

1-1-2013

A Tag And Social Network Based Recommender System

Sogol Naseri
Ryerson University

Follow this and additional works at: <http://digitalcommons.ryerson.ca/dissertations>



Part of the [Computer Sciences Commons](#)

Recommended Citation

Naseri, Sogol, "A Tag And Social Network Based Recommender System" (2013). *Theses and dissertations*. Paper 2061.

This Thesis is brought to you for free and open access by Digital Commons @ Ryerson. It has been accepted for inclusion in Theses and dissertations by an authorized administrator of Digital Commons @ Ryerson. For more information, please contact bcameron@ryerson.ca.

A TAG AND SOCIAL NETWORK BASED RECOMMENDER SYSTEM

by

Sogol Naseri

B.E. in Software Engineering, Azad University, Iran, 2008

A thesis

presented to Ryerson University

in partial fulfillment of the

requirements for the degree of

Master of Science

in the program of

Computer Science

Toronto, Ontario, Canada, 2013

©Sogol Naseri 2013

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my thesis may be made electronically available to the public.

A TAG AND SOCIAL NETWORK BASED RECOMMENDER SYSTEM

Sogol Naseri
Master of Science, Computer Science, 2013
Ryerson University

ABSTRACT

In the era of the Internet, information overload is a growing problem which refers to the inability of a person to make a decision because the amount of information that she/he needs to process is huge. To solve this problem, recommender systems were proposed to apply various algorithms to recognize users' preferences and generate recommendations which are likely match the user's interest on various items. In this thesis, we aim to improve the effectiveness of the recommendation by incorporating the social data into the traditional recommendation algorithms. Hence, we first propose a new user similarity metric that not only considers tagging activities of users, but also incorporates their social relationships, such as friendships and memberships, in measuring the nearest neighbours. Subsequently, we define a new recommendation method which makes use of both user-to-user similarity and item-to-item similarity. Experimental outcomes on a Last.fm dataset show positive results of our proposed approach.

ACKNOWLEDGEMENTS

I am indebted to the kind people around me. Without their help and support, it would not have been possible to write this thesis.

It gives me great pleasure in acknowledging the support of my supervisor Dr. Cherie Ding, for her valuable advice and great patience, whose effort made this thesis possible. I would like to thank her for the time and effort she has put in our weekly meetings and reviewing my thesis many times.

I am very thankful to Dr. Sadeghian, Dr. Abhari and Dr. Ferworn who have reviewed my thesis and given me insightful feedbacks, which enabled me to improve my thesis.

I would like to express my deep appreciations to my family, for their great patience and motivation at all times and inspiring me to follow my dreams. I owe my deepest gratitude to my parents who supported me emotionally, financially and provided comfort to me at all times.

Last, but by no means least, I share the credit of my work with my very special and dear friend, Arash Bahrehmand, who encouraged me, fueled me confidence, provided me with the required resources for this work and helped me to achieve my dreams.

TABLE OF CONTENTS

AUTHOR’S DECLARATION.....	ii
ABSTRACT.....	iii
ACKNOWLEDGEMENTS	iv
LIST OF TABLES	vii
LIST OF FIGURES	ix
LIST OF ACRONYMS	x
CHAPTER 1	1
INTRODUCTION.....	1
1.1 Background and the Problem Statement	1
1.1.1 Background	1
1.1.2 Problem Statement	3
1.2 The Proposed Methodology	5
1.3 Objectives.....	7
1.4 Thesis Outline.....	8
CHAPTER 2	10
LITERATURE REVIEW	10
2.1 Introduction	10
2.2 Collaborative Filtering (CF).....	10
2.2.1 Memory based Collaborative Filtering.....	11
2.2.2 Model based Collaborative Filtering.....	12
2.2.3 Hybrid Collaborative Filtering	12
2.2.4 Overview of Collaborative Filtering Techniques.....	13
2.3 Social Tagging Systems	13
2.4 Collaborative Social Tagging Recommender Systems	16

2.5 Summary	21
CHAPTER 3	22
METHODOLOGY	22
3.1 Preliminaries.....	22
3.2 Measuring the Similarity.....	25
3.2.1 Tag-based Similarity	26
3.2.2 Friendship-based Similarity	29
3.2.3 Membership-based Similarity	31
3.2.4 Overall Similarity	33
3.3 Making Recommendation	34
3.4 A Case Study Illustrating the Recommendation Process	38
3.5 Summary	43
CHAPTER 4	44
EXPERIMENTS	44
4.1 Introduction	44
4.2 Evaluation Metrics	46
4.2.1 Precision	46
4.2.2 Recall.....	47
4.2.3 Datasets	47
4.3 Implementation.....	51
4.4 Preprocessing.....	53
4.4.1 Finding the Tag Set of Each Group.....	53
4.4.2 Finding α and β	53
4.5 Error Evaluation based on 20-80 Method	56
4.6 Summary	61

CHAPTER 5	63
CONCLUSIONS AND FUTURE WORK	63
5.1 Conclusion	63
5.2 Future Work	64
REFERENCES	65

LIST OF TABLES

Table 2-1 Overview of Collaborative Filtering Techniques [26]	13
Table 3-1 A sample user-item matrix	27
Table 3-2 Information of users, items and tags.....	38
Table 3-3 Membership information	38
Table 3-4 Friendship Relations	39
Table 3-5 $TSim_{u,v}$ matrix	39
Table 3-6 Average of each user’s friends’ similarities	39
Table 3-7 Friendship similarity matrix	40
Table 3-8 Membership similarity matrix	40
Table 3-9 Overall similarity matrix	41
Table 3-10 Item similarity matrix	42
Table 3-11 NeighbourItemAvg list	42
Table 4-1 Features of the Last.fm dataset.....	48
Table 4-2 Sample of annotation file.....	49
Table 4-3 Sample of item file	49
Table 4-4 Sample of tag file.....	50
Table 4-5 Sample of group file	50
Table 4-6 Sample of friend file.....	51
Table 4-7 Precision value in the Top 20	55
Table 4-8 Recall value in the Top 20.....	56
Table 4-9 Precision when using our algorithm.....	58
Table 4-10 Recall when using our algorithm.....	58
Table 4-11 Precision when using the Augmenting algorithm	59
Table 4-12 Recall when using the Augmenting algorithm	59

LIST OF FIGURES

Figure 2-1	Social tagging structure[36]	15
Figure 3-1	A model for social network structure.....	24
Figure 4-1	Classes of our implementation.....	52
Figure 4-2	A possible area of variation of α and β	54
Figure 4-3	Improvements on Precision on all of the fusion approaches.....	60
Figure 4-4	Improvements on Recall on all of the fusion approaches	61

LIST OF ACRONYMS

RS: Recommender System

CF: Collaborative Filtering

LDA: Latent Dirichlet Allocation

ARM: Association Rule Mining

IUF: Inverse User Frequency

CHAPTER 1

INTRODUCTION

1.1 Background and the Problem Statement

1.1.1 Background

Recently, Recommender Systems (RS) have played an important role in reducing the information overload, through personalized recommendations based on user preferences and behaviour [1]. Information overload is a predominant and growing issue which leads to an incongruity between the amount of information and the ability to process that information [1, 2].

Nowadays, people are looking for their desired items on the internet, and making their selection decisions based on factors such as ratings and reviews on the candidate items from other customers, recommendation from other people or from recommender systems. The choice of the RS depends upon what a user wants to purchase. Suppose that a colleague suggests you to watch a particular movie and then you visit your favourite online movie website. After entering the name of the movie in the search box, it presents the top-10 movies matched with the searched keywords. In a specific area of the webpage called "Users Who Watched This Movie Also Watched," a list of movies that you might like is displayed. If you are a regular registered user of this type of online movie website, such a personalized list of recommendations will appear automatically as soon as you enter the website. The software application that specifies which items should be recommended to a target user is called a *recommender system*. These types of scenarios are useful for determining several characteristics of such software applications. It is

worth mentioning that in *personalized* recommendations every user gets a special list depending on her/his previous history.

In general a RS is useful to recommend the most relevant items to users based on their interests. These systems use a wide range of techniques for filtering user data and suggest relevant items[3, 4]. Therefore, the way a RS personalizes its recommendation is based on the filtering process. This leads to three types of RS:

- Content-based filtering: recommends to the user resources that have structural similarity with those she/he selected before. The similarity is mainly measured by the information extracted from the content of the item.
- Collaborative filtering (CF): recommends to the user the items selected by the most like-minded neighbours based on her/his neighbours' previous rating histories. If there are u users in a system, each user has $u-1$ neighbours. Therefore, if neighbours share the same interest on one item, that item will be recommended to the user.
- Hybrid model: generally uses the combination of CF and the content-based filtering.

In this study, we apply a special type of collaborative filtering method, which takes tags and social network information into account when predicting the behaviour of users.

Tag-based systems can determine/express user preferences for a resource by providing specific ways for web users to expose their personal opinions in their own words[5]. These systems allow users to annotate resources, especially multimedia content, with keywords which become pronounced in text-based information retrieval[6]. The newer generations of tagging systems are equipped with social networking facilities[7] that allow users to connect themselves to each other through a variety of geo-social factors such as friendship[8], membership, location and so forth.

The social tagging system is a rich environment which enables an analysis to analyze the user interest and the items' attributes from the social relations between the users and then identify the key factors affecting user decisions. These systems consist of three major elements: users, tags, and relationships. Each of these components can potentially be subjected to investigation in order to develop a recommendation method.

In order to study how the combination of the social network information with the traditional collaborative filtering method can affect the accuracy of recommendations, we made use of the Last.fm dataset¹ in which users are free to tag music, find friends and join groups.

1.1.2 Problem Statement

Over the past few years, researchers have taken tremendous strides in attempting to combine human judgment and automated filtering in recommender systems. One of the most effective recommendation approaches is collaborative filtering, which makes recommendations based on previously rated data. Most of the collaborative filtering approaches are based on common ratings or tags. However, measuring the similarity of users only based on common tags is not appropriate, since perhaps two users have many common tags, but these common tags are not assigned to the same items. On the other hand, measuring similarities of users based only on common items would not be a precise measure either, since users might have many common items but assign them different tags or ratings to those items which are in common. To the best of our knowledge, most of the current tag-based systems do not emphasize using only the common tags on common items to measure user similarity, which is indeed the approach we used in our recommender system.

¹ http://www.di.unito.it/~schifane/dataset_lastfm_WSDM.zip.

Recently, the use of social tagging websites has become increasingly popular in recommendation systems, since these types of websites reflect the user behaviour in a more precise way. It is worth pointing out that the application of the social networking information in recommender systems is diverse, and this difference depends on the type of users and resources. For instance each of these recommender systems, such as Last.fm², Facebook³ and Amazon⁴, recommends different resources, which are music, friends and books respectively.

Another one of the most prevalent advantages of using social networking information is the friendship relation with which users define their neighbours explicitly. Although at a first glance recommending items that a user's friends have already bought or listened to appears reasonable, this approach suffers from some critical drawbacks. For instance, perhaps the user has friends with diverse backgrounds. Thus, probably there are some friends whose interests do not match this user's interest. As a matter of fact, humans make friends with others based on different aspects of their personality, so, depending on their particular common aspects; the type and the level of each friendship relation are different. So far there are only a limited number of research efforts, which combine the collaborative filtering and the friendship information which could be used to intensify the similarity value based on friends' data [9]. In this work, we try to identify the friends of a given user and determine how much this user relies on a friend in terms of getting recommendation on selecting a particular item. This friendship relation can help us further improve the accuracy of our recommender system.

Finally, a widespread social networking activity of users is participating in groups. Although a group can be interpreted as a community of users with common interests on a particular

² <http://www.last.fm/>.

³ <https://www.facebook.com/>.

⁴ <http://www.amazon.com/>.

subject, membership alone cannot be a consistent criterion to infer that all users of an active group have exactly the same interest on the group subjects. As a part of our system, we come up with measuring the level of participation of users in the group in order to discriminate among users based on their group activities. In this sense the more two users have the same experience in a group, the more similarity value can be assigned to them. On the other hand the membership is not a reliable measurement on its own since perhaps a user has diverse interests and she/he is a member of various types of groups. So far, in literatures, measuring the level of users' participation in common groups has not received the attention it deserves.

Therefore, a framework should be developed for assessing how interests affect friendships and memberships along with how friendships and membership affect interests. Consequently, our solution to tackle these issues is to consider different aspects of user behaviour by measuring their similarity based on the combination of various user activities such as tagging, making friends and participating in groups. Then based on a thorough consideration of the user similarity calculation, our collaborative filtering-based recommender system could make recommendations in a more accurate way.

1.2 The Proposed Methodology

Collaborative filtering algorithms attempt to find users who are related to each other with the intention of using their shared interests to produce recommendations. In traditional methods, the only considered relation was user similarity, called the “neighbourhood based model”.

In general, we have two types of relationships: implicit and explicit. The former focuses on those relationships which are inferred from user behaviour, such as similarity, and the latter is more related to those relationships directly defined by users, such as friendship and membership. Social relationships not only alleviate some of the limitations of implicit relationships, such as

the data sparsity problem [10], but also can potentially be applied to boost collaborative intelligence in finding key users who really have an impact on the decision making of others. This study will shed light on proposing a new approach for combining implicit and explicit relationships to increase recommendation effectiveness.

