Social Tag-based Community Recommendation using Latent Semantic Analysis

by

Aysha Akther

Thesis submitted to the

Faculty of Graduate and Postdoctoral Studies

In partial fulfillment of the requirements

For the Master's degree in Computer Science

Ottawa-Carleton Institute for Computer Science

School of Electrical Engineering and Computer Science

Faculty of Engineering

University of Ottawa

Abstract

Collaboration and sharing of information are the basis of modern social web system. Users in the social web systems are establishing and joining online communities, in order to collectively share their content with a group of people having common topic of interest. Group or community activities have increased exponentially in modern social Web systems. With the explosive growth of social communities, users of social Web systems have experienced considerable difficulty with discovering communities relevant to their interests. In this study, we address the problem of recommending communities to individual users. Recommender techniques that are based solely on community affiliation, may fail to find a wide range of proper communities for users when their available data are insufficient. We regard this problem as tag-based personalized searches. Based on social tags used by members of communities, we first represent communities in a low-dimensional space, the so-called latent semantic space, by using Latent Semantic Analysis. Then, for recommending communities to a given user, we capture how each community is relevant to both user's personal tag usage and other community members' tagging patterns in the latent space. We specially focus on the challenging problem of recommending communities to users who have joined very few communities or having no prior community membership. Our evaluation on two heterogeneous datasets shows that our approach can significantly improve the recommendation quality.

Acknowledgements

This thesis is based on research, conducted during my graduate studies at the Multimedia Communications Research Laboratory (MCRLab) and DISCOVER Lab of the University of Ottawa. The work presented here is the accumulated result of highly collaborative efforts, and I owe gratitude to the people, I have worked with. First and foremost, I would like to convey my sincere gratitude to my supervisor, Dr. Abdulmotaleb El Saddik, for his guidance, words of encouragement, valuable comments, and support during my entire masters study. Beside's my supervisor, I am grateful to Dr. Heung-Nam Kim, for his suggestions, invaluable help and feedback, hours of meetings and discussions. Without their supports, this thesis would not have been possible. I would like to express my deepest thanks to my parents; Md. Yousuf Ali and Mahfuja Khanam who have been an infinite source of love, support and encouragement all throughout my life. I also want to thank my siblings for their continuous inspiration. Last but not the least, I am grateful to my dearest husband, Kazi Masudul Alam and our two kids Kazi Tahmeed Alam, and Sameeha Alam, for their love, encouragement and sacrifice to follow my dreams.

Dedication

I dedicate this thesis to my parents, my loving husband and our kids.

Contents

List of Tables

List of Figures

Glossary of Terms

ARM Association Rule Mining

HTML Hyper Text Markup Language

IPTV Internet Protocol Television

LDA Latent Dirichlet Allocation

LSA Latent Semantic Analysis

MAP Mean Average Precision

MRR Mean Reciprocal Rank

OSN Online Social Networking

PLSA Probabilistic Latent Semantic Analysis

SVD Singular Value Decomposition

SVM Support Vector Machine

Chapter 1

Introduction

1.1 Background and Motivation

Web 2.0, the second generation of the World Wide Web has changed the relation between user and web pages. Now users are not only passive consumer of static HTML web pages; they are actively interacting with it. Web 2.0 pages allow users to contribute by collaboration and sharing information. Users contribute by providing some content of the Web 2.0 sites and also can have some control over that [43]. Blogs, wikis, podcasting, and online social-networking sites are some examples of Web 2.0 applications. Some important features of web 2.0 sites are social tagging, social bookmarking and user as a contributor. As being an example of Web 2.0 applications, Online Social Networking (OSN) sites perceive all the essential Web 2.0 features. Every user has his/her own profile in an online social environment. User can link with other users based on offline relationship, similar interest or some other factors. Users also create and share items with other users in the OSN. With the collaboration of a collection of users, OSNs become a "collective intelligence" on a specific domain [36]. Different OSNs serve different purposes. Some are designed to share photo; some are designed to share music; some are designed to share video, and some are designed to share scientific articles etc. Some social-networking sites become so popular that people incorporate their daily life practices with their virtual life in the OSNs. Popularity of these social networks is increasing very rapidly. For example, one of the most popular OSN sites Facebook [46] had 500 million active users all over the world in July 2010 [30], whereas at the end of March 2012, it has 901 million monthly active users [44]. So statistics states that the

popularity of OSNs is increasing rapidly. From the rapid growth of popularity of the social networks, we can assume that they are serving millions of users by providing satisfaction of their desires [17].

Most of the OSNs also allow users to form groups or communities. In general, communities tend to be formed by people who share some interests in common; users explicitly join such communities according to topics of interest. Users are actively establishing and joining online communities in order to collectively share their content with a group of people who have common interests [10]. Flickr [53], Vimeo [54], Lastfm [52], and CiteULike [51] are some of the representative OSN services within which such communities play an important role in information sharing. For example, users of CiteULike, one of the outstanding systems for organizing and discovering scholarly papers, collaboratively create shared libraries of articles in a particular field in which they are interested [7]. Likewise, users of Flickr, the best online photo sharing application, participate in groups as a way to communicate with other members around common photo interests [15]. They also assign tags or short key phrases to the shared items within the community. Tags are acting as user given key words that represent users' preferences and also represent topic of the group or community. In modern social Web systems, group or community activities have increased exponentially. Creating new community and joining a community is easy in the virtual environment of social networks. This flexibility helps to enhance diversity in the virtual world. Almost all popular OSNs have many existing communities created by the users, and the number of online communities is increasing every day. For example, Flickr has millions of communities formed by users of similar photo interests [35]. In order to find appropriate communities to join, users need to put a lot of efforts. It became difficult for users to dynamically identify and join appropriate community based on their needs, from thousands of existing communities. In this situation, it is necessary to assist people to find and access appropriate communities that reflect their interests and needs the most. The explosive growth of communities has led researchers to concentrate on identifying (recommending) communities (or groups) that will be of interest to individuals [1, 3, 4, 13, 15, 34, 35, 36]. In our study, we focus on this community recommendation problem.

1.2 Existing Research Problems

Recommender systems emerged in response to the problem of choosing among so many options. When communities are regarded as items, a variety of techniques used for personalized item recommendations, such as Collaborative Filtering (CF), could be applied directly. Collaborative filtering based recommender systems [27] are the earliest one, and are successful in many cases. Despite its simplicity, typical CF for community recommendations would encounter weaknesses, including the sparsity and cold start problems [11]. In practice, unless users are very active, many users still have insufficient membership information. Accordingly, recommender techniques that are based solely on community affiliation may fail to find a wide range of proper communities for users when their available data are insufficient. If community *A* and *B* has no common member, then users of community *A* will never get recommendation for community *B* using CF. This is the problem of sparseness in explicit co-occurrence. Additionally, it is the often case that some users have not previously joined any community, posing the challenging problem of recommending communities to such users. In this situation, their preferences should be inferred implicitly from additional sources. A number of methods have been investigated by researchers to explore users' implicit preferences, though those methods have their own shortcomings. While the methods which infer users implicit preferences are helpful in improving sparsity-related problem, but they can hardly deal with users having no community membership.

1.3 Objective

We address the above-described issues by taking advantage of user-generated tags and thus discovering common topics of interests shared by groups of users. With the current popularity of tag usage, users who post/upload items (e.g., photos on Flickr, videos on Vimeo, music on Last.fm, and papers on CiteULike) actively take the time to add tags to those items for the purpose of describing and characterizing content [8]. If a set of tags are frequently used by members of a certain community, these tags may characterize the main topics of that community. In addition, users sharing similar topics of interest are likely to use similar tags; if such users belong to the same or similar communities, they would have a greater tendency toward such tagging behaviors. In a situation where we attempt to identify communities, in which a particular user is most likely to be interested, tags could have the

potential benefit to represent his/her interests implicitly as well as communities' topics concisely. With this in mind, we analyze not only his/her individual tagging behaviors, but also collective tagging patterns of other users who have belonged to the same communities. To exploit users tagging behavior, we need to understand the inherent topic of the synonymous tags used by the users with different knowledge, needs, or linguistic habit. In these consequences, we adapt Latent Semantic Analysis (LSA) [5], which is the well-known model for dimensionality reduction, for use in the community recommendation scenario. By using LSA, we represent communities in a low-dimensional space, the so-called latent semantic space, so as to capture synonymous tags that refer to the same topic as well as to reduce noise tags. In this phenomenon, it is possible to recommend community *B* to some users of community *A* because of some implicit relationship between them, even though community *A* and *B* has no members in common. Also it might be possible to recommend communities of interest to a cold user by analyzing his/her implicit tagging behavior. In this latent representation, the community recommendation task is viewed as finding topics of interest to users, which would ultimately facilitate better recommendation quality.

1.4 Problem statement

The explosive growth of the number of communities in OSNs also leads to uncertainty together with bringing diversity in virtual life. Vast number of communities creates difficulties for users in choosing the right community to join. Also the number of communities in OSNs is increasing day by day. In such situation, helping individual users' to find appropriate community that matches the users' personality became increasingly important. Because of extensive user contribution in OSNs, it is possible for them to consume contents in a personalized way, so that, the contents meet their interests and needs. In these circumstances, in this study we proposed a latent semantic analysis based community recommendation method for OSNs users, which uses users' social tagging behavior to infer users' implicit preference.

1.5 Thesis Contribution

This research has following primary contributions:

1. Design and develop a latent semantic analysis based recommendation method for online social-networking services. Applying the proposed recommendation method for

communities; we represent communities in latent topic space instead of latent user space. In our proposed method, we construct an implicit query consisting of users' personal tag preference and collective tag preference of other users of the same communities' user has previously joined.

2. Analyze the effectiveness of our proposed recommendation method in various challenging conditions on two heterogeneous actual online social network data.
3. A systematic experimental evaluation of the performance of our proposed recommendation method from different perspectives in comparison with a number of different dimensionality reduction based recommendation methods.

1.6 Scholarly Articles

In addition to meet its objectives as described above, this research undertaking has also lead to a variety of scholarly publications, as listed below.

Papers in Referred Conferences

Aysha Akther, Kazi Masudul Alam, Heung-Nam Kim, Abdulmotaleb El Saddik, Social Network Assisted Personalization with User Context for Recommender Systems, In IEEE Innovations in Information Technology, 2012, March 18-20, UAE

Aysha Akther, Heung-Nam Kim, Majdi Rawashdeh, Abdulmotaleb El Saddik, Applying Latent Semantic Analysis to Tag-based Community Recommendations. Advances in Artificial Intelligence, Springer, 2012, pp. 1—12.

1.7 Thesis Organization

The remainder of this thesis is organized as follows:

Chapter 2 presents an overview of the background literature and related studies. Background literature contains a brief discussion on some concepts that we need to clear for recommending communities to users. Then, we present a brief review of the closely related previous research works.

Chapter 3 presents our proposed community recommendation system description. We present the community recommendation task in OSN environment by steps. We provide detailed descriptions of representing communities in the latent semantic space and thus recommending communities of value to users. We describe each step with an illustrative example.

Chapter 4 gives the details evaluation results of the performance of the proposed recommendation system. We present evaluation of the performance by measuring two standard performance measures. We evaluate performance on various different conditions and from different perspectives. We present the evaluation results in comparison with four other dimensionality reduction based recommendation methods.

Finally, in chapter 5, we conclude by summarizing the overall contribution of our research and discuss some directions to possible future works to enhance the performance.

Chapter 2

Background and Related Works

In this chapter, we provide a brief discussion on various components of recommending communities in online social network scenario in literature review section. In related study section, we briefly discuss some methods applied in the recent years in the community recommendation task. Finally, we provide a summary of the studies and mention how our work differs from prior studies.

2.1 Literature Review

In this section, we present a review of the literature related to recommendation in online social network. Here we briefly describe online social networks, social tagging system, groups/communities in online social networks and necessities of dimensionality reduction based methods for recommendation in online social networks.

2.1.1 Online social network

A social network is "a set of people (or organizations or other social entities) connected by a set of social relationships, such as friendship, coworking or information exchange" [16]. Online social networking is the act of interacting and networking in an online social environment. In the recent years, online social networking emerged as the most popular Web 2.0 application [17, 18]. Facebook [46], Twitter [47], Orkut [49], Youtube [58], Myspace [48], LiveJournal [50], Lastfm [52] are examples of some popular online social-networking websites. Millions of users of the

online social-networking sites have integrated their daily practices with the social-networking sites they use [19]. There are many social networking sites designed from different perspectives. They differ in goals, purposes, and languages. Some of these websites have become more popular than others. Now a day, people are not member of only one social networking site, but they are member of many. Different social-networking sites serve different purposes. Users are the core element of social-networking sites; that is, social-networking sites evolve around users. In general, within all social-networking sites, each user has his/her own profile; he/she can create or upload content and link with some other users based on shared interest or some other online or offline relation. Most of the sites support new users to connect and share their interests, in addition with maintaining existing users. So online social-networking sites can be considered as virtual space for sharing interests among social networkers. Shared interest varies from site to site based on the service the specific OSN provides. Shared interest among social networkers in social networking sites may include photo [53], music [52], movie [55], blog [56], video [54], books [57], scientific articles [51], etc. Furthermore, some of them provide mobile interactions. OSNs are different in their strategies and objectives. Though the interests or activities they support vary, their core technical features remain similar [19]. A number of research already studied various aspects of the online social-networking sites [10, 16, 17, 19, 24, 26, 29, 38, 39, 40, 45]. Article [16], discussed the structure of social network itself and the relations in it, [19] study the history of OSN sites and scholarships in them, [17] study the Web 2.0 environment, its tools, applications, characteristics and the operation of social networks in the Web 2.0 environment, [10] presents a large-scale measurement study and analysis of the structure of four online social networks. Researchers found that the network exhibits small-world behavior and significant local clustering. There are several hundreds of OSN sites [20], supporting a wide range of interests and user activities. By observing the day to day popularity of online social-networking sites, researchers assume that social networking will play an important role in future personal and commercial online interaction and to locate and organize information and knowledge [10].

2.1.2 Social Tagging System

The process of assigning keywords (tags) by the users or social-networkers to the items shared in OSNs is known as Social Tagging System (STS). Social Tagging Systems allow sharing the

individual tagging activities of users into a network of tags and items shared among many users. STS concerns with three different entities: users, items and tags. The tag sharing has advantages in discovery of new item as well as retrieval of previous tagged items [21]. Shared tags are sometimes termed as shared vocabulary. This shared vocabulary is viewed as a description of the content of the tagged items [22]. In one hand, tags are personal annotations of the users to the items, on the other hand they serve as a guideline to new users about the item. Tagging helps an individual to retrieve the tagged items at a later date with known keywords. Tags make the items content searchable. For these benefits, users actively take time to assign tags to items shared in OSNs. In social-networking sites, usually no restrictions are placed to choose tags, so sometimes tags can be noisy or ambiguous. Social tagging has been applied to photos [53], videos [58], music [52], and scientific paper citations [51]. Similar items can be tagged with similar tags and in many cases, similar minded users use similar tags to annotate items. As a result, some items are linked together because of annotating with similar tags, and certain users are also linked because of their similar social interest [8]. A number of researchers studied different aspects of STSs [22, 21, 23, 38, 39]. In the area of recommendation, information incorporated in STS has proven its importance in recommending items, tags and users [11]. In [22], the efficiency of tags in organizing the items they are supposed to encode is studied. Article [21] studied the reasons behind social tagging systems effectiveness. Authors in article [23] studied the relevance of tags in music information retrieval. In [41], authors analyze the structure and usage pattern of social tagging systems in Delicious and compared the difference between collaborative tagging and taxonomies.

2.1.3 Groups or Communities in Online Social Network

Most of the online social-networking sites allow users to form and join special-interest communities or groups. Relationship between users inside the communities is denser, that is users are highly interconnected in a community. So groups or communities are parts of social network, which is tightly-bound and densely-knit [16]. In some social networks initially groups or communities are formed by homogeneous people, maybe people of same demographic area, similar educational level, common language or shared racial, sexual, religious or some other factors. In general, communities or groups are formed based on shared common interest. So

groups or communities can be visualized as clusters within a social network. Most of the groups are free to join, but some require group administrator's permission. Users can create or upload items to the groups, and they can also assign tags to the uploaded items. Each individual user can be a member of as many groups as he/she wants. So groups are overlapping one another and growing continuously in a complex fashion [17] [24]. Users can join new groups through common interest searches, by recommendations or by way of other groups. Article [40] analyzed the method of finding communities in a network. Formation and operation of groups vary widely among social-networking sites. Figure 2.1 visualizes an example of three communities of any social network. The example social network has three overlapped groups. Group ***A*** has two users in common with group ***B*** and one common user with group ***C***.

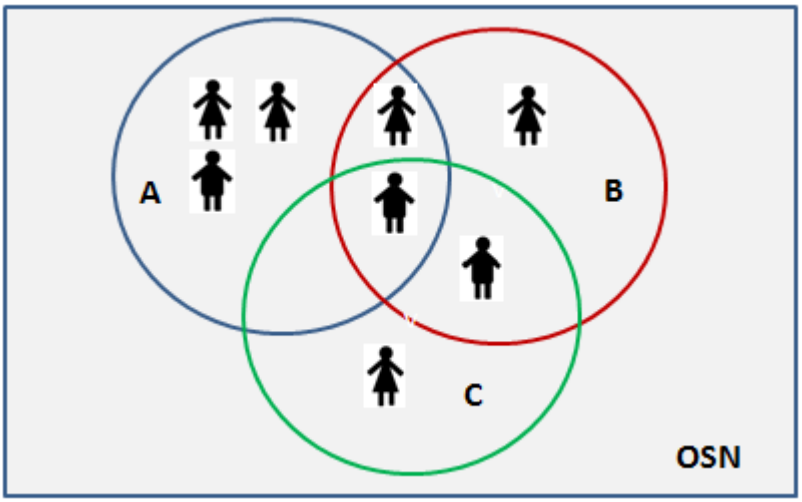

Figure 2.1: Visualization of three commuities in an online social network (OSN).

