Community-Based User Recommendation in Uni-Directional Social Networks

Gang Zhao
School of Computing
National University of Singapore
zhaogang@comp.nus.edu.sg

Mong Li Lee
School of Computing
National University of Singapore
leeml@comp.nus.edu.sg

Wynne Hsu
School of Computing
National University of Singapore
whsu@comp.nus.edu.sg

Wei Chen
School of Computing
National University of Singapore
chenwei@comp.nus.edu.sg

Haoji Hu
Software Engineering Institute
East China Normal University
hjhu@ecnu.cn

ABSTRACT

Advances in Web 2.0 technology has led to the rising popularity of many social network services, e.g., there are over 500 million active users in Twitter. Given the huge number of users, user recommendation has gained importance where the goal is to find a set of users whom a target user is likely to follow. Content-based approaches that rely on tweet content for user recommendation have low precision as tweet contents are typically short and noisy, while collaborative filtering approaches that utilize follower-followee relationships lead to higher precision but data sparsity remains a challenge. In this work, we propose a community-based approach to user recommendation in Twitter-style social networks. Forming communities enables us to reduce data sparsity and focus on discovering the latent characteristics of communities instead of individuals. We employ an LDA-based method on the follower-followee relationships to discover communities before applying the state-of-the-art matrix factorization method on each of the communities. This approach proves effective in improving the conversion rate (by as much as 20%) as demonstrated by the results of extensive experiments on two real world data sets Twitter and Weibo. In addition, the community-based approach is scalable as the individual community can be analyzed separately.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: [Information filtering]

General Terms

Algorithms, Experimentation, Performance

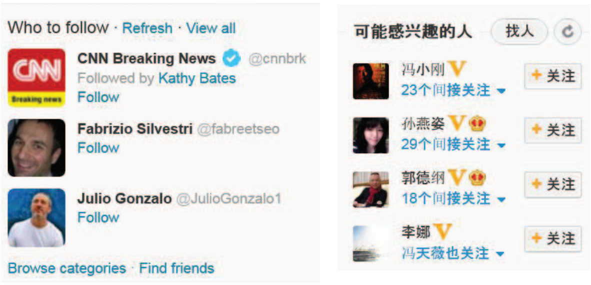
CIKM'13, Oct. 27–Nov. 1, 2013, San Francisco, CA, USA.
Copyright 2013 ACM 978-1-4503-2263-8/13/10 ...$15.00.
http://dx.doi.org/10.1145/2505515.2505533 .

Keywords

Recommender System, User Recommendation, Personalization, Uni-direction Social Network, LDA, Matrix Factoriazation

1. INTRODUCTION

The development of Web 2.0 technology has offered new opportunities and challenges for both service providers and academic researchers. One of the most successful Web 2.0 products is the social network platform, e.g. Facebook and Twitter, which facilitates and enhances relationships among users. The continued success of these social networks relies heavily on their abilities to recommend appropriate and relevant users to drive relationship creation. Figure 1 shows the screen shots of followee recommendations in Twitter and Weibo. If the user actually chooses one of the users from the list of recommended top-K users to follow, then we say that the recommendation is successful.

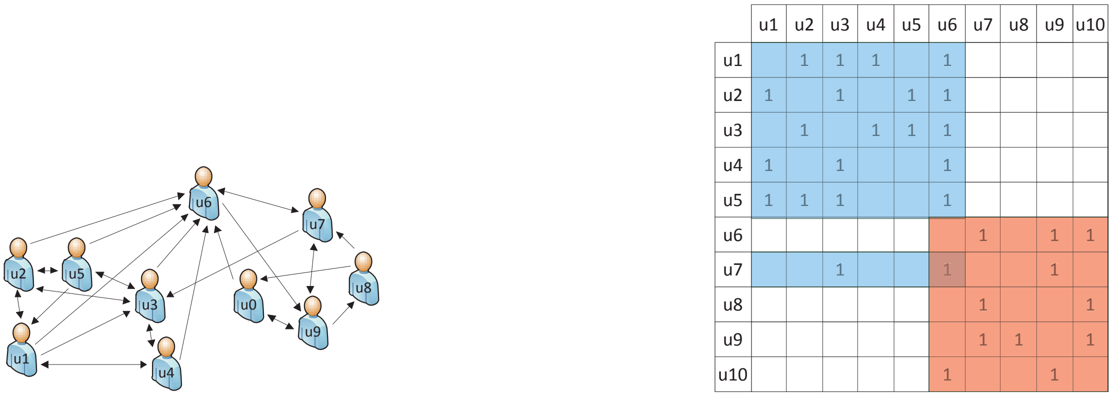

Who to follow · Refresh · View all
CNN Breaking News @cnnbrk
Followed by Kathy Bates
Follow
Fabrizio Silvestri @fabreetseo
Follow
Julio Gonzalo @JulioGonzalo1
Follow
Browse categories · Find friends
可能感兴趣的人
找人
冯小刚
23个间接关注
+关注
孙燕姿
29个间接关注
+关注
郭德纲
18个间接关注
+关注
李娜
冯天薇也关注
+关注

(a) Twitter

(a) Weibo

Figure 1: Screen shots of followee recommender feature in Twitter and Weibo

Existing user recommendation approaches assume that user preference information such as ratings and purchase temporal histories are available to depict their interests [23]. However, this is a challenge in Twitter because of its limited user information. Inferring user preferences from their tweets is also difficult as tweets are inherently noisy (short and peppered with acronyms and abbreviations).



	u1	u2	u3	u4	u5	u6	u7	u8	u9	u10
u1		1	1	1		1				
u2	1		1		1	1				
u3		1		1	1	1				
u4	1		1			1				
u5	1	1	1			1				
u6							1		1	1
u7			1			1			1	
u8							1			1
u9							1	1		1
u10						1			1	

(a) Graph Representation

(b) Matrix representation

Figure 2: Toy Example of a Uni-directional Social Network

The work in [9] examines using combinations of tweet content and follower-followee relationships to recommend users to follow in Twitter. They found that follower-followee relationships are dominant features that capture the interest of users since users actively choose people they are interested in to follow.