Suppose, A is a target user with a sorted nearest neighbours list, $NL = \{B, C, D, E\}$, based on her/his tagging activity on common items with other users. Also A has different kinds of social relationships with these neighbours. For instance, if B and D are friends of A , based on their level of friendships with A , we amplify their similarity values. Therefore, based on friendship information, the order of neighbours of A changes to: $\{B, D, C, E\}$. However, based on membership information and level of participation of neighbours in the groups that A participates in, the order of neighbours in NL changes to: $\{C, B, E, D\}$. We tend to achieve a more accurate order of neighbours based on their tagging activity and their social relationships with the target user. Therefore, in our approach by combining all these information, the order of neighbours is modified in a way that the most like-minded neighbours to the target user are switched to the front. For instance, in our example, NL changes to $\{B, D, C, E\}$ after considering these three types of information.

In our approach nearest neighbours of a user are calculated by a new similarity metric based on her/his tagging history and social activities, whereas most of the traditional collaborative filtering approaches obtain Top k nearest neighbours for a user based on her/his rating using Pearson correlation or cosine correlation formulas [11, 12]. The proposed similarity metric includes the combination of these three main similarity measures: similarity based on common tags on common items, similarity based on friendship and similarity based on membership. Therefore, we need to measure the significance of friendship between similar neighbours for

each user in the system. Moreover, we need to measure the belonging level of each user to a group. Subsequently, a highly preferred item will be recommended to the user based on the interest of the user's neighbours' interests. Finally, a list of sorted items will be suggested to the user based on our top- N recommendation approach which takes into account the item similarity.

1.3 Objectives

In this thesis, we demonstrate the effect of combining collaborative filtering with social network information to enhance the effectiveness of recommender systems. Traditional collaborative filtering algorithms are limited to common items and ratings. In order to achieve a better understanding of user interests, we should consider other aspects of the user's behaviour.

Friendship and membership information are among the most tangible piece of data for judging a user's behaviour. For this reason, we decided to quantify the value of these relations between user pairs in order to evaluate how close they are to each other.

In our framework, the user's interest about an item is measured based on her/his assigned tags. In addition, the similarity of two users, by ignoring their social activities, depends on the closeness of their tagging behaviour for common items. Then social network information is fused into our calculation with the intention of improving the accuracy of our understanding about the user's behaviour.

Our main objectives span these three issues:

- 1) Proposing a novel weighted combination similarity metric which takes into account three main sources: the number of common tags on common items, the significance of friendship and the quantity of shared activities on common groups.
- 2) Proposing unique similarity metrics based on each of the three above mentioned sources.

In tag-based similarity calculation, we consider both common tags and common items. In

friendship based similarity calculations, we extract those friends who are more similar to the target user. Finally in membership based similarity calculations we consider the level of the degree of belonging of each user to a group.

- 3) Designing a collaborative filtering based recommender system which makes use of the user similarity scores generated in the above mentioned method combined with item similarity. The recommended item considers the user's past selection and tagging activities as well as her/his social activities, which makes the result more accurate.

The result of this study can be applied in a variety of recommender systems. These systems should provide users some functionality such as assigning tags, making friends and participating in groups. Hence, our algorithm could be applicable for the Last.fm, Flickr and Del.icio.us sites. It's worth mentioning that even if a system does not support one of these facilities, for example participating in groups, our algorithm still can be useful, since it features a weighted similarity approach. In the mentioned example (i.e., no functionality for users to participate in groups), by assigning the weight on group similarity as zero, the overall similarity can be evaluated based on the combination of the item-tag similarity and friendship-based similarity.

1.4 Thesis Outline

The thesis is organized as follows:

In Chapter 2 we present a state-of-the-art review spanning over multiple aspects of collaborative filtering methods and social tagging systems, and then we briefly survey the related literature.

In Chapter 3 the methodology of our system is presented. Moreover, the procedure of finding nearest neighbours of the target user is explained, which contains three parts: similarity based on common tags on common items, similarity based on friendship and similarity based on

membership. In addition, the overall similarity measurement, combination of these three similarities, is presented. We then provide a method for item recommendation. Finally, to illustrate the similarity computation and the item recommendation algorithm process we provide a use case scenario.

In Chapter 4 we discuss some details about the implementation of our recommendation algorithms on Last.fm dataset, a widespread social networking based website. Moreover, experimental results based on two evaluation metrics (precision and recall) are shown. Finally, this chapter ends up with result analyses and a comparison with one of the most closely related algorithms to this approach.

In Chapter 5 we draw some conclusions about our study. Some suggestions about our possible future work are pointed out.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

There are three types of related work we would like to review in this chapter: i) Collaborative filtering approaches, ii) Social tagging systems, iii) Collaborative Social tagging recommender systems.

2.2 Collaborative Filtering (CF)

These days, people tend to rely more on other people's opinions than their own research efforts. The opinion of the other people is obtainable from their recommendations or comments or feedbacks through verbal words, surveys, reference letters, news, media, travel guides, etc.[13-16]. Discovering the most valuable item from this vast amount of information is a challenging and time consuming process. Recommender systems reduce the information overload problem and help people select their desired items such as web pages, movies, books, articles, products and so forth. Recommender systems apply different algorithms to filter data and provide personalized recommendations based on the user's preferences [17-20].

Collaborative filtering is one of the most popular recommendation techniques, which is used to filter information [21, 22] through collaborative knowledge. The main idea of these systems is that if users shared the same interests in the past[23] - if they viewed same movies, for instance - they will also have similar behaviour in their future choices[23]. For example, user u and user v have a rating history that shows they shared the same interest, and later when user u has rated a movie that v has not yet rated, an intelligent system can recommend this movie also to v [24-

26]. This kind of recommendations, which filters the most promising ones from a vast amount of data so that users indirectly collaborate with each other, is called collaborative filtering.

As explained before, collaborative filtering algorithms mainly consider the rating of users. The process of rating could be explicitly carried out by means of a rating scale, or it could be identified implicitly if the user bought or selected that item[27]. In this system the next preferred item of the user is recognized by finding the opinions of similar users to this user[12, 21]. Then based on the level of similarity between users, the nearest neighbours with the high similarity values are identified. When the nearest neighbours are known, a weighted summation of the nearest neighbours' interests is formed. Based on this weighted summation, a particular item can be evaluated and recommended to the user[21].

A collaborative filtering algorithm is required in order to make an acceptable recommendation in a short period of time on a sparse dataset. This algorithm is categorized to three main groups as explained in the following sub-sections.

2.2.1 Memory based Collaborative Filtering

Memory based algorithms use a user-item database to make a prediction. Similar users are those who have the same interests as the target user. By identifying the nearest neighbours a new prediction based on the neighbours' preferences can be generated for the user[25, 27]. This algorithm calculates the similarity between two users or items [28]. In order to filter the information, nearest neighbours should be recognized based on the Pearson correlation coefficient similarity, cosine similarity or Jaccard similarity measures [11, 12, 29]. Then based on the weighted average of all of the ratings of users or items on an item by a user, or based on the simple weighted average, a prediction can be generated for the user or item [12, 28, 29]. After computing similarities in order to produce top- N recommendation, the k most similar users

or items (the nearest neighbours) should be recognized. Afterwards, according to the nearest neighbours' opinions the most interesting N items for the target user are recommended.

2.2.2 Model based Collaborative Filtering

In the model based collaborative filtering technique, complex patterns are generated on the training data. Then, based on the learned models intelligent predictions are made for the test data or the real world data [26]. Some of the model based collaborative filtering algorithms such as Bayesian models, dependency networks, and clustering models are made to help the memory based algorithm overcome some of its shortcomings [29-31] .

2.2.3 Hybrid Collaborative Filtering

Hybrid collaborative filtering combines the collaborative filtering algorithm with the content based algorithm to generate recommendations.

A content based algorithm makes recommendations based on analyzing textual materials or content of documents, URLs, weblogs, item descriptions, profiles about users' preferences and recognizing consistencies in the content [31, 32]. The importance of the textual content is recognizable through different factors such as: term frequency or inverse document frequency[33].

The success of content-based recommendation depends on the availability of item explanations and a profile that assigns prominence to these features. In the case of a movie store, the characteristics of movies might include the genre, artists, or the director. On the other hand, user profiles may also be learned either by analyzing user behaviour or by inquiring explicitly about the user favourites [26].

All approaches we have seen so far have advantages and of course disadvantages depending on a certain problem condition. One reasonable solution is to combine different methods to

generate more accurate recommendations with a better performance.

2.2.4 Overview of Collaborative Filtering Techniques

All of the discussed techniques of the collaborative filtering algorithms are briefly depicted in Table 2.1[26].

Table 2-1 Overview of Collaborative Filtering Techniques [26]

CF Techniques	Representative Techniques	Advantages	Shortcomings
Memory based CF	<ul style="list-style-type: none"> • Neighbour-based CF • Item-based/user-based • Top N recommendation 	<ul style="list-style-type: none"> • Easy to Implement • New data can be added easily • No need to know the content of the item to being recommended 	<ul style="list-style-type: none"> • Are dependent on human ratings • Sparse dataset decreases the performance • Scalable in a large dataset
Model based CF	<ul style="list-style-type: none"> • Bayesian belief nets CF • Clustering CF • Latent semantic CF • Sparse factor analysis 	<ul style="list-style-type: none"> • Solve the sparsity problem • Improve performance of prediction 	<ul style="list-style-type: none"> • Expensive in model building • There is a tradeoff between prediction performance and scalability
Hybrid based CF	<ul style="list-style-type: none"> • Content-based CF • Memory-based and model-based • Content boosted CF 	<ul style="list-style-type: none"> • Improve predication performance • Overcome the limitation of CF and content-based and other algorithms 	<ul style="list-style-type: none"> • Expensive and complex to implement • External information is needed which is usually unavailable

2.3 Social Tagging Systems

Tags are keywords assigned to an object (photos, music tracks, videos and etc.) to provide a meaningful description for it. Although tags are powerful tools in searching, organizing and discovering items, these keywords also play an important role in recognizing personal interests,

goals, and preferences of users[34]. In order to find whether tags are useful in generating personalized recommender systems, Durao and Dolog [34] present a tag based recommender system which recommends web pages based on their tag similarities. In other words, for the purpose of suggesting personalized resources, an extension method is proposed for computing similarities between tags; in a way that similarity calculations are a combination of the cosine similarity metric with other factors such as: tag frequency, tag popularity and affinity between the user and a tag[34]. Tso-Sutter et al. [35] propose a new method which incorporates tags in the CF algorithm and applies a three two-dimensional correlations for item , tag and user. According to this study the recommendation accuracy is enhanced when the tag information is added to the system. Tags not only are used to organize contents and define a clue why the user liked something, but also are beneficial for users to help them to find their interesting items [36]. Moreover, recently some new studies focus on the further use of tag information for the tag based recommendation systems.

Tagging techniques in the tag-based systems such as: Flickr⁵, Del.icio.us⁶ , Last.fm and CiteU-Like⁷ , provide a rich method for organizing, managing and locating user contents. For example, users of Flickr assign tags to manage their photos and to explore other interesting photos. On Del.icio.us, tags are used to help users organize, share, and discover bookmarks. In some popular recommendation websites such as Del.icio.us and Last.fm, tag and social networking information are associated. In Last.fm people assign tags to the tracks, albums or artist and they can make friends or join their interesting groups. Also, CiteU-Like is a free service to store, share and organize academic papers[37].

⁵ <http://www.flickr.com/>.

⁶ <https://www.delicious.com/>.

⁷ <http://www.citeulike.org/>.

The process of analyzing the relationship between users, items, and tags can be categorized in three main groups:

- 1) Tags connect users and items: Tags are helpful for users to find items and similarly tags are useful for items in a way that items could connect users. For example the “software” tag could be found easily by a user searching for bookmarks about software[36].
- 2) Tags connect items: Different items may share the same tags. For example: in the Last.fm website different songs may share common tags. So it means that these songs could be categorized into the same cluster. Then users could find other songs based on their tag (interest)[36].
- 3) Tags connect users: Different users may use the same tag for items. So users with similar interests will be recognized through their shared common tags[36].

Figure 2.1 depicts the relations between tags, users and items. In Figure 2.1 i_1 , i_2 and i_3 represent items. Similarly, u_1 , u_2 and u_3 are users of the system. In this Figure all the three above connections are combined and relations between tags are extracted.

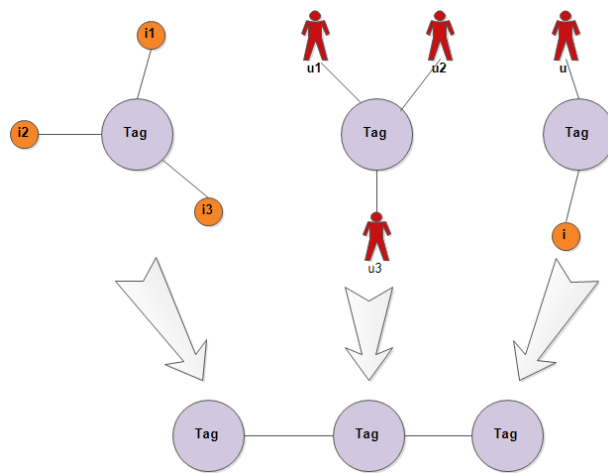


Figure 2-1 Social tagging structure[36]

A social network consists of nodes and connections in which nodes represent particular individuals or organizations. The interconnected nodes are based on different dependency relationships (edges) such as: friends, sexual relations, ideas, conflict, financial exchange, dislike, airline routes, and so on [22, 38]. Some examples of social networks include Facebook, Twitter⁸, LinkedIn⁹, etc.