2.1.4 Dimensionality Reduction based Recommendation for OSN

In an online social network environment, users need various types of recommendations. Users need recommendation to reach new user, to get new item or to find group or community of users' interest and so many. Sifting through the available options and choosing the preferred one among them is tough for them. Recommender systems have emerged in response to this problem. Recommender system analyzes the users' current profile and past behavior in the online environment and provides recommendation to the user as a prediction of his future preference that

are likely to his/her needs [25]. So, in case of online social network, recommender system's goal is to help users to establish new relationship to expand human community [26]. Recommender systems apply statistical and knowledge discovery techniques to the problem of making recommendations [27]. Collaborative filtering [27, 28, 29] based recommendation provide conceptually easy solution to the problem. But typical collaborative filtering based approaches suffered from sparsity, scalability and synonymy problems, which are very common in the online social networks. Some popular online social networks for example, Facebook has over 900 million active users on May 2012 [30], Lastfm claimed over 30 million active users in March 2009 [31]. Using collaborative filtering approach to recommend in such an environment is difficult. Also in many cases, in high-dimensional datasets, all the measured variables are not important to understand the underlying phenomena of interest [32]. The weaknesses of collaborative filtering based approaches led researchers to explore alternative recommendation methods [11]. Dimensionality based approaches are one of the alternatives. Dimensionality reduction is the transformation or mapping of high-dimensional data into a meaningful representation of lower dimensional space. Lower dimensional representation has a minimum number of parameters needed to account for the observed properties of the data [33]. Thus important criteria help dimensionality based methods to deal with a large amount of data.

2.2 Related Works

Though a considerable number of researchers studied various aspects of online social networks, yet research on recommending communities is few. In this section, we study some recent researches on recommending communities in online social network environment.

Authors in article [34] evaluated six different similarity measures to recommend communities to users of Orkut. The recommendation is based on users' current community membership. In their proposed method, all users of a community are provided with same recommendation despite their personal choice. In their experimental result L2 vector normalization showed best performance for recommending communities over other similarity measures.

Chen et al. proposed a fusion method to recommend communities in article [3], which combined information from multiple sources. They named the method *Combinational Collaborative Filtering* (CCF). In the proposed method, a community was viewed both as a bag of participating users and as a bag of words describing that community. According to the authors, by combining these two community information sources, the method overcame some limitations that single information source had.

In the proposed CCF model, they considered a given collection of co-occurrence data consisting of communities $C = \{c_1, c_2, \ldots, c_N\}$, community descriptions from vocabulary $D = \{d_1, d_2, \ldots, d_V\}$, and a set of users $U = \{u_1, u_2, \ldots, u_M\}$. When a community was viewed only as a bag of participating users, they named it C-U model, and it was for community-user co-occurrence analysis, which could be derived from simple PLSA (Probabilistic Latent Semantic Analysis) [60]. The co-occurrence data consisted of a set of community-user pairs (c, u) had value 1 if community c was joined by user u and 0 otherwise. Then a latent class variable $z \in Z = \{z_1, z_2, \ldots, z_K\}$ associated to every community-user pair. When a community is viewed only as a bag of words describing that community, named as C-D model. In C-D model, the latent class variable z represented the topic for a community. The CCF model associated C-U model with C-D model. In the CCF model, each community had a multinomial distribution over topics, and each topic had a multinomial distribution over users and descriptions, respectively. Graphical representation of three models is shown in Figure 2.2.

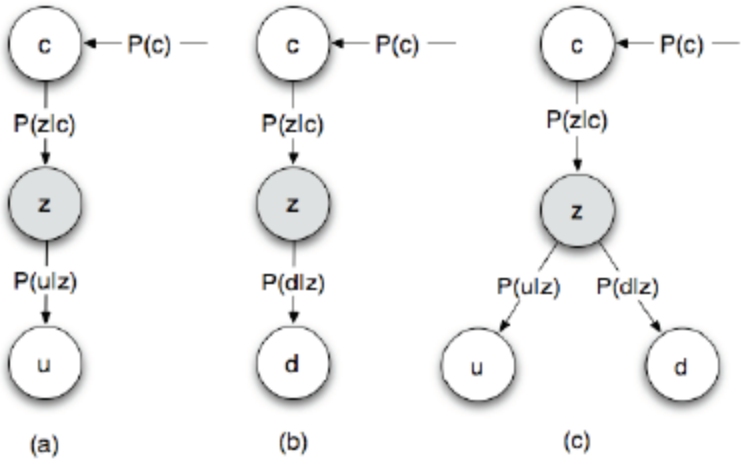

Figure 2.2: Graphical representation of (a)Community-User (C-U) model, (b)Community-Description (C-D) model, (c)Combinational Collaborative Filtering (CCF) model [3].

Authors depicted the CCF model as the joint probability distribution over community, user, and description. Probability distribution of CCF was presented as:

$$P(c,u,d) = \sum_{z} P(c,u,d,z) = P(c)\sum_{z} P(u/z)P(d/z)P(z/c) \tag{2.1}$$

Authors also exploited two strategies to make the model scalable. They employed a hybrid training strategy which used Gibbs sampling for first few iterations then the Expectation-Maximization (EM) algorithm. The second speedup strategy was to parallelize the computation over a number of computers in a network. The experimental results presented in the paper proved the strategies importance for scalability.

CCF provides personalized recommendation by combining community-user co-occurrence information with community description. However in cases, when a user has no community membership, CCF would fail to provide any recommendation, in spite of associating multiple information sources.

In article [35], a system is proposed for Flickr to add photos into proper groups and to recommend tags for those. The authors named the system "SheepDog." They compared two training mechanisms: photo level mechanism and group level mechanism. In photo level mechanism, to collect training photos on a specific topic or concept, they used the topic or concept name as a keyword. For each concept, they selected top k photos on that concept based on interestingness of the photos. In group level mechanism, again concept name was used as a keyword to collect photos. After that, based on some factors, they selected top groups for each concept. Then for each selected group, they sort all photos by interestingness. Finally, the number of photos they picked for each group was proportional to the groups' ranking result. To learn a probability-based model for all concepts, support vector machine (SVM) [37] was used over extracted visual features from the photos.

To recommend appropriate groups for a photo, top *n* concepts are predicted using the training model. Then, for each concept top groups selected by some popularity factors are recommended to the user.

Article [36] proposed a system to identify and recommend Web 2.0 based IPTV (Internet Protocol Television) communities to users by using their social relationships and preferences. To identify potential IPTV communities the system used to motivate users to make communities by recommending them potential groups of users having similar preferences regarding IPTV contents. Existing communities were recommended after social filtering on IPTV contents based on users' social relationships and interests.

They identified two requirements for recommending communities: Potential community identification and Community accessibility. In the first step of recommendation, users' social network information and related data was extracted by the system to retrieve preferences. Next the system organized a semantic social network based on the retrieved user preferences and social-network information. To organize semantic social network, social-network information like familiarity, favorability, and similarity were considered as essential factors. In the semantic social network, they considered target user at the center of the network, and also considered other users having similar preferences and social relationships with the target user. This semantic social network was recommended to target user as a potential community. To recommend existing communities, all the existing communities referred by the users from the semantic social network were filtered by the system based on target users' preferences and the remaining ones were recommended.

Two different approaches to the community recommendation task are investigated in article [4]. One is Association Rule Mining (ARM), and the other is Latent Dirichlet Allocation (LDA). The paper studied the scenarios when these two relations manifest their strengths and weaknesses. ARM, which is a data mining algorithm, used to find association rules based on co-occurring sets of communities then to recommend communities based on the rules. Each user was considered as a transaction and his joined communities were considered as items in ARM. They applied FP-growth algorithm [42] for mining frequent item sets and then association rules were generated from the item sets.

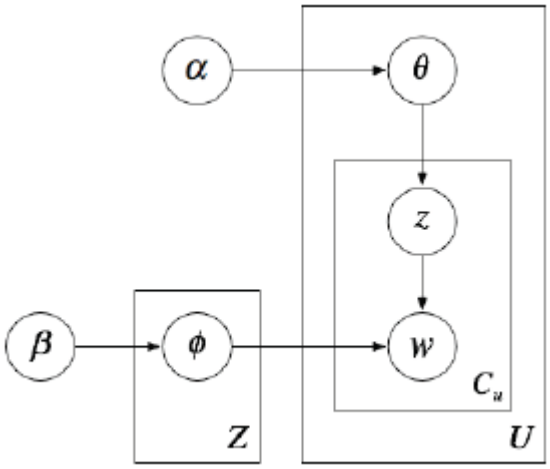

Figure 2.3: LDA model for user-community data [4].

Latent Dirichlet Allocation (LDA) [59], which is actually a machine-learning algorithm, used to model user-community co-occurrences using latent aspects and made recommendations based on the learned model parameters. In LDA, they used user-community binary membership data and Gibbs sampling was used to estimate model parameters. Figure 2.3 shows the LDA model for user-community data. In the Figure, per-user topic distribution Θ, each drawn independently from a symmetric Dirichlet prior α, and the per-topic community distribution ϕ, each drawn from a symmetric Dirichlet prior β. For each occurrence, the topic assignment was sampled from following equation:

$$P(z_i = j \mid w_i = c, z_{-i}, w_{-i}) \propto \frac{C_{cj}^{CZ} + \beta}{\sum_{c'} C_{c'j}^{CZ} + M\beta} \frac{C_{uj}^{UZ} + \alpha}{\sum_{j'} C_{uj'}^{UZ} + K\alpha} \tag{2.2}$$

Where C_{cj}^{CZ} is the number of times community c was assigned to topic j except current instance, and C_{uj}^{UZ} is the number of times topic *j* was assigned to user *u* except current instance. At the beginning, randomly a topic was assigned to each community, and then the assigned topic was updated using Gibbs sampling for each community occurrence. This topic updating process repeated for several iterations. Finally, user-community relations were inferred using Bayes' rule from learned model parameters.

ARM discovered explicit relations between communities from co-occurrences of communities among users. On the other hand, LDA models the implicit relations between communities through the set of latent aspects. Authors performed diverse experiments to discover the strength and weakness of the algorithms in this domain. Experimental results presented in the paper showed that LDA performed better for users having a small number of joined communities of concentrated interests, and ARM was better for users having relatively large number of joined communities of scattered interests. For recommending up to three communities, ARM performed a bit better than LDA. However, LDA performed consistently better than ARM when recommending a list of four or more communities.

In article [1], a group recommendation system is proposed for popular online social network Facebook. The group recommendation system consists of three components: profile feature extraction, classification engine, and final recommendation. They extracted 15 features to characterize a group member. Then hierarchical clustering was used to remove members whose characteristics were not quite relevant with majority of a group. Similarity between members was measured using Euclidian distance. Clustering was done by normalizing each feature value, computing a distance matrix to calculate similarities among all pairs of members, then using unweighted pair-group method using arithmetic averages (UPGMA) on distance matrix to generate hierarchical cluster tree. Based on a user's profile features, group recommendation system found the most suitable groups for a user using decision tree algorithm based on binary recursive partitioning. Their experimental results showed 9% improvement in average accuracy of group recommendation after removing noise from each group using clustering coefficient method over non-clustered data.

Zheng et al. [15] studied various methods, including memory-based, matrix factorization-based, and tensor decompositions-based recommendation methods, to recommend groups to Flickr users. In the paper, the strengths and weaknesses of the approaches were studied at different scenarios. Similar to our work, the tensor decompositions methods employed tagging information. Their experimental results demonstrated that incorporating tags have an advantage in dealing with sparse data, whereas the methods without tags are more suitable with dense data. Their study also

showed that model based recommendation approaches are beneficial than memory based approaches for top k recommendation.

More recently, in article [13], authors proposed two methods for group recommendation by combining user friendship network data with user-community affiliation network. One method based on graph proximity, and the other using latent factors to model users and communities. In the graph proximity method, they employed truncated Katz measure[61] to predict new links between users and groups in the social network. Truncated Katz measure applied on a combined matrix of user friendship network and user-group network where the heterogeneity of two types of networks was controlled by a single parameter that controlled the ratio of the weight of friendship to the weight of group membership. In the latent factor model, another combined matrix was used, which also consider group similarities. In the latent factor model, they also proposed low rank approximation of combined matrix from clustered sub networks of the social network. Experimental results presented in the paper showed that performance of graph proximity model - was better than that of a latent factor model, though the latent factor model showed more scalability than graph proximity method.

2.3 Summary

In this chapter, we discussed a brief background literature that is necessary to understand community recommendation scenario in an online social network. We also discussed methods employed by some previous works in this domain and their results. We provided comparatively detailed discussion on two methods with which we compare the experimental results of our proposed method in chapter 3. Our work differs from most of the earlier works in that the proposed method has been devised to effectively incorporate tags in community recommendations. In doing so, our work viewed the community recommendation task as finding topics of interest. Moreover, our work can successfully recommend communities of value to users even when they have previously joined no (or few) communities.

Chapter 3

A Latent Semantic Model for Community Recommendations

This chapter presents a detail description of the proposed Latent Semantic Analysis based community recommendation method. From section 3.1 to section 3.3, the scenario of the problem domain, notations used in the system and outline of the algorithm are discussed. In section 3.4, section 3.5 and section 3.6 steps of the algorithm are depicted with an illustrative example. Finally section 3.7 provides a summary of the method.

3.1 Problem Domain Scenario

OSNs consist of users, items, tags and communities. Users' in the OSNs upload and share items in the virtual world of social networkers. Items or the contents shared in social-networking sites reflect users' interests. The culture of social-networking sites varies from site to site but their key technological features remain rather consistent [19]. Shared items or contents are different according to the service the social-networking sites provide. For example, in CiteULike items are scientific articles, in Delicious items are web pages, in Last.fm items are music, in Flickr items are photos etc. Users assign tags or keywords to the items to give a short description of the item type, which makes items searchable by keywords. Most OSNs also allow community activities among users. Users of similar tastes can form communities to share their uploaded items among similar minded people. Communities are a collection of users creating shared items. Though some communities can form based on some other characteristic features of the community members, shared interest among users about items is the basis of the connection in most of the communities. The scenario of an example community consisting of four member users, two shared items and some shared tags is presented in Figure 3.1, where the community is formed based on shared interests about item:

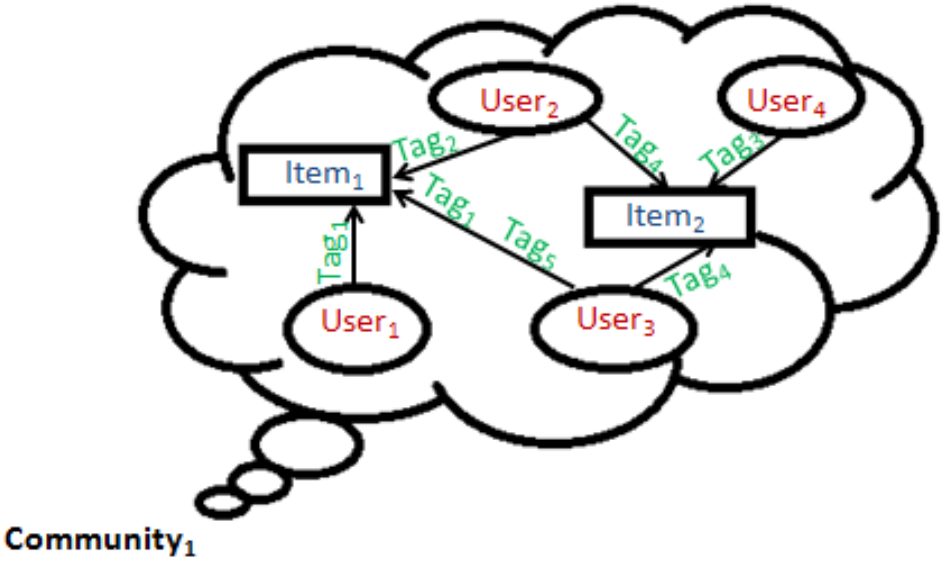

Figure 3.1: Online Social Network Scenario of one community

Different users of a community often assign same or similar tags to items shared in the community. These tags represent users' interests as well as topic of the community. Now a day, most of the popular social-networking sites have so many communities that it has become difficult for users to find the proper one for them. Communities are formed by a number users and one user can be a member of as many communities as he/she want. Tags assigned to the items shared in one community are considered as tags belong to that community. As this study focus on community recommendation based on tags, we are excluding items in the next to the study.

3.2 Problem Formalization

In this study, we regard the problem of the community recommendation as tag-based personalized searches, which tailor search results to individual users, by assuming that a certain user implicitly submits a query comprising a set of tags. Before going into further detail, we introduce some notations. Let $U=\{u_1, u_2,...,u_{|U|}\}$ be the set of distinct users, $T=\{t_1,t_2,...,t_{|T|}\}$ be the set of distinct tags, and $C=\{c_1, c_2,...,c_{|C|}\}$ be the set of distinct communities created/joined by the users. In a social web system, the users can use tags to describe/organize their content and can create/join communities. We assume that individual users' tags can potentially represent their personal interests. Additional notations used in the reminder of this paper are summarized in Table 3.1.

Table 3.1: Meaning of Notation

Notations	Meanings
U	Set of users.
T	Set of tags.
C	Set of communities.
F	User-tag frequency matrix.
M	User-community membership matrix.
N	Tag-community frequency matrix.
$\mathbf{q}_u$	Query vector for a given user u.

.

The user-community membership matrix consists of community-user co-occurrence data. Community recommendation solely depending on user-community co-occurrence data would be unable to recommend any community to a user who has not yet joined any community. Also it is hard to recommend community to users with very few community memberships. Employing users' assigned tag information together with co-occurrence data would facilitate recommendation quality. For this reason, we represent each community, c in latent topic space, where the topic is inferred from tags assigned to a community. To recommend personalized community to each user, we need to infer the latent topic preference of the user. In this study, we generate a query vector $\mathbf{q}_u$, for a given user u, in such a meaningful way that it is possible for the system to recommend a set of communities of value to the given user whether the user has any prior community membership or not.