Figure 2(a) shows a sample Twitter-style social network where the relationships are directional and not necessarily reciprocal. The directed edge $e(u, v)$ indicates that user u is following user v. Each user u has a set of followers F_u and a set of followees G_u. For example, we have $F_{u_1} = \{u_2, u_4, u_5\}$ and $G_{u_1} = \{u_2, u_3, u_4, u_6\}$. Note that we do not have the edge $e(u, v)$ where $u = v$ since a user does not follow him/herself.

Although the follow relationship among users seems disorganized and chaotic, communities exist in these social networks as a user follows another user based on his/her interests. Figure 2(b) gives the matrix representation of the follow relationships in Figure 2(a). The rows and columns denote user ids. An element at row i and column j with a value of 1 indicates that user u_i is a follower of user u_j. In other words, row i is the followee list G_{u_i} for user u_i and column j is the follower list F_{u_j} for user u_j.

By clustering or re-arranging the rows and columns in the matrix, we can obtain 2 communities as indicated by the red and blue submatrice. We observe that:

1. A user may be a follower in more than one communities, indicating his/her multiple interests, e.g., pop music and sports. For example, user 7 is a follower in both the red and blue communities.

2. A user may be a followee in multiple communities, demonstrating his/her influence in these communities. For example, user 6 is a followee in both the red and blue communities.

3. A user may play different roles in different communities. For example, user 6 is both a followee and a follower in the red community. However, s/he is only a followee in the blue community.

The above observations motivate us to utilize a probabilistic approach that leverages both the follower and followee information of users to discover communities. The goal is to form communities of users with similar influence as well as interests. Then, applying state-of-the-art matrix factorization methods and its variants *IF-MF* [13] and *BPR-MF* [18] to each community will lead to better personalized recommendations.

Suppose we want to recommend users to u_{10} to follow. We observe that u_{10} is in the red community. If we apply matrix factorization on the red sub-matrix, we will recommend u_7 to u_{10}. However, if we apply matrix factorization on the entire matrix in Figure 2(b), we will recommend u_3 because u_{10} follows u_6 and u_9, and the majority of the u_6's followers also follow u_3. Our experiments demonstrate that by discovering communities in Twitter-style social network and recommending users to follow within these communities leads to significant improvement in conversion rate, precision and recall over performing matrix factorization on the original dataset (see Section 4.4).

Further, forming communities for user recommendation in a uni-directional social network reduces the sparsity in the matrix which is one of the most serious limitations of contemporay matrix factorization approaches. For example, the densities of the 2 sub-matrix in Figure 2(b) which correspond to the red and blue communities are increased to 48%, 58% respectively compared to the original density of 32%. The proposed approach is also scalable as the matrix factorization of each community (a subset of the original data set) can be performed in parallel (see Section 4.5).

In this work, we utilize the follower-followee relationships in Twitter-style social network and propose a two-step approach to recommend users to follow. We first employ an LDA-based method to discover communities before applying matrix factorization on each of the discovered communities. Based on the results obtained after matrix factorization, we devise two ways to recommend the top-k followees for a target user. Extensive experiments on two real world data sets Twitter and Weibo demonstrate that the proposed approach is scalable and improves the conversion rate by 20% compared to the state-of-the-art matrix factorization based recommendation algorithms [13, 17].

The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 describes our proposed framework. Section 4 gives the results of our experimental study and we conclude in section 5.

2. RELATED WORK

There has been much research on using recommender systems to help users connect with people online [12, 8, 6, 4]. These works are focused on more structured data and restricted domains such as co-authorship links [8], community membership in enterprise social network [4].

The work in [8] profiled users by aggregating information from multiple sources in an enterprise and highlighted users who have contributed in similar ways, e.g., patent authorship, co-author papers or wikis. [4] designed algorithms that utilize content similarity and social network structure in user recommendation. The former is based on the intuition that *if two users both post content on similar topics, then they might be interested in getting to know each other*, while the latter is based on the Friend-of-Friend hypothesis that *if many of my friends consider someone a friend, then I might be interest to know that person too.*

Recent work has examined methods for recommending users to follow in noisy unstructured micro-blogging data such as Twitter [9, 1]. The authors in [9] investigated both content-based approach (users' own tweets, their followers' tweets and followees' tweets) and collaborative filtering approach (users' ID, followers' ID and followees' ID) to profile users. User profiles are indexed and the information retrieval TF-IDF approach is used to rank and recommend users based on a target user profile. They find that the collaborative filtering approach are better at finding relevant followees for a user as users' relationships are more structured than the tweets contents.

Matrix factorization and its variants [15, 13, 17] have become the state-of-the-art collaborative filtering approaches for recommender systems. The work in [13] proposed a matrix factorization method (IF-MF) for implicit feedback data sets. Each user-item (or user-user) pair is associated a confidence variable in the cost function, and each decision is assigned a weight in the learning process. The authors in [17] proposed a probabilistic matrix factorization method (BPR-MF) for implicit feedback data sets. Unlike other matrix factorization approaches that take the unseen items as missing samples, BPR-MF divides the unseen items into negative samples and missing samples. This work also has been applied in KDD Cup 2012 [20] for user recommendation.

The work in [1] designed an algorithm based on the neighborhood of follower/followee relationships to search for candidate users to recommend. This algorithm is based on the hypothesis that, for a target user u, the users followed by the followers of u's followees are candidates to recommend to u. This approach is a variant of the neighborhood item recommendation method [19] where a followee is equivalent to an item. [15] showed that neighborhood approaches perform worse than matrix factorization approach, and this is also confirmed in our experiments.

Several works have utilized LDA to discover groups/ communities in large graphs [5, 22, 3]. The work in [5] explored how to find community structures in large scale undirected social network such as Facebook. [22] proposed a model called Simple Social Network-LDA (SSN-LDA) to model large undirected graphs. The authors map social interactions such as co-authorship and adviser/advisee to words, and people such as authors to documents. The work in [3] designed an LDA-based model to handle popular users and discover communities for followees in directed social network such as Twitter. These works aim to find communities of users which are highly correlated or like-minded, e.g., people who are doing information retrieval research. In contrast, our work considers both follower and followee relationships to discover more coherent communities in order to improve both the effectiveness and efficiency of matrix factorization for user recommendation.