Social tagging systems support social communications and the tagging activity of users during these interactions. In social tagging sites usually the assigned tags help users to revisit their previously visited pages or objects[39]. Moreover, these systems are very powerful in helping users to search for other items interesting to users.

2.4 Collaborative Social Tagging Recommender Systems

Collaborative Filtering (CF) is the most practical and commercially successful approach for recommendation. CF algorithms can simply combine with the social network information by providing the suitable recommendation for the user based on her/his previously liked items. In order to filter the information, nearest neighbours should be recognized using the Pearson correlation coefficient similarity, cosine similarity or Jaccard similarity measures. In this sense, by finding the k nearest neighbours the most interesting items are recommended to users.

In the past few years, application of extracted social data from social web sites has become increasingly popular; in a way, fusing social networking information with recommender systems for increasing the level of personalization has received a significant attention from the research communities[40, 41]. Therefore, some researchers broached the idea of using the trust theory. Golbeck in 2006 [42] presents the benefit of considering the trust level in social network systems to improve the accuracy of movie recommendations. In [43] Ziegler and Lausen present a

⁸ <https://twitter.com/>.

⁹ <http://www.linkedin.com/>.

positive correlation between interest similarity and trust. In doing so, users define their trust to other users in some web sites such as product review websites. Consequently, a good recommendation will be predicted based on the expressed level of trust. In some papers [44-46] social networking information is incorporated with collaborative filtering algorithms to find the nearest neighbours. Moreover, these papers consider the trust factor which predicts the trust level of users to the other users. Trust theory requires the trust value which is obtained from users, by defining how much they trust the people that they know[47].

However, obtaining the trust value based on users' trust relations is not easy[47]. Instead, recommender systems can take advantage of the development of social networking systems. According to several researches [48-50], there is a relation between selected items by a user and selected items by her/his friend. This correlation is useful for item recommendation because friends usually share common tastes and interests and also it is easy to find the trusted users by the given user based on her/his friendship relations. In [48] Ma et al. present a new social recommendation approach combining a rating matrix of user to item with the probabilistic matrix factorization(based on the user's social network) by considering this fact that the user's behaviours on the Web are correlated to her/his social network relations[48]. The experimental results show that social networking information is beneficial when it is combined with the collaborative filtering[40, 41]. Some researchers e.g.: Liu and Yuan in [49] believe that the user's self-defined social networks often help recommender systems to find those users who are most similar to the target user and since measuring similarity is a critical process in CF, it can be concluded that this incorporation can enhance the quality of recommendations. In [49] , a new framework is presented which gathers information from the social network such as friendship relations and gathers information from all the other sources (such as comments that a user assign

to items, favored or disfavored item by a user, etc.) which contain users opinions. In this framework users' opinions and social network information improve the recommendation systems. Groh and Ehmig in [50] compare recommendations from collaborative filtering and social filtering, in a way that in recommendations from CF , nearest neighbours are found based on users' rating activity while in recommendations from social filtering , friends are considered as nearest neighbours . The results of that comparison illustrate that the social filtering approach outperforms the CF approach. Sinha and Swearingen's study [16] shows that by comparing recommendations from friends with generated recommendations via collaborative methods, it could be inferred that friends' recommendations are preferred. In [9], which presents an approach closely related to our algorithm, social rating activities of users are incorporated into the collaborative filtering algorithm. According to Liu and Lee's [9] method, data of users' preference ratings and their social network relations are collected. Then, the nearest neighbours are recognized by the Pearson correlation coefficient similarity metric. Finally, if the social network members are in the list of nearest neighbours, the member's preference is amplified[9]. According to that paper, item recommendation based on social network information and rating history instead of recommendation only based on rating history, improves the recommendation accuracy.

In some research activities a tripartite graph is represented regarding the relations between users, items and tags. For example, Chua et al. in [51] use the effect of social correlation on the user's item adoption prediction. In this paper a social correlation model based on Latent Dirichlet Allocation (LDA) is presented to find the relation between a user-user graph and a user-item adaption graph. Also according to [51] , the social correlation model based on Latent Dirichlet Allocation (LDA) works much better than the pure LDA alone.

Among proposed methods for item recommendations, some were evaluated based on Last.fm datasets. Last.fm is a music recommender web site that incorporates social networking information. In Last.fm people can make friends as well as join their interest groups. Last.fm can predict the most suitable item and recommend it to a user utilizing collaborative filtering and social network information. Ye et al. [52] use Last.fm as an appropriate environment for testing their probabilistic generative model, called the Social Influenced Selection model (SIS), that incorporates user behaviour, social influence and item content in measuring item similarity.

In this regard, although some researchers have applied friendship information, few of them attempted to fuse membership for item recommendation[53, 54] based on the belief that joining groups is a direct indicator of the user's interest comparing friendship activities since making friends can be done for various reasons. Chen et al.[54] used membership information in the Orkut¹⁰ (a social networking site) and compared *Association Rule Mining (ARM)* with *LDA* for community recommendations and came up with the positive effect of *LDA* in community recommendation. According to Spertus et al. [55] membership information is used in the Orkut in order to recommend communities to members. This approach, presents a new collaborative filtering that takes advantage of overlapping membership of pairs in communities. In fact, all the members of a given community get the same recommendation when they visit their community's page.

Vasuki and Natarajan[53] address the affiliation recommendation problem by applying the friendship information and affiliation networks. The experiments were done on the Orkut and YouTube¹¹ web sites. The two proposed models of their approach are *graph proximity* and *latent factors*. These models are used to model users and communities.

¹⁰ <http://www.orkut.com/>.

¹¹ <http://www.youtube.com/>.

Most research activities in this area, believe that opinions of a user's friends on a particular item have a positive effect on the user's decision for selecting that item. Accordingly some researchers believed that combining other data sources with friendship information instead of purely concentrating on friendship can improve the accuracy of recommender systems[56]. Therefore, in [56] both friendship and membership information are used while being combined with traditional CF to predict items more precisely. Moreover, in order to explore the effect of both membership and friendship information, two methods of *random walk graph* with CF and *weighted neighbourhood similarity* are presented [56].The proposed study compares these two methods while those two kinds of social network information are fused on random walk graph with CF and neighbourhood similarity.

Recognizing influential friends is another issue that has some room for improvement. For incorporating social data, our first contribution is recognizing the most influential friends and the second one is utilizing membership information as another source of information. Although we did not use tag information as a semantic source, tags are still considered as the direct indicators for user's opinion on items rather than ratings.

The most relevant studies to our proposed method are [9] and [56]. Liu and Lee's study [9] is based on ratings of users on selected items. In [56] the presented method applied the Last.fm dataset, which contains information of users, the selected items and the tags assigned by users to those items. In this study[56], there is a rating matrix for users and items for inferring rating information from tags. If the user assigns a tag to the selected item, the rating matrix is 1 otherwise the rating is 0.

To the best of our knowledge, most of the papers consider common tags or common items or the combination of these two kinds of information, while our approach focuses on common tags

on common items which we believe can further improve the recommendation accuracy. Moreover, for recommendation algorithms, the rating-based collaborative filtering recommender systems, which are used in other works, use *Weighed Sum of Others' ratings* or *Simple Weighted Average* methods for item recommendation[26]. In tag-based collaborative filtering recommender systems e.g. [57], in the first place, a tag weight is computed for a user or item, and then based on the calculated tag weight, a probability score is calculated to predict items in which the target user might be interested. To sum up, in this paper, we developed a new approach to find nearest neighbours based on combination of CF and social tagging relations to enhance recommendation accuracy. Moreover, we introduce a new recommendation method that applies the item similarity as well as user similarity for suggesting items.

2.5 Summary

From all these reviews we explained the research works related to collaborative filtering systems, tag based collaborative filtering, social based collaborative filtering and social tag-based collaborative filtering recommender systems. We briefly explained about the research works on Last.fm datasets as well. In this regard, we can say that recommendations are not assessed merely by their information value. Rather they are presented within an informal group of users and a social context, which means that there is particular social knowledge behind recommended resources. Although there are some recent research activities discussing the application of social relationships on collaborative filtering systems, more advanced approaches are needed to achieve a better understanding of some activities such as tagging, friend similarities, and cohort similarity in groups.

CHAPTER 3

METHODOLOGY

In this chapter, we present our approach for implementing a recommender system which is based on both social network information and user assigned tags. We mainly address two problems: how to calculate the user similarity in a more accurate way and how to make personalized recommendations. Our user similarity metric not only takes into account the interaction of users with items (in terms of tagging items), but also incorporates the social interactions of users (in terms of friendship and membership). A recommendation for a target user is based on both user and item similarities.

3.1 Preliminaries

In this section, we use a graphic model for the ease of discussion of our proposed social and tag based recommender system. Our model could represent actions performed by users such as tagging items, making friends and joining groups. Tag, friend or group information has been used before in other systems [9, 56]. However, the way we use and combine these different types of information is unique; furthermore, we believe we can improve the accuracy of recommender systems by using such tags and social relations.

Recent studies reveal that the adoption of recommender systems with social networking information as an additional input can enhance the accuracy of recommendations. Social based systems open new possibilities for better understanding of user behaviour based on various types of user relationships. People believe that an item selected by their trusted friends is a reliable indicator when evaluating an item. Thus, utilizing friendship information assists the recommender system to identify which group of items is likely more relevant to a user's taste.

On the other hand, it is generally accepted that people tend to find those communities that share similar interests with themselves. In social network systems, users are generally able to create or join special-purpose groups in order to share their own resources or use those that are generated by other users who have similar interests on a particular subject. Therefore, applying membership information can help us in the process of finding top items, which are more popular among users who are participating in the same community.

Last.fm is among the most popular social network-based services that offer customized songs to users in which users can assign tags to their interested items (e.g., music tracks, artists, albums, etc.), make friends, and joining groups. This music recommender system can be modeled by a graph with three types of nodes and three types of connections represented by edges of the graph. The nodes of this graph represent users, items and groups. The edges of the graph reflect the relationships between nodes. For example, a user selects an item and assigns tags (annotation), a user is a friend of another user (friendship relationship) and a user is a member of a group (membership relationship). In this thesis, we will use the dataset provided by Last.fm to evaluate our proposed algorithms. Our later discussion will mainly focus on music recommender systems; however, the model is generic and can be applied to any recommender system.

Before going into further details, we formalize notations and introduce concepts that will be exploited in the next sections. We present a model for a social network structure using a graph in which its nodes denote users, items, groups, and its edges indicate different kinds of interdependency between users, items and groups. A user is allowed to perform three activities: selecting an item, forming a friendship relation and subscribing to a group/feed. Relationships

among nodes represent friendship between users, selection of an item, and membership of a group. In Figure 3.1, an example of our proposed model is illustrated.

In this example, user u_1 is a friend of u_2 and u_3 and these users select items i_1 , i_2 and i_3 . Moreover, each item node is associated with a tag set which indicates the tags that a user assigned or an item or a group received. In Figure 3.1 each dotted line which connects the user to a particular group specifies that the user is a member of that particular group. Each thick line which connects the user to an item states that this user selects that item.

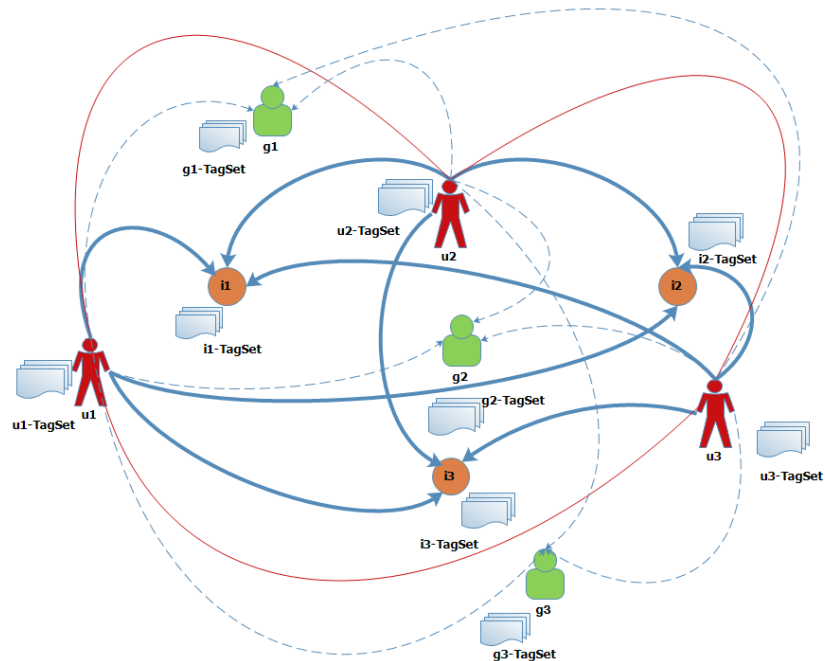


Figure 3-1 A model for social network structure

The aim of utilizing such a model is to simplify the understanding of the interactions between users, items, and groups in the system. For instance, in order to find the target user's interesting items in the future, we not only consider the target user's neighbours' tag sets and the items they have shared with the user but also use those shared items' tag set information.