3.3 Outline of the Proposed Algorithm

This study proposes a Latent Semantic Analysis (LSA) based community recommendation method to users. Communities are represented in latent topic space and users' assigned tags are also represented in latent topic space. Finally, communities are recommended based on user preferred topic and topic preferred by other users of the communities the user has joined. Outline of the community recommendation method is presented in the following steps:

1. Formulate community-tag matrix from user-community membership and user-tag preference data.

2. Apply BM25 weighting schema on community-tag matrix from step 1.

3. Apply Singular Value Decomposition (SVD) on weighted community-tag matrix to represent Communities in Latent topic Space.

4. Represent resulting matrices from step 3 in reduced dimensional space.

5. Formulate user preference matrix by including collective tag preference of other users of the target user's community with target user's personal tag preference.

6. Apply BM25 weighting schema on user tag preference matrix (Query) from step 4.

7. Infer users' latent topic preference from query of step 6.

8. Represent query with latent topic preference from step 7 in reduced dimensional space.

9. Calculate all communities' scores for target user from result of step 4 and 8.

10. Recommend top-*n* communities from sorted list of communities that are not already joined by the target user.

Following sections of this chapter describes verbosely the steps pointed out above with an illustrative example. Some of the sections encompass a description of several steps together.

3.4 Representing Communities in Latent Semantic Space

We begin by defining two matrices from information available in a social web system:

- User-tag frequency matrix $\mathbf{F}_{|U|\times|T|}$ where an entry $F_{u,t}$ represents the number of times that user u has used tag t.
- User-community membership matrix $\mathbf{M}_{|U|\times|C|}$ where an entry $M_{u,c}$ is 1 if user u has belonged to community c and 0 otherwise.

From **F** and **M**, we derive a new tag-community matrix **N**:

$$\mathbf{N}_{|T|\times|C|} = \mathbf{F}^T\mathbf{M} \tag{3.1}$$

where an entry $N_{t,c}$ represents the number of times that members of community *c* have used tag *t*. Consequently, our study represents each community as a (column) vector in the $|T|$ dimensional tag-space of the matrix **N**. Graphical representation of computing community-tag matrix is shown in Figure 3.2.

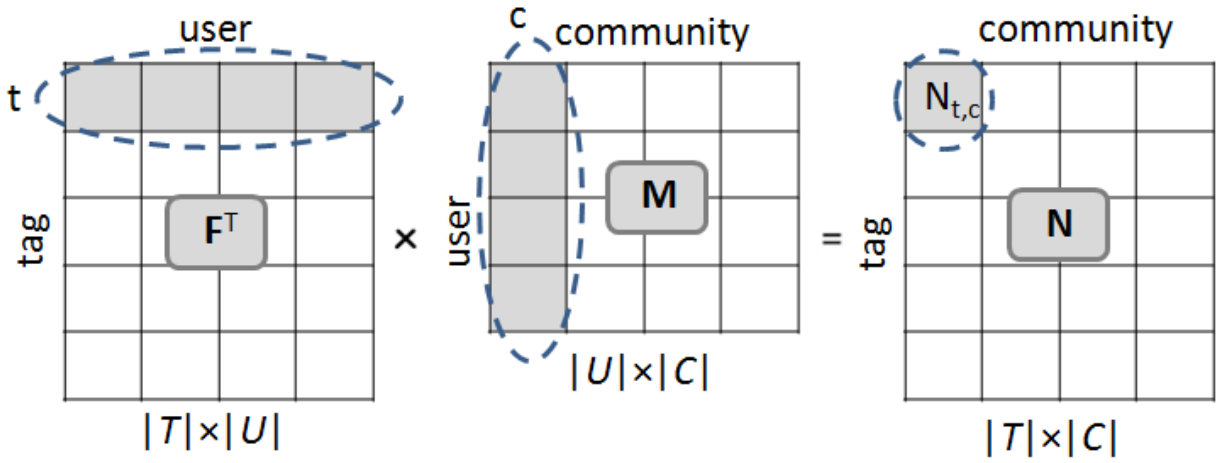

Figure 3.2: Computing the Community-Tag matrix **N**.

In general, various weighting techniques can be applied to both **F** and **N** so as to increase/decrease the importance of tags within/among users and communities, respectively. In our study, we employ the BM25 model [14]. In the weighting method, we treated tags as terms and communities or users as documents depending on matrices we used. In the BM25 method, a weight of a certain tag t in a particular community c was computed as:

$$w_{t,c} = \log\frac{|C|}{n_t}\cdot\frac{N_{t,c}\times(k_1+1)}{N_{t,c}+k_1\times\left(1-b+b\cdot\frac{|c|}{avg(|C|)}\right)} \tag{3.2}$$

where n_t is the number of communities in which tag *t* appears and avg(|*C*|) is the average number of tags in communities. Parameters k_1 and *b* are set to the standard values of 0.75 and 2,

respectively [14]. In an analogue fashion, we also computed a weight of a tag for a given user from the user-tag matrix **F**.

In practice, the matrices of users by tags and tags by communities are extremely sparse since users make frequent use of ambiguous and synonymous tags according to their personal tagging behavior. Additionally, users often use self-referential tags that could result in noise information. To deal with these issues, we exploit Latent Semantic Analysis (LSA), which was originally developed in the context of information retrieval [5]. LSA starts with the term-document matrix and maps both the terms and documents in a concept space. By mapping tags into concepts, LSA reduces noise arose from random choice of tags and synonymy problem. In our study, tags and communities represent terms and documents, respectively. We construct two matrices of reduced dimensionality from the original tag-community matrix **N**. The two constructed reduced matrices show the latent attributes of tags as reflected by their occurrences in communities, and of communities as reflected by the tags that occur within the communities. The tag-community matrix is analyzed by SVD (Singular Value Decomposition) to derive the latent semantic structure model. The first step is to apply SVD to the tag-community matrix **N** to reduce its dimensionality by keeping its first k singular values. Formally, SVD decomposes **N** into three matrices as follows [2]:

$$\mathbf{N}_{|T|\times|C|} = \mathbf{U}_{|T|\times|R|}\mathbf{S}_{|R|\times|R|}\mathbf{V}^{T}_{|R|\times|C|} \tag{3.3}$$

where **U** and $\mathbf{V}^{\mathrm{T}}$ are orthogonal matrices of $|T|$ by $|R|$ and $|R|$ by $|C|$, respectively. Singular matrix **S** is a $|R|\times|R|$ diagonal matrix, where $|R|$ is the rank of the original matrix **N**. **S** has the diagonal entries sorted in a decreasing order of their singular values. In the left matrix **U**, each tag is represented by a row vector and in the right matrix $\mathbf{V}^{\mathrm{T}}$, each community is represented by a column vector.

The original matrix **N** can be approximated by taking the k largest singular values of the matrix **S** such that $k < |R|$. Specifically, we can write as:

$$\tilde{\mathbf{N}}_{k} = \mathbf{U}_{k}\mathbf{S}_{k}\mathbf{V}^{T}_{k} \tag{3.4}$$

where $\mathbf{U}_k$ and $\mathbf{V}_k^T$ contain the first *k* columns of **U** and the first *k* rows of $\mathbf{V}^T$, respectively. The rank *k* approximation results in $\mathbf{V}_k$, dimensionality reduction for community vectors. This resultant $\mathbf{V}_k$ can represent inter-relationships among tags with respect to communities and can remove unneeded "noise" information. In our study, this derived k-dimensional representation is used for personalized community recommendations. In this latent space, communities which share frequently co-occurring tags exhibit a similar representation, even if they have no tags in common. An intuitive view of this approximation is depicted in Figure 3.3.

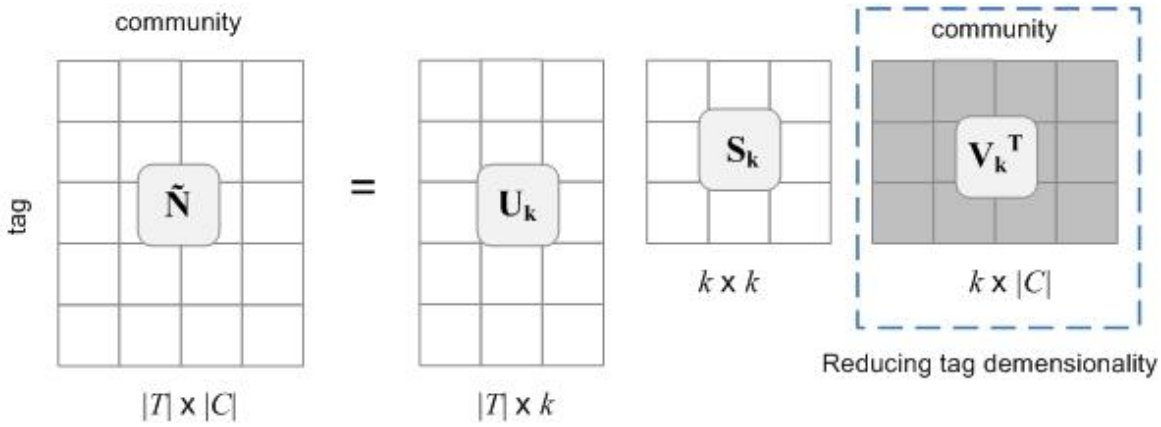

Figure 3.3: The rank k approximation of the tag-community matrix **N**.

3.4.1 Example: Representing Communities in Latent Semantic Space

To illustrate a simple example of building the latent community recommendation model, consider six users' tag assignments on four communities, shown in Figure 3.4.

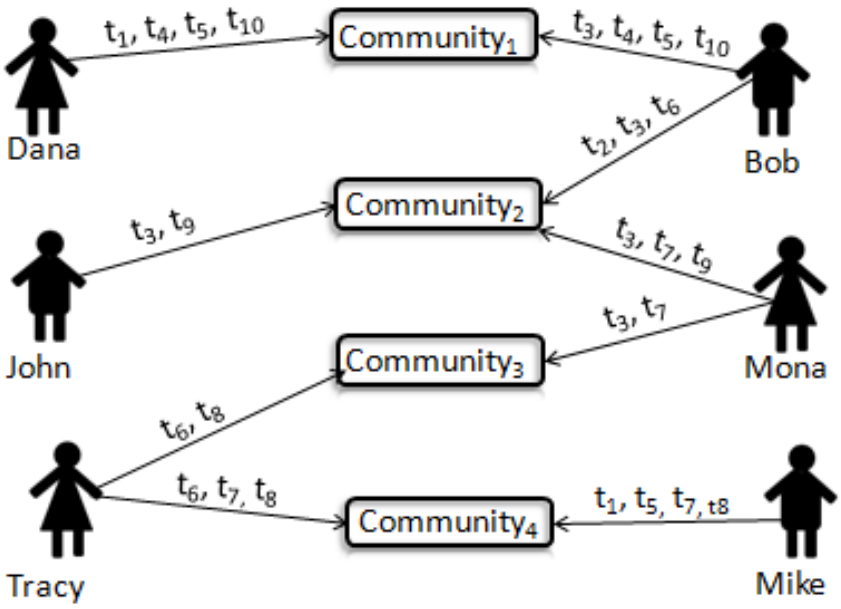

Figure 3.4: An example of tag assignments of six users to four communities.

In this example, users' tag assignments to communities mean that users assigned these tags to the items those are shared in these communities. We will follow this example to describe each step in the recommendation process through rest of the study. In the example, John has only one community membership and Bob has two community memberships. John is a member of only *community*$_2$ and Bob is a member of *community*$_1$ and *community*$_2$. John assigned tags t_3 and t_9 to community$_2$, whereas Bob assigned tags t_3, t_4, t_5, t_{10} to *community*$_1$ and t_2, t_3, t_6 to *community*$_2$. To derive user-tag matrix from users' tag assignments, we aggregate users' tags over communities. The aggregated user-tag matrix is shown in Table 3.2.

Table 3.2: An example of user-tag matrix, **F**.

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
Dana	1			1	1					1
John			1						1	

Tracy						2	1	2		
Bob		1	2	1	1	1				1
Mona			2				2		1	
Mike	1				1		1	1		

We can also easily derive user-community membership matrix, which shown in Table 3.3.

Table 3.3: An example user-community membership matrix, **M**.

	$Community_1$	$Community_2$	$Community_3$	$Community_4$
Dana	1			
John		1		
Tracy			1	1
Bob	1	1		
Mona		1	1	
Mike				1

Some social-networking sites provide tag-community information explicitly, but some other ones do not provide the information explicitly. So we need to derive tag-community matrix implicitly from user-tag matrix and user-community membership matrix. From the given example, we can derive tag-community matrix using Equation 3.1. Each entry in the tag-community matrix implies the number of times a specific tag is assigned by the member users of that specific community. For example, in $community_3$, tag count for t_3 is five, because t_3 is assigned once by John, twice by Bob and twice by Mona. The resulting tag-community matrix of our example is shown in Table 3.4.

Table 3.4: An example tag-community matrix, **N**.

	$Community_1$	$Community_2$	$Community_3$	$Community_4$
t_1	1			1
t_2	1	1		
t_3	2	5	2	
t_4	2	1		
t_5	2	1		1
t_6	1	1	2	2
t_7		2	3	2
t_8			2	3
t_9		2	1	
t_{10}	2	1		

We employ BM25 scoring on tag-community matrix by using Equation 3.2. To infer the latent topics of each community we employ SVD to the BM25 scored tag-community matrix. Applying Equation 3. 3 on BM25 scored tag-community matrix, the decomposed matrices are shown in Figure 3.5.

$$\begin{bmatrix} 0.082 & 0.000 & 0.000 & 0.050 \\ 0.082 & 0.045 & 0.000 & 0.000 \\ 0.165 & 0.225 & 0.054 & 0.000 \\ 0.165 & 0.045 & 0.000 & 0.000 \\ 0.165 & 0.045 & 0.000 & 0.050 \\ 0.082 & 0.045 & 0.054 & 0.100 \\ 0.000 & 0.090 & 0.082 & 0.100 \\ 0.000 & 0.000 & 0.054 & 0.150 \\ 0.000 & 0.090 & 0.027 & 0.000 \\ 0.165 & 0.045 & 0.000 & 0.000 \end{bmatrix} = \begin{bmatrix} 0.175 & 0.035 & 0.326 & 0.314 \\ 0.209 & -0.119 & 0.020 & 0.166 \\ 0.614 & -0.063 & -0.573 & 0.147 \\ 0.360 & -0.261 & 0.228 & -0.343 \\ 0.384 & -0.084 & 0.347 & 0.480 \\ 0.280 & 0.339 & 0.206 & -0.396 \\ 0.198 & 0.557 & -0.215 & -0.317 \\ 0.095 & 0.635 & 0.304 & 0.261 \\ 0.127 & 0.099 & -0.400 & 0.245 \\ 0.360 & -0.261 & 0.228 & -0.343 \end{bmatrix} \begin{bmatrix} 0.430 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.227 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.182 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.021 \end{bmatrix} \begin{bmatrix} 0.786 & -0.393 & 0.458 & -0.132 \\ 0.556 & 0.120 & -0.757 & 0.321 \\ 0.171 & 0.430 & -0.175 & -0.869 \\ 0.210 & 0.804 & 0.432 & 0.352 \end{bmatrix}$$

$\mathbf{N}_{|T|\times|C|}$ $\mathbf{U}_{|T|\times|R|}$ $\mathbf{S}_{|R|\times|R|}$ $\mathbf{V}^{T}_{|R|\times|C|}$

Figure 3.5: SVD result on scored tag-community matrix, **N**.

The scored tag-community matrix **N**, is decomposed to left orthogonal matrix **U**, where tags are represented by latent attributes, singular matrix **S**, consisting of singular values of matrix **N** and

right orthogonal matrix $\mathbf{V}^T$, communities are represented by latent topics. The rank of our example tag-community matrix is 4. So, the singular matrix contains four non-zero diagonal entries. Though, the last diagonal entry has a very low value.

The original tag-community matrix can be approximated by keeping only first *k* singular values. Appropriate selection of *k* is very important to approximate the original matrix because a small value for *k* may lose some important information; on the other hand, large *k* value may include noisy information. We consider *k*=3 for our example matrix, thus the original tag-community matrix is approximated from reduced dimensional left orthogonal matrix, singular matrix and right orthogonal matrix, as shown in Figure 3.6. By selecting *k*=3, tag dimensions of the original matrix is reduced from 10 to 3. In the right orthogonal matrix $\mathbf{V}_k^T$, each community is represented by three topics in the corresponding column vector.

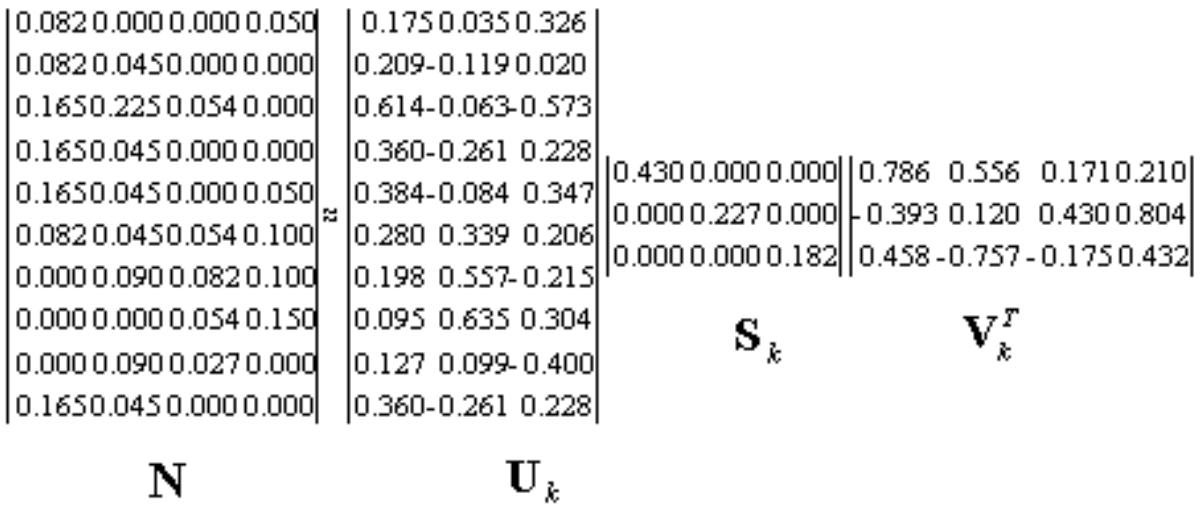

Figure 3.6: Reduced dimensional SVD result on scored tag-community matrix, **N**.

