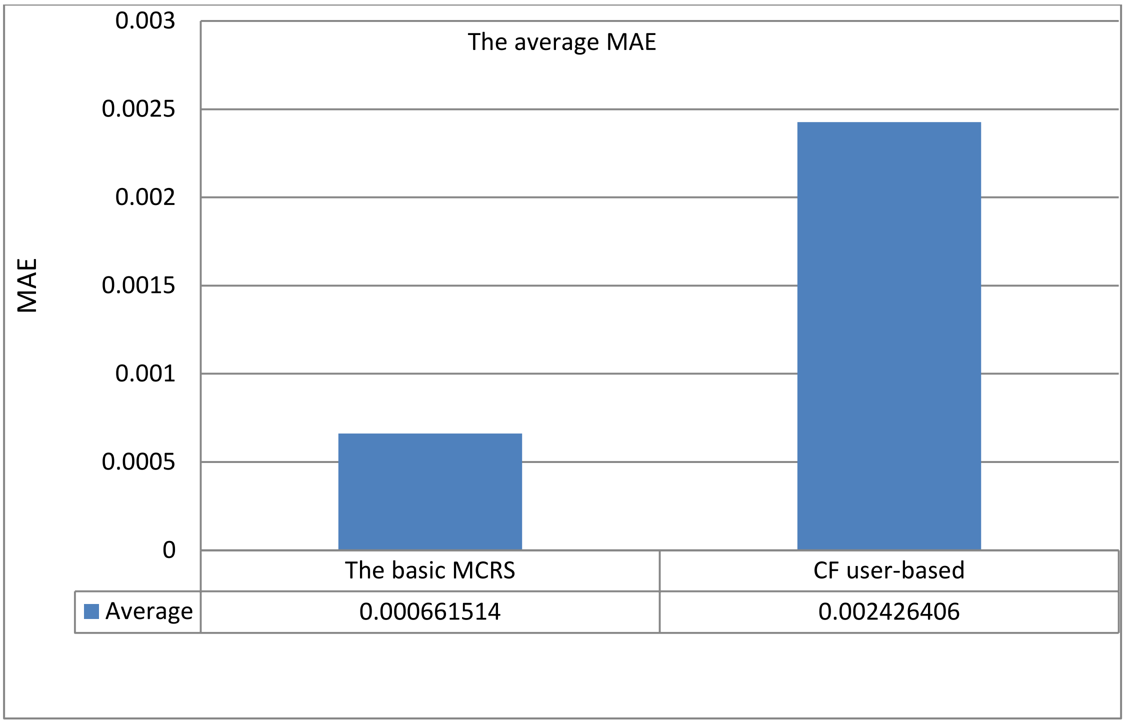


Figure 4-6 The average MAE of the basic MCRS VS CF user-base algorithm in the case of excluding the active user's accessed items in the recommendation results

the mean absolute error (MAE):

On the other hand, the basic MCRS and CF user-based algorithm are compared using the mean absolute error (MAE) which is calculated from the twelve tests results. MCRS has the MAE average (0.000661514); it is less than the CF user-based algorithm by (0.001764892) as represented in Figure 4-6. This means the basic MCRS outperforms the conventional Collaborative filtering user-base algorithm.

This means that the basic MCRS outperforms the CF user-based algorithm using mean average precision and the mean absolute error (MAE).

4.7. Chapter summary

This chapter introduces the general area of recommending items to users, using Collaborative filtering techniques and the general idea of using the time factor to weight ratings of users for items before they are used in the recommendation processes. Then, we find some limitations in these techniques. The first limitation comes from ways of calculating the similarities between users and items. These similarities are based on users' rating for items. These ratings are not accurate; because users' opinions vary with time. Thus, the similarities calculation violates the accuracy of the recommendation. Markov chain techniques have a limitation. They consider the sequences of accessing items. The transition matrix in these techniques is the probabilities of accessing item that follows a sequence of items. They use sets of a of k items, while users can access more than k items. Also users that access less than k items can't benefit from these techniques. We illustrate ways of addressing this limitation and give the motivation of designing new technique that can be used to recommend items to users. The new technique is Markov Chain recommendation system. It based on users rating for items, in the same session or period of time, that taken implicitly from users' preferences. These ratings are used to calculate the transition matrix and the initial matrix. The new technique outperforms the conventional collaborative filtering recommendation system. We use a data set from MovieLens for the evaluation. The evaluation is done using the mean absolute error, and precision and recall.

Chapter 5

THE ENHANCEMENT OF (MCRS) USING THE TIME FACTOR

5.1.Introduction

Chapter 4 illustrates in details the basic Markov Chain Recommendation Systems that can be used to generate suggestions for items to users. The experimental results prove that the basic MCRS outperforms the conventional user-based Collaborative filtering Recommendation System. The new technique is based on users' preferences for items; it consists of the active user's initial vector, and the transition matrix. Users of web applications access items in sessions or period of time, and we use the feature ***(accessing items by the same user in the same session or period of time)*** to design the basic MCRS. This feature is used to calculate the active user's initial vector that contains the probabilities of all items. The accessed items, by the active user, have the same probability in the initial vector, and it equals to one out of the number of all accessed items by the active user. The items that not accessed by the active user have zero probabilities in the initial vector.

Example:

Consider the set of all items is S={A,B,C,D,E,F}, and the active user U has accessed items A,C and F. We can represent the initial vector of the user U as follows:

I = (1/3 ,0 ,1/3 ,0 ,0 , 1/3). Items B, D and F are not accessed by the active user, and they have zero probabilities and need to be predicted.

The same feature ***(accessing items by the same user in the same session or period of time)*** is used to calculate the transition matrix that contains rows of items. In this matrix, any row represents an item and contains the probabilities of accessing other items with it. If the active user accesses only one item then the result is the row of that item, and the recommendation list contains items of the highest probabilities, in this row. On the other hand, if he has accessed more than one item then the result is the summation of the rows of these items i.e. the result is the vector product of the initial vector and the transition matrix. The recommendation list is taken from this result.

This chapter introduces an enhancement of the basic MCRS using the general weights of items. More enhancement is done using the period weights of items.

The rest of the chapter illustrates the limitation of the basic MCRS in Section 5.2. Section 5.3 gives the motivation of the enhancement of the proposed techniques. In Section 5.4 the general weights MCRS is designed. Section 5.5 illustrates the periods weighted MCRS. The experimental design is implemented in Section 5.6. The experimental results are given in Section 5.70. Then the chapter is summarised in Section 5.7.

5.2.The limitation of the basic MCRS

The basic MCRS technique outperforms the conventional Collaborative Filtering recommendation systems. However, more enhancements can be done using the time factor because items popularities vary with time. Items normally can be divided into three classes with respect to their popularities:

- First class: items that their popularities increase with time.
- Second class: items that their popularities decrease with time.
- Last class: items that are not affected either positively or negatively time.

According to these three classes, the basic MCRS has two limitations:

- The technique can recommend items that are not popular at the moment of the recommendation. Because some items are popular in the long term, but they are become not popular at the last period of time. These items may be popular in general, but they are not popular in the last period of time

- The technique can excludes some items from the recommendation list because some items are not popular in the long term, but they are popular in the short term, and users are interesting of them i.e. these items may not be popular in general, but they are popular in the last period of time.

These two limitations can violate the accuracy of the recommendation. To recommend the actually needed items to the active user, the basic MCRS can be enhanced using the time factor.

5.3.The motivation of MCRS enhancement using the time factor

Markov Chain Recommendation System (MCRS) is based on users' preferences for items. From users preferences for items, we can generate items popularities. As more users access an item its popularity increase and vice versa. There are two factors that can increase items popularities:

- **Items live time**: Items live time is the time interval when users have been accessing these items. Some items are submitted in the web earlier; thus, their live times are long. The recent submitted items have short live time. Items that have long live time might be more accessed by users.

- **The interesting items to users**: Users rate for the interesting items, even if they have short live time.

MCRS can be used to recommend items to users, and the recommendation result can be one of these situations, according to the mentioned factors:

- The Basic MCRS can be used to generate a list of interesting items by users in general, using the active user's accessed items. However, many items can be interesting to users in general, but they are not recommended to the active user. Because, the recommendations are based on the initial vector. In this case some interesting items by users are not recommended by the system. Therefore, this limitation can be solved if the result is weighted using the general weights of items before the generation of the recommendation list. More details are given in Section 5.4.

- The basic MCRS can be used to generate a list of the most rated items by users, but some of these items are not interesting in the last period of time. They are popular because they are rated in the long term. We can solve this problem if the result is weighted using the period weights of items before the generation of the recommendation list. More details are given in Section 5.45.5.

5.4.The general weights MCRS

Users of web applications are faced by the challenge of retrieving the actually interesting items. The challenge comes from the information overload problem. The number of available items (e.g. movies) that can be accessed (e.g. viewed) is very big. Users cannot retrieve all items to identify the interesting ones. Recommendation systems work on behalf of users to generate suggestions for items that might be of interest to users. MCRS recommends items to users according to the list items that have been accessed by the active user. However, items popularities vary with time. As more users access an item its popularity increases and vice versa.

Consider a web site provides movies (A,B,C) to users. For new users, the question is which movie is suitable to be viewed first? The popularities of items identify the best choice of items to be viewed first. The most viewed items by users are the most popular. Our target is how to calculate items popularly?

For example:

If ten users have accessed three movies A,B and C as represented in

Figure 5-1 ; then, what is the most popular one?

User1 → (A, B)	User3 → (A, C)
User2 → (B, C)	User4 → (A, C)
User5 → (A)	User6 → (B)
User7 → (C)	User8 → (C, A)
User9 → (A, B, C)	User10 → (A)

→ **The user views movies**

The set of movies is (A, B, C)

Figure 5-1 : The most accessed items by all users.

The answer:

We have ten users. This mean any item has ten chances to be viewed. The most accessed items by the ten users is the most popular.

- Movie A viewed 7 times out of 10.
- Movie B viewed 3 times out of 10.
- Movie C viewed 6 times out of 10.

The most popular movie are A and C respectively.

The general weights of items is the vector W, that contains weights of $item_j$ (j=1,2,3,...,n). The vector W can be calculated using the following equation:

$$W=\{W_j : W_j = \frac{(\text{Count of all users}'\text{sessions that contain item}_j)}{(\text{A count of all users}'\text{sesions })} \text{ and } j = 1,2,3,...,n\} \quad 5\text{-}1$$

Table **3-4** (page 71) and procedure three (**Figure 5-2**, page 107) can be used to calculate the general weights of items. The general weights are the summation of (items' columns) divided by the number of all records. The general weights do not consider the time factor which will be discussed later in weights of items in a period of time, section 5.5. The weight of item depends on users' preferences on it. When more users access an item its weight increases and vice versa. The general weights of items can be used to enhance the basic MCRS. Before recommending a list of items to the active users, the basic MCRS can be weighted by the general weight of items. The weighted MCRS is compared with the basic MCRS for the evaluation.

The general weights MCRS:

$$G\text{-}MCRS= I*T_{(i,j)} .W \quad 5\text{-}2$$

Procedure three:

 # Users-items table (Table **3-4***) is used to generate the general weights of items).*

"Items" is the list of all items in the given dataset.

Weights = null;

for all records of all users.

Weights = Weights + (fields of Items in the current record);

end for

The general Weights= Weights/(the number of all records)

Figure 5-2 : The creation of the general weights of items.

5.5.The period weights MCRS

Users view any movie at specific point of time. For simplicity, the time interval, in which users have viewed movies, is divided into periods. If ten users have viewed three movies A,B and C (

Figure 5-3) ; then, what is the most popular one in the last period of time?

The popularities of movies in the last period of time are different from the general popularities of movies.

- In general, as represented in
- Figure 5-3:

    - ✓ Movie A viewed 7 times out of 10.
    - ✓ Movie B viewed 3 times out of 10.
    - ✓ Movie C viewed 6 times out of 10.

    The most popular movie are A and C respectively.

- In the last period of time p3, as represented in
- Figure 5-3:

    - ✓ Movie A viewed 3 times out of 4.
    - ✓ Movie B viewed 1 times out of 4.
    - ✓ Movie C viewed 3 times out of 4.

    The most popular movies are A and C; all have the same popularities.

In general, A is most popular than C. In the last period of time the popularity of C is increased, and it become equal to the popularity of A.

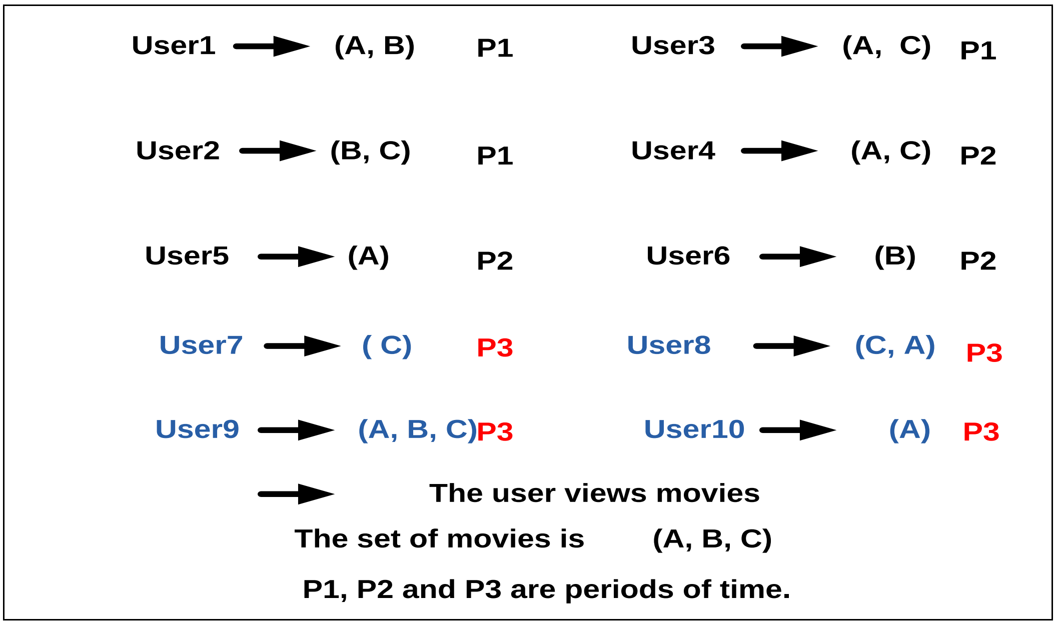


Figure 5-3 : The most viewed movies by all users per periods of time.