In the following sections we provide more details on our recommender system. Our proposed approach has two main parts:

1. Measuring the similarity.
2. Making a recommendation.

3.2 Measuring the Similarity

There are three types of the collaborative filtering algorithms: memory based, model based and hybrid model. In the model based algorithm users' ratings on items are collected in order to learn a suitable model. Then by using the constructed model this algorithm predicts ratings on items [22]. Building a proper model is not easy. Compared with model based algorithms, memory based algorithms are easy to implement and are more practical. The memory based algorithm finds the similarity between users or between items by using users' ratings on items. In the meantime, the nearest neighbours of a target user are defined and according to the nearest neighbours, the most interesting items are recommended to the user. In our approach we present a new user similarity approach for finding the nearest neighbours.

The aim of the user similarity calculation is to analyze the relationship between users. If two users have a similar profile, there is a high chance that they will act similarly in the future. Also the more we analyze a user's behaviour by considering all different aspects, the better we understand the user's tastes and the more accurately we can predict the user's preferences [1].

There are various techniques we can use to calculate the similarity between users. However, these metrics usually concentrate only on one or two factors[1]. In order to have a successful measurement of the similarity, a similarity metric must reflect a user understanding of the item space from different perspectives. In this thesis, the similarity calculation consists of three parts:

1. Tag-based similarity
2. Friendship-based similarity
3. Membership-based similarity

3.2.1 Tag-based Similarity

Several methods have been proposed on how to improve an item recommendation in social tagging systems. However, these approaches consider either common tags or common items between users. Measuring the similarity of users only based on common tags is not appropriate, since it is possible that although two users have many common tags, most of those tags are not assigned to any shared items between these two users. Measuring the similarity of users only based on common items might also not be a precise measure, since although these users have many common items; it's possible that they have assigned different tags to those items.

Our first similarity metric is utilizing the tag information and we only focus on common tags which are assigned to common items. In equation (3.1), we define the similarity value between two users u and v .

$$TSim_{u,v} = \frac{\sum_{i \in (I_u \cap I_v)} \left(\frac{|T_{uv_i}|^2}{|T_{u_i}| * |T_{v_i}|} \right)}{\text{Max}(|I_u|, |I_v|)} \quad (3.1)$$

Where T_{u_i} is a set of tags that user u assigned to item i and T_{v_i} is a set of tags that user v assigned to item i , T_{uv_i} is a set of common tags, which both users u and v assigned to item i . I_u indicates the set of all the items to which user u assigned tags and I_v indicates the set of all the items that user v assigned tags to. Item i is in the intersection set of I_u and I_v . It means that item i is a common item between user u and v . In addition, $\text{Max}(|I_u|, |I_v|)$ indicates the maximum of the number of items selected by user u and the number of items selected by user v . In this equation,

$\frac{|T_{uv_i}|^2}{|T_{u_i}| * |T_{v_i}|}$ measures how the opinion of user u and the opinion of user v are similar on item i

(common item between u and v). The greater overlap between tags assigned by users u and v to item i implies a greater similarity for the opinions of users u and v on item i .

The rationale behind using $Max (|I_u|, |I_v|)$ as a denominator is to normalize our similarity metric since in some situations the summation of $\frac{(|T_{uv_i}|)^2}{|T_{u_i}| * |T_{v_i}|}$ may be greater than 1. Suppose both

users u and v have selected n items. If all of these items are the same and also the tag set of each common item is exactly the same, the value of $\sum_{i \in (I_u \cap I_v)} \left(\frac{(|T_{uv_i}|)^2}{|T_{u_i}| * |T_{v_i}|} \right)$ is n . On the other hand, For the

sake of comparison, the similarity of two users should be normalized to produce a value less than or equal to 1. In order to determine an appropriate denominator we analyze four candidates for the denominator:

- The minimum number of items selected by two users
- The number of items selected by one of the users
- The number of common items selected by two users
- The maximum number of items selected by two users

Suppose that we have three users who have assigned tags to items as shown in Table 3.1.

Table 3-1 A sample user-item matrix

	i_1	i_2	i_3	i_4
u_1	t_1, t_3	t_2, t_4		
u_2	t_1, t_3	t_2, t_4		
u_3	t_1, t_3	t_2, t_4	t_5	t_6

In the following section, we measure the similarity of u_1 with u_2 and u_3 based on the four candidates of the denominator.

1. $Min (|I_u|, |I_v|)$: If we divide the numerator by the minimum value then we have:

$$TSim_{u1,u2} = \frac{\frac{2}{2} * \frac{2}{2} + \frac{2}{2} * \frac{2}{2}}{Min(2,2)} = 1$$

$$TSim_{u1,u3} = \frac{\frac{2}{2} * \frac{2}{2} + \frac{2}{2} * \frac{2}{2}}{Min(2,3)} = 1$$

It means that the similarity of u_1 and u_2 is equal to the similarity of u_1 and u_3 . However, it is quite obvious that the former similarity should be higher than the latter because two users u_1 and u_2 assign the exact same tags to two items.

2. $|I_u|$ or $|I_v|$: If denominator is the number of items that user u selected or the number of items that user v selected, $TSim_{u,v}$ is not equal to $TSim_{v,u}$ and the similarity is not symmetric.
3. $(|I_u \cap I_v|)$: If the denominator is the number of common items between the two users then a significant problem may occur. If user u and user v do not share any common items, the denominator is 0.
4. $Max (|I_u|, |I_v|)$: In this case the similarity values are as follows, which are more reasonable values:

$$TSim_{u1,u2} = \frac{\frac{2}{2} * \frac{2}{2} + \frac{2}{2} * \frac{2}{2}}{Max(2,2)} = 1$$

$$TSim_{u1,u3} = \frac{\frac{2}{2} * \frac{2}{2} + \frac{2}{2} * \frac{2}{2}}{Max(2,3)} = \frac{2}{3}$$

This example explained our conclusion to take the maximum number as the best option for the denominator.

3.2.2 Friendship-based Similarity

A social network plays an important role in reflecting the shared interests between entities in a social based system. To deal with the information overload problem in the sense of huge amount of choices in selecting items, a user may trust her/his friends' opinions in order to filter irrelevant information. Recently there has been a growing body of development on social networking websites that provide an abundance of available social data. Many researchers have investigated the integration of the social networking information with the neighbourhood based collaborative filtering to increase the accuracy of recommender systems. Two types of relationships, friendship and membership are the most popular ones in social network websites[56]. Social relationships not only alleviate some limitations of implicit relationships (such as the data sparsity problem [58]), but also can potentially be applied to boost collaborative intelligence in finding key users who really have an impact on decision making of others.

In this section we propose a new approach for combining the implicit and explicit relationships to increase the recommendation effectiveness. It has been proved that the combination of social tagging and friendship can improve the performance of an item recommendation system [59]. A friendship relation has a significant effect on the similarity of users. Even so, we cannot say that the similarity of users should be simply amplified if there is a friendship relation between them. The reason is that two users may be friends in a social tagging system but they may not share any common interest on most of the items. Hence, we need to find out those friends who have been trusted mostly by the user and who have shared the similar interests with the user[9].

Firstly, for a user u in the system we calculate $AvgF_u$ (the average of the tag-based similarities of all the friends of user u) and then we amplify the tag-based similarity if user v is

among those friends of u that are mostly similar to her/him. The amplification is applied if the tag-based similarity is greater than this average value. Based on equation (3.2) the friendship similarity is amplified.

$$FSim_{u,v} = TSim_{u,v} \frac{1}{1+(TSim_{u,v}-AvgF_u)} \quad (3.2)$$

In this equation $TSim_{u,v}$ is the similarity of users u and v based on their tagging activity. $AvgF_u$ is the average of the tag-based similarities of all the friends of user u . Since, the value of $AvgF_u$ may be different than $AvgF_v$, the similarity based on friendship is an asymmetric similarity.

Algorithm 1 presents the pseudo code for computing the similarity based on the friendship information.

Algorithm1. Pseudo code for computing user similarity based on friendship information

```

(1) Float SimFriendShip (User u , User v)
(2) {
(3)     if (v is friend of u && TSimu,v > AvgFu)
(4)     {
(5)          $FSim_{u,v} = TSim_{u,v} \frac{1}{1+(TSim_{u,v}-AvgF_u)}$ ;
(6)     }
(7)     else
(8)         FSimu,v = 0 ;
(9)     return FSimu,v;
(10) }
```

The similarity function in line (1) of the algorithm.1 accepts two users u and v as inputs, generates a friendship-based similarity score for those users who meet particular conditions as specified in line (3). Basically in line (3), we check if users u and v are friends and if $TSim_{u,v}$ is greater than the value $AvgF_u$. If this condition is true then in line (5) based on the amount of the difference between $TSim_{u,v}$ and $AvgF_u$, the amplification will be performed on $TSim_{u,v}$ and this

value will be assigned to $FSim_{u,v}$. Otherwise in line (8), $TSim_{u,v}$ will not be amplified and a “0” will be assigned to $FSim_{u,v}$. According to algorithm 1, the amount of augmentation for $TSim_{u,v}$ is not a constant value. It is based on the amount of difference from the average similarity value.

The augmentation formula in line (5) shows that by increasing the denominator, the $TSim_{u,v}$ (which is less than 1) to the power of a very small number will be changed to a bigger number which is still less than 1(after the normalization). That is, the more difference between $TSim_{u,v}$ and $AvgF_u$, the bigger the amplification value.

3.2.3 Membership-based Similarity

The membership information reflects the behaviour of a user in her/his shared community and users in a group usually share similar interests on some common subjects in the group. As users join a group based on their interests such as genre of music, it is more likely that all the like-minded people join the same group. Our objective is to use this additional information and fuse it into the collaborative filtering algorithm.

Social networks these days are grouping those users who have a common interest into one community. In the social community, users with the same idea or interest interact with each other. Based on our observation, people who are in a same group in a virtual environment will likely have the same interest. However it is not always true because some people may randomly join a group and most of the time they are not attentive to this group’s interest. There should be another factor to measure the level-of-belonging of each user to a group, for example based on the common tags that are used between a user and a group. Hence, if the belonging level of two users to a common group is very high, the probability that the two users are similar to each other increases.

For computing this similarity, the first step is to make a tag set for each group. Each group contains the users who are interested in this group. Besides, each user in the system has a tag set containing all the tags that she/he assigned to her/his selected items. Each group's tag set contains the assigned tags of all the members. We should control the tag frequency in each group's tag set. This means some tags should be removed from the group's tag set. The reason is that there might be some tags which are assigned by only a few users and they cannot reflect the common interests of group members. Thus, these tags should not have any effect on a group's tag set. In order to find the belonging level of a user u to a group g , equation (3.3) is defined.

$$ms(u, g) = \frac{\sum_{ta \in (T_u \cap T_g)} freq(u, ta)}{\sum_{tu \in (T_g)} freq(g, tu)} \quad (3.3)$$

Where ta represents a tag which is in the intersection of tag set of user u (T_u) and tag set of group g (T_g). $freq(u, ta)$ defines the frequency of assigned tag of ta by user u . $\sum freq(u, ta)$ defines the sum of the frequencies of all the tags from user u which are in the intersection of tag set of user u and tag set of group g . tu represents a tag of tu which belongs to tag set of group g . $freq(g, tu)$ defines the frequency of tu occurring in group g . $\sum freq(g, tu)$ defines the sum of the frequencies of all the tags in group g . Finally, in order to calculate the similarity of users u and v based on their membership information, equation (3.4) is shown below,

$$MSim_{u,v} = \frac{\sum_{g_i \in (G_u \cap G_v)} ms(u, g_i) * ms(v, g_i)}{|G_u \cap G_v|} \quad (3.4)$$

Where G_u is a set of groups that user u joined. Similarly, G_v is a set of groups that user v joined. g_i presents a group which is in the intersection of group sets of user u and group sets of user v . It means g_i is a shared group between user u and user v . Also, $G_u \cap G_v$ is a set of all shared groups between users u and v . According to equation (3.4) the similarity of a user and the neighbour user will be calculated based on the membership information.