Subsequently, this reduced dimensional community and tag representation will be used for further calculation of recommendation.

3.5 Representing User Preferences in Latent Semantic Space

To identify communities that will be of interest to a given user, we first generate an implicit query that consists of a set of tags, potentially representative of his/her interests. The basic premise underlying the query is that the user is likely to prefer tags that have been previously used by him/her or by other users who have belonged to the same communities he/she has joined. For a given user *u*, a query is represented as a vector $\mathbf{q}_u$ in which a value of an entry, denoted $\mathbf{q}_u(t)$, is obtained by:

$$\mathbf{q}_u(t) = F_{u,t} + \sum_{c \in C(u)} N_{t,c} \tag{3.5}$$

where *C(u)* is the set of communities that user *u* has joined. Using the matrices defined in the previous section, we can express query vectors for all users in matrix form as:

$$\mathbf{Q}_{|T| \times |U|} = \mathbf{F}^T + \mathbf{N}\mathbf{M}^T \tag{3.6}$$

where each column vector in **Q** represents the corresponding column user's query. That is, the query vector **q** for a particular user is composed of his/her personal tags in addition to all tags of communities to which he/she has belonged. For example, if user u joined communities c_1, c_2, and c_3, then we add user *u*'s personal tags to all tags of communities c_1, c_2 and c_3.

For recommending communities to a given user, we capture how the tags contained in that user's query vector appear in communities on the latent semantic space. To this end, we transform a query vector $\mathbf{q}_u$ into a new reduced vector in the same *k*-dimensional space. Formally, the reduced vector for a given user *u* is given by

$$\tilde{\mathbf{q}}_u^T = \mathbf{q}_u^T \times \mathbf{U}_k \times \mathbf{S}_k^{-1} \tag{3.7}$$

where $\mathbf{S}_k^{-1}$ is the inverse of the singular matrix $\mathbf{S}_k$ [2].

3.5.1 Example: Representing User Preferences in Latent Space

In the step of user preference calculation, we consider user's personal preferences and preferences of other users of the same community that the target user belongs to. So the query submits for a user comprises of tags assigned by the user in the past and tags of other users of the communities, he has joined. In our given example, John's personal tags are t_3 and t_9. John is a member of only $community_2$. So, collective tags for John comprised of all tags assigned by all members of $community_2$. Bob and Mona are also member of $community_2$. So john's collective tag preference also consists of Bob's and Mona's tag preferences. We denote the tags assigned by all users who have belonged to the same communities; the target user has joined as accumulated tag- user. If John is our target user, then his collective tag preference is: $t_2 + 5t_3 + t_4 + t_5 + t_6 + 2t_7 + 2t_9 + t_{10}$. Accumulated tag-user matrix for all users is shown in Table 3.5.

Table 3.5: Accumulated user-tag matrix

	Dana	John	Tracy	Bob	Mona	Mike
t_1	1	0	1	1	0	1
t_2	1	1	0	2	1	0
t_3	2	5	2	7	7	0
t_4	2	1	0	3	1	0
t_5	2	1	1	3	1	1
t_6	1	1	4	2	3	2
t_7	0	2	5	2	5	2
t_8	0	0	5	0	2	3
t_9	0	2	1	2	3	0
t_{10}	2	1	0	3	1	0

Accordingly, relevant column vector of the accumulated tag-user matrix composed of that users' collective tag preference and relevant column vector of the tag-user matrix composed of that users' personal tag preference. We applied BM25 scoring to both accumulated tag-user matrix and tag-user matrix separately. The scoring helps to increase/decrease the effects of tags to users'

query. BM25 scored accumulated tag-user and tag-user matrices are shown in Table 3.6 and Table 3.7.

Table 3.6: Bm25 scored accumulated tag-user matrix.

	Dana	John	Tracy	Bob	Mona	Mike
t_1	0.046	0.000	0.046	0.023	0.000	0.082
t_2	0.046	0.030	0.000	0.046	0.026	0.000
t_3	0.092	0.152	0.091	0.161	0.182	0.000
t_4	0.092	0.030	0.000	0.069	0.026	0.000
t_5	0.092	0.030	0.046	0.069	0.026	0.082
t_6	0.046	0.030	0.183	0.046	0.078	0.163
t_7	0.000	0.061	0.229	0.046	0.130	0.163
t_8	0.000	0.000	0.229	0.000	0.052	0.245
t_9	0.000	0.061	0.046	0.046	0.078	0.000
t_{10}	0.092	0.030	0.000	0.069	0.026	0.000

Table 3.7: BM25 scored tag-user matrix.

	Dana	John	Tracy	Bob	Mona	Mike
t_1	0.389	0.000	0.000	0.000	0.000	0.344
t_2	0.000	0.000	0.000	0.272	0.000	0.000
t_3	0.000	0.429	0.000	0.545	0.482	0.000
t_4	0.389	0.000	0.000	0.272	0.000	0.000
t_5	0.389	0.000	0.000	0.272	0.000	0.344
t_6	0.000	0.000	0.652	0.272	0.000	0.000
t_7	0.000	0.000	0.326	0.000	0.482	0.344
t_8	0.000	0.000	0.652	0.000	0.000	0.344
t_9	0.000	0.429	0.000	0.000	0.241	0.000
t_{10}	0.389	0.000	0.000	0.272	0.000	0.000

Finally, target user's query is calculated using Equation 3.5. Final query for a specific user consists of the addition of corresponding column vector of BM25 scored tag-user matrix and corresponding column vector of BM25 scored accumulated tag-user matrix. If John is the target

user, he assigned t_3 and t_9 to $community_2$. So, in John's query, t_3 and t_9's scores from corresponding column vector of the tag-user matrix are added with the corresponding column vector's tags scores in scored accumulated tag-user matrix. So, John's personal tags are getting highest scores in his final query. Finally, John's query is: $0.030t_2 + (0.152+0.429)t_3 + 0.030t_4 + 0.030t_5 + 0.030t_6 + 0.061t_7 + (0.061+0.429)t_9 + 0.030t_{10}$. Query matrix for all users is calculated using Equation 3.6, presented in Table 3.8.

Table 3.8: Query matrix for all users.

	Dana	John	Tracy	Bob	Mona	Mike
t_1	0.435	0.000	0.046	0.023	0.000	0.425
t_2	0.046	0.030	0.000	0.318	0.026	0.000
t_3	0.092	0.581	0.091	0.706	0.664	0.000
t_4	0.481	0.030	0.000	0.341	0.026	0.000
t_5	0.481	0.030	0.046	0.341	0.026	0.425
t_6	0.046	0.030	0.835	0.318	0.078	0.163
t_7	0.000	0.061	0.555	0.046	0.612	0.507
t_8	0.000	0.000	0.881	0.000	0.052	0.589
t_9	0.000	0.490	0.046	0.046	0.319	0.000
t_{10}	0.481	0.030	0.000	0.341	0.026	0.000

So, we observe that, in the query, everyone's personal tags are getting highest preferences. This phenomenon helps in getting higher scores to users' preferred topics. On the other hand, the presence of similar users' tags in the query help to recommend communities on the similar topic. This type of query formation has advantages in cases when the target user has no personal tag, it can recommend communities of similar users' taste and if the target user has personal tags, but no joined communities, it can recommend communities based on personal tags.

To capture how the tags contained in the user's query appear in communities on the latent semantic space, we transform the query in the same reduced k (k=3) dimensional space using Equation 3.7. Reduced dimensional query for all users is shown in Table 3.9. Each column vector

of the reduced dimensional query matrix represents corresponding users' preference for latent topics.

Table 3.9: Reduced dimensional query matrix for all users.

	Dana	John	Tracy	Bob	Mona	Mike
$Topic_1$	1.595	1.116	1.197	2.291	1.466	1.023
$Topic_2$	-1.198	0.149	5.045	-0.663	1.631	3.033
$Topic_3$	2.665	-2.805	1.541	-0.433	-3.218	2.141

Reduced dimensional query vectors of each user will be used for calculating scores for all communities.

3.6 Top-n Community Recommendations

We are considering the task of recommending communities to a given user. This problem can be posed as a problem of ranking various communities in the order of the given user's interest in joining them. Thus, we describe a method of assigning scores to various communities in order to rank them.

Once we obtain a reduced query vector for a given user *u*, we speculate as to how much the user would prefer a particular community *c* by the dot product of two vectors:

$$\mathbf{r}_u(c) = \mathbf{v}_c^T \cdot \tilde{\mathbf{q}}_u^T \tag{3.8}$$

where $\mathbf{v}_c^T$ refers to the *c*th row (community) vector in the dimensionality reduction matrix $\mathbf{V}_k$. The ranking scores of all communities for user *u* can also be expressed in matrix form as

$$\mathbf{r}_u = \mathbf{V}_k \times \tilde{\mathbf{q}}_u \tag{3.9}$$

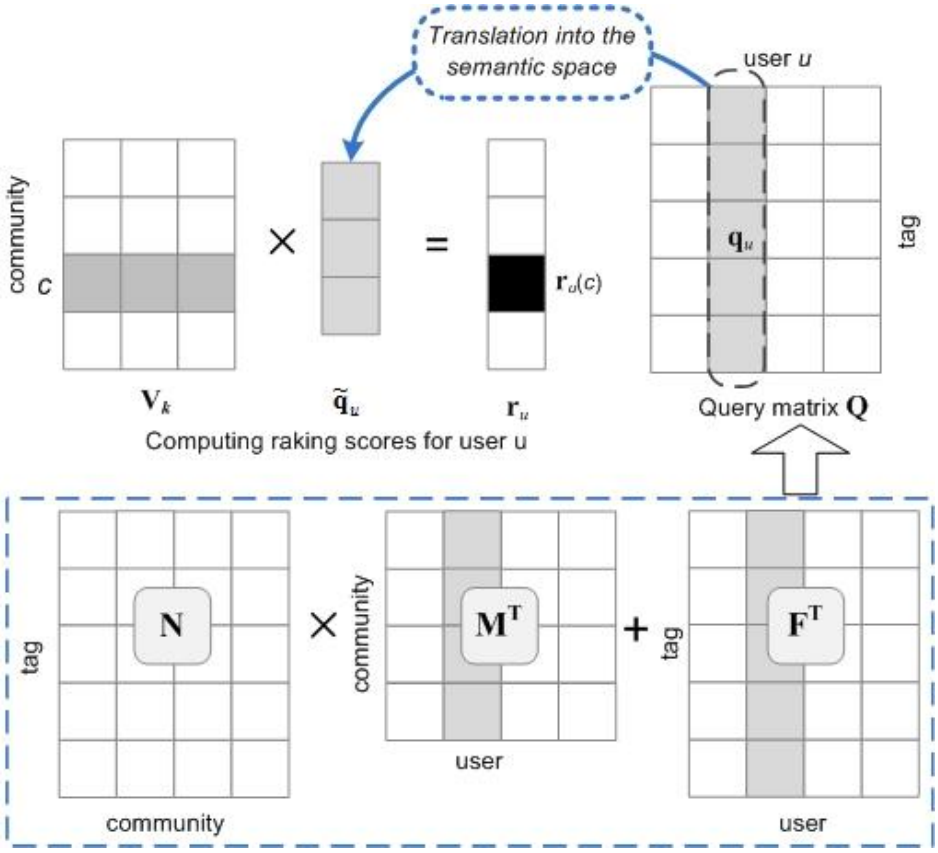

Figure 3.7: Computing ranking scores for recommending communities to a given user u.

In Equation 3.9, $\tilde{\mathbf{q}}_u$ provides the latent preference of user *u* toward tags in the semantic space. As mentioned earlier, the original query vector $\mathbf{q}_u$ reflects user *u*'s personal tags as well as his/her community members' tags. Accordingly, our approach computes ranking scores of communities in a situation where the user has not belonged to any communities, but has used some tags, as well as in a situation where the user has joined some communities, but has not previously used any tags. In addition, by representing communities and a query in the low-dimensional semantic space, our method can be beneficial not only in reducing the computational cost at the recommendation phase, but also in alleviating sparsity-related problems in recommender systems.

Once $\mathbf{r}_u$ is estimated with respect to user *u*, we identify a set of *n* ordered communities having the highest score values. Those communities not previously joined by user *u* are regarded as communities in which he/she is most likely to be interested, and are ultimately recommended to him/her. Figure 3.7 illustrates the process of computing communities' ranking scores using LSA.

3.6.1 Example: Top-n Community Recommendations

From our example, we can calculate a score for each community for any given user from his latent preference represented by the implicit query. John's preference for $community_3$ is speculated from the dot product of *community*$_3$'s row vector from $\mathbf{V}_k$ and john's query vector as: (0.171×1.116) + (0.430×0.149) + (-0.175×2.805) = 0.746. From John's personal tag usage tendency and other users of *community*$_2$'s tag usage tendency, *community*$_3$'s latent value is 0.746. Though John never joined *community*$_3$, our implicit query gives it a higher score by analyzing John's latent preference. Thus, there is a good chance that if $community_3$ is recommended to him, he would join it. Analogously, all users' preference for all communities is speculated from Equation 3.9. Calculated scores for each community for each user are shown in Table 3.10.

Table 3.10: Community score for all users.

	Dana	John	Tracy	Bob	Mona	Mike
$Community_1$	2.947*	-0.468	-0.338	1.863*	-0.965	0.592
$Community_2$	-1.275	2.761*	0.107	1.521*	3.448*	-0.686
$Community_3$	-0.709	0.746	2.104*	0.182	1.515*	1.104
$Community_4$	0.522	-0.856	4.971*	-0.239	0.230	3.576*

(* communities that are previously joined by the user)

So, as a result of John's query, *Community*$_3$ is recommended to him, as it is not already joined by John, and it has the highest score. In the same way top n highest scored communities that are not already joined by the target user, are recommended to him/her.

The algorithm of the total community recommendation process is presented below:

Algorithm: LSA based community recommendation method

Input:

F = User-tag frequency matrix
M = User-community membership matrix
k = Number of reduced dimensions
n = Number of requested communities

Output:
List of *n* communities sorted in descending order of score for each user

Main Procedure:

Generate tag-community matrix **N** from **F** and **M** using equation 3.1
BM25score(**N**)
Compute **U**, **S** and $\mathbf{V}^T$ by applying singular value decomposition on **N** using equation 3.3
Calculate $\mathbf{U}_k$, $\mathbf{S}_k$ and $\mathbf{V}_k^T$ by rank *k* approximation using equation 3.4

//Calculate query vector
Calculate query vector $\mathbf{q}_u$ for each user using equation 3.5
Represent each users' query in *k* dimensional space in $\tilde{\mathbf{q}}_u^T$ using equation 3.7

//Compute score vector $\mathbf{r}_u$ for each user
for u=1 to $N_{|U|}$ do
Calculate scores for all communities for user *u* in $\mathbf{r}_u$ using equation 3.9
end for

//Generate recommendation list
for u=1 to $N_{|U|}$ do
Remove the communities that user *u* has joined in the row vector $\mathbf{r}_u$
Sort $\mathbf{r}_u$ in the descending order of score
end for

3.7 Summary

We propose a dimensionality reduction based community recommendation method for users in social-networking sites. To recommend communities, we form an implicit query for each user which is representative of target users' latent topic preference as well as preferences of other users of the communities the user has already joined. Major advantage of the proposed method is, it can recommend communities if the target user has not joined any community before but assigned tags or target user never assign tag but has community membership. Also the dimensionality reduction approach helps to alleviate scalability. Though the method is presented only for recommending communities, it is applicable to other domains like item recommendation and user recommendation too.

Chapter 4

Experimental Evaluations

In this chapter, we empirically evaluate our recommendation approach and compare performance against a number of state of the art dimensionality reduction based recommendation methods. We present a brief description of the datasets we used for conducting experiments, a brief description of the evaluation metrics we used to measure the performance of the algorithms. Afterwards, we present results of various experiments, like, sensitivity of the algorithms on changing parameter *k*, experimental results of all algorithms at different levels of sparsity, experimental results for different user behavior and experimental results on some special conditions are presented in the subsequent sections. While presenting the results we have provided necessary discussion about the results alongside.

4.1 Datasets

We evaluated the performance of the algorithms with two heterogeneous datasets; one is CiteULike [51] dataset and other is Lastfm [52] dataset.

Table 4.1: Description of the datasets used in our experiments

	$\|U\|$	$\|T\|$	$\|C\|$	**F**	**M**	**N**
CiteULike	685	2,310	525	13,593	2,139	37,518
Lastfm	2,948	3,563	888	223,254	37,574	878,528

CiteULike is an online service for tagging, managing, and discovering scholarly references. Users of CiteULike can also create and join groups according to their research topics of interest. We downloaded the latest snapshot[1] of social tagging data and group membership data in June 2011. Subsequently, we refined those to conduct experiments that were more meaningful. Eventually, the cleaned dataset used in this study contained 2,139 non-zero entries in the user-community membership matrix **M** and 13,593 non-zero entries in the user-tag frequency matrix **F**. From both matrices, we generated the tag-community matrix **N**. In this dataset, an average user belonged to 3.1 communities while using 19.8 distinct tags.

On the other hand, Lastfm is a social music service which aggregates musical tastes of users where users can upload, share, tag their favorite artist's song. In March 2009, it claimed 30 million active users. In Lastfm, users can also form groups to share their musical tastes. We downloaded the social tagging data and group membership data of Lastfm in June 2011. Then we refined those to conduct experiments. Finally, the refined dataset contained 37,574 nonzero entries in the user-group membership matrix **M** and 223,254 nonzero entries in the user-tag frequency matrix **F**. From these two matrices, we generate tag-community matrix **N** which contains 878,528 nonzero entries. In Lastfm dataset, an average user belonged to 12.7 communities while using 75.7 distinct tags. Table 4.1 shows the statistics of the datasets.