MCRS can use users' preferences on items in specific time interval P which can be divided into equal periods of time p (e.g. days, weeks or months).

The weights of items in period p_t is the vector w_{p_t} where :

$$w_{p_t}=\{w_{jt} : w_{jt} = \frac{(\text{count of all users}'\text{sessions that contains item}_j \text{ in period } p_t)}{(\text{count of all users}'\text{sessions in period } p_t)} \text{ and } j = 1,2,3,\ldots,n\} \qquad 5\text{-}3$$

The weight of items vary with time. More users can access an item in the old periods of time; the same item can be accessed by less number of users in the recent periods and vice versa. For more enhancement, weight of items in the last period of time can be used to weight the result of the basic MCRS. **Procedure four** (Figure **5-4**) can be used to calculate the period weights of items.

Procedure four:

 # Users-items table (Table **3-4***) is used to generate the period weights of items).*

"Items" :the list of all items in the given dataset.

"Periods" :the number of all periods of the time.

Period-eights = null;

for all records of all users.

If the period is (the last period of time in the data)

period-weights = period-weights + (fields of Items in the current record);

end if

end for

The period Weights of items= period-weights /(the number of all records)

Figure 5-4 : The creation of the period weights of items.

The period weights of items give the popularities of items in the last period of time. We need to identify the probability of accessing all items in the last period. Thus, the vector of period weights of items can be considered as the period initial vector in the last period.

There are three cases:

- The basic MCRS result.
- The probability of accessing items in general.
- The probability of accessing items in the last period of time.

The period weights MCRS is:

(The basic MCRS result) **and** (the probability of accessing items in general) **and** (the probability of accessing items in the last period of time)

$$\text{P-MCRS} = (I * \mathbf{T}_{(i,j)}).\boldsymbol{W}.(\,\mathbf{w}_{\mathbf{p_t}} * \mathbf{T}_{(i,j)}) \qquad 5\text{-}4$$

The period weights MCRS is compared with the basic MCRS and The general weights MCRS for the evaluation.

5.6.Experimental design

We use MovieLens dataset to conduct the experiments for the evaluation of the general weights MCRS and the period weights MCRS models. To do that, we divide the time interval of the data into months, and we consider any two months as one period. The starting month is p out of 137. The initial value of p= 70. In the first experiment the training data starts from at the first month and end at the 70th month. The next two months are used for testing. The last two months in the training data are used to generate the period weights of items. All the training data is used to generate the general weights of items. Then, for the next experiment we increment p by 3. The number of all experiments is twelve.

The basis MCRS technique and the enhanced techniques using the general weights and the period weights of items can be used to recommend items to an active user. The active user accesses only small subset (A) of items. The subset (A) can be used in the recommendation processes. The rest of items that not accessed by the active user can be considered as B.

To evaluate the general weight MCRS and the period weights MCRS, we can use the set A that accessed by the active user to recommend items from the set B, which is hidden, to the active user. The training data can be used to recommend items to the active user as follows:

- The set A can be used by the basic MCRS to generate "MCRS result".
- It can be used by the general weight MCRS to generate "G-MCRS result".
- It can be used by the period weight MCRS to generate "P-MCRS result".

The testing data can be used to retrieve the actually accessed items as follows:

- The set A is used to retrieve "The actually accessed items" from the testing data i.e. we retrieve all records that contains any item accessed by the active user. Then, we can calculate the accessibility of items by the summation of all retrieved records. Then, we can normalize such that the summation of the accessibility of all items equal to one.

The evaluation can be done using the accuracy. It can be done using the mean average precision. In these cases, the k highest probability items can be taken from the actually accessed items and the different results. The best result has the highest accuracy and mean average precision.

5.7.Experimental results

The basic MCRS technique is enhanced twice. First, the general weights MCRS. In this case the general weights of items are calculated from the training data in the twelve tests. These weights are used to enhance the result of MCRS. The enhancement result is the scalar product of the basic MCRS result and the general weights of items. Second, the enhancement is done using the period weights of items. The period weights of items are calculated from the last two periods in the training data in all of the twelve tests. The periods weights MCRS is the scalar product of the periods weights of items and the result of the basic MCRS.

The evaluation of the enhanced MCRS's is done using the accuracy and the mean average precision. Twelve user's accessed items are used in the twelve tests. The same user, the same training data, and the same testing data are used by the basic MCRS and the enhanced techniques. The general weights of items are calculated in the twelve tests using the training data; and the periods weights of items are calculated using the last two periods of the training data in any test.

The mean average precision

The mean average precision of the general weights MCRS is **0.868223287**; its better than the basic MCRS by **0.015476038**. This means the general weights MCRS outperforms the basic MCRS. The mean average precision of the period weights MCRS is **0.873066328** it is better than the basic MCRS by **0.004843041** and better than the general weights MCRS by **0.020319079**. This means the period weights MCRS outperforms the basic MCRS and the general weights MCRS using the mean average precision (Figure **5-5**).

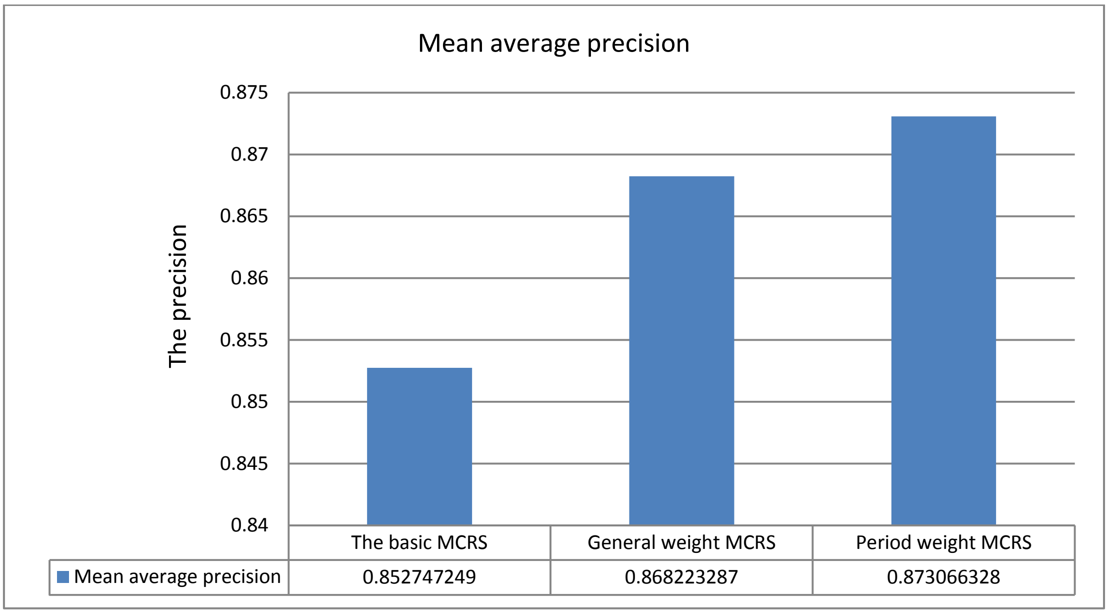


Figure 5-5 : Mean average precision of the basic MCRS VS the general weights MCRS and the period weights MCRS.

The average accuracy

The average accuracy of the general weights MCRS is 0.834666667; it's greater than the basic MCRS accuracy by 0.017. This means the general weights MCRS outperforms the basic MCRS. The accuracy of the period weights MCRS is

0.839666667 it is greater than the basic MCRS by 0.005 and greater than the general weights MCRS by 0.022. This means the period weights MCRS outperforms the basic MCRS and the general weights MCRS using the accuracy (Figure **5-6**).

Recommendation systems have been used by many websites to ease the selection of the next actually needed items to their users. The basic MCRS can be used to recommend items to users. It is enhanced twice. First, it is enhanced using the general weights of items which calculated using the training data. Second, MCRS is enhanced using the period weights of items that calculated from the last two periods in the training data. The evaluation of MCRS is done using the average precision, and the accuracy. Firstly, we prove that the general weight MCRS outperforms the basic MCRS. Secondly, we found that the period weights MCRS is better than the basic MCRS, and the general weights MCRS.

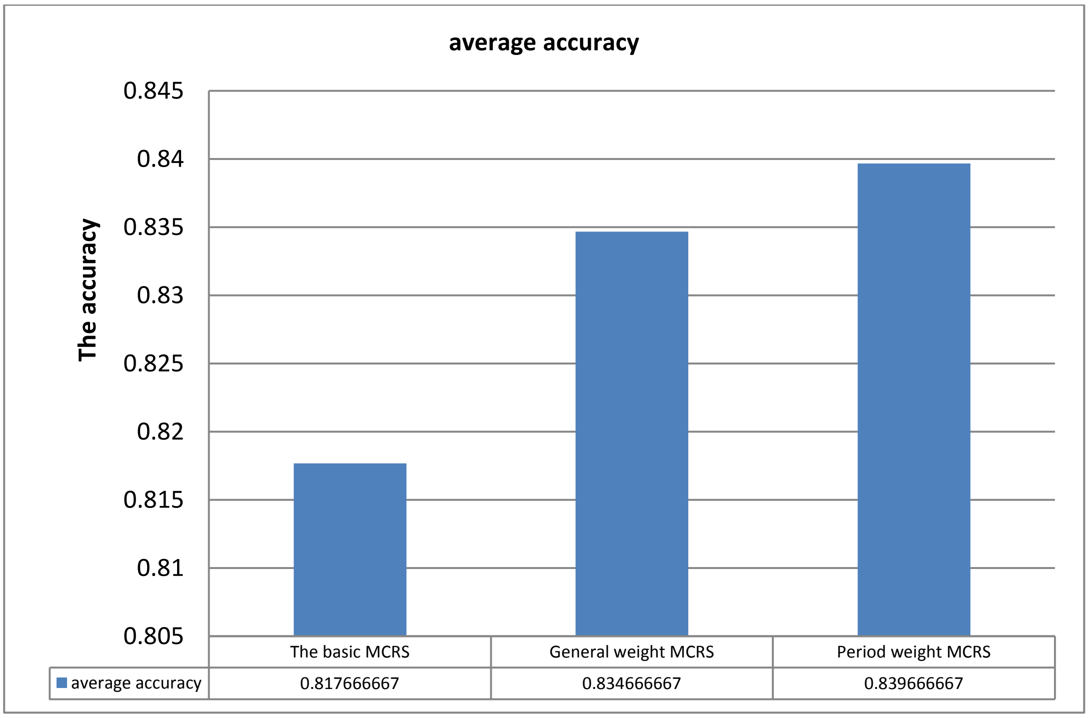


Figure 5-6 : Accuracy of the basic MCRS VS the general weights MCRS and the period weights MCRS.

5.8. Chapter summary

The basic MCRS is based on users' preferences for items to recommend items to users. We find some limitations violate the accuracy of the recommendation using MCRS. These limitations are caused by the variation of users' opinions and items popularities with time. The time factor is used to enhance the basic MCRS. We use the time of users' preferences for items to calculate items popularities in general. Also, we can find items' popularities in the last period of time.

In this chapter, we enhance the basic MCRS twice as follows:

- In the first enhancement we use items' popularities in general.
- In the second enhancement we use items' popularities in the last period of time.

The experimental results show that the time factor affects positively or negatively in the recommendation.

The enhancements using the time factor outperform the basic MCRS.

Chapter 6

THE ENHANCEMENT OF (MCRS) USING THE FRIEND FEATURE

6.1.Introduction

The world is become one community, as the internet provides to users at web applications the interaction between each other, with no consideration to distances and time factors. People can interact and communicate with each other through the globe in no time. Social media provides many features that can be used in these interactions that duplicate the amount of information; hence, they complicate the problem of information overload. Recommendation systems have been used, by web applications, to cope with this problem. In chapter two, we review the literature related to Collaborative filtering recommendation systems, ways of using social media features and trends analysis in recommendation systems. We review ways of using Markov model in recommendation systems and identify their strengths and weaknesses. In chapter four, we discuss the limitation of these web applications and identify the motivation of designing new recommendation systems. A new recommendation systems, based on Markov model, are designed and evaluated. Then, they are enhanced in chapter 5, using the time factor. However, they still have more limitation and need more enhancements. In this chapter, we represent this limitation and give a motivation of a new enhancement using the friends feature of social media.

The rest of this chapter investigates the limitation of MCRS and its enhancement using the time factor in section 6.2. Section 6.3 illustrate the motivation of MCTS enhancement using the friend feature. In section 6.4 we introduce the enhancement of Markov Chain recommendation system using the friend feature. We design the experiment for the evaluation purpose in section 6.5. Section 6.6 is the experimental results. The chapter is summarised in section 6.7.

6.2. The limitation of MCRS and its enhancements using the time factor

The goal of recommendation systems is to generate a list of interesting items (e.g. movies, books, articles, etc.) to users. Recommendation systems are applied by websites to ease their usage and to recommend the actually interesting items to users. There are many recommendation techniques that have been used by these websites. This thesis discusses the limitation of the conventional recommendation systems and finds the motivation of new techniques. Then, the thesis proposes a new recommendation system in chapter 4. The new technique is based on users' preferences for items and the Markov model. The evaluation proves that it outperforms the conventional recommendation techniques. In chapter 5, we find some limitation of our proposed technique, and discuss the motivation of the enhancement using the time factor. The enhancement was designed and evaluated. In this chapter, we find a new limitation of MCRS and motivation of extra enhancement using the friends feature.

The basic MCRS, and its enhancements using the time factor, are based on users' preferences for items. All users' preferences are used to generate the transition matrix, the general weights of items and the period weights of items. However, friends or groups of users became friends because they have strong relationship that links between them. It ties those users to formulate one community. Users with friendship are likely to be interested in the same subset of items; while, website uses the basic MCRS or its enhancements to recommend the most popular items with no consideration of their friends’ activities.