3.2.4 Overall Similarity

As mentioned before, the ultimate goal of this study is to fuse the social networking information such as friendship and membership into the collaborative filtering algorithm, in order to enhance the accuracy of recommendations. Firstly, $TSim_{u,v}$ is calculated and then the amplifying method is applied in order to take into account the friendship for those who have a strong relationship with a target user. Afterwards, membership information is incorporated for the purpose of taking advantage of the shared interests in a group. In order to compute the overall similarity we have defined two parameters, α and β , to adjust the weight of different factors.

$$Sim_{u,v} = \alpha * TSim_{u,v} + (1 - \alpha) (\beta * FSim_{u,v} + (1 - \beta) * MSim_{u,v}) \quad (3.5)$$

In equation (3.5), the precise value of α and β should be determined empirically. To keep the overall similarity value between 0 and 1 we consider $0 < \alpha, \beta < 1$. In equation (3.5), α is applied to adjust the weight between tag-based similarity ($TSim_{u,v}$) and the social networking information. Then β adjusts the relative weights between these two types of social relationships which are the similarity based on friendship ($FSim_{u,v}$) and the similarity based on membership ($MSim_{u,v}$). In this sense, the bigger α is, the greater the weight of the tagging activity is. In doing so, tagging activity plays a more important role. On the other hand, a bigger β value implies that the friendship-based similarity plays a more important role in the overall similarity. Since each system has different features, for example maybe in a system, membership information is more

reliable than friendship information, by adjusting these two values; we determine which factor plays a more important role in our decision about computing the similarity value. After the computation of $Sim_{u,v}$ for finding neighbours, the next step is to recommend items to users by predicting each item's ratings [56]. In brief, in the above equation the similarity of the neighbour user and the given user is computed.

3.3 Making Recommendation

One of the most important steps in recommendation systems is predicting the future behaviour of a user. At first, a subset of similar users to a target user based on their similarities is calculated and then the weighted aggregation of their ratings is applied to make recommendations for the user[60].

In this step, the system makes recommendations for all of its users. In each system, the type of recommendation can be different depending on the type of users and items. In our system, we have generated a recommendation list for a target user which is sorted based on her/his interests. The first step is making a tag set for each item that will be necessary in finding the recommended items.

As an output, our system provides the target user u with a sorted list of items that she/he will likely select in the future. We predict how much the target user u likes those items which are tagged by her/his nearest neighbours. The user's interest level for a particular item depends on two main components:

1. How similar is the neighbour v (who tagged the item) to the user u .
2. How similar is the item i selected by the neighbour v to the items that are tagged by the user u .

In order to find the similarity of items by considering their tag frequencies , we used the weighted Jaccard similarity method [61] with some modifications. The original Jaccard similarity method does not consider the tag frequency, and therefore we make some modifications. According to equation (3.6), there is a vector v_i for each item i and a vector v_j for item j in which each element of each of these vectors is a pair of (Tag, Frequency) representing the assigned tag to this item and its frequency. Thus, $v_i(t).Fq$ determines the frequency of tag t (a common tag between item i and j) on item i . Similarly, $v_j(t).Fq$ determines the frequency of tag t on item j . $Min(v_i(t).Fq, v_j(t).Fq)$ returns the minimum of two frequency values of tag t in vector v_i and vector v_j . Also, $Max(v_i(t).Fq, v_j(t).Fq)$ returns the maximum of two frequency values of tag t in vector v_i and vector v_j .

$$SimItem_{i,j} = \frac{\sum_{t \in (v_i \cap v_j)} Min(v_i(t).Fq, v_j(t).Fq)}{\sum_{t \in (v_i \cap v_j)} Max(v_i(t).Fq, v_j(t).Fq) + \sum_{ta \in (v_i \cup v_j - v_i \cap v_j)} Max(v_i(ta).Fq, v_j(ta).Fq)} \quad (3.6)$$

In equation 3.6 the sum of the minimum frequencies which is the sum of the minimum of two frequency values of common tags between item i and item j , is the numerator. In the denominator, for all of the common tags between item i and item j we compute the sum of the maximum frequency values of these common tags. Moreover, in denominator of equation 3.6, ta is a tag which is not shared between item i and item j . $Max(v_i(ta).Fq, v_j(ta).Fq)$ returns the maximum of two frequency values of tag ta in vector v_i and vector v_j . Finally, we find the sum of the frequency values of those tags which are not common between items i and item j and add this sum value to the sum of the maximum frequencies to get the denominator. Therefore, based on user and item similarity we predict those items that the target user will probably select in the future.

Algorithm 2 presents the pseudo code for the recommendation algorithm.

Algorithm 2.Pseudo code for making item recommendation

```
(1) For (each user  $u$  in system)
(2) {
(3)   For (each  $v$  as a nearest neighbour of user  $u$ )
(4)     {
(5)       Generate a Nlist for  $v$ ;
(6)       For (each item  $i$  in Nlist)
(7)         {
(8)           For (each item  $j$  in target user's item list)
(9)             {
(10)              ItemInterests [ $i$ ] .Add(SimItem $_{i,j}$  * Sim $_{u,v}$ )
(11)            }
(12)          NeighbourItemAvg.Add( $i$  , ItemInterests [ $i$ ].Avg());
(13)        }
(14)      }
(15)    For (for each  $g_i$  as NeighbourItemAvg.Groupby( $i$ ) in NeighbourItemAvg)
(16)      {
(17)        TopN.Add( $g_i$ .Key,  $g_i$ .Max());
(18)      }
(19)    Sort the TOPN list based on the max values;
(20)    Return items in TopN list;
(21) }
```

According to algorithm 2, in line (1) for each user u in this system and for each of her/his neighbour as v in line (3), we have a *Nlist* in (5) which is a list of items of v which are not tagged by u . In line (6) to line (11) for each item i of user v we compute ItemInterests [i] vector which is a vector of the probability of interest of user u in this item. Each element of this vector is computed by comparing this item with each item j of the user u based on the presented equation in line (10). Thus, the number of elements of vector ItemInterests [i] depends on how many items user u selected. Based on the equation on line (10), ItemInterests [i] is implemented as a list where each element of this list contains the value of multiplied similarities (SimItem $_{i,j}$ * Sim $_{u,v}$) for a particular item i of user v . In line (12) we find the average of prediction value for item i of

user v and save it in `NeighbourItemAvg` which is a list that each elements of this list is a pair of $(i, \text{ItemInterests}[i].\text{Avg}())$. Therefore, in this list we have selected item i by user v and $\text{ItemInterests}[i].\text{Avg}()$ which is the average prediction value of item i for a particular v . In line (15) to (18) we find the maximum of those average values for each group of item i . Then we save the value and item index in TOPN list. In line (19) we sort the TOPN list based on those average values and in line (20) we recommend those items to user u .

In our approach, we use both the item similarity and the user similarity in the item recommendation part. The motivation behind combining the item similarity in the item recommendation is explained in the following paragraph.

1. If we recommend items only by considering the user similarity:

If a nearest neighbour user selects an item, it is not reasonable to recommend this item to the target user only because the nearest neighbour user selected this item. The necessary and sufficient condition for recommending an item to the user could be explained in a way that the neighbour user should select this item and be interested in the selected item.

2. If we recommend items only by considering the item similarity:

If one of the items selected by the nearest neighbour user is very similar to the items of the given user, this item should not be switched to the first position in the TOPN list because this might be wrong.

Thereupon, we should always consider the similarity of the user with her/his nearest neighbour users combined with the item similarity when recommending items.

3.4 A Case Study Illustrating the Recommendation Process

Here we use an example to illustrate our algorithm. In this example, we suppose that there are 3 users, 5 items, 10 tags, and 3 groups. Table 3.2 is a sample of users, items and their assigned tags. Table 3.3 illustrates groups and users who join these groups. Similarly, Table 3.4 shows users and their friends.

Table 3-2 Information of users, items and tags

User	Item	Tag
U ₁	I ₁	{t ₁ , t ₂ , t ₃ }
U ₁	I ₂	{t ₄ , t ₈ }
U ₂	I ₁	{t ₁ , t ₂ , t ₃ , t ₁₀ }
U ₂	I ₄	{t ₂ , t ₃ , t ₆ }
U ₃	I ₂	{t ₄ , t ₅ , t ₇ }
U ₃	I ₃	{t ₁ , t ₅ , t ₇ , t ₉ }
U ₃	I ₄	{t ₃ , t ₆ , t ₉ }
U ₃	I ₅	{t ₂ , t ₄ , t ₈ }

Table 3-3 Membership information

Group	User
G ₁	U ₁ , U ₂ , U ₃
G ₂	U ₁ , U ₂
G ₃	U ₂ , U ₃

Table 3-4 Friendship Relations

User	Friends
U ₁	U ₂ , U ₃
U ₂	U ₁
U ₃	U ₁

Using our approach, the similarity of users will be calculated based on equation (3.1) which computes the similarity based on tagging activity. Table 3.5 represents a matrix of the user similarity for these three users. The similarity of each user with her/himself is not calculated.

Table 3-5 $TSim_{u,v}$ matrix

	U ₁	U ₂	U ₃
U ₁	-	0.375	0.042
U ₂	0.375	-	0.111
U ₃	0.042	0.111	-

After calculating the user similarity matrix based on the tagging activity, the next step is to calculate the $AvgF_u$ value for each user u . As Table 3.6 shows, $AvgF_u$ is calculated for each user.

Table 3-6 Average of each user's friends' similarities

User	AvgF
U ₁	0.208
U ₂	0.375
U ₃	0.042

As a result, the similarity matrix based on the tagging activity changes to the values as shown in Table 3.7 based on the friendship relationship using Algorithm 1. In Table 3.7 for each user u , the similarity value of two users u and v is amplified if u and v are friends and the tag-based similarity of u and v is greater than the $AvgF_u$. The friendship-based similarity is 0 for those who do not meet the required conditions.

Table 3-7 Friendship similarity matrix

	U_1	U_2	U_3
U_1	-	0.431	0
U_2	0	-	0
U_3	0	0	-

After calculating the similarity based on the friendship, the next step is calculating the similarity based on the membership. The first step is making a tag set for each group containing all the members' tags and their tag frequencies, which are presented below:

$$G_1 = \{t_1 (3), t_2 (4), t_3 (4), t_4 (3), t_5 (2), t_6 (2), t_7 (2), t_8 (2), t_9 (2), t_{10} (1)\}$$

$$G_2 = \{t_1 (2), t_2 (3), t_3 (3), t_4 (1), t_6 (1), t_8 (1)\}$$

$$G_3 = \{t_1 (2), t_2 (3), t_3 (3), t_4 (2), t_5 (2), t_6 (2), t_7 (2), t_8 (1), t_9 (2), t_{10} (1)\}$$

In this example, we consider all of the tags in each group for controlling the tag frequency in each group's tag set. The matrix of the membership similarity is presented in Table 3.8.

Table 3-8 Membership similarity matrix

	U_1	U_2	U_3
U_1	-	0.117	0.104
U_2	0.117	-	0.199
U_3	0.104	0.199	-

The next step is to calculate the overall similarity based on equation (3.4). In this step the precise value of α and β should be determined which are 0.6 and 0.7 respectively. Thus the matrix of the overall similarity is:

Table 3-9 Overall similarity matrix

	U_1	U_2	U_3
U_1	-	0.360	0.037
U_2	0.239	-	0.090
U_3	0.037	0.090	-

Afterwards we make the item similarity matrix in order to have the similarities of items. In order to find the item similarity based on equation (3.5) we should provide each item with its assigned tags and their frequencies.

The tag set of each item is represented below:

$$I_1 = \{t1 (2), t2 (2), t3 (2), t10 (1)\}$$

$$I_2 = \{t4 (2), t5 (1), t7 (1), t8 (1)\}$$

$$I_3 = \{t1 (1), t5 (1), t7 (1), t9 (1)\}$$

$$I_4 = \{t2 (1), t3 (1), t6 (1), t9 (1)\}$$

$$I_5 = \{t2 (1), t4 (1), t8 (1)\}$$

Then we can compute the item similarity matrix as shown in Table 3.10 based on equation (3.5).

Table 3-10 Item similarity matrix

	I₁	I₂	I₃	I₄	I₅
I₁	-	0	0.1	0.3	0.111
I₂	0	-	0.286	0	0.5
I₃	0.1	0.286	-	0.111	0
I₄	0.3	0	0.111	-	0.125
I₅	0.111	0.5	0	0.125	-

Suppose that in this example the given user is U_1 . This user has selected $\{I_1, I_2\}$ and U_2, U_3 are the nearest neighbours. According to algorithm 2, for each of these neighbours there is an Nlist which is a list of the items that U_1 has not selected yet. For example, the Nlist for the first neighbour U_2 is: $\{I_4\}$. We should find ItemInterests $[I_4]$ that is $[\text{SimItem}_{I_4, I_1} * \text{SimOverall}_{u_1, u_2}, \text{SimItem}_{I_4, I_2} * \text{SimOverall}_{u_1, u_2}]$. Therefore, $\text{ItemInterests } [I_4] = [0.3 * 0.360, 0 * 0.360]$. After comparing all items of U_1 with each item i of each neighbour v , NeighbourItemAvg could be generated. As a result, the average of values of ItemInterests $[I_4]$ is 0.054 that we save it in NeighbourItemAvg list for I_4 and U_2 . In doing so, the NeighbourItemAvg list for this example is presented in Table 3.11.

Table 3-11 NeighbourItemAvg list

I₄	0.054
I₃	0.007
I₄	0.005
I₅	0.011

In this step for each group of item i in NeighbourItemAvg list, we find the maximum average for this item. For example, for group of I_4 the maximum value is 0.054. We keep all the items with their maximum average value in TOPN list.