The work in [21] explored how user-item subgroups can improve the performance of recommender systems. They design a multi-class co-clustering approach that utilizes the explicit ratings to group the users and items. This idea can be extended to find follower-followee subgroups and we evaluate this approach with the proposed LDA-based method. Our experiment results show that the LDA-based method outperforms the co-clustering approach.

3. PROPOSED FRAMEWORK

Our proposed framework comprises two main phases. The first phase utilizes an LDA-based method to determine the topic distribution of the users. Communities are formed by grouping users whose probability of a given topic is above some threshold. The second phase applies matrix factorization on each community to generate a list of candidate followees. We then combine these candidate lists to obtain the top-k users for a target user to follow. Before we describe the details of each phase, we summarize the symbols used in Table 1.

Table 1: Meanings of symbols used

Symbol	Meaning
u	A Twitter user
U	The set of all Twitter users
f	A follower
F	The set of all followers
g	A followee
G	The set of all followees
$e(f, g)$	A follow edge from f to g
E	The set of all edges $e(f, g)$ $f \in F$, $g \in G$
z	A topic
Z	The set of all topics
c	A community
C	The set of all community
$c.F$	The set of followers in community c
$c.G$	The set of followees in community c
$c.E$	The set of edges $e(f, g)$ in a community c, $f \in c.F$, $g \in c.G$

3.1 Discover Communities

Latent Dirichlet Allocation (LDA) [2] has been proposed for modeling the topic distribution of a set of documents D. Similar to Probabilistic Latent Semantic Indexing ($PLSI$) [11], each document in the LDA model is represented as a mixture of a fixed numbers of topics Z, with topic z having a probability $Pr(z|d)$ in document d. Each topic is a probability distribution over a finite vocabulary of words W, with word w having probability $Pr(w|z)$ in topic z.

Given the parameters α and β where α is a vector of dimension $|Z|$ and β is a vector of dimension $|W|$, the document generation process is as follows:

1. Choose the number of topics.

2. Choose $\theta \sim Dir(\alpha)$

3. For each word w_n

    - Choose a topic $z_n \sim Multinomial(\theta)$
    - Choose word w_n from $Pr(w_n|z_n, \beta)$

LDA has been shown to be effective in document classification and recently, it has been applied to uni-directional social network such as Twitter to group users based on their follower relationship [3]. In this work, we propose to incorporate both the follower and followee relationships into the LDA model to discover communities. We map both followees and followers into the same space so that the communities obtained will link users based on their interests (followees) and influence (followers).

Let U be the set of users and E be the set of directed edges connecting the users in a social network. An edge $e(f, g) \in E$ implies that user f follows user g. Let $F \subset U$ be the set of followers and $G \subset U$ be the set of followees defined as:

$$\begin{aligned} F &= \{u \mid u \in U \wedge \exists g \in U \wedge \exists\ e(u, g) \in E\} \\ G &= \{u \mid u \in U \wedge \exists f \in U \wedge \exists\ e(f, u) \in E\} \end{aligned} \quad (1)$$

Just as one has a topic in mind when choosing a word for a document, likewise a user has an interest in mind when following another user in Twitter. Hence, each follower f can be regarded as a document consisting of a list of followees g. We denote $Pr(z|f)$ as the multinomial probability of topic z given a follower f, and $Pr(g|z)$ as the multinomial probability of a followee g given z.

Since a user u can be both a follower f and a followee g, s/he is associated with two documents d_f and d_g. The content of d_f is the list of followees of u, while the content of d_g is the list of followers of u, denoted as follows:

$$\begin{aligned} d_f &: \{u \mid u \in U \wedge \exists\ e(f, u) \in E\} \\ d_g &: \{u \mid u \in U \wedge \exists\ e(u, g) \in E\} \end{aligned} \quad (2)$$

Therefore our document corpus D is given by

$$D = \bigcup_{f \in F} d_f \quad \cup \quad \bigcup_{g \in G} d_g \quad (3)$$

We apply LDA on D to generate a pre-defined number of topics Z. Figure 3 depicts the graph model for this representation.

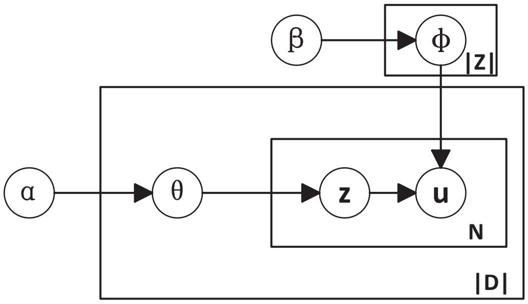


Figure 3: Graphical Model Representation

For each topic $z \in Z$, we form a community c such that the followers and followees in c, denoted as $c.F$ and $c.G$ respectively, are given by

$$\begin{aligned} c.F &= \{f \mid f \in F \wedge Pr(z|d_f) > \gamma\} \\ c.G &= \{g \mid g \in G \wedge Pr(z|d_g) > \gamma\} \end{aligned} \quad (4)$$

where γ is some threshold.

The edges in c, denoted as $c.E$, represent the follower-followee relationships in c and is given by

$$c.E = \{e(f, g) \mid e(f, g) \in E \wedge f \in c.F \wedge g \in c.G\} \quad (5)$$

The output for this phase is a set of communities C where $|C| = |Z|$.

3.2 Recommend Followees

After discovering the communities, the next phase is to generate candidate followees from these communities for recommendation. Matrix factorization is first proposed in [15] for recommender systems and has been applied to predict user ratings for items. This approach has been adapted to handle binarized user preference for items in implicit feedback data sets (IF-MF) [13].