6.3.The motivation of MCRS enhancement using the friends feature.

The list of items in a recommendation can be of interest to the active user because the recommendation is generated according to the initial vector or the list of items that accessed by him. However, the active user's friends might be interested in other items. Therefore, if the result of the recommendation is weighted by the popularities of items that are of interest to the active users' friends; then, the result might be more accurate.

Context aware recommendation systems use the friend feature in pre-filter or post-filter techniques. On the other hand, we can mix many factors including friends feature to enhance MCRS. These factors are:

- The rating of all users for items.
- The time when users access these items.
- Ratings of the active user's friends for items.

The new technique is designed and evaluated later in this chapter.

6.4.The friends weighted MCRS

In any web application, users can view movies in a given time interval. On the other hand, we can concentrate on movies that have been viewed by friends. If ten users have viewed three movies A,B and C (

Figure 6-1) ; then:

- What are the most popular items in general?

- What are the most popular items according to movies that have been viewed by the active user's friends?

The popularities of movies can be given as follows:

In general, •

- Figure 6-1:

  ✓ Movie A viewed 7 times out of 10.
  ✓ Movie B viewed 3 times out of 10.
  ✓ Movie C viewed 6 times out of 10.

  The most popular movie are A and C respectively.

For movies viewed by the active user's friends "friends2" , in •

- Figure 6-1, we find the following facts:

  ✓ Movie A viewed 5 times out of 6.
  ✓ Movie B viewed 3 times out of 6.
  ✓ Movie C viewed 2 times out of 6.

  The most popular movie are A and B respectively.

This means popularities of items that accessed by friends can affect positively or negatively the items popularities.

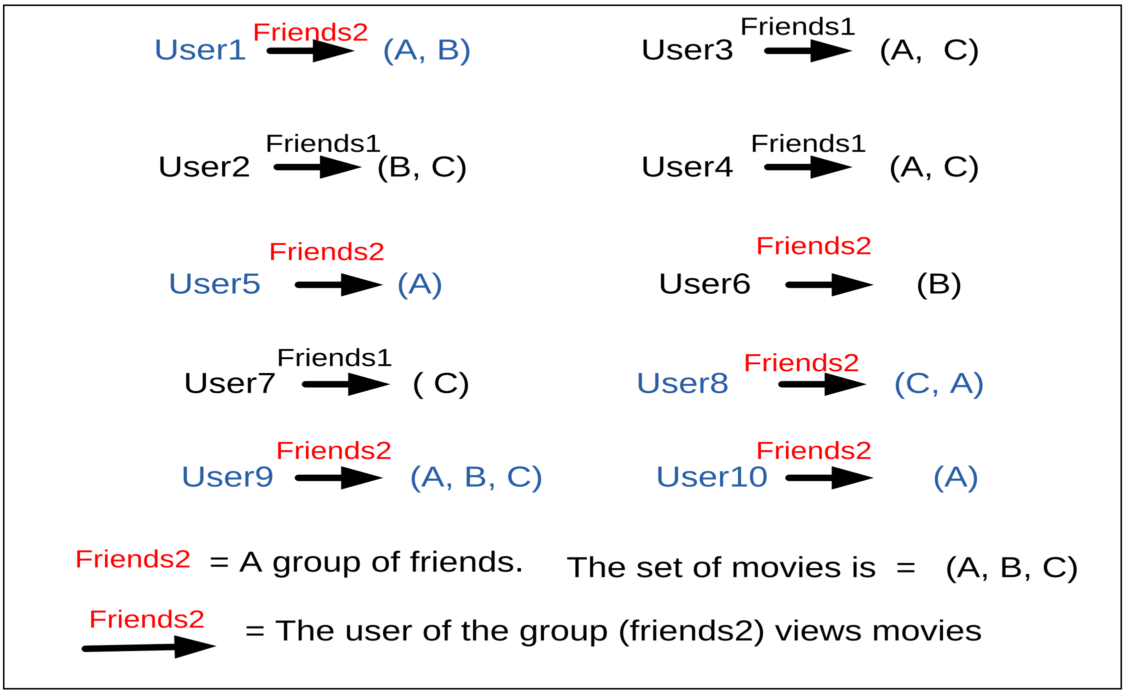


Figure 6-1 : The most viewed movies by the active user friends.

In Figure 6-1, there are two group of users; users that are belong to the friendship "friends1", and users that are belong to the friendship "friends2".

Friends feature simulates users' real world with respect to the flexibility and endless locations in real time interaction [24]. The feature is used in recommendation systems. Julien et. al. [147] develop a model-based recommendation system that based on friends preferences. **They introduced a social embedded collaborative filtering**. In their solution, they consider the user's profile as a mixture of his own and his friends' profiles. In our solution, we use the user's friends preferences for items to generate weights of items, with respect to the friendship feature. **We propose a new MCRS in Chapter 4**, and now we use friends feature and the time factor to enhance our proposed solution. In the dataset users' friends are given in a table in the following format:

Table 6-1: The table contains users Ids and their friends Ids.

User Id	User friend Id
2	275
2	428
2	515
2	761
2	1585
2	1625
2	1869
3	78
3	255
3	460

The weights of friends' items are calculated by filtering the data in the last period of time, by items of the active user's friends and friends of friends. The weights of friends in the period p_t are given by the vector fw_t where :

$$\mathbf{fw_t=\{fw_{jt} : fw_{jt} = \frac{(\text{count of all sessions of the active user's friends that contains item}_j \text{ in the period } p_t)}{(\text{count of all sessions of the active user's friends in the period } p_t)} \text{ and } j = 1,2,3,\dots,n\}} \quad \textbf{6-1}$$

Where n is the number of items, t=1,2,3,...,p and p is the number of the periods of time. Before identifying the recommendation list, the result of the basic MCRS is weighted using the weights of items that accessed by the active user's friends according to these four cases:

- The basic MCRS result.
- The probability of accessing items in general.
- The probability of accessing items in the last period of time.
- The probability of accessing items by the active user's friends in the same periods.

The friends' weights MCRS is:

(The basic MCRS result) **and** (the probability of accessing items in general) **and**

(The probability of accessing items in the last period of time) **and** (the probability of accessing items by the active user's friends)

$$\text{P-MCRS}= (I^*\mathbf{T}_{(\mathbf{i},\mathbf{j})}).\boldsymbol{W}.(\mathbf{w}_{\mathbf{p_t}} * \mathbf{T}_{(\mathbf{i},\mathbf{j})}).(\mathbf{fw}_{\mathbf{p_t}} * \mathbf{T}_{(\mathbf{i},\mathbf{j})}) \qquad 6\text{-}2$$

The friends' weight MCRS is compared with the basic MCRS, the general weights MCRS and the period weights MCRS for the evaluation.

Procedure four:# *Users-items table (Table* **3-4***) and friends table (Table* **6-1***) are used to generate the period weights of items).*

retrieve "User" from the Friends table.

"Friends" = the set of friends ids from the "User" retrieved data.

retrieve " Friends " from the Friends table.

"Friends of friends " = the set of friends ids from the " Friends" retrieved data.

"Items" :the list of all items in the given dataset.

"Periods" :the number of all periods of the time.

Friends-weights = null

retrieve " Friends of friends" from the users-items table.

for all records that contains "Last period of time"

Friends-weights = Friends-weights + (fields of Items in the record);

end for

Figure 6-2 : The creation of the period weights of items.

6.5.Experimental design

A LastFm dataset is used to conduct the experiments to evaluate the MCRS [148]. Data statistics are as follows: 1,892 users, 17,632 artists, 12,717 bi-directional user–friend relations, i.e. 25,434 (user_i, user_j) pairs, avg. 13.443 friend relations per user, 92,834 user–listened-to artist relations, i.e. tuples [user, artist, listening count], avg. 49.067 artists most listened to by each user, avg. 5.265 users who listened to each artist 11,946 tags, 186,479 tag assignments (tas), i.e. tuples [user, tag, artist], avg. 98.562 tas per user, avg. 14.891 tas per artist, avg. 18.930 distinct tags used by each user, avg. 8.764 distinct tags used for each artist. The dataset is released in the framework of the *2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems* (*HetRec 2011*).

The time of users' activities is divided into months. The total number of months is 69, from August 2005 to May 2011.

The time interval is 69 months. In the first experiment we have to start by a suitable data, p months : (1) The training data is p moths. (2) The testing data is the next x months after p. (3) The last period is the last x months in p. In the next experiment p is incremented by y months; the number of all experiments is twelve. The initial value of p=40 months and the maximum value of p =64 months , the value of x=2 months , and the value of y=2 months.

In the first experiment p=40. The training data set contains users' preferences in period from the first month to the40th month. the months (41 and 42) are used for test. The months (39 and 40) are used to calculated the period weights and friends weights of items.

In the second experiment p is incremented by 2, so p=42. The training data set contains users' preferences in period from the first month to the 42th month. the months (43 and 44) are used for test. The months (41 and 42) are used to calculated the period weights and friends weights of items. The number of the experiments is 12.

The data of friends “Fs”, of the active user, is used in the experiment here. Fs have accessed a set of items in the dataset “Is”. To evaluate the proposed enhancement of MCRS using Friends feature, “Is” can be split into two subsets. Let $A \subset I_s$ be the subset of items that have been accessed by an active user and B the subset of the remaining items. A can be used in the recommendation processes. B can be considered as the pool of items from which recommended items will be drawn. The MCRS can use set A to recommend sets of items $S_{(1)}$, using the basic MCRS; A can be used to recommend the list of items $S_{(2)}$, using the enhanced MCRS using friends feature. To evaluate the results, twelve experiments are conducted and the recommended lists are compared with B using precision recall, AUR curve.

6.6.Experimental results

To evaluate the friends weights MCRS. We use dataset from LastFm website. Twelve users Ids are used in twelve tests. The same user accessed items, the same training data, and the same testing data are used by the following techniques to give these results:

- The basic MCRS (MCRS-result).
- The enhancement of MCRS technique using the general weights of items (G-MCRS-result). The general weights of items are calculated in the twelve tests using the training data.
- The enhancement of MCRS technique using the period weights of items (P-MCRS-result). The periods' weights of items are calculated using the last two periods of the training data in any test.
- The enhancement of MCRS technique using the friends weights of items (F-MCRS-result). The friends' weights of items are calculated using items that have been accessed by the active user's friends in the last two periods of the training data in any test.
- We retrieve the items actually accessed by the active user, from the test data (Really-Data), in any test to compare it by the mentioned techniques.

The mean average precision

We calculate the precision and recall of all techniques comparing the with the actually accessed items by the active user**. (Figure 6-3, Figure 6-4, Figure 6-5, Figure 6-6)** give four samples from the twelve tests. In these figure the highest curve gives the best result. It is clear that all the enhancements outperform the basic MCRS. Because, it's curve is the lowest one.

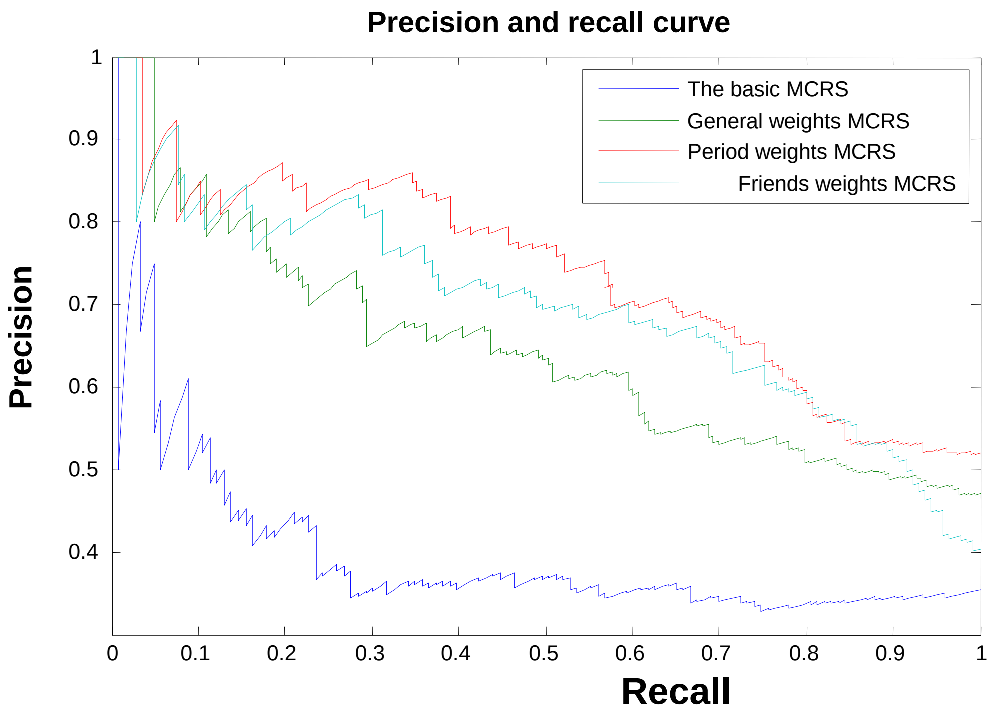


Figure 6-3 : : Sample (1), Precision and recall of the basic MCRS and its enhancements