Hence, the TOPN list contains $[(I_4, 0.054), (I_3, 0.007), (I_5, 0.011)]$. Finally, the TOPN list should be sorted by the average values. Based on sorted TOPN: $[(I_4, 0.054), (I_5, 0.011), (I_3, 0.007)]$, the recommended items are I_4, I_5 and I_3 .

3.5 Summary

In this chapter, we have explained the design of our recommender system, the algorithms that are used in finding similar users, as well as the recommendation algorithm itself. The user similarity consists of three parts. The first part is based on common tags on common items, the second one is based on the friendship, and the third one is based on the membership. By combining collaborative filtering with social networks our algorithm becomes more effective when recommending items. In the next chapter, we will do empirical evaluation on our proposed algorithms.

CHAPTER 4

EXPERIMENTS

4.1 Introduction

In order to measure the performance of our proposed recommender system, some experiments were conducted with different parameter settings. In all of the experiments the dataset was divided into two parts: a training set and a testing set. Each of these datasets was generated by randomly selecting a number of users' tagging history from the original dataset. The main purpose of creating the training dataset is to learn the user behaviour for predicting the future behaviour of users. Then by using the testing set, the accuracy of the algorithm can be measured through the comparison between the recommended items with the items users actually selected in the testing set. Thus, it is obvious that the less the difference between the recommended datasets and the already tagged dataset (test dataset), the more accurate the proposed algorithm is.

In this chapter, our proposed algorithm is compared with a most similar recommendation algorithm which we call the Augmenting algorithm. In other words, we implemented a recommender system that is one of the most closely related to our own approach, in order to compare the performance of our proposed algorithm with this previously proposed algorithm.

The augmenting algorithm [56] based on weighted neighbourhood similarity used friendship and membership information and combined them with the collaborative filtering algorithm. This algorithm utilized the extracted information from the Last.fm web site. In [56] the weighted neighbourhood similarity method was compared with another fusing method which is based on the random walk graph, and the results showed that the random walk graph method

attained 8% improvement compared with the augmenting algorithm. In this thesis, we also compare our method with the augmenting algorithm. Later we will show that our algorithm generates a higher improvement over the augmenting algorithm than the random walk graph method. Another advantage of our approach is that it is vector based and thus it is more efficient than the graph based approach.

Because we compare our algorithm with the weighted neighbourhood similarity method in augmenting algorithm, in this whole thesis we use the word ‘‘Augmenting’’ algorithm instead of ‘‘weighted neighbourhood similarity’’. The augmenting method finds the similar users based on the cosine similarity method. Similar to our approach, two other kinds of user similarity measures regarding users’ social relations (friendship and membership) are computed individually using cosine similarity method. Finally, based on equation (4.1) the overall user similarity is computed.

$$Sim_{ui+fri+mem}(u_a, u_b) = \lambda Sim_{ui}(u_a, u_b) + (1 - \lambda)(\beta Sim_{mem}(u_a, u_b) + (1 - \beta) Sim_{fri}(u_a, u_b)) \quad (4.1)[56]$$

Where $Sim_{ui}(u_a, u_b)$ is the similarity of two users u_a and u_b based on their tagging activities, $Sim_{mem}(u_a, u_b)$ is the similarity of users based on their membership information and $Sim_{fri}(u_a, u_b)$ is the similarity of users based on their friendship. After the computation of similar users for finding nearest neighbours, the next step is recommending items to users. For item recommendation the augmenting algorithm[56] used equation 4.2.

$$r_{i,m} = \frac{\sum_{j=1}^N Sim(u_i, u_j) * r_{j,m}}{\sum_{j=1}^N Sim(u_i, u_j)} \quad (4.2)[56]$$

Where $r_{i,m}$ represents the rating that user u_i assigns on the item m (if the user assigns a tag to an item the rating is 1 otherwise rating is 0). Similarly, $r_{j,m}$ presents the rating that user u_j assigns on the item m .

In order to compare the effectiveness of our algorithm and the augmenting algorithm, both algorithms should be tested on the same dataset. To do the testing, a proper dataset should be selected and also an appropriate subset from that dataset should be extracted in order to generate a dataset that shows all aspects of user behaviour.

4.2 Evaluation Metrics

In order to measure the error rates in this tag based recommender system, we introduce two types of evaluation metrics which are the most commonly used metrics for the prediction accuracy. We use the precision/recall metrics to measure the performance of item recommendations. These two classic metrics are also used for measuring the quality of information retrieval tasks in general[23].

4.2.1 Precision

Precision is one of the classic metrics which is used to classify the n most appropriate items for a certain user. Both precision and recall use a fraction of the number of hits which is the number of correctly recommended relevant items[23]. Equation (4.3) shows the precision formula.

$$P_u = \frac{|hits_u|}{|recSet_u|} \quad (4.3) [23]$$

Where $|hits_u|$ is the number of correctly recommended items to user u and $|recSet_u|$ is the total number of recommended items to user u [23]. As a result, as the total number of

recommended items increases, the precision value decreases[62]. In doing so, the high precision value illustrates that more relevant than irrelevant outcomes are returned.

4.2.2 Recall

Recall is another classic metric which is applied for measuring error rates in collaborative recommender systems[23]. Equation (4.4) shows the recall formula.

$$R_u = \frac{|hits_u|}{|testSet_u|} \quad (4.4) [23]$$

Where $|hits_u|$ is the number of correctly recommended items to user u and $|testSet_u|$ is the number of items in the test set of user u . This number represents the total number of items that should be recommended to u . Therefore, high recall value explains that most of the relevant outcomes are returned.

4.2.3 Datasets

One of the most important steps in evaluating these algorithms is choosing a proper dataset. Since our method applies some new factors, we selected a dataset with two features. The first feature is that the dataset covers all of the possible situations (various factors) of the proposed approach. The second feature is that our dataset should be practical for other algorithms in order to evaluate the improvement of our proposed algorithm. Furthermore, we were looking for a dataset that has all the features such as user selected items and their assigned tags, friendship and membership information. In order to cover all these requirements we chose the Last.fm dataset which is a popular and standard dataset for music recommender systems. Last.fm is a music recommender web site that makes a profile for each user based on her/his previously listened-to songs and incorporates social networking information. In the Last.fm people can make friends as

well as join their insert groups. Last.fm can predict the most suitable item and recommend it to a user utilizing the collaborative filtering and social network information.

There are many algorithms that are tested using this dataset. Among all of Last.fm's datasets, we selected the one that provides us with the required information. Table 4.1 presents a comprehensive description of the attributes of our dataset.

Table 4-1 Features of the Last.fm dataset

Item	#Instances
crawled (active) users	99,405
Annotations	10,936,545
Items	1,393,559
Tags	281,818
Groups	66,429
Friends	1,048,576

According to Table 4.1, 99,405 users were crawled in 2009. Below we will explain every feature covered in this dataset.

- **Annotation:** this file has the information of users and their selected items and related tags. In this file a list of annotations is in the form of triples <user_id, item_id, tag_id>. Table 4.2 represents a sample of that file.

Table 4-2 Sample of annotation file

3	523	6
3	523	274202
6041	33152	360
6041	33164	360
15512	1309593	119

According to the first row of this table user 3 selected item 523 and assigned tag 6.

- Items: In this file a list of items is in the form of pairs <item_id, item_description>. Table 4.3 shows a sample of item file.

Table 4-3 Sample of item file

18	black 9
19	dark side
20	murder squad
21	mr. x
22	lil chill

The first row of Table 4.3 means that item 18 is black 9.

- Tags: In this file a list of tags is in the form of pairs <tag_id, tag_description>. Table 4.4 demonstrates tag file information.

Table 4-4 Sample of tag file

1	celtic
2	dark metal
3	gothic
4	gothic metal
5	liv kristin

The first row of Table 4.4 shows that tag_id 1 means celtic.

- Groups: In Group file we have a list of groups with users' membership information. Each line is a pair of <group-id, user_id>. Table 4.5 shows a sample of group file.

Table 4-5 Sample of group file

1	7382
1	21190
1	24274
1	31089
1	40425

The first row of Table 4.5 shows that user_id 7382 is a member of group-id 1.

- Friends: It contains friendship information of Last.fm. Each line is a pair of <user_id,user_id> which means these two users are friends. Table 4.6 shows a sample of friend information.

Table 4-6 Sample of friend file

1	2
1	3
1	4
1	5
1	6

According to Table 4.6 user_id 1 is a friend of user_id 2 and vice versa.

4.3 Implementation

In order to implement and test our recommendation algorithm, the C# programming language and the LINQ technique were used. The configuration of the system we used to run the code is shown below:

CPU: Intel Xeon

RAM: 16 Gigabyte

Operating System: Windows 7

Software: Visual Studio 2010 with C# 4.0 and LINQ

LINQ provides an easy way to querying on any source of data such as arrays, collections of objects, database or XML files. It also presents some powerful rules that help provide some queries that support joining, aggregation, sorting, filtering functions, etc. These rules are called language-level rules and there is no need to compile them to get the result. Figure 4.1 shows all the classes which are used in our implementation.

4.4 Preprocessing

4.4.1 Finding the Tag Set of Each Group

For this step 10% of our dataset is selected. In our implementation, in order to find the tag frequency in each group's tag set, we only consider tags which occur more frequently. The reason for removing those tags is that there might be some tags that are assigned to a group only a few times, and these tags cannot represent the common interests of group members. We consider them as outliers. For each user there is a tag frequency list which contains objects in pairs of Tag-ID and its frequency. Similarly, there is a tag frequency list for each group which is generated from all of the tag frequency lists of users who are members of this group. We sort the tag frequency list of each group. Then, the top 50% tags of this sorted tag frequency list are considered in our further calculation.

4.4.2 Finding α and β

In order to achieve the best performance of our approach some preprocessing in terms of initialization of some variables is needed. According to equation (3.4) a suitable scale for both α and β is 0-1. Figure 4.2 illustrates the possible values of the combination of α and β in a square with the length of 1. In other words, several combinations of α and β will fit in this square. In Figure 4.2 there are some dots which are shown as a sample. These are some of the possible areas of combinations of α and β .

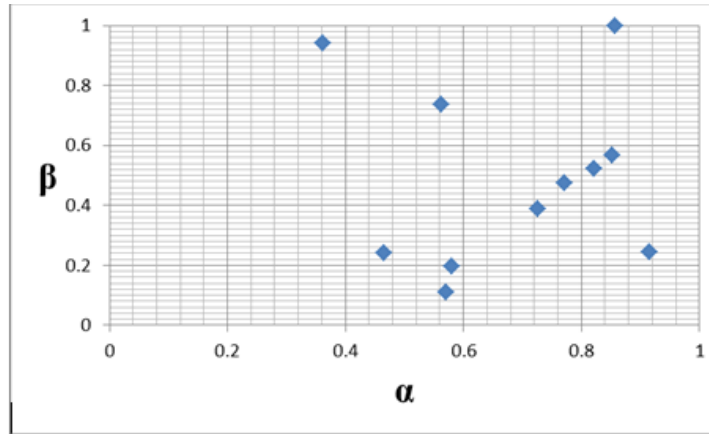


Figure 4-2 A possible area of variation of α and β

In the direction of discovering the most appropriate value of α and β , we vary these values in an increment of 0.1 to find the best combinations of α and β , which is the combination that has the highest precision value. The highest precision value indicates that most of the relevant items are returned to the user which shows the effectiveness of the proposed approach. In Table 4.7, we examined the precision values based on equation (4.3) while returning the Top 20 recommendations. For this step, again, 10% of the dataset was selected. Among the selected data, 80% of the dataset was used as the training set and 20% was testing dataset. With 80% of that dataset (training set) we predict the probability of recommended items and with the 20% remaining (test set) we evaluate our approach to check if the recommended item is the item that the target user selected in test set. Table 4.7 shows the precision values for different α and β combinations while our algorithm returns 20 items to the certain user.

Table 4-7 Precision value in the Top 20

$\beta \backslash \alpha$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.1	18.03	18.28	19.47	20.72	20.59	19.32	18	19.62	19.16
0.2	19.73	20.51	19.02	19.24	20.1	20.09	19.92	20.03	20.89
0.3	20.22	19.42	20.82	21.12	19.01	21.1	20.92	19.22	20.45
0.4	19.35	20.69	21.24	20.92	21.06	21.13	21.34	20.65	19.19
0.5	21.64	18.82	19.99	21.27	21.08	22.06	19.51	19.96	20.31
0.6	21.34	19.51	18.97	21.47	21.31	22.15	20.28	21.97	18.49
0.7	19.22	19.76	20.11	21.09	22.21	22.56	22.19	19.3	19.83
0.8	19.89	20.44	18.53	18.7	21.18	21.54	22.38	20.01	19.87
0.9	18.4	19.28	19.26	19.08	21.84	21.04	20.54	21.75	18.35

Table 4.7 shows the performance of our algorithm reaches its peak when α is 0.6 and β is 0.7. This means that the tagging activity contributes 60%, the friendship and membership relations contribute 28% and 12% respectively in the overall similarity calculation. We have a similar table for Recall (Table 4-8) with similar results for α and β . So these will be the final values we use for the later experiment.