Here, we apply the IF-MF method by considering $f \in F$ as users and $g \in G$ as items and construct the matrix M in the model as follows. For each community $c \in C$, the matrix M has dimensions $|c.F| \times |c.G|$. Each entry $M[f, g]$ has a value of 1 if there is an edge $e(f, g) \in c.E$, otherwise $M[f, g] = 0$.

After matrix factorization, we obtain two matrices, namely $P^{|c.F| \times L}$ and $Q^{L \times |c.G|}$, where $P^{|c.F| \times L}$ denotes the mappings of followers in the reduced latent space of L dimensions and $Q^{L \times |c.G|}$ denotes the mappings of followees to the same reduced latent space. In other words, each follower f is associated with a vector $p_f \in P^{|c.F| \times L}$, while each followee g is associated with a vector $q_g \in Q^{L \times |c.G|}$.

Then for a follower f, we obtain the score that s/he will follow g in community c. This is given by the inner product of p_f and q_g as follows:

$$score(f, g, c) = \langle p_f, q_g \rangle \quad (6)$$

A target user f may belong to more than one community. Thus we will have a different candidate followee recommendation list from each community. Here, we propose two ways to compute the final score that a target user $f \in F$ will follow $g \in G$ from these lists.

We can take the maximum score among the scores in the communities that both f and g belong to.

$$maxScore(f, g) = \underset{c \in C}{Max}(score(f, g, c)) \quad (7)$$

Alternatively, we can sum up the scores in all the communities that f and g appear in as follows:

$$sumScore(f, g) = \sum_{c \in C} (score(f, g, c) \times Pr(c|f)) \quad (8)$$

where $Pr(c|f)$ is the probability that f belongs to the community c.

Note that $Pr(c|f)$ is $Pr(z|d_f)$ in the LDA model where z is the latent topic corresponding to community c.

Finally, we sort these scores for each follower f and output the top-K followees to recommend to f.

Algorithm 1 summarizes our proposed approach. We call our method Community-Based Matrix Factorization (*CB-MF*). We first obtain the set of followers and followees from the follower-followee relationships (lines 1-3). Then we obtain the document corpus and apply LDA to generate a predetermined number of topics (lines 4-11). Lines 12 to 18 shows how to construct each community with its followers, followees and associated edges. Then we perform matrix factorization on each community (lines 19 to 24). Lines 25-28 aggregates the scores from each community and we obtain a ranked list of recommended followees for each follower.

```
Algorithm 1: CB-MF Algorithm
   input  : 1. Set of follower-followee relationships
               $E = \{e(f,g)\}$,
            2. Number of communities $N$,
            3. Number of latent factors $L$,
            4. Threshold $\gamma$
   output: Ranked recommendation list
1  $F \leftarrow \{f \mid \exists e(f,g) \in E\}$;
2  $G \leftarrow \{g \mid \exists e(f,g) \in E\}$;
3  $U \leftarrow F \cup G$;
4  $D = \emptyset$;
5  foreach $f \in F$ do
6      $d_f = \{u \mid u \in U \wedge \exists\ e(f,u) \in E\}$
7      $D = D \cup \{d_f\}$;
8  foreach $g \in G$ do
9      $d_g = \{u \mid u \in U \wedge \exists\ e(u,g) \in E\}$
10     $D = D \cup \{d_g\}$;
11 $Z \leftarrow LDA(D, N)$;
12 $C = \emptyset$;
13 foreach $z \in Z$ do
14     $c \leftarrow \emptyset$
15     $c.F = \{f \mid f \in F \wedge Pr(z|d_f) > \gamma\}$;
16     $c.G = \{g \mid g \in G \wedge Pr(z|d_g) > \gamma\}$;
17     $c.E = \{e(f,g) \mid e(f,g) \in E \wedge f \in c.F \wedge g \in c.G\}$;
18     $C = C \cup \{c\}$;
19 $R = \emptyset$;
20 foreach $c \in C$ do
21     construct matrix $M_c$;
22     $IF\text{-}MF(M_c, L)$;
23     $R_c = \{score(f,g,c) \mid f \in c.F \wedge g \in c.G\}$
24     $R = R \cup \{R_c\}$;
25 $Result = \emptyset$;
26 foreach pair $(f,g)$ do
27     compute $sumScore(f,g)$ (or $maxScore$) according
       to Equation 7 (or 8);
28 Return the ranked lists of followees for each follower;
```

4. EXPERIMENTAL STUDY

In this section, we report the results of the extensive experiments we have carried out to evaluate both of the effectiveness and efficiency of our proposed *CB-MF* method. We compare the performance of our method with the following methods:

1. *TopPop*. This is a baseline algorithm which ranks users according to their number of followers and recommends the top-K most popular users to follow.
2. *FoF*. This is based on the Friend-of-Friend hypothesis, that is, if a particular person is followed by many followees of a target user, then s/he might be interested to follow this person too. In other words, we find the top-K most highly ranked followees of a target user's followees.
3. *NB-based* [1]. This is an implementation of the neighborhood based algorithm in [1]. Given a target user u and its set of followees G_u, we find the set of followers $F = \{u \mid \exists e(u,g) \in E \wedge \exists g \in G_u\}$. For each $f \in F$, we find the set of followees G_f and take the union. Then we find the top-K users with the most occurrences to recommend to u.
4. *LDA-based* [3]. This is an implementation of the LDA model described in [3] which map followers to documents and followees to words. Each followee g is scored using Equation 9 and we recommend the top-K followees with the highest score.

$$Pr(g|f) = \sum_{z \in Z} Pr\,(g|z)\,Pr\,(z|f) \qquad (9)$$

5. *IF-MF* [13]. This is the state-of-the-art matrix factorization method for implicit feedback data sets.
6. *BPR-MF* [17]. This is a probabilistic matrix factorization method for implicit feedback data sets.

We implement the methods using Python. We code the LDA model according to [10], and use the C# implementation provided in [7] for the methods *BPR-MF* and *IF-MF*. All the experiments are carried out on an Intel(R) Core(TM) i7-2600 with 3.4 GHz, 8 GB RAM, 64 bit Microsoft Windows 7 operating system.