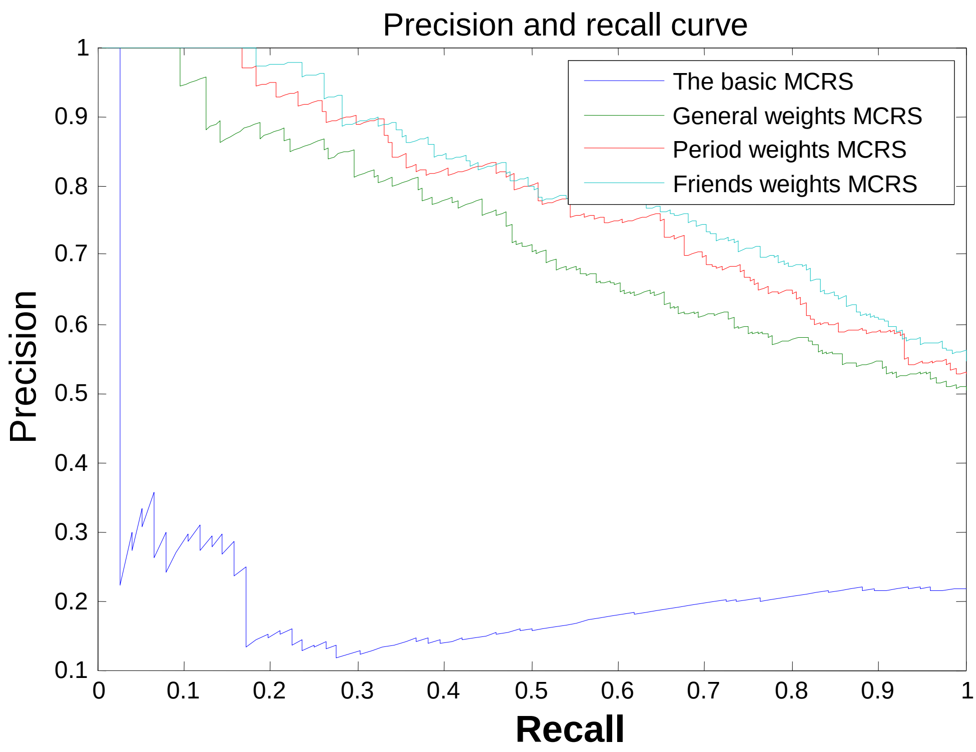


Figure 6-4 : Sample (2), Precision and recall of the basic MCRS and its enhancements

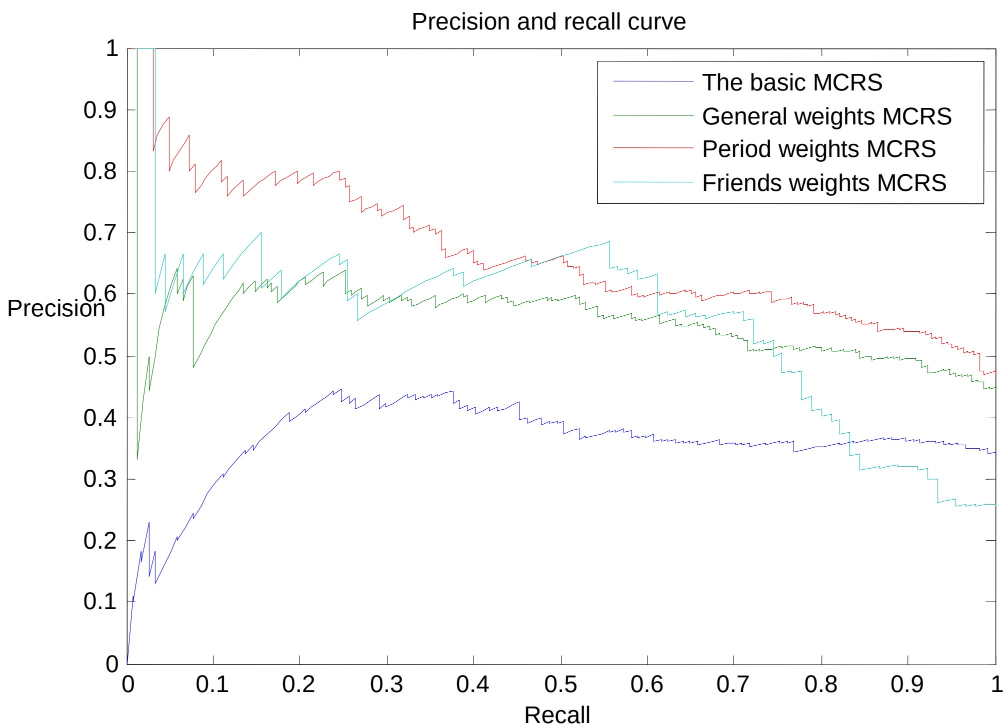


Figure 6-5 : Sample (3), Precision and recall of the basic MCRS and its enhancements

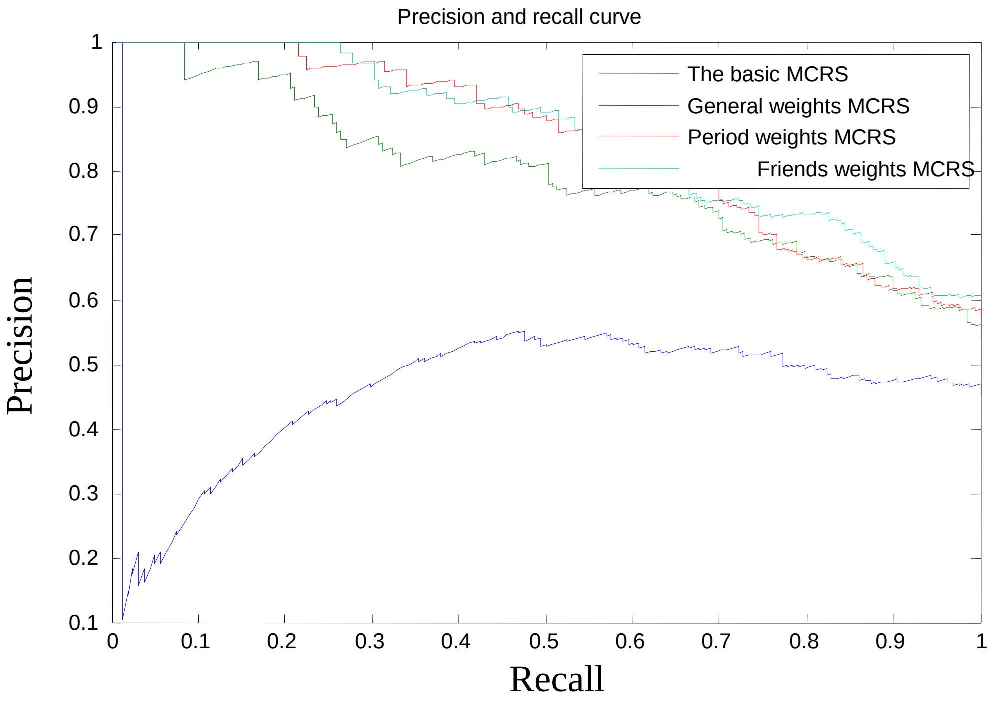


Figure 6-6 : Sample (4), Precision and recall of the basic MCRS and its enhancements

We find the average precisions of the techniques, as shown in (Figure 6-7 , Figure 6-8 and Table 6-2). It is clear that we can use the basic MCRS to recommend movies to the active user, but we can get a better result if we use its enhancements. These curves indicate that, the enhancements, using the time factor, effect positively and increase the precision and recall, the same as the enhancement using the friends feature.

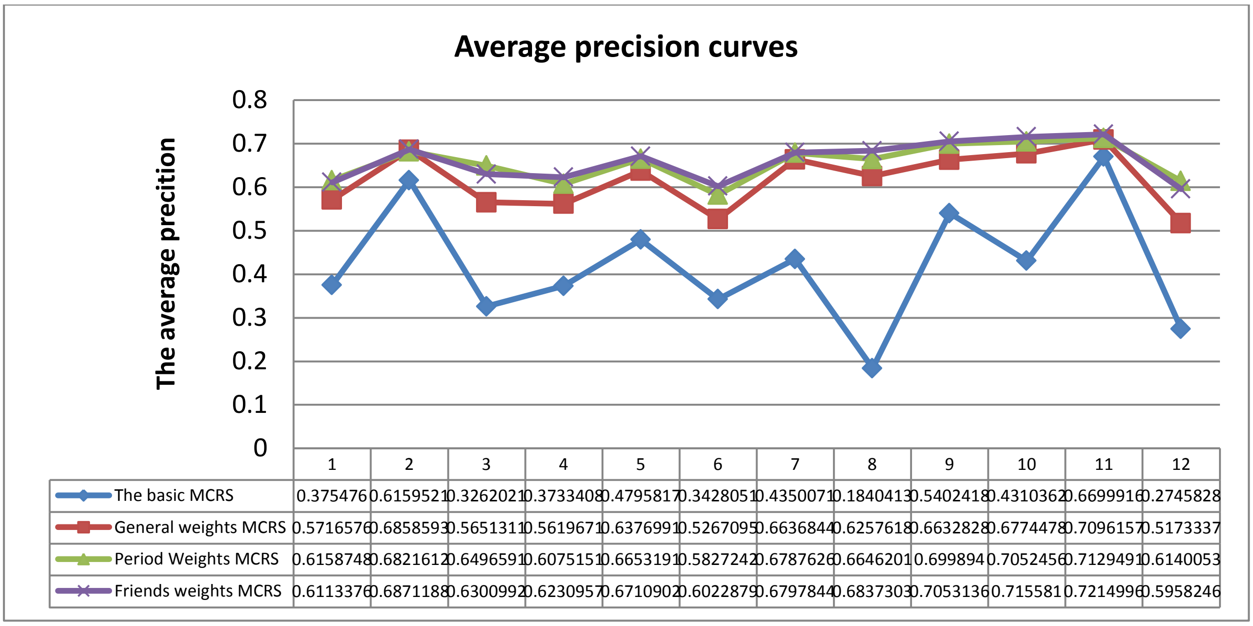


Figure 6-7 : The average precision of the basic MCRS and its enhancements in the twelve tests

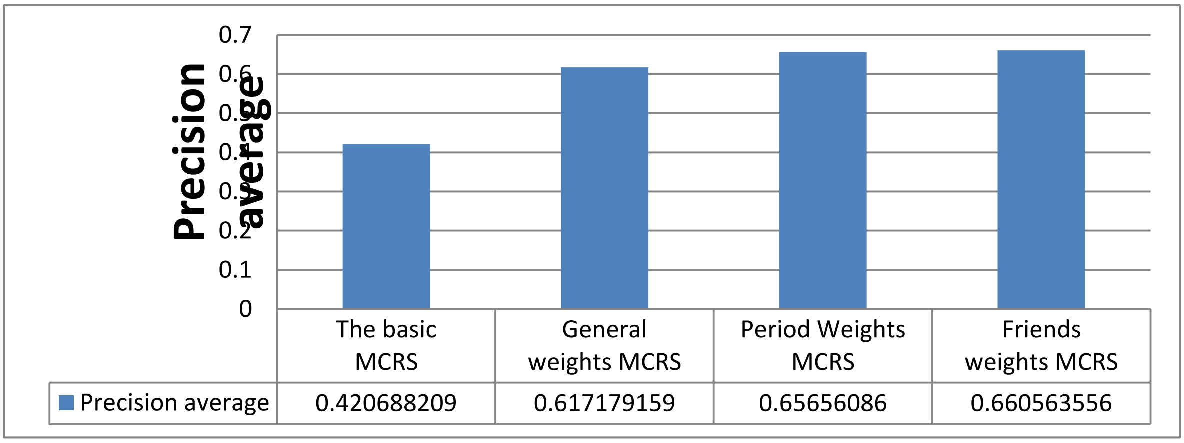


Figure 6-8 : The average precision's means of the basic MCRS and its enhancements in the twelve tests

The mean average precision of friends weights MCRS is **0.660563556** (Figure 6-7 and Figure 6-8); it is better than the basic MCRS by **0.239875347.** This means that, the friends weights MCRS outperforms the basic MCRS using the mean average precision.

The mean average precision of friends weights MCRS is better than the mean average precision of the general weights MCRS by **0.043384397** and better than the mean average precision of the period weights MCRS by **0.004002696**. This means the friends weights MCRS outperforms the enhancements of MCRS, using the time factor (Figure 6-7 , Figure 6-8 and Table 6-2).

Table 6-2 Summary of statistics of Precision and recall of the basic MCRS and its enhancements

Techniques	The basic MCRS	General weights MCRS	Period Weights MCRS	Friends weights MCRS
Precision average	0.420688209	0.617179159	0.65656086	**0.660563556**
VS Friends weights MCRS	**0.239875347**	**0.043384397**	**0.004002696**	

The precision and recall calculations are based on the rank of the list of the recommended items. Thus, it is not accurate; since, users not consider sequences of accessing items. On the other hand, the basic MCRS and its enhancements are based on accessing items by the same user in the same period of time and not consider the rank of accessing items. Therefore the evaluation using precision and recall is not enough. Thus, we do more evaluation using AUR curve and accuracy.

Test of Significance:

Table 6-3 : The precision average of The basic MCRS and The general W-MCRS in 12 experiments

	The basic MCRS	**The general W-MCRS**
1	0.3755	0.5717
2	0.6160	0.6859
3	0.3262	0.5651
4	0.3733	0.5620
5	0.4796	0.6377
6	0.3428	0.5267
7	0.4350	0.6637
8	0.1840	0.6258
9	0.5402	0.6633
10	0.4310	0.6774
11	0.6700	0.7096
12	0.2746	0.5173
The main	**0.42068333**	**0.61718**

The means of The basic MCRS and The general W-MCRS are significantly different at $p < 0.05$.

Summary

	The basic MCRS	The general W-MCRS
Mean	0.4207	0.6172
Variance	0.0195	0.0043
Stand. Dev.	0.1396	0.0656
n	12	12
t	**-4.4118**	
degrees of freedom	**22**	
critical value	**2.074**	

Area under ROC curve:

A receiver operating characteristic (ROC), or ROC curve, is curve that gives the performance of a binary classifier, recommended items are classified as interested or not, of a system as its classification is varied. The x-axis of the curve is the true positive (recommended and interested) rates and the y-axis is the false positive rates (not recommended but interested).

We plot the Area under ROC (Receiving Operation Characteristics) curve of results of the basic MCRS technique and its enhancements comparing with the actually accessed items by the active users friends. **(Figure 6-9,** Figure 6-10**, Figure 6-11, Figure 6-12)** give four samples from the twelve tests. In these figures the highest curve gives the best result. It is clear that the enhancements , of the new technique, using the general weights of items, the period weights of items, and friends' weights outperform the basic MCRS. Because, the curve of the basic MCRS in the mentioned figures is the lowest one.