Table 4-8 Recall value in the Top 20

$\beta \backslash \alpha$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.1	32.74	33.21	33.6	32.21	35.36	34.23	35.6	33.83	33.63
0.2	32.38	33.69	32.08	33.98	33.75	34.45	34.07	33.28	33.65
0.3	33.52	32.85	33.11	32.39	33.86	33.92	35.52	34.16	32.05
0.4	32.36	34	32.95	32.81	34.48	34.42	33.61	33.92	32.34
0.5	32.31	32.34	32	33.01	34.23	34.36	35.04	33.37	33.12
0.6	33.2	33.09	33.05	32.43	37.48	36.45	37.1	34.16	34.8
0.7	33.61	32.34	33.67	33.06	35.37	37.85	36.03	33.95	32.04
0.8	33.39	32.16	32.53	32.57	36.29	35.69	35.58	33.2	32.85
0.9	32.78	32.51	33.91	32.05	35.3	37.39	37.11	33.62	33.75

4.5 Error Evaluation based on 20-80 Method

In the 20-80 testing method, 80% of the dataset is selected randomly as the training set and the remaining 20% of the dataset is selected as the testing set. The recommender system recommends a list of ordered items to the test user which has not been selected by this user before. According to the training set information we predict the interest probability of users to those items which are not selected by users. In the test set we have the information of users and their selected items. Thus, in the evaluation part we check for each user if the recommended items based on the training set are the same as the items selected by the user in the test set or not. In other words, based on precision and recall metrics we evaluate the accuracy of our algorithm.

In details, based on the prediction scores which are sorted in a descending order, the ordered Nlist which we propose in the recommendation algorithm will be recommended to the user. If the test user has already selected or tagged the recommended item which is in the Nlist, then the

item would be counted as a hit. Therefore for each group of test user's dataset we compute the average precision and recall. These numbers are used to measure the accuracy of the recommendation algorithm.

As we explained before, we implement another algorithm which combines collaborative filtering and friendship and membership information (augmenting algorithm). Our goal is to show that our recommendation algorithm improves the accuracy of item recommendation from the augmenting algorithm.

We implemented the augmenting algorithm and based on our implementation on the whole dataset that we have, the suitable value of λ and β are found similar to the calculation for computing α and β in our recommended approach. Thus, λ and β are also set as 0.6 and 0.4 respectively. These two algorithms are evaluated and compared using the 80-20 method. Table 4.9 shows the result of evaluation values on precision for our algorithm. Firstly, this value was calculated when the similarity was based purely on the tagging activity (Sim_{TUI}) without considering the social networking information. Secondly, the precision value was calculated when the similarity was based on the combination of the tagging activity with the friendship information ($Sim_{TUI+fri}$). Thirdly, precision was calculated when the similarity was the combination of the tagging activity with the membership information ($Sim_{TUI+mem}$). Finally, the last precision value was calculated when the similarity was the combination of these three types of information ($Sim_{TUI+fri+mem}$).

The results of precision values from our algorithm are presented in Table 4.9. In our algorithm when the number of recommended items increases, it is more possible that the recommended items are the desired items of users. The precision of recommending Top1 item when combining the social information may not be improved compared to the case when we

only consider the tagging information because there is a certain level of randomness in the results. However, when we consider more recommended items, normally combining the social relation could improve the recommendation accuracy in terms of the precision value. Sometimes the friendship information may perform better, and sometimes the membership information may perform better. Overall, when we recommend 10 or 20 items to users, combining all three types of information gives us the best results. We could see similar results in the recall value as shown in Table 4.10.

Table 4-9 Precision when using our algorithm

Precision	Top 1	Top 2	Top 5	Top 10	Top 20
<i>Sim_{TUI}</i>	30.15	35.22	26.27	24.27	18.85
<i>Sim_{TUI+fri}</i>	28.48	38.02	27.68	23.78	19.51
<i>Sim_{TUI+mem}</i>	27.17	33.38	31.1	24.41	19.57
<i>Sim_{TUI+fri+mem}</i>	29.49	32.76	29.32	30.95	21.92

Table 4-10 Recall when using our algorithm

Recall	Top 1	Top 2	Top 5	Top 10	Top 20
<i>Sim_{TUI}</i>	6.68	11.03	21.63	34.19	31.61
<i>Sim_{TUI+fri}</i>	6.63	12.66	24.52	37.27	31.12
<i>Sim_{TUI+mem}</i>	5.94	12.45	25.81	35.25	31.86
<i>Sim_{TUI+fri+mem}</i>	5.78	12.40	24.91	38.43	35.17

From the results shown in Table 4.11 for augmenting algorithm, when we add the social information such as friendship or membership, the precision value could always be improved compared to the case when we only consider the tag information. Friendship information normally could provide a better result than the membership information. Combining three of them achieves the best result for recommending Top10 items.

The results of recalls for the augmenting algorithm are shown in Table 4.12. We can get similar conclusion on the recall value. Combining the tagging activity of the user with the social information can also improve the performance on recall values.

Table 4-11 Precision when using the Augmenting algorithm

Precision	Top 1	Top 2	Top 5	Top 10	Top 20
<i>Sim_{UI}</i>	29.47	27.21	23.2	17.23	10.45
<i>Sim_{UI+fri}</i>	29.86	30.3	28.94	18.21	12.45
<i>Sim_{UI+mem}</i>	29.85	36.83	28.84	18.29	12.35
<i>Sim_{UI+fri+mem}</i>	29.63	33.78	28.75	20.75	11.3

Table 4-12 Recall when using the Augmenting algorithm

Recall	Top 1	Top 2	Top 5	Top 10	Top 20
<i>Sim_{UI}</i>	5.95	11.45	20.9	28.95	26.15
<i>Sim_{UI+fri}</i>	6.84	13.10	24.69	30.61	25.83
<i>Sim_{UI+mem}</i>	5.32	11.37	24.56	29.47	23.16
<i>Sim_{UI+fri+mem}</i>	6.49	11.97	24.58	30.7	24.39

If we compare the last rows of Table 4.9 and Table 4.11, or Table 4.10 and Table 4.12, it can be seen that our algorithm is more accurate than the Augmenting algorithm while returning 5, 10 and 20 items.

Figure 4.3 and Figure 4.4 show the improvement of our algorithm compared with the augmenting algorithm while returning items from Top 1 to Top20. According to Figure 4.3, our algorithm can achieve an improvement of 10.62% in Top20 precision values. Moreover, based on Figure 4.4 our algorithm attains an enhancement of 10.78% in the Top20 recall values.

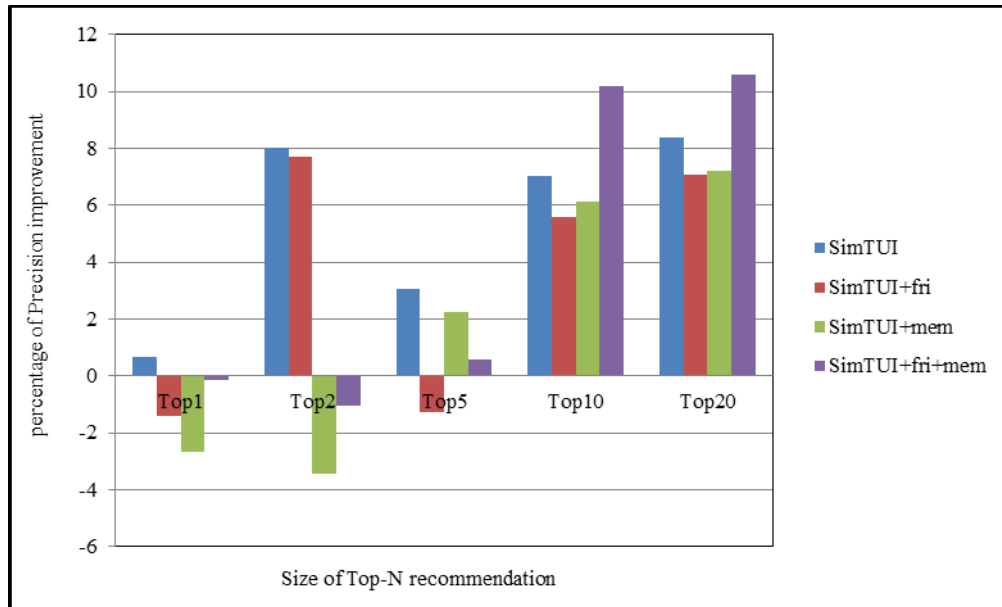


Figure 4-3 Improvements on Precision on all of the fusion approaches

According to Figure 4.3 in some situations when combining friendship and membership the improvements are not tangible. The reason is that the chance of recommended items to be the interesting items of the user is extremely low when the number of recommended items is limited. However, combining the tagging activity with the friendship and membership information in Top10 and Top20 causes an improvement in our results. According to Top10 and Top20 values, finding the similar users based on their tagging activity and their friendship relations improves the recommendation results. Also the third bar in Top10 and Top20 shows that in our algorithm, purely membership information combined with the tagging activity of users, enhances the recommendation results. Subsequently, combining the tagging activity with the friendship and membership information in Top10 and Top20 could bring an even higher improvement in our results.

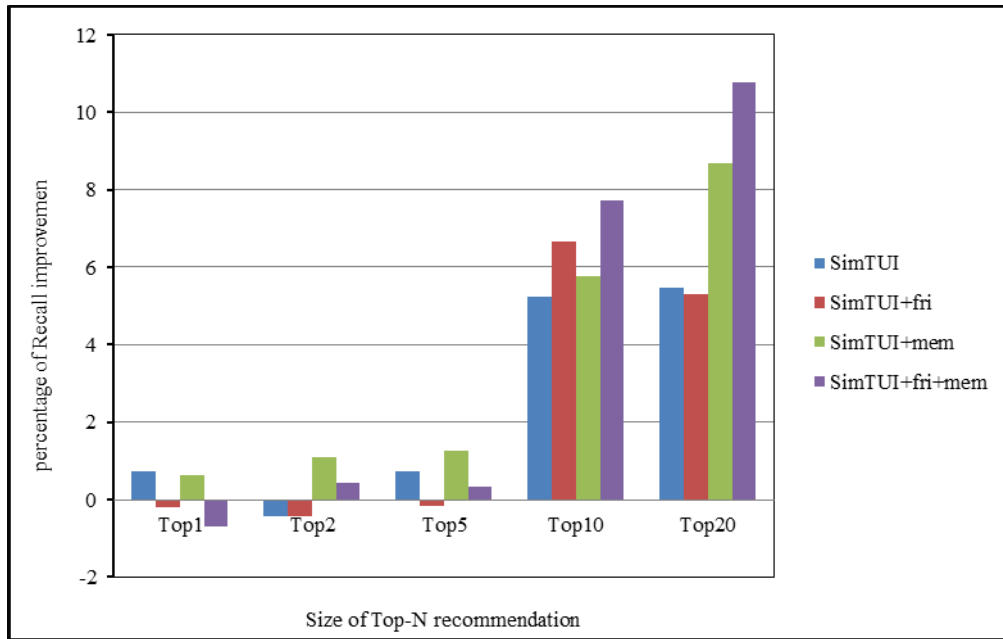


Figure 4-4 Improvements on Recall on all of the fusion approaches

In Figure 4.4 using the purely membership information combined with the tagging activity was beneficial from Top1 to Top20 and we had significant improvements. In addition, the friendship and membership information combined with the tagging activity was helpful and worthy in Top10 and Top20 and cause better results on recall.

Overall based on Figure 4.3 and Figure 4.4, we have achieved a significant improvement on Top10 and Top20 precision and recall values. Thus, it shows that combining social relation information with the tagging activity enhances the performance of our algorithm.

4.6 Summary

In this chapter, we explained our dataset and presented some samples of different files in our dataset. We explained the implementation process and our experiments. Our experiments results prove that the proposed algorithm positively affects the precision and recall. We could see that the combination of the user's tagging activity with friendship and membership information could

enhance the accuracy of the tag based recommender systems. Based on our analyses and evaluations, it could be observed that users get better recommendations when the number of recommended items is increased.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

5.1 Conclusion

Social tagging systems provide recommendations to users based on what tags other users have assigned to items. In this study, we developed a similarity metric, based on social tagging information, to model three types of relationships: tagging similarity, friendship, and membership. Moreover, we have proposed a new recommendation method, which applies user similarity for finding the most relevant items to a target user's taste, and also takes item similarity into consideration for sorting recommended items.

The main focus of this thesis is enhancing the recommendation accuracy while integrating the CF method with the social networking information. Our approach use both implicit and explicit relations to improve the accuracy of recommender systems. The implicit relations are concluded from the user based CF method utilizing user behaviour and the tagging activity with considering not only users' shared items but also users' shared tags. In other words, our implicit information is inferred from shared tags on shared items. Explicit information is gathered from the users' social relations including users' friendship and membership information.

In this study we tried to separate the influential friends of the target user (those friends whose tastes are more similar to the target user's taste) from the non-influential friends of the target user (those users whose tastes are not similar to the target users' taste). Also, the membership information was useful in finding the similarity of users based on the two factors: their shared groups and their belonging level to those shared groups. Consequently, the nearest neighbours of the target user were found by combining implicit relations (similarity of users

based on common tags on common items) with explicit relations (similarity of users based on their friendship and membership relations). Furthermore, to recommend items, we considered items similarity as well as user similarity scores. To the best of our knowledge, the work is one of the first efforts which combine the similarity of users based on their shared tags on shared items with their similarity based on friendship and membership information, and also recommends the items by considering the user similarity and the item similarity. Our experimental results show that our proposed approach is effective.