Thus, the datasets we used for experimental evaluation are downloaded from two totally different social-networking sites providing different services. Also user behavior is different in the datasets. CiteULike dataset is very sparse where Lastfm dataset is comparatively denser. We evaluate the performance of our proposed method on these two heterogeneous datasets to draw a generalized conclusion about the performance of our method.

4.2 Evaluation Methodology

To evaluate the recommendation quality, we adopted two different measures: Mean Reciprocal Rank (MRR) and Mean Average Precision (MAP).

[1] http://www.citeulike.org/faq/data.adp

Mean Reciprocal Rank (MRR):

Mean Reciprocal Rank (MRR) is a method of evaluating an ordered list of results from a query where the results are ordered in descending order by probability of correctness. The reciprocal rank of the result of a given query is the multiplicative inverse of the rank of the first correct answer. The mean reciprocal rank is the average of the reciprocal ranks of all results for a number of queries. For our system evaluation, Mean Reciprocal Rank (MRR) is defined as follows [6]:

$$MRR = \frac{1}{|U|}\sum_{u=1}^{|U|}\sum_{c \in T_u}\frac{1}{r(c)} \tag{4.1}$$

where T_u is a set of test communities for user u and $r(c)$ refers to the rank of community c in T_u. Mean Reciprocal Rank evaluation metric is precision oriented and look at top n results. The higher the MRR value, the more accurately the algorithm ranks (recommends) relevant communities to users. In all our experiments, we fixed the number of recommendations to top 10 for MRR measure.

Mean Average Precision (MAP):

To evaluate the recommendation quality, we also adopted the Mean Average Precision (MAP), which measures a precision and recall at every position in a ranked list of recommended communities. Formally, MAP is defined as follows [14]:

$$MAP = \frac{1}{|U|}\sum_{u=1}^{|U|}\frac{1}{c_u}\sum_{n=1}^{c_u} P_n \times rel_n \tag{4.2}$$

where c_u is the total number of the test data for user u and P_n is precision at position n. rel_n is a binary variable that equals 1 if the recommended community at rank n is a relevant community (i.e., appear in the test data) and equals 0 otherwise.

MRR represents correctness of ranking of the recommended communities at top n and MAP represents both recall and precision aspect, and it is sensitive to entire ranking list.

4.3 Baseline Algorithms

To compare the performance of our approach, we took four other dimensionality reduction based approaches as baseline methods. We compared the results achieved by our approach with following four baseline methods:

– Probabilistic Latent Semantic Analysis (denoted PLSA):

In this approach, the user-community co-occurrence data is analyzed in a statistical technique [60]. The set of user-community pair (*c,u*) is assumed to be generated independently. A latent class variable *z* is introduced for each user-community pair, so that user *u* and community *c* is rendered conditionally independent. The resulting mixture model is written as:

$$P(c,u) = \sum_z P(c,u,z) = P(u)\sum_z P(c/z)P(z/u) \tag{4.3}$$

where *z* represents topic for a community. For recommending communities, after learning the model parameters the user-community relationship is inferred using Bayesian rules. The probability of a user u_i joining community c_j, $P(c_j/u_i)$ is calculated as [3]:

$$P(c_j / u_i) = \sum_z P(c_j / z)P(z / u_i) \tag{4.4}$$

– Combinational Collaborative Filtering (denoted CCF):

In this approach, each community has a multinomial distribution over topics and each topic has a multinomial distribution over users and tags, respectively. This approach is discussed in Chapter 2 in description of related study section in detail. In this approach, the probability of a user u_i joining community c_j, $P(c_j/u_i)$ is calculated as:

$$\begin{aligned} P(c_j \mid u_i) &= \frac{\sum_z P(c_j, u_i, z)}{P(u_i)} \\ &= \frac{P(c_j)\sum_z P(u_i / z)P(z / c_j)}{P(u_i)} \\ &\propto P(c_j)\sum_z P(u_i / z)P(z / c_j) \end{aligned} \tag{4.5}$$

Unlike the approach followed in article [3], we keep $P(c_j)$ in calculating community scores, because it is not a uniform prior for all users.

– Latent Dirichlet Allocation (denoted LDA):

This approach also discussed in chapter 2 related study section in detail. In this approach user-community co-occurrence data is only observed data. Unlike PLSA model, in this model two symmetric Dirichlet priors are added for per-user topic distribution and per-topic community distribution respectively. After learning the model parameters, user-community relationships are inferred using Bayesian rule.

– SVD-based Collaborative Filtering (denoted SVDCF):

A SVD based CF recommendation is presented in [11]. In SVD a user-community matrix is decomposed into three matrices **U**, **S** and $\mathbf{V^T}$, where **U** and $\mathbf{V^T}$ are orthogonal matrices and **S** is a diagonal matrix composed of all singular values of the root matrix sorted in decreasing order. By keeping only *k* highest singular values of the diagonal matrix and only *k* rows of **U** and *k* columns of $\mathbf{V^T}$, its dimensionality is reduced. Then a score for each user for each group is calculated from the product of $\mathbf{U}_k\mathbf{S}_k^{1/2}$ and $\mathbf{S}_k^{1/2}\mathbf{V}_k^T$. If $\mathbf{M_{u,c}}$ is the main user-community matrix then:

$$\mathbf{M}'_{u,c} = (\mathbf{U}_k\mathbf{S}_k^{1/2})_{u:}(\mathbf{S}_k^{1/2}\mathbf{V}_k^T)_{:c} \tag{4.6}$$

Each entry of matrix $\mathbf{M'_{u,c}}$ gives the score for a user u to join to a specific community c. We followed the recommendation method described in [15]. A number of communities are recommended to the target user based on highest scores and which are not already joined by the target user.

4.4 Experimental Setup

The experiments were designed to carry out the following procedure: we randomly selected 20% of each user's membership data from the entire dataset and subsequently used those as the test set

for them. The remaining 80% community membership data of all users were employed as the training set for recommending them communities. When a certain user had less than five membership data, we used one membership data as the test data. This procedure was repeated five times and thus the average results with standard deviations were reported. Next section describes the experimental results for approximating the parameter *k*, number of topics. In the subsequent experimental results, we denote our proposed recommendation method as LSA-Comb-Query (LSA-combined-Query).

In Latent Dirichlet Allocation (LDA) approach, we carried out 100 iterations to learn the model parameters for CiteULike dataset and 300 iterations to learn the model parameters for Lastfm dataset respectively. The default value for symmetric Dirichlet prior α, for per-user topic distribution is 50/*k* (*k* is the number of topics), and default value for symmetric Dirichlet prior β, for per-topic community distribution is 0.1 for both the datasets.

For CCF and PLSA, we used necessary number of iterations needed for convergence. In CCF approach, we only employed EM (Expectation Maximization) to infer the parameter values.

4.5 Experiments with *k* Approximations

The number of dimension is a critical factor for the effectiveness of the low dimensional representation [2]. Since our proposed recommendation method and all the other comparison methods are dimensionality reduction based, so selecting the optimal *k* value for all methods was very important for our experiment. Our intention was to select the number of dimensions that is small enough to avoid over-fitting and large enough to capture all the latent relationships. As there is no direct analytical way to find out the optimal value [5], so we need to determine the optimal value through experimental evaluation. In case of our method, we investigated how sensitive our performance was with regard to the number of tag dimensions, *k*. We expected that the value of *k* could be significant factors, affecting our recommendation quality. In case of other comparison methods *k* represents user dimensions. The optimal value for lower dimension can vary from dataset to dataset; it can be even different for different type of method. In this sub section we present, experimental evaluation of the datasets for all comparison recommendation methods using evaluation metric MRR and MAP.

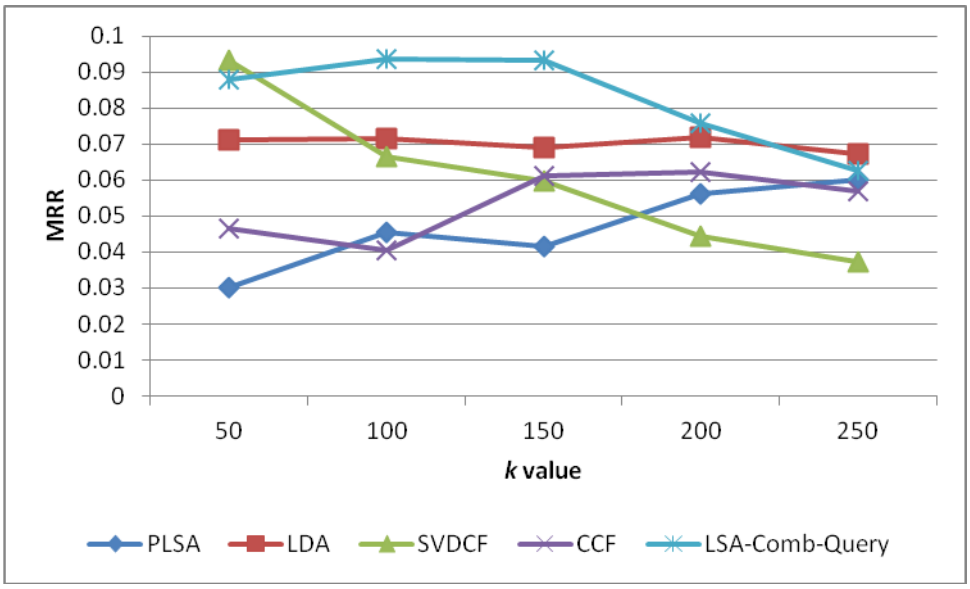

Figure 4.1: MRR values according to the variation of the parameter k value for the CiteULike dataset.

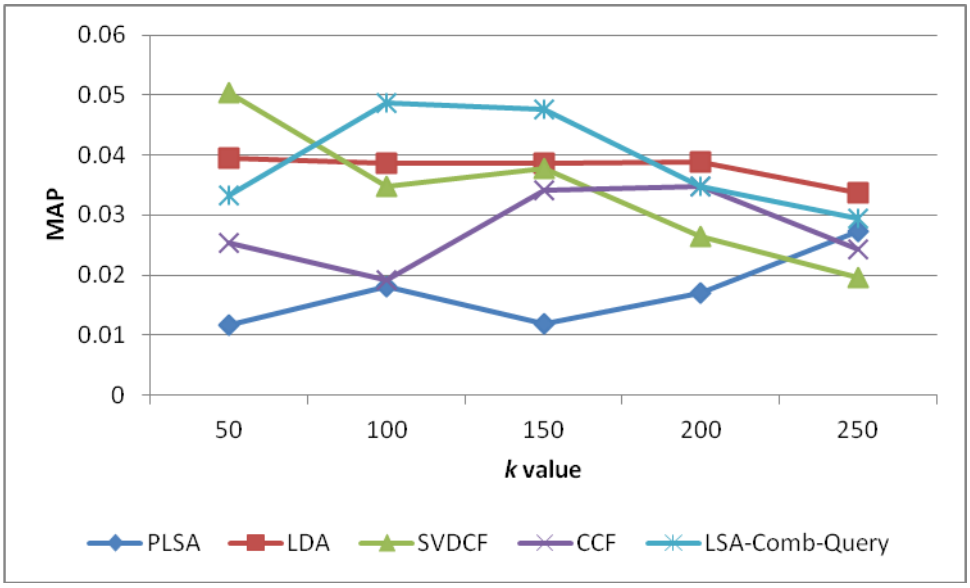

Figure 4.2: MAP values according to the variation of k value for the CiteULike dataset.

we measured the MRR and MAP values of CiteULike dataset by changing the *k* value from 50 to 250 with an increment of 50. Figure 4.1 and Figure 4.2 depicts the results of the experiment for MRR and MAP respectively. The x-axis on the graphs refers to the value of *k* and the y-axis refers to MRR at top 10 recommendations and MAP value accordingly. Our recommendation method is

denoted as LSA-Comb-query. For example, in case of our recommendation method, when *k* was set to 50 (i.e., $\mathbf{V}_{50}$), tag dimensions of **N** were reduced from 2310 (i.e., the total number of tags) to 50. The experimental results demonstrated that in case of our method, the quality of the recommendation peaks at *k*=100, suggesting that this value was optimal dimensionality of the latent semantic space. After this value, the quality deteriorated. While a reduction in *k* may remove much noise information, if *k* is too small, the resultant model $\mathbf{V}_k$ would lose important information for identifying suitable communities for users; thus, this results in lower recommendation quality. On the other hand, if *k* is too large, superfluous noise information can be included in $\mathbf{V}_k$, in turn leading to poor recommendation quality, as observed in Figure 4.1 and Figure 4.2. In Figure 4.1, higher MRR values stands for more communities from test dataset are recommended within top 10 recommendation list. In Figure 4.2, higher value for MAP stands for higher precision and recall at every position in the entire ranked list of recommended communities. In considering these points, for our method, we selected *k*=100 as the number of tag dimensions for $\mathbf{V}_k$ in the subsequent experimental evaluations. For other comparison recommendation methods *k* represents user dimensions. For example, when *k*=50 user dimension reduced from 685(total number of users) to 50. For the subsequent experiments, we selected that particular value for *k,* at which the recommendation quality peaks, for each particular comparison method. For SVDCF, we selected *k*=50, for PLSA *k*=250, for CCF *k*=150 and for LDA *k*=100, as the recommendation quality peaks at these values of *k* for the specific methods in case of CiteULike dataset.

Figure 4.3 and Figure 4.4 shows MRR at top 10 recommended communities and MAP results for varying *k* values for Lastfm dataset. From both MRR and MAP results, it is clear that our method performed best for *k*=20; that is when tag dimensionality reduces from 3,563 (total number of tags) to 20. LDA, CCF, PLSA also performed best for *k*=20 only SVDCF performed best for *k*=50. For LDA, CCF, PLSA and SVDCF, *k* represents user dimensionality as mentioned earlier. Surprisingly, all methods had better recommendation quality at very lower value of *k*. As *k* increases unexpected noise information included in $\mathbf{V}_k$, and results in poor recommendation quality.

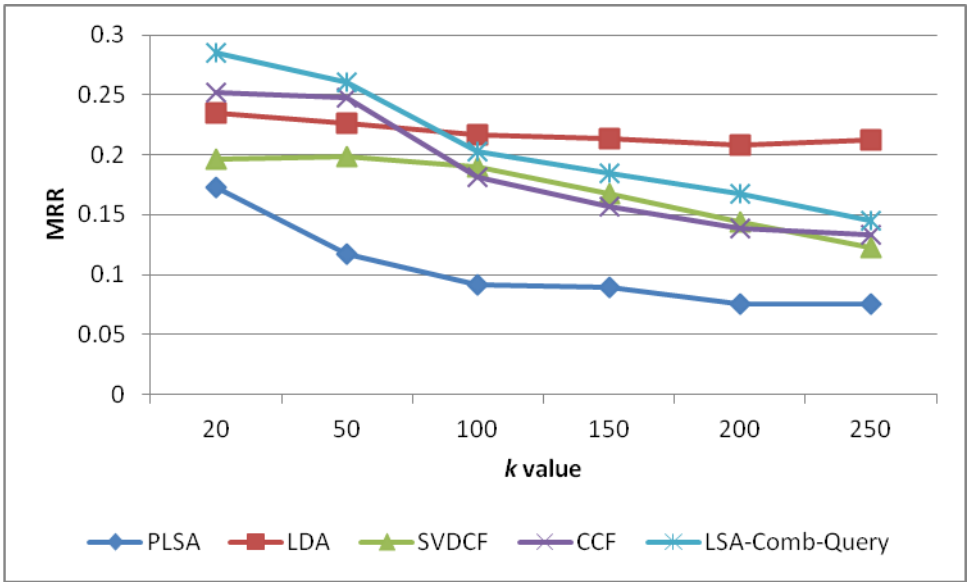

Figure 4.3: MRR result according to the variation of *k* value for the Lastfm dataset.

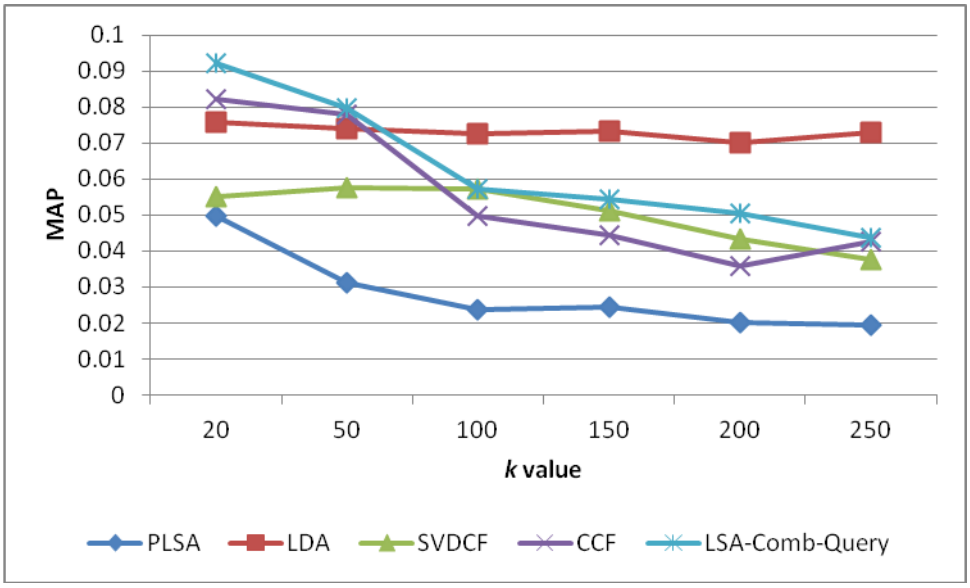

Figure 4.4: MAP result according to the variation of *k* value for the Lastfm dataset.