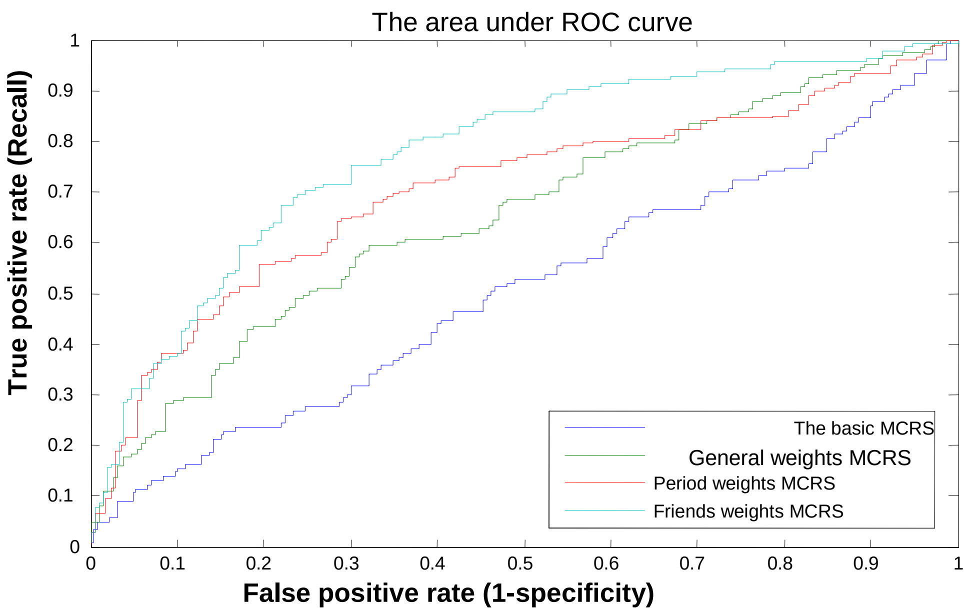


Figure 6-9 : Sample (1), area under ROC curve of the basic MCRS and its enhancements

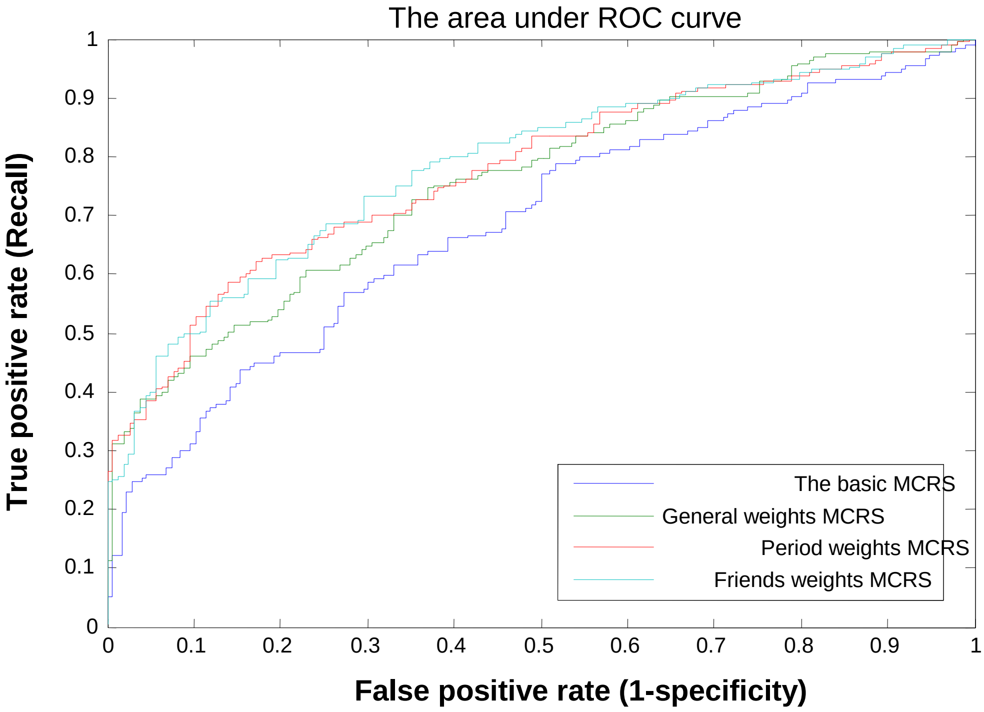


Figure 6-10 : Sample (2), area under ROC curve of the basic MCRS and its enhancements

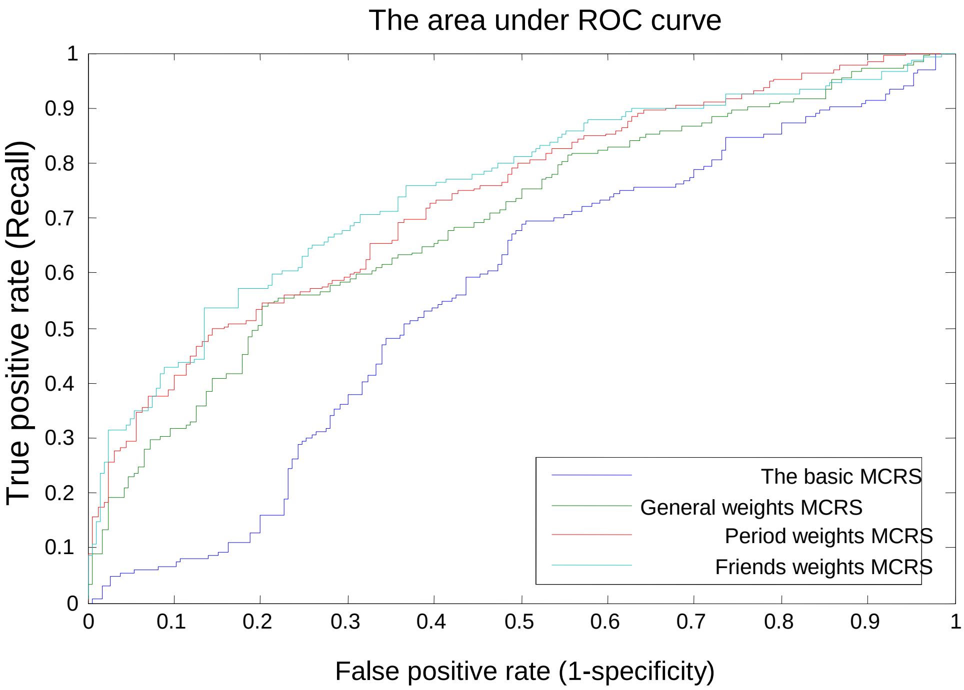


Figure 6-11 : Sample (3), area under ROC curve of the basic MCRS and its enhancements

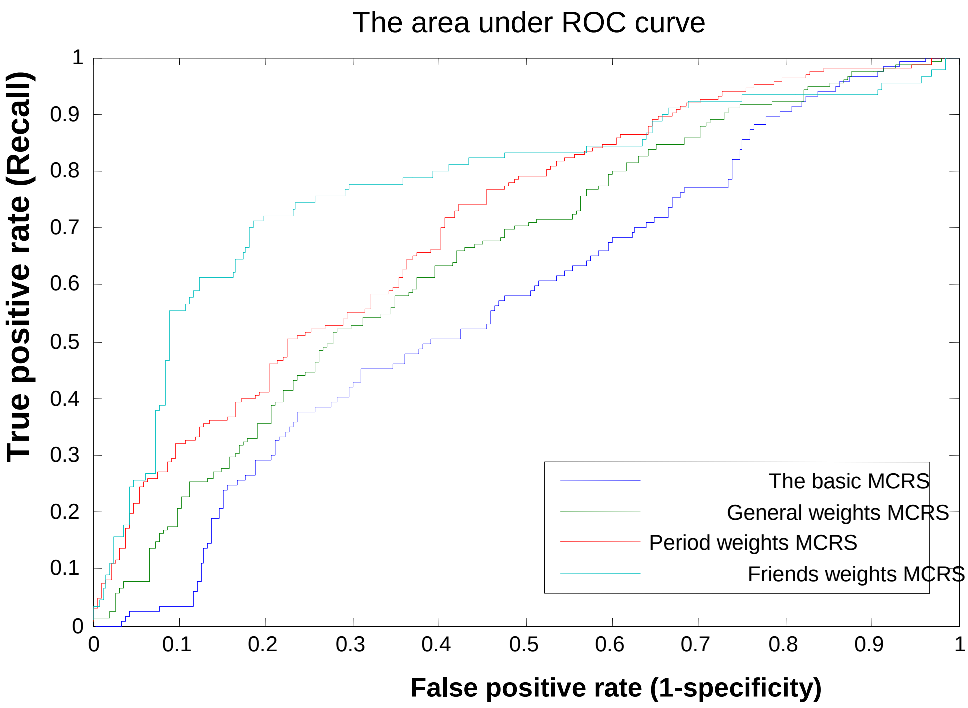
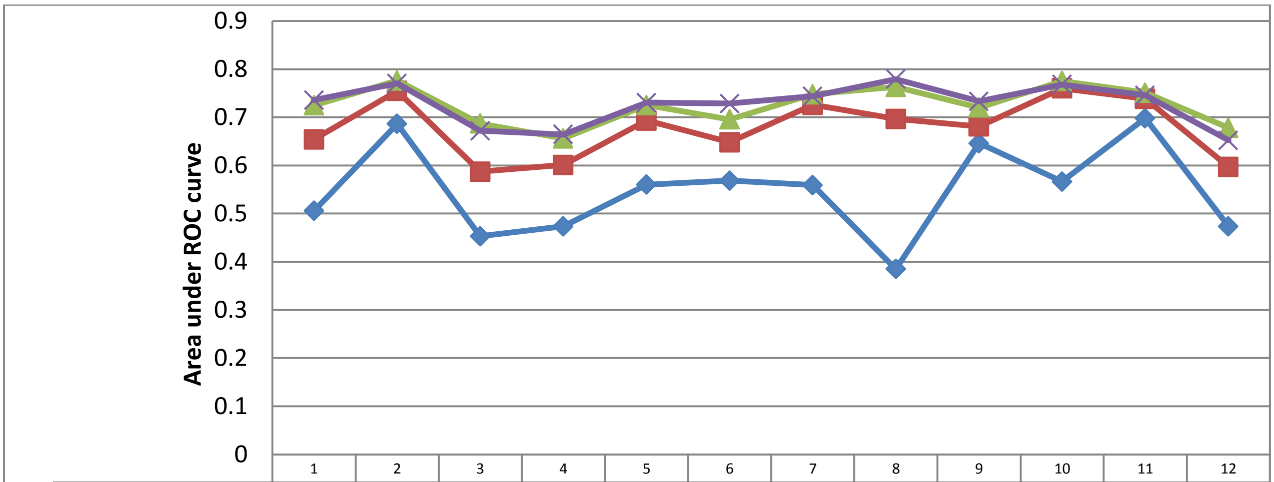


Figure 6-12 : Sample (4), area under ROC curve of the basic MCRS and its enhancements

We find the average of AUR of the basic MCRS technique and its enhancements, as represented in Figure 6-13. It is clear that we can use the basic MCRS, and its enhancements to recommend movies to the active user. The upper curve represents the result of recommending items to users using the friends feature, and it is the best result. The next one is the result of the enhancements of the basic MCRS using period's weights of items. The third cure from the top is the result of the general weights MCRS. The lowest curve represents the AUR of the basic MCRS. All techniques can be used to recommend items to user, but the enhancement using the friends feature gives the best result, as given is Figure 6-13.



	1	2	3	4	5	6	7	8	9	10	11	12
The basic MCRS	0.505640915	0.686259143	0.452897433	0.47376317	0.560352741	0.568431092	0.559148158	0.385084518	0.646070766	0.566653481	0.697924927	0.473450509
General weights MCRS	0.654276435	0.754793571	0.587124183	0.601078431	0.693452381	0.648172043	0.725702653	0.696773772	0.68119864	0.760064412	0.738536585	0.597079038
Period Weights MCRS	0.724770342	0.776233199	0.687149372	0.656151007	0.725078802	0.695553584	0.747670322	0.763129479	0.72018364	0.775267064	0.751017662	0.67797551
Friends weights MCRS	0.735774353	0.770416379	0.672174141	0.664456493	0.730467729	0.728391802	0.743610774	0.779295659	0.733166667	0.768070775	0.745634987	0.652219595

Figure 6-13 : The average of AUR, area under ROC curve, of the basic MCRS and its enhancements

The mean of average AUR of friends weights MCRS is **0.751817605** (**Figure 6-14** and **Table 6-4**); it is better than the mean of average AUR of the basic MCRS by **0.203844534** and better than the mean of average AUR of the general weights MCRS by **0.073629926**. And better than the mean of average AUR of the general weights MCRS by **0.023043351**. This means the friends weights MCRS outperforms the basic MCRS, the general weights MCRS, and the period weights MCRS using the mean average AUR **(Figure 6-9,** Figure 6-10**, Figure 6-11, Figure 6-12)** .

This means friend's feature enhancement outperforms the basic MCRS and its enhancement using the time factor.