4.1 Experimental Data Sets

We use two real world Twitter-style data sets for our experiments. The first data set is the social network data used in [16] which is obtained from Twitter[1]. The second data set is the social network data which we crawled from Weibo[2], the biggest Chinese micro-blog system in China.

We pre-process these data sets to anonymize the user ids and improve the data set density by removing users who have less 10 followers/followees. Table 2 gives the statistics of the two data sets after pre-processing.

Table 2: Statistics of Twitter and Weibo data sets

Statistic	Twitter	Weibo
$\|F\|$	130,352	168,561
$\|G\|$	114,997	150,761
$\|U\|$	142,624	169,750
$\|E\|$	10,242,503	40,358,104
$\underset{g \in G}{Max}(\|E(*,g)\|)$	31,952	55,948
$\underset{f \in F}{Max}(\|E(f,*)\|)$	26,663	2,053
Sparsity	99.93%	99.84%

Figures 4 and 5 show the characteristics of the Twitter and Weibo data sets respectively. The figures depict the number of users who have same number of followers or followees.

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

[2] http://www.weibo.com

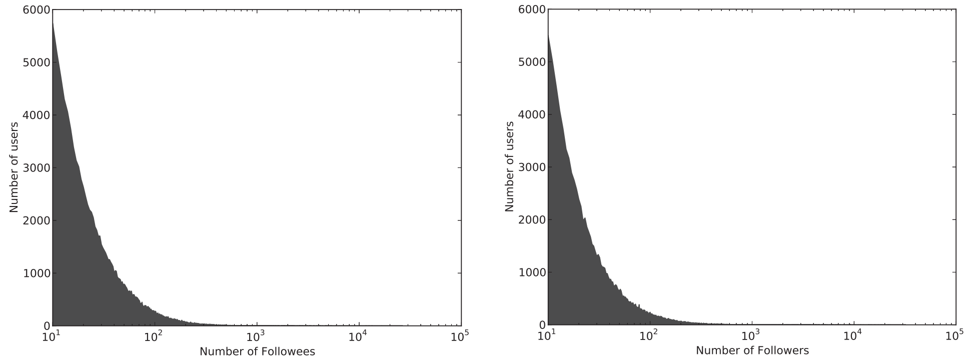


Figure 4: Characteristics of Twitter Data Set

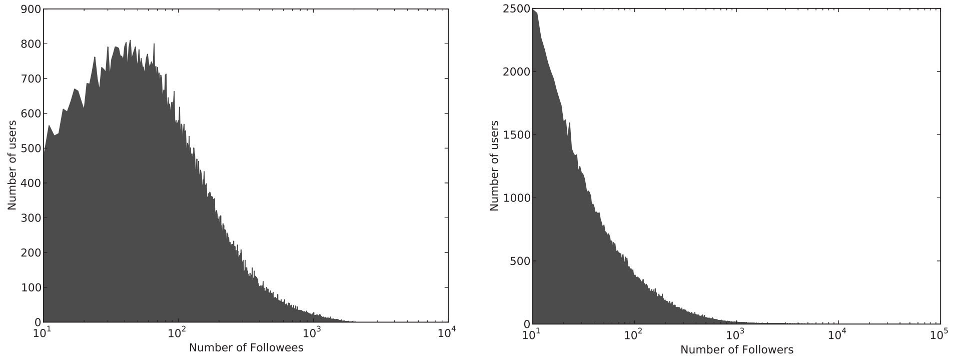


Figure 5: Characteristics of Weibo Data Set

As expected, both data sets have long tails, indicating that a small number of users have large number of followers or followees. For the Weibo data set, we see that more users have around 100 followees instead of 10 primarily because Weibo provide features such as batch following to encourage a user to have more followees. The difference in the number of followees in the two data sets is due to the different policies in Twitter and Weibo. Twitter allows users to have more followees as long as their number of followers increase. On the other hand, Weibo places a limit on the number of followees that a user can have (< 3000).

4.2 Evaluation Metrics

Our goal is to recommend top-k users for a target user to follow. For each follower, we randomly choose 10% followees s/he has followed as testing data, and keep the rest as training data. Our evaluation metrics include conversion rate, NDCG [14], precision, recall and F1 score.

Conversion rate is a commonly used metric in recommender systems to determine if a user has obtained at least one good recommendation. If L is the list of recommended k followees and L' is the list of k followees actually followed by the user, then the conversion rate is given by:

$$Conversion\ Rate = \begin{cases} 1 & \text{if } |L \cap L'| > 0 \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

We compare the conversion rates of the various algorithms by taking the average of values computed for each test user.

Normalized Discounted Cumulative Gain (NDCG) is a widely used metric for a ranked list. $NDCG_k$ is defined as:

$$NDCG_k = \frac{1}{IDCG_k} \times \sum_{i=1}^{k} \frac{2^{b_i - 1}}{log_2(i+1)} \quad (11)$$

where b_i is a binary value, 1 if the item at position i is hit item and 0 otherwise, $IDCG_k$ is the maximum $NDCG_k$ that corresponds to the optimal ranking list so that perfect NDCG can be 1.

The standard definitions for precision and recall are:

$$Recall = \frac{|L \cap L'|}{|L'|} \quad (12)$$

$$Precision = \frac{|L \cap L'|}{|L|} \quad (13)$$

We also report the *F1* score, which is the harmonic mean of precision and recall, defined as:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (14)$$

4.3 Sensitivity Experiments

We first examine how the various parameters affect the performance of our proposed *CB-MF* method. We fix the number of latent factors $L = 16$, and vary the threshold γ and number of communities N.