Experimental results show that, two datasets exhibit remarkably different behavior. Though Lastfm dataset has higher tag dimensions than CiteULike, it has better recommendation quality at lower value of *k*. We believe, the reason is the difference in user behavior due to different type of items shared in the sites. In CiteULike, scientific articles covering vast topics are shared within communities. In Lastfm shared music are not covering so many topics. So it required reduced

number of dimensions to cover enough topics for better recommendation quality. Furthermore, CiteULike is significantly sparser and accordingly has much less dependencies.

4.6 Effect of Tag Weights

In the following experiment, we examined the effect of weighting metrics on recommendation quality. We applied two different weighting methods to the tag-community matrix **N** and the user-tag frequency matrix **F**: the TF-IDF weighting method and BM25 weighting method. For comparison purposes, we also reported the result when using **N** and **F**, themselves without using any weighting schema. In the weighting methods, we treated tags as terms and communities or users as documents depending on matrices we used. In the TF-IDF method, we determined the frequency of a tag in a specific document (a user or a community) compared to the inverse proportion of that tag over all documents (all users or all communities) [12]. The BM25 method is described in section 3.4.

Table 4.2: MRR and MAP values with standard deviation according to different weighting methods (CiteULike dataset).

k=100	Tag Freq.	TF-IDF	BM25
MRR	0.086 ±0.003	0.089±0.002	0.093±0.008
MAP	0.048± 0.002	0.047± 0.002	0.047± 0.004

Table 4.2 shows the MAP and MRR results of our method on CiteULike dataset. From the result, we observed that the TF-IDF weighing schema did not contribute to the improvement on MRR and MAP compared to the simple tag frequency case. When the BM25 weights were used, the MRR value slightly improved, but difference appeared comparatively insignificant.

4.7 Performance at Different Level of Sparsity

In this section, we evaluated our LSA-based approach in comparison with four other dimensionality reduction based methods, LDA, PLSA, CCF and SVDCF. We evaluated the performance of the methods at different levels of sparsity. We randomly selected 20% data from each user's community membership data and used the remaining 80% data as training data for sparsity level 1. In level 2, we randomly withheld 20% data from training data of level 1 and used the remaining data as level 2 training data. In level 3, we again withheld each user's 20% community membership data from level 2 training data and used the remaining data as level 3 training data. For all the three sparsity levels, we used the same 20% withheld data of level 1 as test data. Note that since many of the users in CiteULike dataset joined a small number of communities, as a result training membership data were very sparse. Hence for CiteULike dataset, we experimented at three levels of sparsity. Since users in Lastfm have more community membership than CiteULike, so for Lastfm, we experimented on four levels of sparsity.

Figure 4.5 shows the results of MRR obtained via the five methods on CiteULike dataset at three sparsity levels. Our method is denoted as LSA-Comb-query. As can be seen from the graph, our method exhibited the best performance at sparsity level 1 and sparsity level 2. That is, the LSA-based method provides more desirable communities with a higher rank in the recommended community set, and thus can make better recommendations than the other methods.

Though all methods are based on dimensionality reduction, LDA, PLSA, SVDCF reduced user dimensions to infer the latent topics of the communities. At sparsity level 1, training data contained enough membership information to infer users' latent preference from membership data. Hence SVDCF performs similar as LSA based proposed method. However, as sparsity increases, training data became too sparse to infer users' latent preference from membership data, consequently, SVDCF's performance decreases drastically. LDA and PLSA both used statistical approaches to find model parameters. At level 1 and level 2, LSA based approach outperforms LDA and PLSA. But at sparsity level 3 training data turn into too sparse to infer latent semantics from decomposed matrices. So, at level 3 statistical approaches are doing better than the proposed

one. Though CCF employed both user-community affiliation data **M** and Tag-frequency matrix **F**, it results in a low performance for CiteULike dataset.

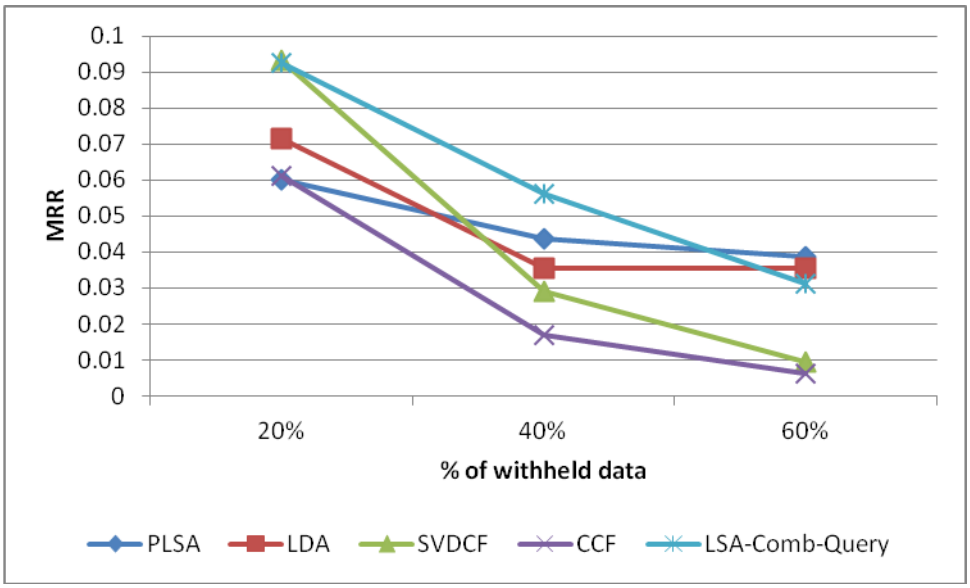

Figure 4.5: MRR value at different sparsity levels for CiteULike.

Figure 4.6 shows that MAP values of the comparison methods also exhibits similar performance. At sparsity level 1, SVDCF outperformed our method. But SVDCF's performance degraded more rapidly than our method as level of sparsity increases. MAP result of our approach is consistent with MRR result shown in Figure 4.5. That means, LSA based proposed method not only ranks test communities at higher position also it shows better MAP through the entire ranking list.

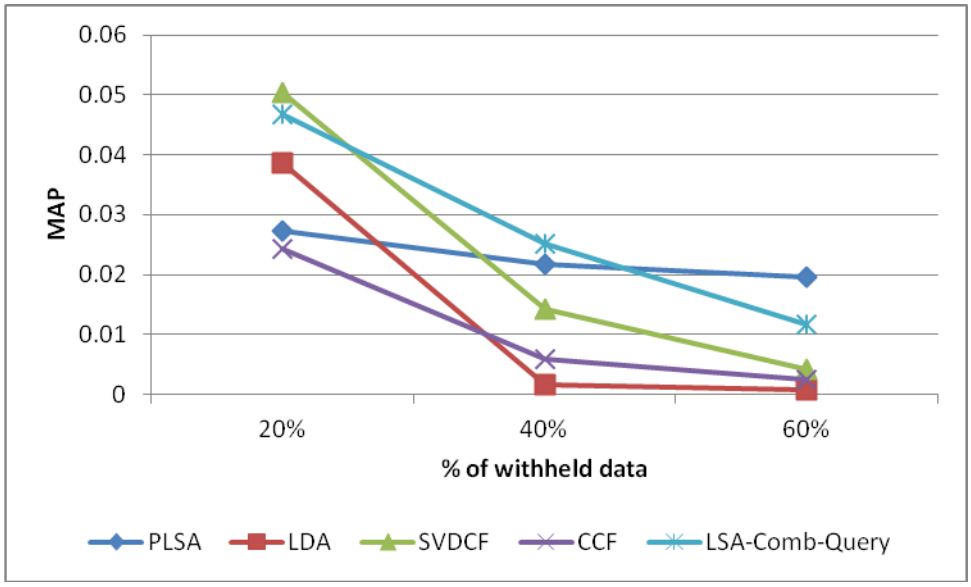

Figure 4.6: MAP value at different sparsity levels for CiteULike.

Figure 4.7 and Figure 4.8 shows MRR and MAP results of LSA based proposed method with four other comparison methods at four different sparsity levels for Lastfm dataset. At first three levels of sparsity, our proposed method outperforms all other comparison methods.

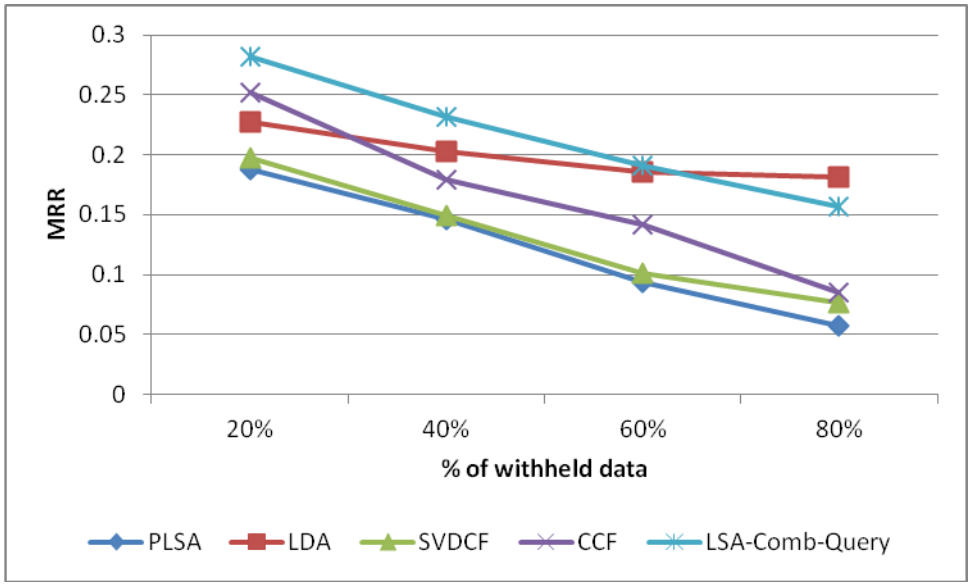

Figure 4.7: MRR values at different sparsity levels for Lastfm.

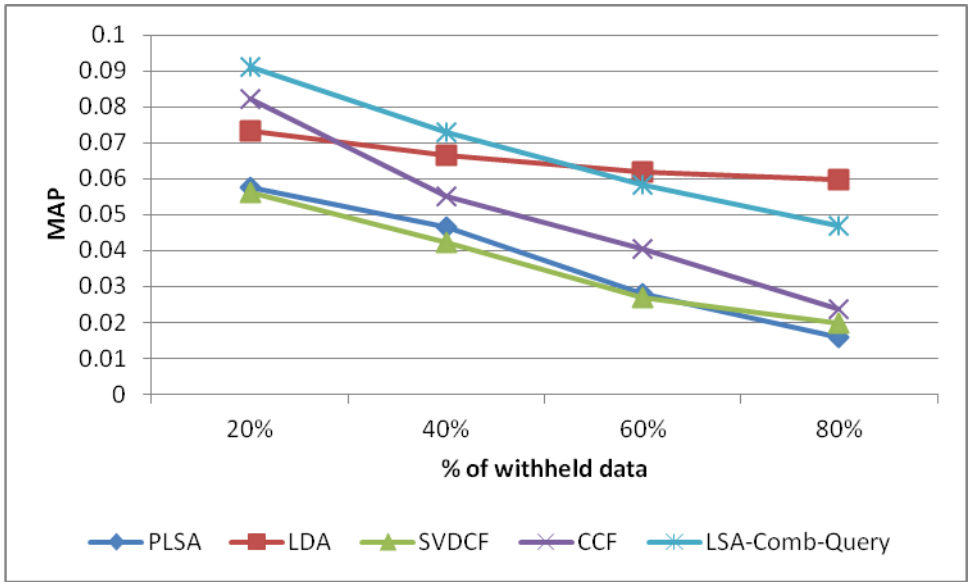

Figure 4.8: MAP values at different sparsity levels for Lastfm.

The above results of our proposed method not only confirm more desirable communities at higher ranks in the recommended community set but also proof better precision and recall through the entire recommendation list in comparison with other methods. Better recommendation quality of our proposed method proofs the superiority of our method over others.

Although all methods exploit dimensionality reduction, our proposed method identified more suitable communities with a higher rank in a recommended list than did other methods. The main difference between other comparison methods and our method is that other comparison methods infer latent user preferences from user-community matrix **M**, whereas our proposed method did the tag-community matrix **N**. In our method, a community was represented as a *k*-dimensional topic (tag) space instead of *k*-dimensional user space. This latent topic representation indeed helped improve the recommendation quality. Also our implicit query consisting of users' personal tag preference and collective tag preference of other users of the communities that target user joined, helped to rank users' more desirable communities in higher positions. This comparison experiment clearly confirmed the benefits of our approach. CCF shows better performance comparing LDA, PLSA and SVDCF in case of Lastfm dataset. It proofs the benefits of tag usage. Still our proposed LSA based method outperforms CCF. Though in CiteULike case SVDCF performed slightly better than our proposed method but in Lastfm case SVDCF performed the worst.

At the last sparsity level, our method's performance decreases because the training data contains very less information to infer acceptable result from that. Statistical approaches show better performance at this level.

The evaluation results of our proposed method on two heterogeneous datasets proof the superiority of LSA based proposed recommendation method and it is not sensitive to any particular type of dataset.

4.8 Performance for different user behavior

In this section, we evaluate the performance of our proposed recommendation approach comparing with other recommendation methods on the basis of different user behavior. We analyze performance of the methods from two different behavioral aspects. Firstly, from users' tendency to join communities and secondly, from users' tendency to assign tags. Following two sub-sections present the analysis of results from two aspects.

4.8.1 Performance for Varying Number of Joined Communities

In this section, we evaluate performance of our proposed method comparing with other methods based on different user types. We divide users based on their number of joined communities. Figure 4.9 and Figure 4.10 shows the graphical representation of CiteULike dataset and Lastfm dataset as a distribution of the number of communities per user respectively.

We can observe that users' behavior is totally different in these two datasets. In CiteULike, more than half of the total users are very inactive and having only two joined communities. On the other hand, users in Lastfm are comparatively more active. Even some users have more than 25 joined communities in lastfm. Based on the number of joined communities by each user, we divide users into three groups: cold user, medium user and active user. Since users show diverse behavior in different sites, thus definition of different user group is also different for different datasets. For CiteULike dataset, we define users having less than or equal two joined communities as cold user, users having greater than two and less than four joined communities as medium user and users

having greater than four joined communities as active user. On the other hand, for Lastfm dataset we define cold users having less than or equal five joined communities, medium users having six to 25 joined communities and active users having greater than 25 joined communities. For this experiment, we used sparsity level 1 data of both datasets.

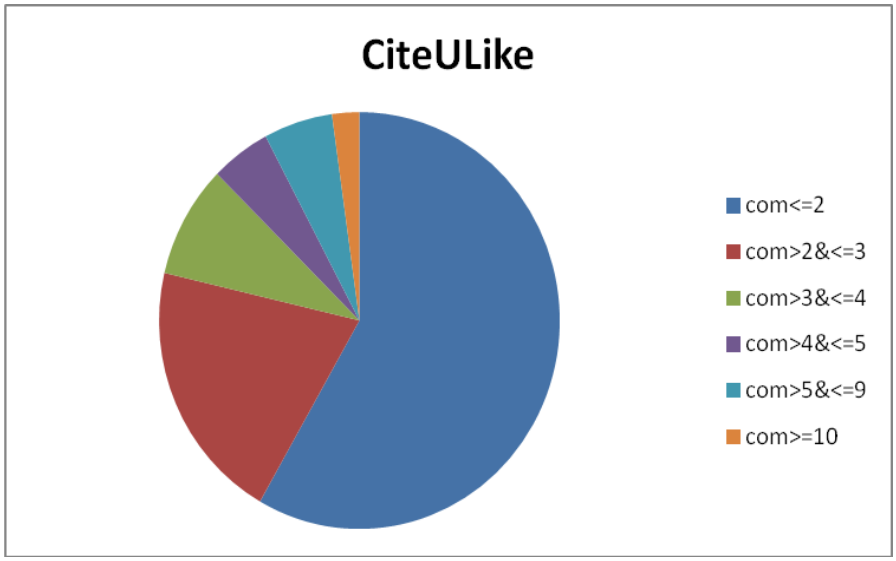

Figure 4.9: Distribution of the number of communities per user on the CiteULike dataset.

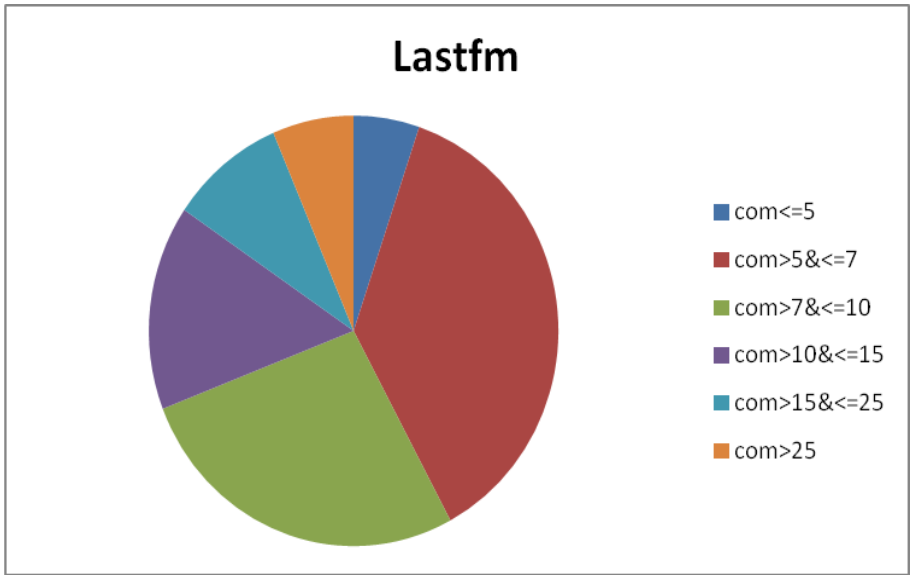

Figure 4.10: Distribution of the number of communities per user on the Lastfm dataset.