Table 6-4 Summary of statistics of AUR curve of the basic MCRS and its enhancements

techniques	The basic MCRS	General weights MCRS	Period Weights MCRS	Friends weights MCRS
Mean of average AUR Curve	0.547973071	0.678187679	0.728774254	**0.751817605**
VS of Friends weights MCRS	***0.203844534***	***0.073629926***	***0.023043351***	

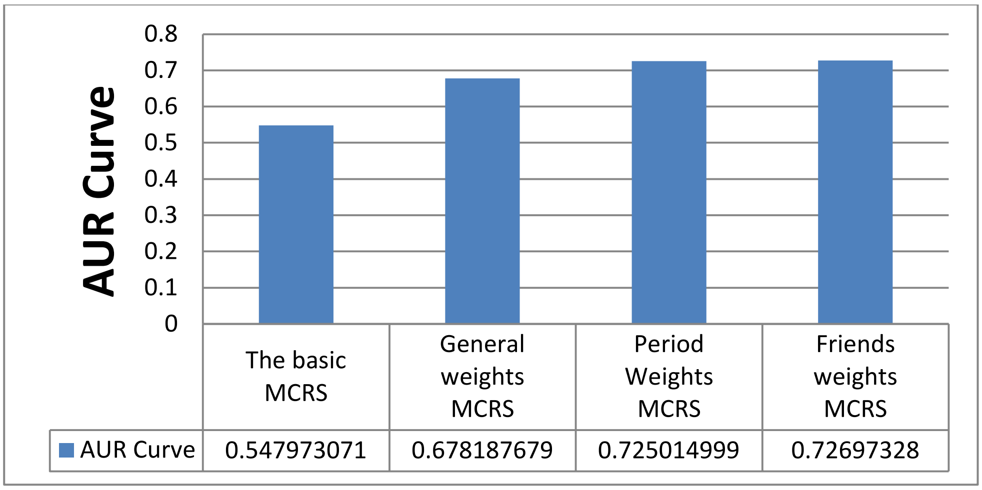
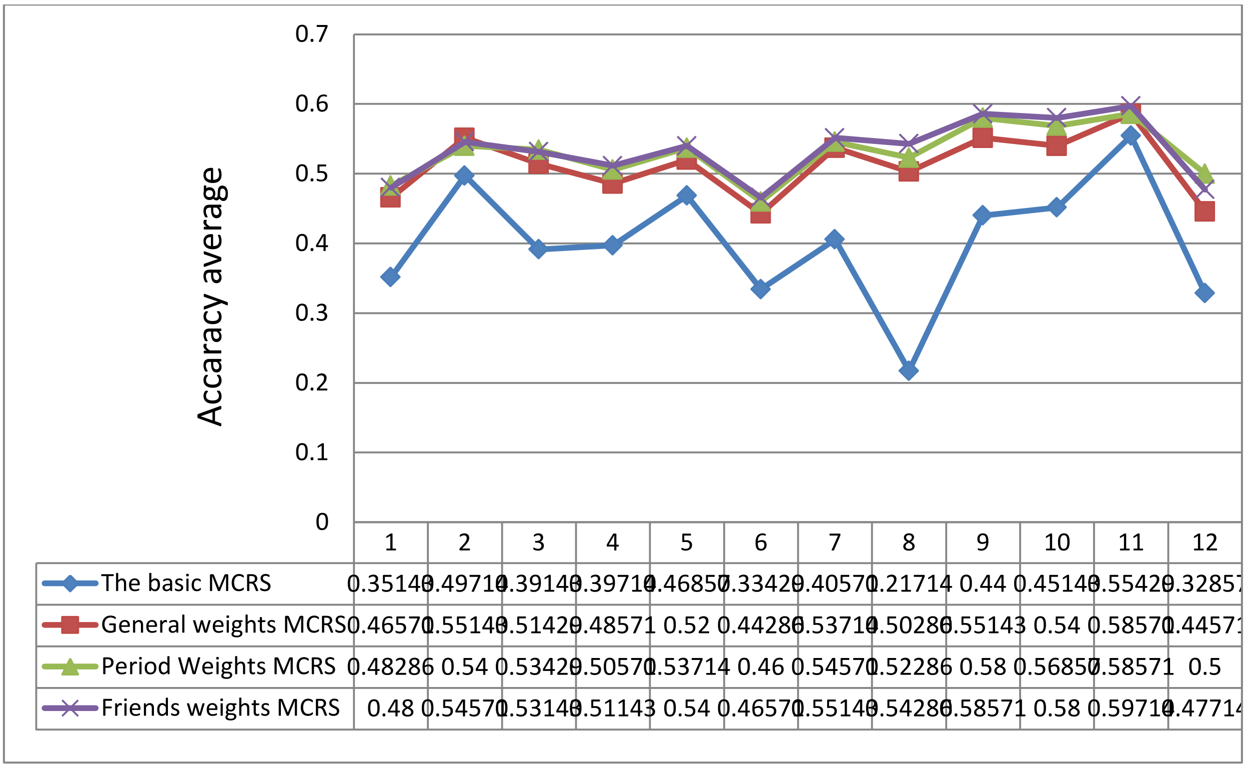


Figure 6-14 : The mean of average AUR, area under ROC curve, of the basic MCRS and its enhancements

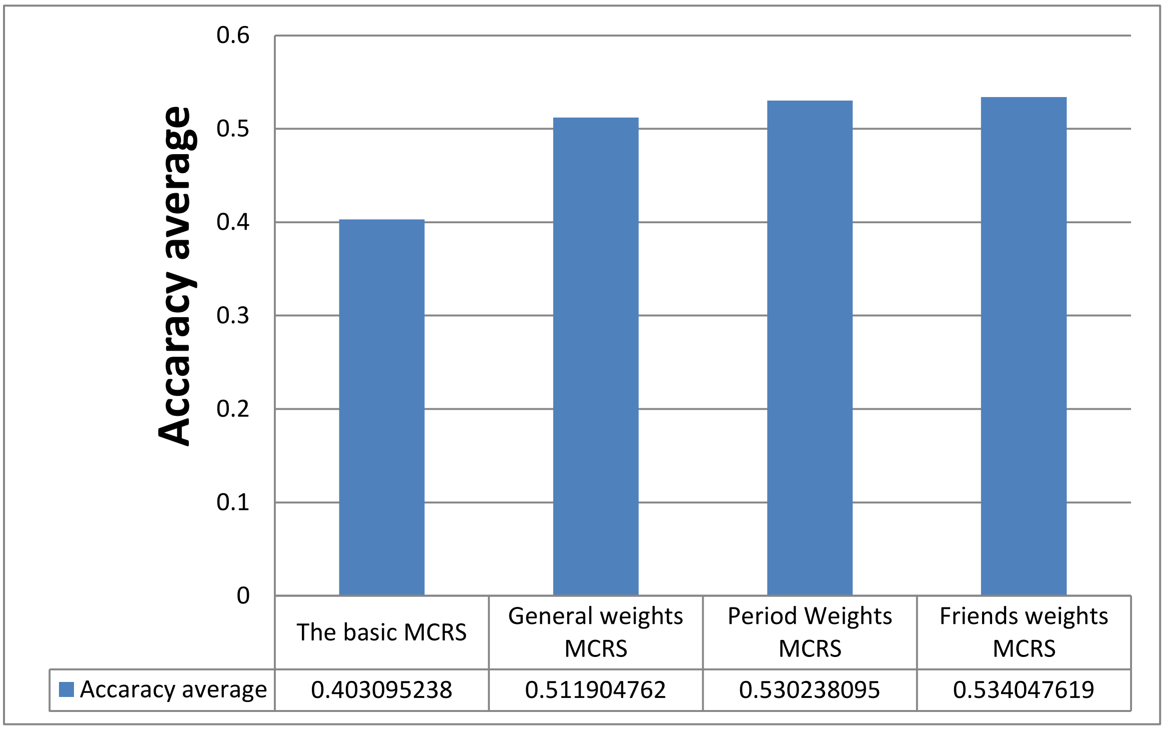
Accuracy:

Accuracy is the number of the recommended and interested items divided by the number of all items.



	1	2	3	4	5	6	7	8	9	10	11	12
The basic MCRS	0.3514	0.4971	0.3914	0.3971	0.4685	0.3342	0.4057	0.21714	0.44	0.4514	0.5542	0.3285
General weights MCRS	0.4657	0.5514	0.5142	0.48571	0.52	0.4428	0.5371	0.5028	0.55143	0.54	0.5857	0.4457
Period Weights MCRS	0.48286	0.54	0.5342	0.5057	0.53714	0.46	0.5457	0.52286	0.58	0.5685	0.58571	0.5
Friends weights MCRS	0.48	0.5457	0.5314	0.51143	0.54	0.4657	0.5514	0.5428	0.58571	0.58	0.5971	0.4771

Figure 6-15 : The mean accuracy average curves, of the basic MCRS and its enhancements



	The basic MCRS	General weights MCRS	Period Weights MCRS	Friends weights MCRS
Accaracy average	0.403095238	0.511904762	0.530238095	0.534047619

Figure 6-16 : The mean accuracy average of the basic MCRS and its enhancements

The mean accuracy average of friends weights MCRS **is 0.534047619** (Figure 6-16, Table **6-5**); it is better than mean average accuracy of the basic MCRS by **0.130952381**. This means that, the friends weights MCRS outperforms the basic MCRS using the mean average precision.

The mean accuracy average of friends weights MCRS is better than the mean average accuracy of general weights MCRS by **0.022142857** and better than the period weights MCRS by **0.003809524**. This means the friends weights MCRS outperforms the enhancements of MCRS, using the time factor (Figure **6-16**, **Table 6-5**).

Table 6-5 Summary of statistics of mean average accuracy of the basic MCRS and its enhancements

Techniques	The basic MCRS	General weights MCRS	Period Weights MCRS	Friends weights MCRS
Mean accuracy average	0.403095238	0.511904762	0.530238095	0.534047619
Friends weights MCRS	**0.130952381**	**0.022142857**	**0.003809524**	

Rest of Significance:

Table 6-6 : The accuracy average

	The accuracy average											
The basic MCRS	0.3514	0.4971	0.3914	0.3971	0.4686	0.3343	0.4057	0.2171	0.4400	0.4514	0.5543	0.3286
The general W-MCRS	0.4657	0.5514	0.5143	0.4857	0.5200	0.4429	0.5371	0.5029	0.5514	0.5400	0.5857	0.4457

Mean of The basic MCRS = 0.403083 (n = 12)

Mean of The general W-MCRS = 0.5119 (n = 12)

Assuming equal variances:

Combined standard error = 0.028805

degree of freedom df= 22

t = 3.777725

One sided P = 0.0005

Two sided P = 0.001

95% confidence interval for difference between means = -0.168554 to -0.049079

Power (for 5% significance) = 99.9%

Assuming unequal variances:

Combined standard error = 0.028805

df = 16.229584

t(d) = 3.777725

One sided P = 0.0008

Two sided P = 0.0016

95% confidence interval for difference between means = -0.168554 to -0.049079

Power (for 5% significance) = 94.2%

Comparison of variances:

TWO SIDED F TEST IS SIGNIFICANT

6.7.Chapter summary

Chapter4 represents and evaluates the basic MCRS that based on users' preferences for items. Markov model is used to design the new techniques. Markov model states, in the new technique, are items that can recommend to users. When, any user accessed items, he can access with it, in the same list, many items. The basic MCRS invests the feature of accessing items by the same user in the same period to generate the list of recommendation. It uses all users' preferences for all items to generate the transition matrix. The initial vector of Markov Model is based on the active user's accessed items. MCRS aims to generate suggestions from items to the active user using his previous accessed items. The result of the basic MCRS is the vector product of the initial vector and the transition matrix. The new technique outperforms the conventional Collaborative filtering recommendation systems. However, items popularities, in general and in periods, vary with the time. Therefore, the basic MCRS is enhanced, using the general weights and the period weights of items, in Chapter 5. These two enhancements outperform the basic MCRS. The last enhancement is designed in this chapter. In the new enhancement, the period weights MCRS is weighted by friends weights of items; it outperforms the basic MCRS and its enhancements using the time factor. The evaluation is done using mean average precision, accuracy, and AUR curve. They use MovieLens dataset to conduct experiments.

Chapter 7

CONCLUSION AND FUTURE WORK

7.1.Conclusion

This thesis proposes new recommendation techniques based on users' preferences for items, the time factor, and friends feature. The new techniques are the basic MCRS and its enhancements using the time factor and the friends feature. This thesis addresses the limitation of the conventional collaborative filtering techniques that based on users and items' similarities. However, users' opinions and items popularities vary with time. As any two users can be similar to each other in long term (two years), but the same these two users can be not similar to each others in short terms (one month). Therefore, the time factor must be considered in similarity calculations.

Our new techniques divide the time interval, when users have accessed the items, into periods. We use users' preferences for items in these periods to design the basic MCRS (Markov Chain Recommendation System). The basic MCRS is enhanced three times. firstly, we enhance it using the general weights of items. Secondly, we enhance it using the time factor. Finally, we enhance it using the friends' features. Therefore, we have four contributions:

- The basic MCRS.
- The general weights MCRS.
- The period weights MCRS.
- Friends weights MCRS.

The basic MCRS

It consists of these components:

- States: It can be any item, which recommended to users.
- The relation between items: It can be calculated using the feature 'accessing items by the same user in the same period of time'
- The initial vector: It represents the active user accessed items that have the same chance in the recommendation process.
- The transition matrix: It contains the probabilities of accessing all items with any items.

The basic MCRS is the vector product of the initial vector and the transition matrix. The technique recommends items that have the most probabilities to satisfy the active user based on his previous accessed items.

The general weights MCRS

The feature, 'accessing items by the same user in the same session', is used to calculated the general weights of items. Before generating the recommendation list the basic MCRS is weighted using the general weights of items. The general weights of items vary with the time. These variations in the general weights of time increase the accuracy of MCRS.

The period weights MCRS

The feature, 'accessing items by the same user in the same session' in the last period of time, is used to calculated the period initial vector that is used to calculate the period weights of items. Before generating the recommendation list the general weighted MCRS is weighted by the period weights of items. The variation of the period weights of items are better than the variations in the general weights of items. Hence, they have most positive effects on MCRS results.

The friends weights MCRS

Using the same steps, we calculate the friends weights of items to weight the period weights MCRS before generating the recommendation lists.

Users of the same friends group can access the same set of items. This feature is used to enhance the period weights MCRS. The active user's friends' preferences are collected and used to calculate weights of the friends' accessed items that are used to weight the period weights MCRS before items recommendation. Because some items can be popular in general at the same time may be not interesting for a specific group of friends.

The new techniques and its enhancements outperform the conventional Collaborative filtering techniques. The evaluation is done using precision and recall, accuracy, Mean absolute error, and area under ROC curve. we use datasets from MovieLens and LastFm.

7.2.Future work

The basic MCRS and its enhancements open up new research topics that can be summarized as follows:

Time series:

The basic MCRS can be studied to know how it can be used in time series e.g. days of the week, months of the year, seasons, etc; because users might have many activities depending on these time series.