We measure the $F1$ score for k=3 using the two ways of combining the lists of candidate followees from each com-

Table 3: Performance on Twitter for varying γ and N

	N=5		N=10		N=15		N=20	
γ	$F1_{sum}$	$F1_{max}$	$F1_{sum}$	$F1_{max}$	$F1_{sum}$	$F1_{max}$	$F1_{sum}$	$F1_{max}$
0.01	0.0695	0.0612	0.0725	0.0638	0.0735	0.0650	0.0637	0.0572
0.02	0.0722	0.0632	**0.0740**	0.0681	0.0708	0.0602	0.0649	0.0580
0.04	0.0682	0.0593	0.0692	0.0595	0.0690	0.0597	0.0650	0.0581
0.08	0.0657	0.0584	0.0690	0.0595	0.0652	0.0579	0.0593	0.0521

Table 4: Performance on Weibo for varying γ and N

	N=5		N=10		N=15		N=20	
γ	$F1_{sum}$	$F1_{max}$	$F1_{sum}$	$F1_{max}$	$F1_{sum}$	$F1_{max}$	$F1_{sum}$	$F1_{max}$
0.01	0.0385	0.0313	0.0436	0.0372	**0.0440**	0.0375	0.0410	0.0326
0.02	0.0377	0.0308	0.0428	0.0350	0.0423	0.0333	0.0418	0.0330
0.04	0.0359	0.0293	0.0348	0.0290	0.0402	0.0327	0.0401	0.0323
0.08	0.0298	0.0231	0.0351	0.0298	0.0343	0.0270	0.0360	0.0285

munity (Equations 7 and 8). Tables 3 and 4 show the results for the Twitter and Weibo data sets respectively. We see that the $F1$ scores obtained by summing the weighted scores from the candidate lists ($F1_{sum}$) is higher compared to taking the maximum scores ($F1_{max}$). Further, a larger value for N improves the performance of $CB\text{-}MF$ on the larger Weibo data set.

Based on the results in Tables 3 and 4, we obtain the optimal parameter settings for the rest of the experiments. We use $\gamma = 0.02$, $N = 10$ for the Twitter data set, and $\gamma = 0.01$, $N = 15$ for the Weibo data set.

4.4 Comparative Experiments

Next, we compare the performance of the various user recommendation methods. We set the number of latent factors $L = 16$ for the matrix factorization based methods ($BPR\text{-}MF$, $IF\text{-}MF$). Our $CB\text{-}MF$ calls $IF\text{-}MF$ for each community with the same L setting.

Figures 6 and 7 show the Conversion Rate, NDCG, Precision and Recall for the Twitter and Weibo data sets respectively. From the results in both data sets, it is clear that the matrix factorization based methods ($BPR\text{-}MF$, $IF\text{-}MF$ and $CB\text{-}MF$) outperform the methods that do not utilize matrix factorization (*TopPop*, FoF, LDA-*based* and NB-*based*).

Among the 3 matrix factorization based methods, the proposed $CB\text{-}MF$ gives the best performance. All the methods perform better of Weibo compared to Twitter in terms of conversion rate. This is mainly because that the density of Weibo data set is higher then Twitter data set. For state-of-the-art matrix factorization approaches $IF\text{-}MF$ and $BPR\text{-}MF$, $IF\text{-}MF$ performs better than $BPR\text{-}MF$ on both data sets. This is because $IF\text{-}MF$ can better handle the data set sparsity.

We also observe that FoF outperforms the NB-*based* algorithm. This is because the recommendations given by NB-*based* for a target user who follows popular users will be similar to the baseline *TopPop*. The LDA-*based* method is better than *TopPop*, FoF and NB-*based* mainly because it is able to discover and utilize the hidden characteristics of followees and followers for recommendation.

Overall, our proposed community-based approach improves the conversion rate in Weibo by about 15%, and leads to a significant 30% increase in the conversion rate for Twitter. This is because our approach applies matrix factorization on communities which have lower sparsity compared to the original data set. Figure 8 compares the sparsity of the original data sets and the communities obtained, clearly indicating that reducing data sparsity can help improve the effectiveness of user recommendation.

Comparison of Community Discovery Methods. We also examine the impact of using different community discovery methods on the conversion rate. We compare our approach to find communities with the following two methods:

1. $LDA\text{-}Followee$ [3]. This is an LDA-based model which utilizes only follower relationships.

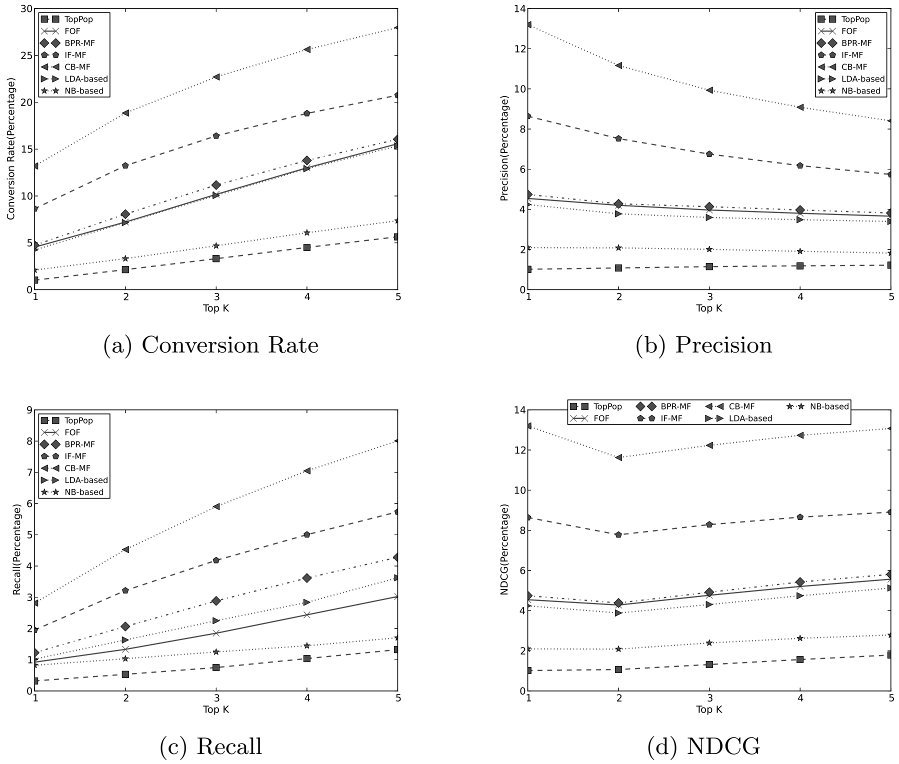
2. *MCoC* [21]. This is a multi-class co-clustering method to find user-item subgroups for item recommendation. We use this method to find follower-followee subgroups.