Figure 4.11 and Figure 4.12 shows MRR and MAP results of CiteULike dataset based on users' total number of joined communities.

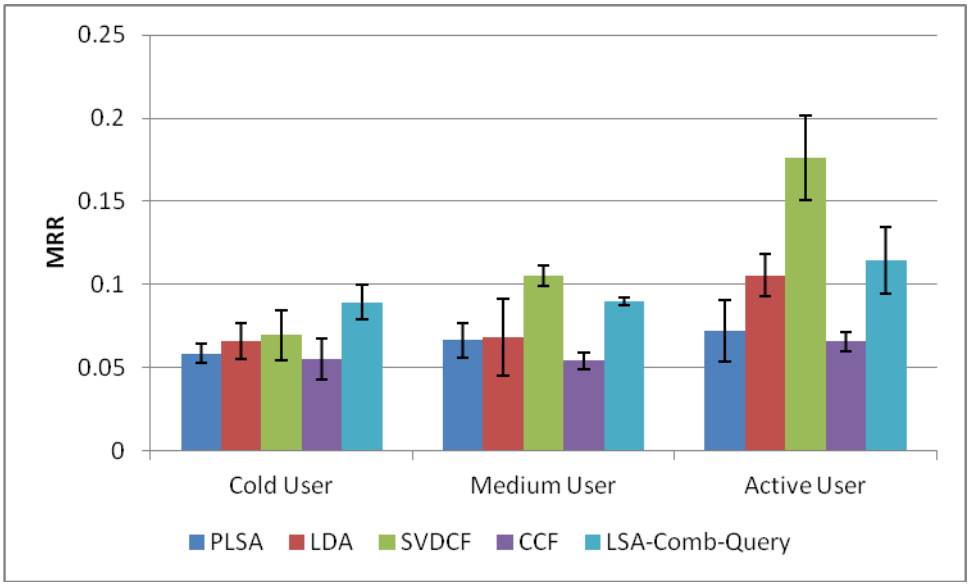

Figure 4.11: MRR result for different type of user (CiteULike).

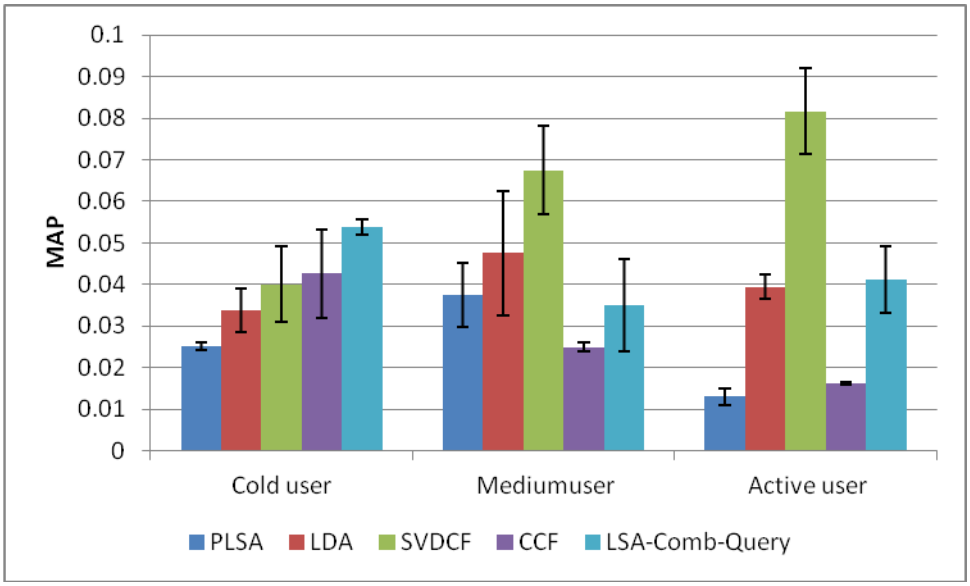

Figure 4.12: MAP result for different type of user (CiteULike).

We observe that based on different user types MRR and MAP results also varies. For CiteULike dataset, as the number of joined communities by users increased, SVDCF performed better. But for cold users, LSA based our proposed method shows best performance for both MRR and MAP.

In Lastfm dataset, user behavior is different than that of CiteULike; users are more active in Lastfm than users in CiteULike. Our proposed method shows best performance for all types of user in Lastfm. LSA based proposed method performs best for both (MRR and MAP) evaluation metric that confirms the best recommendation quality of the method. Figure 4.13 and Figure 4.14 shows MRR and MAP results for Lastfm dataset for different user types.

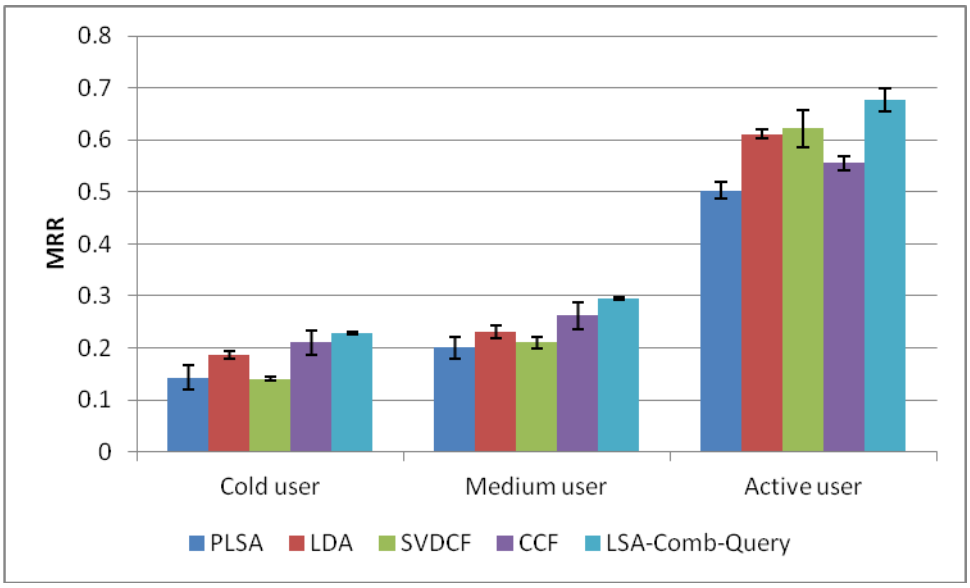

Figure 4.13: MRR result for different type of user (Lastfm).

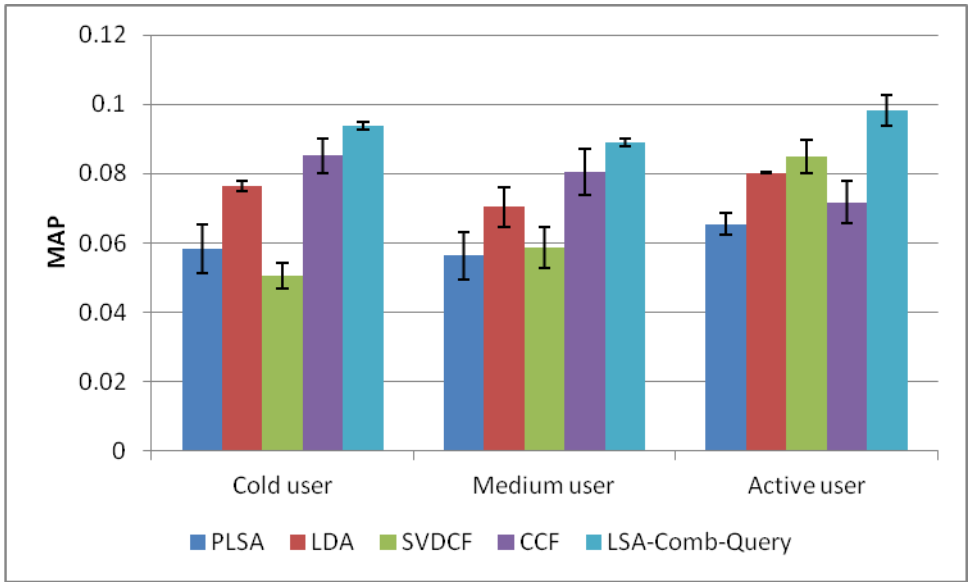

Figure 4.14: MAP result for different type of user (Lastfm).

When users' have more joined communities then all methods are benefited. But recommending cold users is challenging for every method. Cold users have a small number of joined communities, so it is difficult to infer their preference from only community membership information. In our proposed method, we represent communities in latent topic space and to infer user preferences, we employed a special query. Our method could recommend communities of value to cold users, as our implicit query vector consists not only of a given user's personal tags, but also of like-minded users' tags. As demonstrated in the MRR and MAP results, such an implicit query was concise and stable enough to characterize a particular user's preferences.

4.8.2 Performance for Varying Number of Assigned Tags

In this section, we evaluate the performance of our proposed recommendation method for varying number of user assigned tags. Since all comparison methods are not employing tag information, so we are not considering results from all methods here. We compare the performance of our approach only with CCF approach, since it is also employing tag information.

Depending on users' nature, number of assigned tags varies from user to user. Figure 4.15 and Figure 4.16 depicts the graphical representation of CiteULike and Lastfm dataset as a distribution of the number of assigned tags per user.

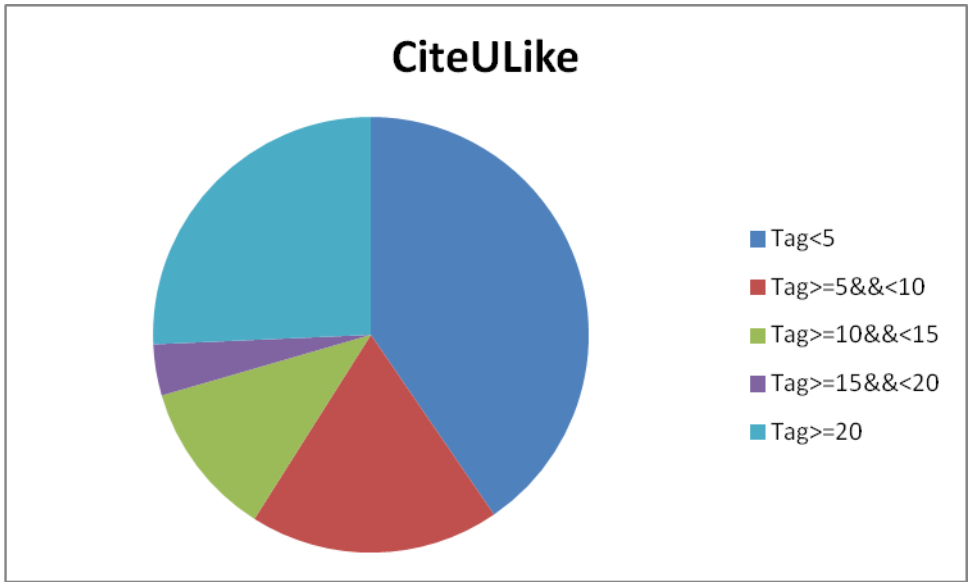

Figure 4.15: Distribution of number of assigned tags per user on the CiteULike dataset.

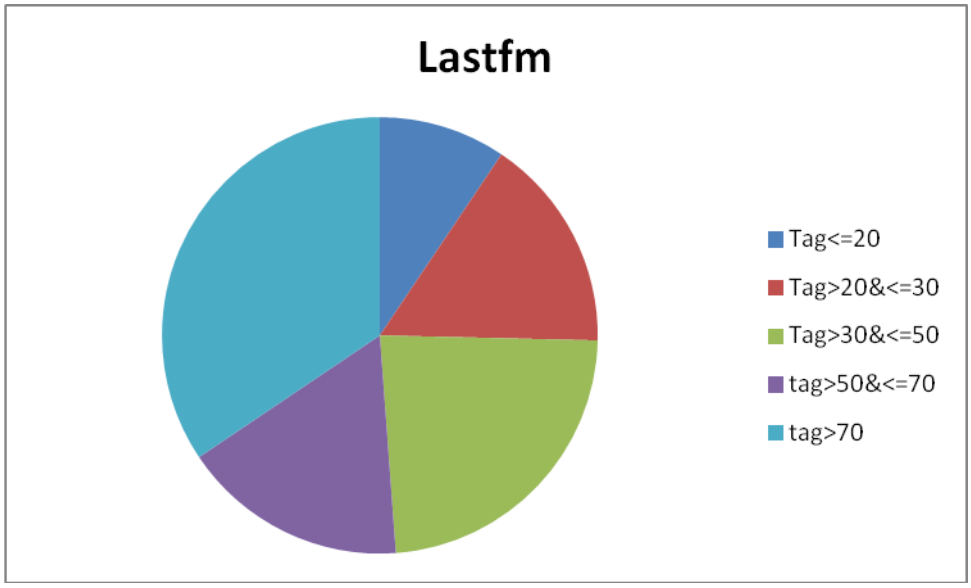

Figure 4.16: Distribution of number of assigned tags per user on the Lastfm dataset.

Like previous section, we again divide total users into three categories namely: cold user, medium user and active user. But in this section, the definitions of different categories are different than the previous ones. For CiteULike dataset, we define users having less than five assigned tags as cold user, users having more than or equal five tags and less than 15 tags as medium user, and users having more than or equal 15 assigned tags as active user. As user behavior in Lastfm is different than that of CiteULike, so definitions of the categories are different in Lastfm. In Lastfm,

users having less than or equal 30 assigned tags as cold users, users having more than 30 and less than or equal 70 assigned tags as medium user and users having more than 70 assigned tags as active user.

Figure 4.17 and Figure 4.18 portrays the performance of CiteULike dataset for evaluation metric MRR and MAP respectively.

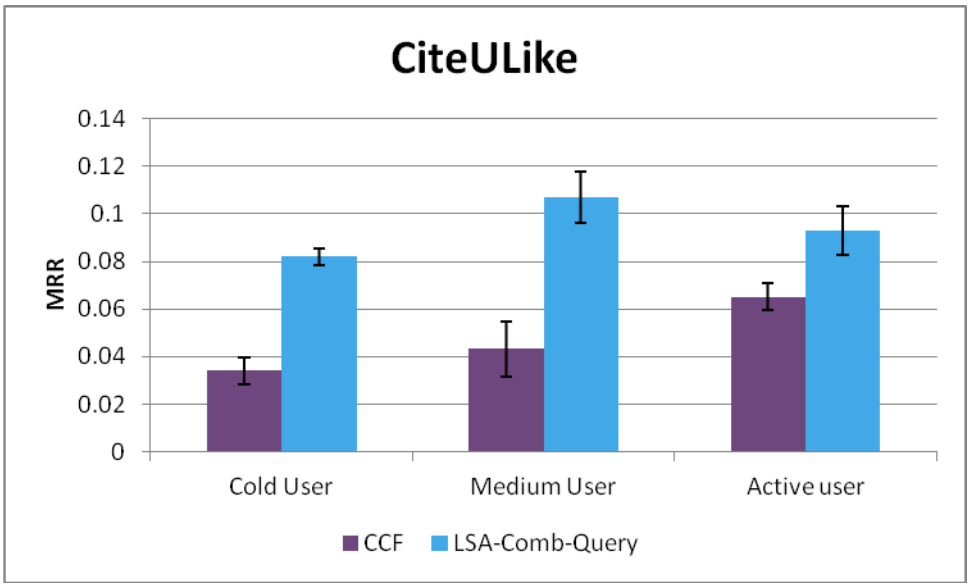

Figure 4.17: MRR result for different user type according to users' tag usage pattern for CiteULike dataset.

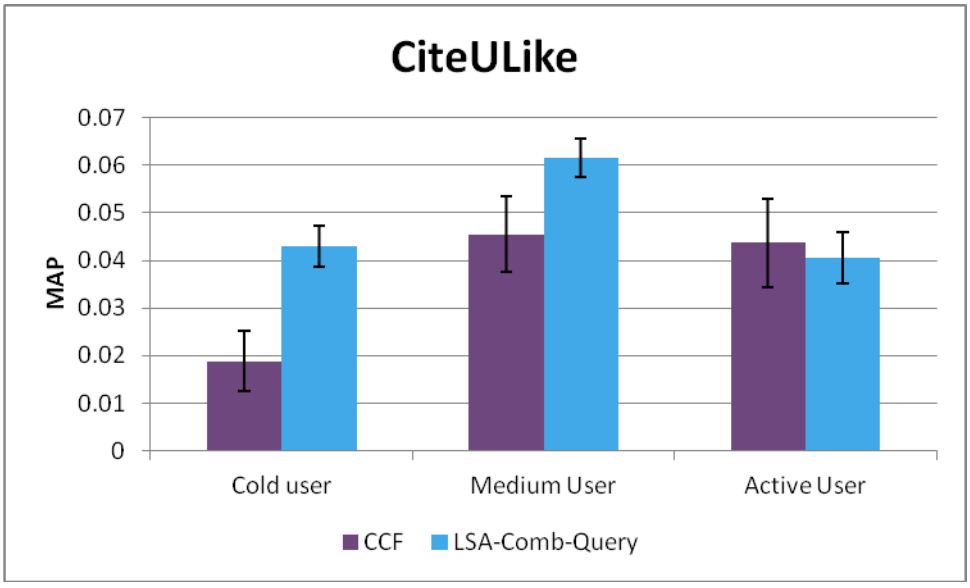

Figure 4.18: MAP result for different user type according to users' tag usage pattern for CiteULike dataset.

As can be seen from Figure 4.17, our proposed method outperforms CCF approach for all categories of users in the sense of MRR result for top 10 recommended communities. Performance increases for medium users than that of cold users. But for active users, performance degrades than medium users. We guess that active users' assigned tags encompass diverse topics that make it difficult to infer the right topic of preference. From Figure 4.18, we observe that though our proposed approach outperforms CCF for cold users and medium users, for active users, CCF has better MAP result than that of our proposed approach.

Figure 4.19 and Figure 4.20 presents MRR and MAP results for Lastfm dataset respectively. In case of Lastfm, our proposed approach outperforms CCF approach for all categories of users. Though in Lastfm case number of user assigned tags is more than that of CiteULike, but in Lastfm assigned tags encompass fewer topics. Like CiteULike, in Lastfm, MRR and MAP results for active users slightly decreases than medium users. As active users' assigned tags encompass a vast number of topics, performance degrades.