Items' clusters:

Social media provides a huge amount of information, leading to the problem of information overload. The conventional recommendation systems as well as MCRS cope with the problem. However, the numbers of users and items are increasing exponentially. Clustering techniques are used in the conventional recommendation systems to cope with the sparsity problem by grouping items or users to clusters. Thus, more studies can be done to investigate ways of using clustering techniques in MCRSs based on social media.

References

[1] G. Linden, B. Smith, and J. York, “Amazon.com Recommendations: Item-to-Item Collaborative Filtering,” *IEEE Internet Comput.*, vol. 7, no. 1, pp. 76–80, 2003.

[2] A. Das, “Google News Personalization : Scalable Online,” *WWW 2007 / Track Ind. Pract. Exp.*, pp. 271–280, 2007.

[3] A. Edmunds and A. Morris, “The problem of information overload in business organisations: a review of the literature,” *Int. J. Inf. Manage.*, vol. 20, no. 1, pp. 17–28, Feb. 2000.

[4] P. Lops, M. De Gemmis, and G. Semeraro, *Recommender Systems Handbook.* Boston, MA: Springer US, 2011.

[5] L. Lü, M. Medo, C. H. Yeung, Y.-C. Zhang, Z.-K. Zhang, and T. Zhou, “Recommender systems,” *Phys. Rep.*, vol. 519, no. 1, pp. 1–49, 2012.

[6] A. Hossein, N. Rafsanjani, N. Salim, and A. R. Aghdam, “Recommendation Systems : a review,” *Int. J. Coputational Engneering Res.*, vol. 03, no. 5, pp. 47–52, 2013.

[7] M. J. Pazzani and D. Billsus, “Content-Based Recommendation Systems,” *Springer-Verlag Berlin Heidelb. 2007*, no. 4321, pp. 325–341, 2007.

[8] M. Sood and H. Kaur, “Survey on News Recommendation,” *Int. J. Adv. Res. Electr. Elictronics Instrum. Engneering*, vol. 3, no. 6, pp. 9972–9977, 2014.

[9] G. Linden, B. Smith, and J. York, “Amazon . com,” no. February, 2003.

[10] D. Almazro, G. Shahatah, L. Albdulkarim, M. Kherees, R. Martinez, and W. Nzoukou, “A Survey Paper on Recommender Systems,” Jun. 2010.

[11] Y. Koren, “Collaborative filtering with temporal dynamics,” in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, pp. 447–456.

[12] S. Vucetic and Z. Obradovic, “Collaborative Filtering Using a Regression-Based Approach,” *Knowl. Inf. Syst.*, vol. 7, no. 1, pp. 1–22, Feb. 2005.

[13] W. West, “Collaborative Filtering in Twitter Recommendation Systems.”

[14] X. Cai, M. Bain, A. Krzywicki, W. Wobcke, Y. S. Kim, P. Compton, and A. Mahidadia, “Collaborative Filtering for People to People Recommendation in Social Networks.”

[15] X. Su and T. M. Khoshgoftaar, “A Survey of Collaborative Filtering Techniques,” *Adv. Artif. Intell.*, vol. 2009, no. Section 3, pp. 1–19, 2009.

[16] F. Ricci, L. Rokach, and B. Shapira, *Recommender Systems Handbook*. Boston, MA: Springer US, 2011.

[17] S. Perugini, M. a Gonçalves, and E. a Fox, “Recommender Systems Research: A Connection-Centric Survey,” *J. Intell. Inf. Syst.*, vol. 23, no. 2, pp. 107–143, 2004.

[18] Y. S. Y. Song and C. L. Giles, “Efficient user preference predictions using collaborative filtering,” *2008 19th Int. Conf. Pattern Recognit.*, 2008.

[19] P. Symeonidis, a Nanopoulos, a Papadopoulos, and Y. Manolopoulos, “Collaborative recommender systems: Combining effectiveness and efficiency,” *Expert Syst. Appl.*, vol. 34, no. 4, pp. 2995–3013, May 2008.

[20] J. a. Konstan and J. Riedl, “Recommender systems: from algorithms to user experience,” *User Model. User-adapt. Interact.*, vol. 22, no. 1–2, pp. 101–123, Mar. 2012.

[21] J. Ben Schafer, D. Frankowski, J. Herlocker, and S. Sen, “Collaborative Filtering Recommender Systems,” pp. 291–324.

[22] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, “Item-based collaborative filtering recommendation algorithms,” *Proc. tenth Int. Conf. World Wide Web WWW 01*, vol. 1, no. 3, pp. 285–295, 2001.

[23] M. Muñoz-organero, G. A. Ramírez-gonzález, and P. J. Muñoz-merino, “A Collaborative Recommender System Based on Space-Time Similarities,” pp. 81–87, 2010.

[24] T. Sandholm and H. Ung, “Real-time, location-aware collaborative filtering of web content,” in *Proceedings of the 2011 Workshop on Context-awareness in Retrieval and Recommendation*, 2011, pp. 14–18.

[25] Y. Ding and X. Li, “Time weight collaborative filtering,” in *Proceedings of the 14th ACM international conference on Information and knowledge management*, 2005, pp. 485–492.

[26] T. Q. Lee and Y. Park, “A Time-based Recommender System using Implicit Feedback,” *Expert Syst. Appl.*, vol. 34, no. 4, pp. 3055–3062, 2008.

[27] Z. Zhang, T. Zhou, and Y. Zhang, “Tag-Aware Recommender Systems : A State-of-the-art Sur- vey arXiv : 1202 . 5820v1 [cs . IR] 27 Feb 2012.”

[28] R. Kumar, “Context-aware Social Popularity based Recommender System,” vol. 92, no. 2, pp. 37–42, 2014.

[29] J. M. Somwanshi and P. Y. B. Gurav, “Cloud-Based Mobile Multimedia Recommendation System with User Behavior Information,” pp. 6830–6834, 2014.

[30] N. N. Liu, L. He, and M. Zhao, “Social temporal collaborative ranking for context aware movie recommendation,” *ACM Trans. Intell. Syst. Technol.*, vol. 4, no. 1, pp. 15:1–15:26, Feb. 2013.

[31] H. Molina, “Context-Based Recommendation Approach for Measurement and Evaluation Projects,” *J. Softw. Eng. Appl.*, vol. 03, no. 12, pp. 1089–1106, 2010.

[32] P. G. Campos, A. Bellogín, F. Díez, and J. E. Chavarriaga, “Simple time-biased KNN-based recommendations,” in *Proceedings of the Workshop on Context-Aware Movie Recommendation*, 2010, pp. 20–23.

[33] X. Amatriain, A. Jaimes, N. Oliver, and J. M. Pujol, *Recommender Systems Handbook*. Boston, MA: Springer US, 2011.

[34] S. M. Allen and R. M. Whitaker, “Uttering : Social Micro-Blogging without the Internet,” pp. 58–64, 2010.

[35] L. Lü, M. Medo, C. H. Yeung, Y.-C. Zhang, Z.-K. Zhang, and T. Zhou, “Recommender Systems,” p. 97, Feb. 2012.

[36] G. Shani, D. Heckerman, and R. I. Brafman, “An MDP-Based Recommender System ∗,” vol. 6, pp. 1265–1295, 2005.

[37] S. Rendle, C. Freudenthaler, and L. Schmidt-thieme, “Factorizing Personalized Markov Chains for Next-Basket Recommendation,” 2010.

[38] M. Deshpande and G. Karypis, “Item-based top- N recommendation algorithms,” *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 143–177, 2004.

[39] X. Su and T. M. Khoshgoftaar, “A Survey of Collaborative Filtering Techniques,” *Adv. Artif. Intell.*, vol. 2009, no. Section 3, pp. 1–19, 2009.

[40] J. Tang, X. Hu, and H. Liu, *Social recommendation: a review*, vol. 3, no. 4. 2013.

[41] X. Sha, D. Quercia, M. Dell’Amico, and P. Michiardi, “Trend makers and trend spotters in a mobile application,” *Proc. 2013 Conf. Comput. Support. Coop. Work - CSCW ’13*, p. 1365, 2013.

[42] G. Adomavicius and Y. K. Y. Kwon, “New Recommendation Techniques for Multicriteria Rating Systems,” *IEEE Intell. Syst.*, vol. 22, no. 3, pp. 1–28, 2007.

[43] G. Guo, J. Zhang, and N. Yorke-smith, “A Novel Bayesian Similarity Measure for Recommender Systems,” pp. 2619–2625, 2008.

[44] S. Kaushik and P. Tomar, “Evaluation of Similarity Functions by using User based Collaborative Filtering approach in Recommendation Systems,” vol. 21, no. 4, pp. 194–200, 2015.

[45] P. B. Thorat, “Survey on Collaborative Filtering , Content-based Filtering and Hybrid Recommendation System,” vol. 110, no. 4, pp. 31–36, 2015.

[46] D. M. Boyd and N. B. Ellison, “Social Network Sites: Definition, History, and Scholarship.,” *J. Comput. Commun.*, vol. 13, no. 1, pp. 210–230, 2007.

[47] S. Yu and S. Kak, “A Survey of Prediction Using Social Media,” *CoRR*, vol. abs/1203.1, 2012.

[48] H. Sun, Y. Peng, J. Chen, C. Liu, and Y. Sun, “A New Similarity Measure Based on Adjusted Euclidean Distance for Memory-based Collaborative Filtering,” *J. Softw.*, vol. 6, no. 6, pp. 993–1000, Jun. 2011.

[49] I. Konstas, V. Stathopoulos, and J. M. Jose, “On social networks and collaborative recommendation,” *Proc. 32nd Int. ACM SIGIR Conf. Res. Dev.*

Inf. Retr. - SIGIR '09, p. 195, 2009.

[50] I. Guy, N. Zwerdling, I. Ronen, D. Carmel, and E. Uziel, "Social Media Recommendation based on People and Tags," no. Lc, pp. 194–201, 2010.

[51] G. Go, J. Yang, H. Park, and S. Han, "Using Online Media Sharing Behavior as Implicit Feedback for Collaborative Filtering," in *Social Computing (SocialCom), 2010 IEEE Second International Conference on*, 2010, pp. 439–445.

[52] A. Lobzhanidze, W. Zeng, P. Gentry, and A. Taylor, "Mainstream media vs. social media for trending topic prediction - an experimental study," in *Consumer Communications and Networking Conference (CCNC), 2013 IEEE,* 2013, pp. 729–732.

[53] A. Kupavskii, L. Ostroumova, A. Umnov, S. Usachev, P. Serdyukov, G. Gusev, and A. Kustarev, "Prediction of retweet cascade size over time," *Proc. 21st ACM Int. Conf. Inf. Knowl. Manag. - CIKM '12,* p. 2335, 2012.

[54] T. D. Wang, C. Plaisant, B. Shneiderman, N. Spring, D. Roseman, G. Marchand, V. Mukherjee, and M. Smith, "Temporal Summaries: Supporting Temporal Categorical Searching, Aggregation and Comparison," *Vis. Comput. Graph. IEEE Trans.*, vol. 15, no. 6, pp. 1049–1056, 2009.

[55] Y. Ding and X. Li, "Time Weight Collaborative Filtering Categories and Subject Descriptors," *CIKM '05 Proc. 14th ACM Int. Conf. Inf. Knowl. Manag.*, no. ACM New York, NY, USA ©2005, pp. 485–492, 2005.

[56] T.-Q. Lee and Y. Park, "A Time-Based Recommender System Using Implicit Feedback.," in *CSREA EEE,* 2006, pp. 309–315.

[57] G. P. Basharin, A. N. Langville, and V. a. Naumov, "The life and work of A.A. Markov," *Linear Algebra Appl.*, vol. 386, pp. 3–26, Jul. 2004.

[58] D. P. Foster and L. H. Ungar, "Spectral dimensionality reduction for HMMs," *Representations*, no. 1989, pp. 1–26, Mar. 2012.

[59] Y. Huang, T. Huang, K. Wang, and W. Hwang, "A Markov-based Recommendation Model for Exploring the Transfer of Learning on the Web," vol. 12, pp. 144–162, 2009.

[60] L. Xiong, X. Chen, T.-K. Huang, J. G. Schneider, and J. G. Carbonell,

"Temporal Collaborative Filtering with Bayesian Probabilistic Tensor Factorization.," in *SDM*, 2010, pp. 211–222.

[61] C. Desrosiers and G. Karypis, *Recommender Systems Handbook*. Boston, MA: Springer US, 2011.

[62] U. B. Mir, M. Nazir, and A. Yadav, "System Based on Apache Lucene .," vol. 3, no. 2, pp. 18–20, 2014.

[63] J. Kunegis, A. Said, and W. Umbrath, "The Universal Recommender," p. 17, Sep. 2009.

[64] D. G. Medhi and J. Dakua, "MovieReco : A Recommendation System," pp. 564–567, 2007.

[65] S. Yang, B. Long, A. Smola, and H. Zha, "Collaborative competitive filtering: learning recommender using context of user choice," *SIGIR '11 Proc. 34th Int. ACM SIGIR Conf. Res. Dev. Inf.*, pp. 295–304, 2011.

[66] T. A. Dang and E. Viennet, "Collaborative filtering in social networks: A community-based approach," *2013 Int. Conf. Comput. Manag. Telecommun.*, pp. 128–133, Jan. 2013.

[67] M. Munoz-Organero, G. A. Ramiez-Gonza ez, P. J. Munoz-Merino, and C. Delgado Kloos, "A Collaborative Recommender System Based on Space-Time Similarities," *Pervasive Comput. IEEE*, vol. 9, no. 3, pp. 81–87, 2010.

[68] N. Delannay and M. Verleysen, "Collaborative Filtering with interlaced Generalized Linear Models," no. April, pp. 25–27, 2007.

[69] S. M. Kywe, E. Lim, and F. Zhu, "A Survey of Recommender Systems in Twitter," pp. 1–14, 2011.

[70] J. A. Gulla and R. C. Erdur, "A Survey on Challenges and Methods in News Recommendation."

[71] Y. Koren and R. Bell, "Advances in Collaborative Filtering," pp. 43–52.

[72] G. Alzamzmi, Y. Albright, and N. Hanumanthagowdaa, "Comprehensive Survey of Collaborative Filtering Technique," pp. 3–5.