The *MCoC* code provided by the authors could not scale on the large Weibo data set. For the Twitter data set, we had to further improve the density by filtering out users who have less than 100 followers or followees. The resulting data set has 19305 followers and 16782 followees, and the data set sparsity is improved to 98.62%.

We apply the same matrix factorization approach $IF\text{-}MF$ with $L = 16$ on the communities obtained by the different methods. Figure 9 shows the results on both Twitter and Weibo data sets. We observe that our LDA-based model which utilizes both follower and followee relationship outperforms both $LDA\text{-}Followee$ and *MCoC*, indicating that the communities obtained by our model are able to capture the user influence and interests.

4.5 Scalability Experiments

In this last set of experiments, we examine the scalability of the proposed approach. Matrix factorization is computationally expensive, especially when the number of latent factors increases. We advocate that $CB\text{-}MF$ can be an alternative form of parallelization for matrix factorization. The run time of $CB\text{-}MF$ is given by the time needed to discover communities and the maximum time obtained from running $IF\text{-}MF$ on each of the community in parallel.



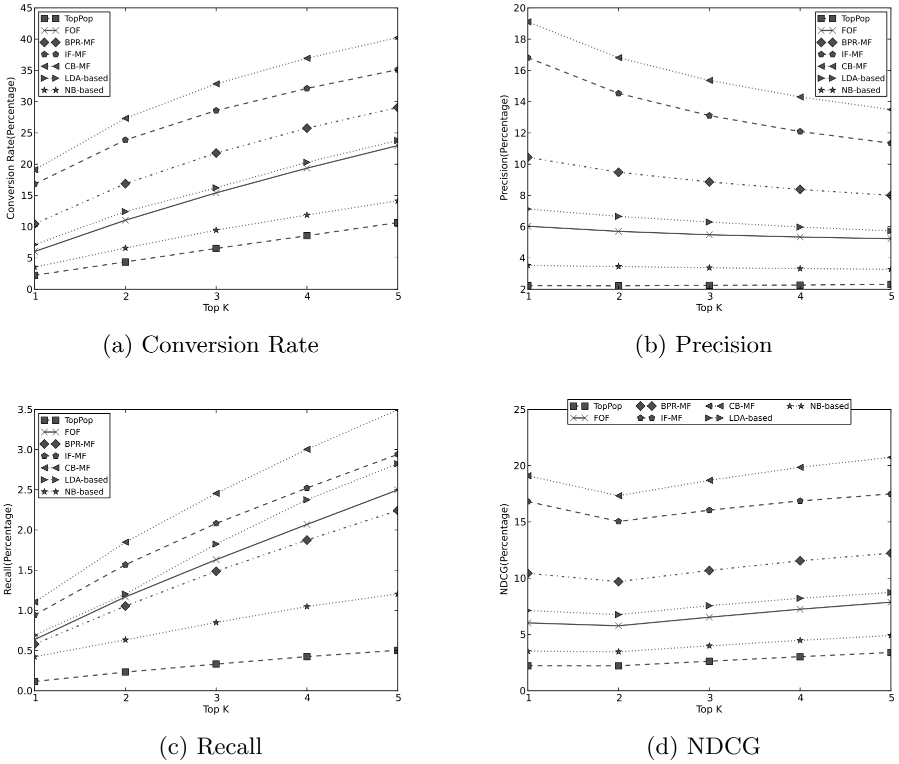
(a) Conversion Rate

(b) Precision

(c) Recall

(d) NDCG

Figure 6: Comparative study on Twitter data set



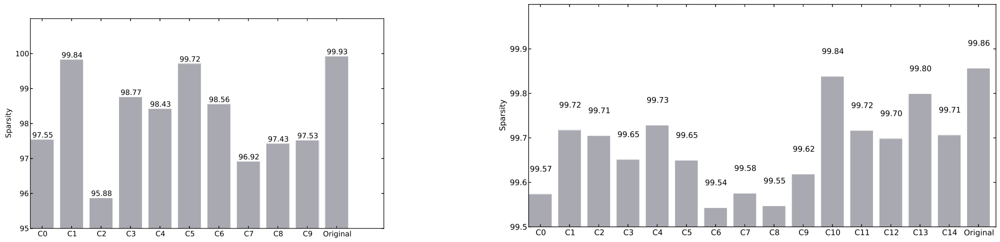
(a) Conversion Rate

(b) Precision

(c) Recall

(d) NDCG

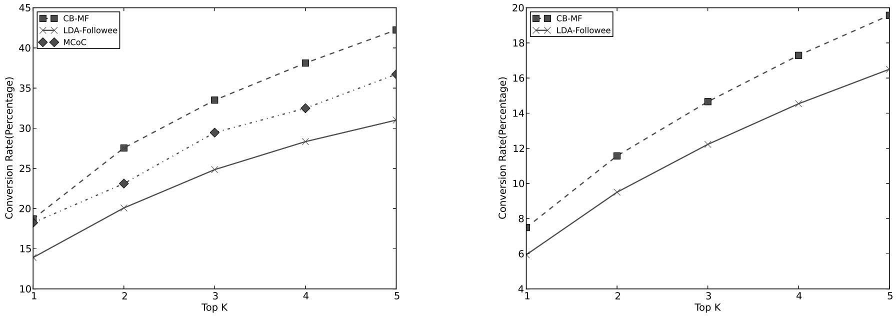
Figure 7: Comparative study on Weibo data set