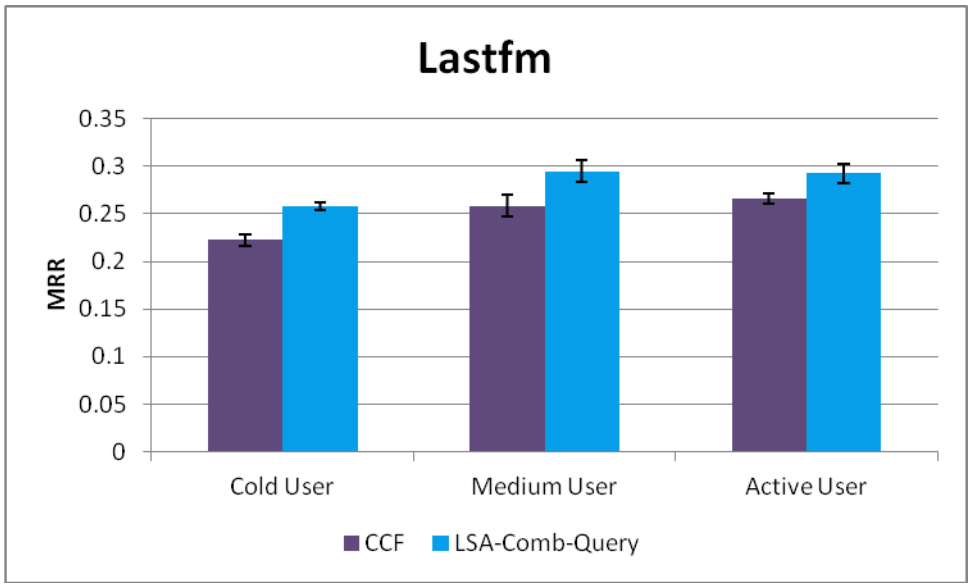

Figure 4.19: MRR result for different user type according to users' tag usage pattern for Lastfm dataset.

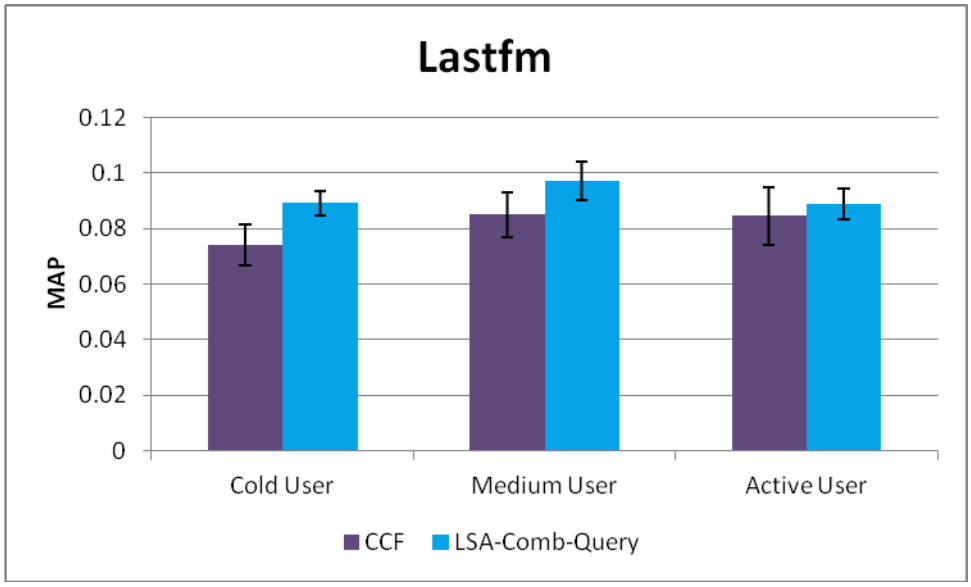

Figure 4.20: MAP result for different user type according to users' tag usage pattern for Lastfm dataset.

We observed that, overall performance of our proposed recommendation method is better than the other method. Our implicit query composed of both personal tags and collective tags of user joined groups helps to congregate users' latent preference. Mainly cold and medium users are benefited by our implicit query. MRR and MAP results of the datasets confirm better recommendation quality of our proposed approach.

4.9 Performance for users having no community membership

In this section, we evaluate the performance of our proposed recommendation approach for more complicated situation, when a target user has not joined any communities. To recommend communities to this type of user is a great challenge for all recommendation approaches. Recommendation approaches, which provide recommendation solely by user-community co-occurrence analysis would fail to recommend any community in this situation. LDA, PLSA, SVDCF and CCF could not recommend for users having no membership information. Because LDA, PLSA and SVDCF represent communities in user space, they could not recommend communities for such users. Though CCF approach employed user-tag information, it failed to recommend communities to such users. CCF employed user-tag information to infer topics for communities only.

In this experiment, we randomly select a number of users and withheld all the joined communities of these users. Then we used the remaining data as train data and evaluate the performance of the algorithm for users whose joined communities were withheld. For CiteULike dataset, we withheld 100 users all joined communities and used them as test data. For Lastfm dataset we randomly select 500 users and withheld their community membership data and used them as test data. Table 4.3 shows the MRR and MAP result for both datasets with standard deviations.

Table 4.3: Performance for users having no joined communities.

	MRR	MAP
CiteULike	0.02397±0.00118	0.05854±0.00138
Lastfm	0.12278±0.00097	0.08945±0.00126

This experiment clearly shows the advantages of the proposed method. Results shown in Table 4.3 obtained for users who have no joined communities. Though they have no user-community co-occurrence data, our method is able to infer their preferable communities. Two special steps of our recommendation algorithm help to recommend in such a complex situation. We represent communities in latent topic space instead of user space. And our implicit query is representative of users' preferable topic. These representations help to infer communities of user's preferable topic though the user has no prior community membership.

4.10 Summary

In this chapter, we evaluated our proposed recommendation method in comparison with four other dimensionality reduction based recommendation methods. We evaluated the results with two different evaluation metrics. We also verified the results for different conditions and from different perspectives. Most of the cases, our proposed recommendation method outperformed other comparison methods. Specifically, our proposed recommendation method showed better performance for recommending communities to cold users, which is a challenge for all other methods. Also our method shows significant performance for recommending users with no prior community membership, where other methods depending on co-occurrence data failed. Community representation in latent tag space and implicit query consists of users' personal tag

preference and collective tag preference of other users of the users' joined communities facilitated our proposed recommendation method to perform better in challenging conditions. The experimental results proof the superiority of the proposed method in recommending communities over other methods.

Chapter 5

Conclusion and Future work

In this chapter, we briefly summarize our contribution to the community recommendation problem in conclusion section. We briefly focus on the elucidation of the problems in existing systems stated in introduction. In the future work section, we stated some directions to possible future research to this problem.

5.1 Conclusion

In social Web systems, communities are becoming widely used as a way of sharing rich information and media content. In this paper, we presented a new method for modeling communities via user-generated tags and recommending communities of value to individual users in the latent semantic space.

The major advantage of our approach is that it has the ability to recommend communities regardless of whether users have explicitly joined communities. It can also take advantage of dimensionality reduction to alleviate the sparsity-related limitations. We evaluated performance of our method with two totally different domain datasets (CiteULike and Lastfm) and compared the performance of our method with four different state of the art recommendation methods. For measuring performance, we evaluated all results with two standard performance measurement approaches MAP and MRR. The experimental evaluation with the CiteULike and Lastfm dataset

clearly demonstrates the benefits of our LSA-based approach, which achieves improvements in both MAP and MRR over existing alternatives in most of the cases. Specifically, for recommending communities to cold users, our method showed significant improvements over existing alternatives. Also, our method is able to recommend in a very complex situation when a user has no explicit community membership. Evaluation with different user behavior (in different domain dataset, users' tag usage pattern is different according to type of items e.g., scientific articles in CiteULike and music in Lastfm) also proofs the benefit of our method for recommending communities to users.

5.2 Future Work

There are several directions for future work in this research. Firstly, we plan to employ some tag cleaning process to reduce noisy tags. Secondly, in our method, very active users' topic of preference might divert due to collective tags of all users of his joined communities. To handle this, we can weight collective tags from other users and users' own tags unequally and the value of the weighting parameter might vary according to user's activeness. Thirdly, instead of using binary valued user-community membership data, we can use integer or real value according to strength of relationship between user and community. Fourthly, we plan to evaluate the performance of our method for different domains like for recommending items or users. We also plan to evaluate our method's performance on some other social networking services which have lots of cold users having very few community memberships.

Bibliography

1. Baatarjav, E.-A., Phithakkitnukoon, S., Dantu, R. (2008). Group recommendation system for Facebook. In: the OTM Confederated International Workshops and Posters on On the Move to Meaningful Internet Systems: 2008 Workshops, pp. 211–219.
2. Berry, M.W., Dumais, S.T., O'Brien, G.W. (1995). Using Linear Algebra for Intelligent Information Retrieval, SIAM Review 37(4), pp. 573-595.
3. Chen, W.-Y, Zhang, D., Chang, E.Y. (2008). Combinational Collaborative Filtering for Personalized Community Recommendation. In: 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 115–123.
4. Chen, W.-Y., Chu, J.-C., Luan, J., Bai, H., Wang, Y., Chang, E.Y. (2009). Collaborative Filtering for Orkut Communities: Discovery of User Latent Behavior. In: 18th International Conference on World Wide Web, pp. 681–690.
5. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman R. (1990). Indexing by Latent Semantic Analysis. Journal of the American Society for Information Science 41(6), pp. 391–407.
6. Deshpande, M., Karypis, G. (2004). Item-based Top-N Recommendation Algorithms, ACM Transactions on Information Systems 22(1), pp. 143–177.
7. Lee, D.H., Brusilovsky, P. (2010). Interest Similarity of Group Members: The Case Study of Citeulike. In: the WebSci10: Extending the Frontiers of Society On-Line.
8. Li, X., Guo, L., Zhao, Y.E. (2008). Tag-based Social Interest Discovery. In: 17th International Conference on World Wide Web, pp. 675–684.
9. Markines, B., Cattuto, C., Menczer, F., Benz, D., Hotho, A., Stumme, G. (2009). Evaluating Similarity Measures for Emergent Semantics of Social Tagging. In: 18th International Conference on World Wide Web, pp. 641–650.
10. Mislove, A., Marcon, M., Gummadi, K.P., Druschel, P., Bhattacharjee, B. (2007). Measurement and analysis of online social networks. In: 7th ACM SIGCOMM Conference on Internet Measurement, pp. 29–42.

11. Sarwar, B., Karypis, G., Konstan J., Riedl, J. (2000). Application of Dimensionality Reduction in Recommender System–A Case Study, In: ACM WebKDD 2000 Web Mining for E-Commerce Workshop.
12. Vallet, D., Cantador, I., Jose, J.M. (2010). Personalizing Web Search with Folksonomy-based User and Document Profiles, in: Proceedings of 32nd European Conference on Information Retrieval, pp. 420-431.
13. Vasuki, V., Natarajan, N., Lu, Z., Savas, B., Dhillon, I. (2011). Scalable Affiliation Recommendation using Auxiliary Networks. ACM Transactions on Intelligent Systems and Technology 3(1), Article 3.
14. Xu, S., Bao, S., Fei, B., Su, Z., Yu, Y. (2008). Exploring Folksonomy for Personalized Search. In: 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 155–162.
15. Zheng, N., Li, Q., Liao, S., Zhang, L. (2010). Which Photo Groups Should I Choose? A Comparative Study of Recommendation Algorithms in Flickr. Journal of Information Science 36(6), pp. 733–750.
16. Garton, L. and Haythornthwaite, C. and Wellman, B. (1997). Studying Online Social Networks. Journal of Computer-Mediated Communication 3(1).
17. Lai, L.S.L. and Turban, E. (2008). Groups formation and operations in the Web 2.0 environment and social networks. Group Decision and Negotiation 17(5), pp. 387-402.
18. Wellman, B. (2005). Community: from neighborhood to network. Communications of the ACM 48(10), pp. 53-55.
19. Boyd D. M., Ellison, N.B. (2007). Social network sites: Definition, history, and scholarship. Journal of Computer-Mediated Communication 13(1), pp. 210-230.
20. http://en.wikipedia.org/wiki/List_of_social_networking_websites (accessed on 1/6/2012)
21. Furnas, G.W. and Fake, C. and Von Ahn, L. and Schachter, J. and Golder, S. and Fox, K. and Davis, M. and Marlow, C. and Naaman, M. (2006). Why do tagging systems work? CHI'06 extended abstracts on Human factors in computing systems (ACM), pp. 36-39.
22. Chi, E.H. and Mytkowicz, T. (2008). Understanding the efficiency of social tagging systems using information theory. Proceedings of the nineteenth ACM conference on Hypertext and hypermedia, pp. 81-88.

23. Lamere, P. (2008). Social tagging and music information retrieval. Journal of New Music Research 37(2), pp. 101-114.
24. Backstrom L., Huttenlocher D., Kleinberg J., Lan X. 2006. Group formation in large social networks: membership, growth and evolution. In: Proceedings of the 12th international conference on knowledge discovery and data mining. Philadelphia, Pennsylvania, USA, pp. 20–23.
25. Schafer, J.B. and Konstan, J. and Riedi, J. (1999). Recommender systems in e-commerce. Proceedings of the 1st ACM conference on Electronic commerce. Pp. 158—166.
26. Kazienko, P. and Musia K. (2006). Recommendation framework for online social networks. Advances in Web Intelligence and Data Mining, pp. 111-120.
27. Sarwar, B. and Karypis, G. and Konstan, J. and Riedl, J. (2000). Analysis of recommendation algorithms for e-commerce. Proceedings of the 2nd ACM conference on Electronic commerce, pp. 158-167.
28. Konstan, J.A. and Miller, B.N. and Maltz, D. and Herlocker, J.L. and Gordon, L.R. and Riedl, J. (1997). GroupLens: applying collaborative filtering to Usenet news. Communications of the ACM, 40(3), pp. 77-87.
29. Shardanand, U. and Maes, P. (1995). Social information filtering: algorithms for automating "word of mouth". Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 210-217.
30. http://en.wikipedia.org/wiki/Facebook (Accessed on 1/6/2012)
31. http://en.wikipedia.org/wiki/Last.fm (Accessed on 1/6/2012)
32. Fodor, I.K. (2002). A survey of dimension reduction techniques. Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, volume 9, pp. 1-18.
33. Van Der Maaten, LJP and Postma, EO and Van Den Herik, H. J. (2007). Dimensionality reduction: A comparative review. Published online by Citeseer, volume 10, pp. 1-35.
34. Spertus, E. and Sahami, M. and Buyukkokten, O. (2005). Evaluating similarity measures: a large-scale study in the orkut social network. Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining. pp. 678-684.
35. Chen, H.M. and Chang, M.H. and Chang, P.C. and Tien, M.C. and Hsu, W.H. and Wu, J.L. (2008). SheepDog: group and tag recommendation for flickr photos by automatic search-based learning. Proceedings of the 16th ACM international conference on Multimedia. pp. 737-740.

36. Ko, H.G. and Choi, S.H. and Ko, I.Y. (2009). A community recommendation method based on social networks for web 2.0-based IPTV. 16th International Conference on Digital Signal Processing, pp. 1-6.
37. C.-C. Chang and C.-J. Lin. (2001). LIBSVM: a library for support vector machines. Software available at http://www.csie.ntu.edu.tw/˜cjlin/libsvm.
38. Bao, S. and Xue, G. and Wu, X. and Yu, Y. and Fei, B. and Su, Z. (2007). Optimizing web search using social annotations. Proceedings of the 16th international conference on World Wide Web. pp. 501-510.
39. Biancalana, C. and Micarelli, A. (2009). Social tagging in query expansion: A new way for personalized web search. International Conference on Computational Science and Engineering, CSE'09. Volume (4), pp. 1060-1065.
40. Yang, B. and Liu, D. and Liu, J., (2010). Discovering Communities from Social Networks: Methodologies and Applications. Handbook of Social Network Technologies and Applications. Published by Springer. pp. 331-346.
41. Golder, S.A. and Huberman, B.A., (2006). Usage patterns of collaborative tagging systems. Journal of Information Science. Volume 32(2), pp. 198-208.
42. Han, J. and Pei, J. and Yin, Y. and Mao, R., (2004). Mining frequent patterns without candidate generation: A frequent-pattern tree approach. Data Min. Knowledge Discovery. Volume 8(1), pp. 53-87.
43. OReilly, T. (2007). What is Web 2.0: Design patterns and business models for the next generation of software. http://mpra.ub.uni-muenchen.de/4578/.
44. http://newsroom.fb.com/content/default.aspx?NewsAreaId=2, (Accessed on 01/07/2012)
45. Singh, S. (2007). Social Networks and Group Formation Theoretical Concepts to Leverage. Citeseer.
46. http://www.facebook.com
47. https://twitter.com/
48. http://www.myspace.com/
49. www.orkut.com/
50. www.livejournal.com/
51. www.citeulike.org/
52. www.last.fm/

53. www.flickr.com/
54. http://vimeo.com/
55. www.flixster.com/
56. www.blogster.com/
57. www.goodreads.com/
58. http://www.youtube.com/
59. Blei, D.M. and Ng, A.Y. and Jordan, M.I. (2003). Latent dirichlet allocation. The Journal of Machine Learning Research. Volume 3, pp 993—1022.
60. Hofmann, T. (2001). Unsupervised learning by probabilistic latent semantic analysis. Machine Learning. Volume 42(1). pp. 177—196.
61. Hudak, P.L. and Amadio, P.C. and Bombardier, C. and Beaton, D. and Cole, D. and Davis, A. and Hawker, G. and Katz, J.N. and Makela, M. and Marx, R.G. (1996). Development of an upper extremity outcome measure: The DASH (Disabilities of the Arm, Shoulder, and Head). Volume 29(6), pp. 602—608.