[73] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, “Item-based collaborative filtering recommendation algorithms,” *Proc. tenth Int. Conf. World Wide Web - WWW ’01*, pp. 285–295, 2001.

[74] N. Feed and S. Updates, “Facebook Features Photo and video uploads,” 2010.

[75] M. Jiang, P. Cui, R. Liu, Q. Yang, F. Wang, W. Zhu, and S. Yang, “Social contextual recommendation,” *Proc. 21st ACM Int. Conf. Inf. Knowl. Manag. - CIKM ’12*, p. 45, 2012.

[76] L. Quijano-Sanchez, J. a. Recio-Garcia, B. Diaz-Agudo, and G. Jimenez-Diaz, “Social factors in group recommender systems,” *ACM Trans. Intell. Syst. Technol.*, vol. 4, no. 1, pp. 1–30, Jan. 2013.

[77] Y. Kim and K. Shim, “TWITOBI: A Recommendation System for Twitter Using Probabilistic Modeling,” *2011 IEEE 11th Int. Conf. Data Min.*, pp. 340–349, Dec. 2011.

[78] D. Asanov, “Algorithms and Methods in Recommender Systems,” *Other Conf.*, 2011.

[79] J. Ruff, “Information Overload : Causes , Symptoms and Solutions,” no. Eppler, pp. 1–13, 2002.

[80] D. Lewandowski, “Web searching , search engines and Information Retrieval,” pp. 1–15, 2005.

[81] P. INNOVATION, “E-Commerce : Purchasing and Selling Online – What You Need to Consider,” 2013.

[82] J. Ben Schafer, J. Konstan, and J. Riedl, “Recommender Systems in E-Commerce,” 1993.

[83] N. Manouselis, C. Costopoulou, and A. B. Sideridis, “Studying how e-markets evaluation can enhance trust in virtual business communities Studying How E-Markets Evaluation Can Enhance Trust in Virtual Business Communities,” 2006.

[84] S. Asur and B. A. Huberman, “Predicting the Future with Social Media,” *CoRR*, 2010.

[85] F. Hernández del Olmo and E. Gaudioso, “Evaluation of recommender

systems: A new approach," *Expert Syst. Appl.*, vol. 35, no. 3, pp. 790–804, Oct. 2008.

[86] J. Liu, P. Dolan, and E. R. Pedersen, "Personalized news recommendation based on click behavior," in *Proceedings of the 15th international conference on Intelligent user interfaces*, 2010, pp. 31–40.

[87] A. Poriya, "Non-Personalized Recommender Systems and User- based Collaborative Recommender Systems," vol. 6, no. 9, pp. 22–27, 2014.

[88] J. Liu, P. Dolan, and E. Pedersen, "Personalized news recommendation based on click behavior," *Proc. 15th Int. …*, pp. 31–40, 2010.

[89] K. Bradley and B. Smyth, "Improving Recommendation Diversity," *Business*, pp. 75–84, 2001.

[90] M. Blattner and M. Medo, "Recommendation systems in the scope of opinion formation: a model," p. 8, Jun. 2012.

[91] F. Ricci, L. Rokach, and B. Shapira, *Recommender Systems Handbook*. Boston, MA: Springer US, 2011.

[92] I. Konstas, V. Stathopoulos, and J. M. Jose, "On social networks and collaborative recommendation," in *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, 2009, pp. 195–202.

[93] S. M. Kywe, "A Survey of Recommender Systems in Twitter," 2012.

[94] G. Shani and A. Gunawardana, "Evaluating Recommendation Systems," *Recomm. Syst. Handb.*, vol. 12, no. 19, pp. 1–41, 2011.

[95] G. Shani and A. Gunawardana, "Evaluating recommendation systems," *Recomm. Syst. Handb.*, pp. 257–298, 2011.

[96] O. Chacaltana, J. Distler, and A. Trimm, "Tinkertoys for the Twisted E_6 Theory," p. 52, Jan. 2015.

[97] M. Sharma, "A Survey of Recommender Systems : Approaches and Limitations," pp. 1–9, 2013.

[98] X. Su and T. M. Khoshgoftaar, “A Survey of Collaborative Filtering Techniques,” *Adv. Artif. Intell.*, vol. 2009, no. Section 3, pp. 1–19, 2009.

[99] O. S. Revankar and Y. V Haribhakta, “Survey on Collaborative Filtering Technique in Recommendation System,” vol. 4, no. 3, pp. 85–91, 2015.

[100] K. Yu, A. Schwaighofer, V. Tresp, X. Xu, and H.-P. Kriegel, “Probabilistic memory-based collaborative filtering,” *Knowl. Data Eng. IEEE Trans.*, vol. 16, no. 1, pp. 56–69, 2004.

[101] M. R. McLaughlin and J. L. Herlocker, “A collaborative filtering algorithm and evaluation metric that accurately model the user experience,” *Proc. 27th Annu. Int. Conf. Res. Dev. Inf. Retr. - SIGIR ’04*, p. 329, 2004.

[102] J. Pandey, “A Survey on Recommender system using collaborative filtering Techniques,” vol. 2, no. January, pp. 241–245, 2015.

[103] P. Resnick, P. Bergstrom, and J. Riedl, “GroupLens : An Open Architecture for Collaborative Filtering of Netnews,” pp. 175–186, 1994.

[104] G. Qian, S. Sural, Y. Gu, and S. Pramanik, “Similarity between Euclidean and cosine angle distance for nearest neighbor queries,” in *Proceedings of the 2004 ACM symposium on Applied computing - SAC ’04*, 2004, p. 1232.

[105] H. Parvin, H. Alizadeh, and B. Minaei-bidgoli, “MKNN : Modified K-Nearest Neighbor,” in *Proceedings of the World Congress on Engineering and Computer Science, WCECS*, 2008, pp. 22–25.

[106] G. Qian, S. Sural, Y. Gu, and S. Pramanik, “Similarity between Euclidean and cosine angle distance for nearest neighbor queries,” in *Proceedings of the 2004 ACM symposium on Applied computing - SAC ’04*, 2004, p. 1232.

[107] M. D. Ekstrand, “Collaborative Filtering Recommender Systems,” *Found. Trends® Human–Computer Interact.*, vol. 4, no. 2, pp. 81–173, 2010.

[108] J. Redpath, D. H. Glass, S. Mcclean, and L. Chen, “User-based Collaborative Filtering : Sparsity and Performance.”

[109] J. L. Herlocker, J. a. Konstan, L. G. Terveen, and J. T. Riedl, “Evaluating collaborative filtering recommender systems,” *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 5–53, 2004.

[110] M. Degemmis, P. Lops, and G. Semeraro, *A content-collaborative recommender that exploits WordNet-based user profiles for neighborhood formation*, vol. 17, no. 3. 2007.

[111] I. Paik and H. Mizugai, “Recommendation System Using Weighted TF-IDF and Naive Bayes Classifiers on RSS Contents,” pp. 5–6, 2010.

[112] R. Burke, “Hybrid Recommender Systems: Survey and Experiments,” *User Model. User-adapt. Interact.*, vol. 12, no. 4, pp. 331–370, Nov. 2002.

[113] G. Adomavicius and A. Tuzhilin, “Toward the Next Generation of Recommender Systems : A Survey of the State-of-the-Art and Possible Extensions,” vol. 17, no. 6, pp. 734–749, 2005.

[114] R. Burke, “Hybrid Recommender Systems : Survey and Experiments †,” pp. 1–29.

[115] A. Bellogín, I. Cantador, F. Díez, P. Castells, and E. Chavarriaga, “An empirical comparison of social, collaborative filtering, and hybrid recommenders,” *ACM Trans. Intell. Syst. Technol.*, vol. 4, no. 1, pp. 1–29, Jan. 2013.

[116] P. C. Vaz, D. M. de Matos, B. Martins, and P. Calado, “Improving an Hybrid Literary Book Recommendation System through Author Ranking,” pp. 1–4, Mar. 2012.

[117] D. M. Boyd and N. B. Ellison, “Social network sites: definition, history, and scholarship,” *IEEE Eng. Manag. Rev.*, vol. 38, 2010.

[118] W. Chen and S. Fong, “Social network collaborative filtering framework and online trust factors: A case study on Facebook,” *2010 Fifth Int. Conf. Digit. Inf. Manag.*, pp. 266–273, Jul. 2010.

[119] Z. Wang, W. Zhu, P. Cui, L. Sun, and S. Yang, “Social Media Recommendation,” pp. 1–20.

[120] T. A. Dang and E. Viennet, “Collaborative filtering in social networks: A community-based approach,” *Comput. Manag. Telecommun. (ComManTel), 2013 Int. Conf.*, pp. 128–133, 2013.

[121] A. Jameson and B. Smyth, “Recommendation to Groups,” *Adapt. Web*, pp. 596–627, 2007.

[122] J. Masthoff, “Group modeling : Selecting a sequence of television items to suit a group of viewers,” 2001.

[123] M. O. Connor, D. Cosley, J. A. Konstan, and J. Riedl, “PolyLens : A Recommender System for Groups of Users.”

[124] Z. Wang, J. Liao, Q. Cao, H. Qi, and Z. Wang, “Friendbook: A Semantic-Based Friend Recommendation System for Social Networks,” *Mob. Comput. IEEE Trans.*, vol. 14, no. 3, pp. 538–551, 2015.

[125] E. Dattaprasad S., N. Jain, S. S. Mazire, A. B. Patil, and P. D. T. Kurian, “Friend Recommendation System Based on Lifestyle,” vol. 5, no. 2, pp. 347–349, 2015.

[126] L. H. Ungar and D. P. Foster, “Clustering Methods for Collaborative Filtering What is Collaborative Filtering ?,” pp. 114–129.

[127] G. Guo, J. Zhang, D. Thalmann, and N. Yorke-smith, “ETAF : An Extended Trust Antecedents Framework for Trust Prediction,” *2014 IEEE/ACM Int. Conf. Adv. Soc. Netw. Anal. Min.*, 2014.

[128] G. Guo, J. Zhang, and N. Yorke-smith, “TrustSVD : Collaborative Filtering with Both the Explicit and Implicit Influence of User Trust and of Item Ratings,” *AAAI Publ. Twenty-Ninth AAAI Conf. Artif. Intell.*, 2015.

[129] D. Mican, L. Mocean, and N. Tomai, “Building a Social Recommender System by Harvesting Social Relationships and Trust Scores Between Users.”

[130] I. Guy, N. Zwerdling, I. Ronen, D. Carmel, and E. Uziel, “Social media recommendation based on people and tags,” in *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, 2010, pp. 194–201.

[131] G. Go, J. Yang, H. Park, and S. Han, “Using Online Media Sharing Behavior as Implicit Feedback for Collaborative Filtering,” *2010 IEEE Second Int. Conf. Soc. Comput.*, pp. 439–445, Aug. 2010.

[132] L. L. Yu, S. Asur, and B. A. Huberman, “What Trends in Chinese Social Media,” *CoRR*, vol. abs/1107.3, 2011.

[133] D. Mican, L. Mocean, and N. Tomai, “Building a Social Recommender

System by Harvesting Social Relationships and Trust Scores between Users.," in *BIS (Workshops)*, 2012, vol. 127, pp. 1–12.

[134] Z.-K. Zhang, T. Zhou, and Y.-C. Zhang, "Tag-aware recommender systems: a state-of-the-art survey," *J. Comput. Sci. Technol.*, vol. 26, no. 5, pp. 767–777, Sep. 2011.

[135] L. Balby Marinho, A. Nanopoulos, L. Schmidt-Thieme, R. Jäschke, A. Hotho, G. Stumme, and P. Symeonidis, "Social Tagging Recommender Systems," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. New York: Springer, 2011, pp. 615–644.

[136] A. Krzywicki, W. Wobcke, X. Cai, A. Mahidadia, M. Bain, P. Compton, and Y. S. Kim, "Interaction-Based Collaborative Filtering Methods for Recommendation in Online Dating."

[137] M. C. Pham, Y. Cao, R. Klamma, and M. Jarke, "A Clustering Approach for Collaborative Filtering Recommendation Using Social Network Analysis.," *J. UCS*, vol. 17, no. 4, pp. 583–604, 2011.

[138] L. B. Marinho, I. Nunes, T. Sandholm, C. N brega, J. Arajo, and C. E. S. Pires, "Improving location recommendations with temporal pattern extraction," in *Proceedings of the 18th Brazilian symposium on Multimedia and the web*, 2012, pp. 293–296.

[139] N. Lathia, S. Hailes, and L. Capra, "Temporal collaborative filtering with adaptive neighbourhoods.," in *SIGIR*, 2009, pp. 796–797.

[140] D. Stirzaker, "Markov Chains," pp. 396–477, Oct. 2003.

[141] M. Jacobsen, "Markov Chains and Markov Processes," no. 1, pp. 1–3, Sep. 2006.

[142] T. Konstantopoulos, "MARKOV CHAINS AND RANDOM WALKS," 2009.

[143] T. Markov, "FINITE-STATE MARKOV CHAINS," vol. 103.

[144] X. Wu, Q. Liu, E. Chen, L. He, J. Lv, C. Cao, and G. Hu, "Personalized Next-song Recommendation in Online Karaokes," 2013.

[145] M. Eirinaki, M. Vazirgiannis, and D. Kapogiannis, "Web path recommendations based on page ranking and Markov models," *Proc. seventh*

ACM Int. Work. Web Inf. data Manag. - WIDM '05, p. 2, 2005.

[146] F. Fouss, S. Faulkner, M. Kolp, A. Pirotte, and M. Saerens, "MARKOV-CHAIN MODEL," 2001.

[147] J. Delporte, A. Karatzoglou, and T. Matuszczyk, "Socially Enabled Preference Learning from Implicit Feedback Data."

[148] I. Cantador, P. Brusilovsky, and T. Kuflik, "2nd Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec 2011)," in *Proceedings of the 5th ACM conference on Recommender systems*, 2011.

[149] K. Stefanidis, I. Ntoutsi, K. Nørvåg, and H.-P. Kriegel, "A Framework for Time-Aware Recommendations.," in *DEXA (2)*, 2012, vol. 7447, pp. 329–344.