5.2 Future Work

As a further line of research, it would be extremely interesting to study the use of the semantic information of those tags. Thus, we may extend our approach to a novel semantic-based method with a hybrid approach which applies combination of CF and the content-based filtering to check if it could further improve the performance. It means that we need to analyze the semantic meaning and context of social tags to find the similar users or similar items[63]. Another interesting direction is to apply the inverse user frequency (IUF) concept which assumes that generally liked items are less important in similarity computing than the less common items.

Although recommender systems provide impressive solutions for recommending preferred resources to users, these techniques fail to evaluate the fluctuating behaviour of users[1, 57]. Another direction we would like to consider is to extend our approach to integrate the time dimension as a measure to assess the importance of an item-tag pair. Moreover, we can push forward the use of the friendship relations by considering the transitive relationship (friend of friend) between users. Therefore, a new graph can be created which defines more broad relationships between users.

REFERENCES

- [1] R. Rafeh and A. Bahrehmand, "An adaptive approach to dealing with unstable behaviour of users in collaborative filtering systems," *Journal of Information Science*, vol. 38, pp. 205-221, 2012.
- [2] J. Jacoby, D. E. Speller, and C. A. Kohn, "Brand choice behavior as a function of information load," *Journal of Marketing Research*, pp. 63-69, 1974.
- [3] K. Bischoff, C. S. Firan, W. Nejdl, and R. Paiu, "Bridging the gap between tagging and querying vocabularies: Analyses and applications for enhancing multimedia IR," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 8, pp. 97-109, 2010.
- [4] O. Celma, "Foafing the music: Bridging the semantic gap in music recommendation," in *Proceedings of the 5th International Semantic Web Conference (ISWC)*, pp. 927-934, 2006.
- [5] R. Pan, G. Xu, and P. Dolog, "Improving recommendations in tag-based systems with spectral clustering of tag neighbors," in *Proceedings of the 3rd International Conference on Computer Science and Its Applications CSA 2011 and 2011 FTRA World Convergence Conference*, pp. 355-364, 2012.
- [6] J. Illig, A. Hotho, R. Jäschke, and G. Stumme, "A comparison of content-based tag recommendations in folksonomy systems," in *Proceedings of the 1st International Conference on Knowledge Processing in Practice, KPP*, pp. 136-149, 2011.
- [7] A. Dattolo, F. Ferrara, and C. Tasso, "On social semantic relations for recommending tags and resources using folksonomies," *Human-Computer Systems Interaction: Backgrounds and Applications 2*, vol. 98, pp. 311-326, 2012.
- [8] X. Luo, Y. Ouyang, and Z. Xiong, "Improving neighborhood based Collaborative Filtering via integrated folksonomy information," *Pattern Recognition Letters*, vol. 33, pp. 263-270, 2011.
- [9] F. Liu and H. J. Lee, "Use of social network information to enhance collaborative filtering performance," *Expert Systems with Applications*, vol. 37, pp. 4772-4778, 2010.
- [10] A. Bahrehmand and R. Rafeh, "Proposing a New Metric for Collaborative Filtering Open Access," *Journal of Software Engineering and Applications*, vol. 4, pp. 411-416, 2011.
- [11] A. Agresti and L. Winner, "Evaluating agreement and disagreement among movie reviewers," *Chance*, vol. 10, pp. 10-14, 1997.
- [12] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: an open architecture for collaborative filtering of netnews," in *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pp. 175-186, 1994.
- [13] U. Shardanand and P. Maes, "Social information filtering: algorithms for automating "word of mouth"," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 210-217, 1995.
- [14] P. Resnick and H. R. Varian, "Recommender systems," *Communications of the ACM*, vol. 40, pp. 56-58, 1997.
- [15] A. Ansari, S. Essegaiar, and R. Kohli, "Internet recommendation systems," *Journal of Marketing research*, vol. 37, pp. 363-375, 2000.

- [16] R. Sinha and K. Swearingen, "Comparing recommendations made by online systems and friends," in *Proceedings of the Delos-NSF workshop on personalization and recommender systems in digital libraries*, 2001.
- [17] Y. H. Cho, J. K. Kim, and S. H. Kim, "A personalized recommender system based on web usage mining and decision tree induction," *Expert Systems with Applications*, vol. 23, pp. 329-342, 2002.
- [18] C. Y. Kim, Y. H. Cho, D. H. Kim, and J. K. Lee, "A Multimedia Contents Recommender System for Customized Mobile Web Service," *Proceedings of The Korea Society of Management Information Systems*, pp. 808-814, 2003.
- [19] J. K. Kim, H. K. Kim, and Y. H. Cho, "A user-oriented contents recommendation system in peer-to-peer architecture," *Expert Systems with Applications*, vol. 34, pp. 300-312, 2008.
- [20] L.-H. Li, F.-M. Lee, and S.-C. Chan, "The blog-article recommendation system (BARS)," in *Proceedings of the International MultiConference of Engineers and Computer Scientists IMECS*, pp. 771-776, 2008.
- [21] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems (TOIS)*, vol. 22, pp. 5-53, 2004.
- [22] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 17, pp. 734-749, 2005.
- [23] M. Z. Dietmar Jannach , Alexander Felferning, Gerhard Friedrich, *Recommender Systems : an introduction*. New York: Cambridge University Press, 2011.
- [24] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Application of dimensionality reduction in recommender system-a case study," in *In Proceedings of the ACM WebKDD 2000 Web Mining for E Commerce Workshop*, 2000.
- [25] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, "Eigentaste: A constant time collaborative filtering algorithm," *Information Retrieval*, vol. 4, pp. 133-151, 2001.
- [26] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in Artificial Intelligence*, vol. 2009, pp. 1-20, 2009.
- [27] B. N. Miller, J. A. Konstan, and J. Riedl, "PocketLens: Toward a personal recommender system," *ACM Transactions on Information Systems (TOIS)*, vol. 22, pp. 437-476, 2004.
- [28] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*, pp. 285-295, 2001.
- [29] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pp. 43-52, 1998.
- [30] C. Basu, H. Hirsh, and W. Cohen, "Recommendation as classification: Using social and content-based information in recommendation," in *Proceedings of the national conference on artificial intelligence*, pp. 714-720, 1998.
- [31] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, and M. Sartin, "Combining content-based and collaborative filters in an online newspaper," in *Proceedings of ACM SIGIR workshop on recommender systems*, pp. 1-11, 1999.
- [32] M. J. Pazzani, "A framework for collaborative, content-based and demographic filtering," *Artificial Intelligence Review*, vol. 13, pp. 393-408, 1999.

- [33] T. Zhu, R. Greiner, and G. Häubl, "Learning a model of a web user's interests," in *User Modeling 2003*, ed: Springer, pp. 65-75, 2003.
- [34] F. Durao and P. Dolog, "A personalized tag-based recommendation in social web systems," *Adaptation and Personalization for Web*, vol. 2, p. 40, 2009.
- [35] K. H. L. Tso-Sutter, L. B. Marinho, and L. Schmidt-Thieme, "Tag-aware recommender systems by fusion of collaborative filtering algorithms," in *Proceedings of the 23rd ACM symposium on Applied computing*, pp. 16-20, 2008.
- [36] C. Wu and B. Zhou, "Analysis of tag within online social networks," in *Proceedings of the ACM 2009 international conference on Supporting group work*, pp. 21-30, 2009.
- [37] X. Zhou, Y. Xu, Y. Li, A. Josang, and C. Cox, "The state-of-the-art in personalized recommender systems for social networking," *Artificial Intelligence Review*, vol. 37, pp. 119-132, 2012.
- [38] B. S. Solomon, D. Duce, and R. Harrison, "Methodologies for using Social Media Collaborative Work Systems," in *Requirements Engineering for Social Computing (RESC), 2011 First International Workshop on*, pp. 6-9, 2011.
- [39] J. Peng, D. D. Zeng, H. Zhao, and F.-y. Wang, "Collaborative filtering in social tagging systems based on joint item-tag recommendations," in *Proceedings of the 19th ACM international conference on Information and knowledge management*, pp. 809-818, 2010.
- [40] A. Nocera and D. Ursino, "An approach to providing a user of a "social folksonomy" with recommendations of similar users and potentially interesting resources," *Knowledge-Based Systems*, vol. 24, pp. 1277-1296, 2011.
- [41] P. De Meo, A. Nocera, G. Terracina, and D. Ursino, "Recommendation of similar users, resources and social networks in a Social Internetworking Scenario," *Information Sciences*, vol. 181, pp. 1285-1305, 2011.
- [42] J. Golbeck, "Generating predictive movie recommendations from trust in social networks," in *Proceedings of the 4th International Conference on Trust Management, iTrust*, pp. 93-104, 2006.
- [43] C.-N. Ziegler and G. Lausen, "Analyzing correlation between trust and user similarity in online communities," in *Trust management*, ed: Springer, pp. 251-265, 2004.
- [44] P. Avesani, P. Massa, and R. Tiella, "A trust-enhanced recommender system application: Moleskiing," in *Proceedings of the 2005 ACM symposium on Applied computing*, pp. 1589-1593, 2005.
- [45] Y. Ryu, H. Kim, Y. Cho, and J. Kim, "Peer-oriented content recommendation in a social network," in *Proceedings of the Sixteenth Workshop on Information Technologies and Systems*, pp. 115-120, 2006.
- [46] P. Massa and P. Avesani, "Trust-aware recommender systems," in *Proceedings of the 2007 ACM conference on Recommender systems*, pp. 17-24, 2007.
- [47] Q. Yuan, L. Chen, and S. Zhao, "Factorization vs. regularization: fusing heterogeneous social relationships in top-n recommendation," in *Proceedings of the fifth ACM conference on Recommender systems*, pp. 245-252, 2011.
- [48] H. Ma, H. Yang, M. R. Lyu, and I. King, "Sorec: social recommendation using probabilistic matrix factorization," in *Proceedings of the 17th ACM conference on Information and knowledge management*, pp. 931-940, 2008.
- [49] B. Liu and Z. Yuan, "Incorporating social networks and user opinions for collaborative recommendation: local trust network based method," in *Proceedings of the Workshop on Context-Aware Movie Recommendation*, pp. 53-56, 2010.

- [50] G. Groh and C. Ehmig, "Recommendations in taste related domains: collaborative filtering vs. social filtering," in *Proceedings of the 2007 international ACM conference on Supporting group work*, pp. 127-136, 2007.
- [51] F. C. T. Chua, H. W. Lauw, and E.-P. Lim, "Predicting item adoption using social correlation," in *SIAM Conference on Data Mining*, pp. 367-378, 2011.
- [52] M. Ye, X. Liu, and W.-C. Lee, "Exploring Social Influence for Recommendation-A Probabilistic Generative Model Approach," in *Proceedings of the 35th Annual ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR*, pp. 671-680, 2012.
- [53] V. Vasuki, N. Natarajan, Z. Lu, and I. S. Dhillon, "Affiliation recommendation using auxiliary networks," in *Proceedings of the fourth ACM conference on Recommender systems*, pp. 103-110, 2010.
- [54] W.-Y. Chen, J.-C. Chu, J. Luan, H. Bai, Y. Wang, and E. Y. Chang, "Collaborative filtering for orkut communities: discovery of user latent behavior," in *Proceedings of the 18th international conference on World wide web*, pp. 681-690, 2009.
- [55] E. Spertus, M. Sahami, and O. Buyukkokten, "Evaluating similarity measures: a large-scale study in the orkut social network," in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pp. 678-684, 2005.
- [56] Q. Yuan, S. Zhao, L. Chen, Y. Liu, S. Ding, X. Zhang, *et al.*, "Augmenting collaborative recommender by fusing explicit social relationships," in *Workshop on Recommender Systems and the Social Web, Recsys 2009*, pp. 46-56, 2009.
- [57] N. Zheng and Q. Li, "A recommender system based on tag and time information for social tagging systems," *Expert Syst. Appl.*, vol. 38, pp. 4575-4587, 2011.
- [58] Z. Huang, H. Chen, and D. Zeng, "Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering," *ACM Transactions on Information Systems (TOIS)*, vol. 22, pp. 116-142, 2004.
- [59] 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*, pp. 195-202, 2009.
- [60] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 230-237, 1999.
- [61] M. Braunhofer, M. Kaminskis, and F. Ricci, "Recommending music for places of interest in a mobile travel guide," in *Proceedings of the fifth ACM conference on Recommender systems*, pp. 253-256, 2011.
- [62] Z. Huang, D. Zeng, and H. Chen, "A comparison of collaborative-filtering recommendation algorithms for e-commerce," *Intelligent Systems, IEEE*, vol. 22, pp. 68-78, 2007.
- [63] I. Fernández-Tobías, I. Cantador, and A. Bellogín, "Semantic disambiguation and contextualisation of social tags," vol. 7138 LNCS, ed, pp. 181-197, 2012.