(a) Twitter Data Set

(b) Weibo Data Set

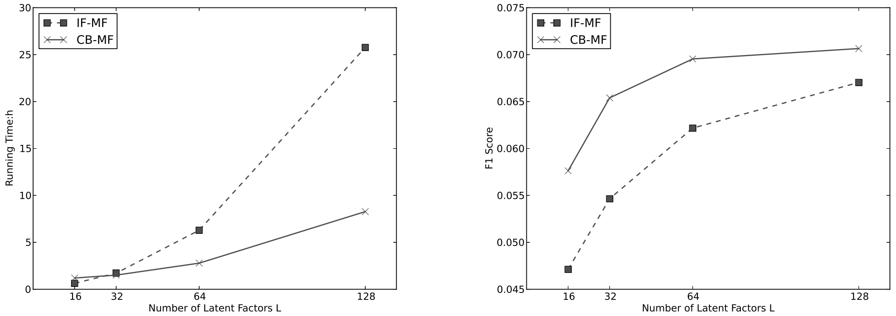
Figure 8: Sparsity of original dataset vs. discovered communities



(a) Twitter Data Set

(b) Weibo Data Set

Figure 9: Effect of different community discovery methods on conversion rate



(a) Runtime

(b) F1 Score

Figure 10: Effect of L on runtime and F1 (Weibo dataset)

We compare the performance of $CB\text{-}MF$ and $IF\text{-}MF$ on the larger Weibo data set. Figure 10 shows the run time and the F1 scores as we vary the number of latent factors L from 16 to 128. The results clearly demonstrate the effectiveness of the proposed community-based matrix factorization approach and its ability to scale. Although the F1 scores of both methods increase with L, the running time for $CB\text{-}MF$ remains reasonably stable while the run time for $IF\text{-}MF$ grows significantly.

5. CONCLUSION

In this paper, we have investigated using both follower and followee relationships to discover communities to improve user recommendation in uni-directional social networks. We have introduced a two-phase approach where we first utilize the LDA model to discover communities, and then applied matrix factorization on each community found. We carried out extensive experiments to evaluate the performance of our approach on two real world uni-directional social network data sets, Twitter and Weibo. The results indicate that the proposed $CB\text{-}MF$ method significantly outperforms state-of-the-art recommender algorithms. We have further shown that the community-based approach is a good alternative form of parallelization for matrix factorization. Future research direction includes developing our approach on MapReduce framework.

6. REFERENCES

[1] M.G. Armentano, D.L. Godoy, and A.A. Amandi. A Topology-based Approach for Followees Recommendation in Twitter. In *9th Workshop on Intelligent Techniques for Web Personalization and Recommender Systems*, 2011.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. In *the Journal of Machine Learning Research*, 3:993–1022,2003.

[3] Y. Cha and J. Cho. Social-network Analysis Using Topic Models. In *Proceedings of the 35th International ACM SIGIR Conference*, pages 565–574, 2012.

[4] J. Chen, W. Geyer, C. Dugan, M. Muller, and I. Guy. Make New Friends, but Keep the Old: Recommending People on Social Networking Sites. In *Proceedings of the ACM SIGCHI Conference* , pages 201–210, 2009.

[5] E. Ferrara. Community Structure Discovery in Facebook. in *International Journal of Social Network Mining*, 1(1):67–90, 2012.

[6] J. Freyne, M. Jacovi, I. Guy, and W. Geyer. Increasing Engagement through Early Recommender Intervention. In *Proceedings of the 3rd ACM Conference on Recommender Systems*, pages 85–92, 2009.

[7] Z. Gantner, S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. MyMediaLite: A Free Recommender System Library. In *Proceedings of the 5th ACM Conference on Recommender Systems* , 2011.

[8] I. Guy, I. Ronen, and E. Wilcox. Do You Know?: Recommending People to Invite into Your Social Network. In *Proceedings of the 14th International Conference on Intelligent User Interfaces*, pages 77–86, 2009.

[9] J. Hannon, M. Bennett, and B. Smyth. Recommending Twitter Users to Follow Using Content and Collaborative Filtering Approaches. In *Proceedings of the 4th ACM Conference on Recommender Systems*, pages 199–206, 2010.

[10] M.D. Hoffman, D.M. Blei, and F. Bach. Online Learning for Latent Dirichlet Allocation. *Advances in Neural Information Processing Systems*, 23:856–864, 2010.

[11] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd International ACM SIGIR Conference*, pages 50–57, 1999.

[12] W.H. Hsu, A. King, M. Paradesi, T. Pydimarri, and T. Weninger. Collaborative and Structural Recommendation of Friends Using Weblog-based Social Network Analysis. In *AAAI*, 2006.

[13] Y. Hu, Y. Koren, and C. Volinsky. Collaborative Filtering for Implicit Feedback Datasets. In *Proceedings of the 8th IEEE International Conference on Data Mining*, pages 263–272, 2008.

[14] K. Järvelin and J. Kekäläinen. Cumulated Aain-based Evaluation of IR Techniques. *ACM Transactions on Information Systems* , 20(4):422–446, 2002.

[15] Y. Koren, R. Bell, and C. Volinsky. Matrix Factorization Techniques for Recommender Systems. In *IEEE Computer*, 42(8):30–37, 2009.

[16] H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter, A Social Network or A News Media? In *Proceedings of the 19th International Conference on WWW*, pages 591–600, 2010.

[17] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, pages 452–461, 2009.

[18] R. Salakhutdinov and A. Mnih. Bayesian Probabilistic Matrix Factorization Using Markov Chain Monte Carlo. In *Proceedings of the 25th International Conference on Machine Learning*, pages 880–887, 2008.

[19] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based Collaborative Filtering Recommendation Algorithms. In *Proceedings of the 10th International Conference on WWW*, pages 285–295, 2001.

[20] KDD Cup'12 Workshop. http://www.kddcup2012.org/workshop. 2012.

[21] B. Xu, J. Bu, C. Chen, and D. Cai. An Exploration of Improving Collaborative Recommender Systems via User-item Subgroups. In *Proceedings of the 21st International Conference on WWW*, pages 21–30, 2012.

[22] H. Zhang, B. Qiu, C. L. Giles, H. C. Foley, and J. Yen. An Lda-based Community Structure Discovery Approach for Large-scale Social Networks. In *Intelligence and Security Informatics*, pages 200–207, 2007.

[23] G. Zhao, M. L. Lee, W. Hsu, and W. Chen. Increasing Temporal Diversity with Purchase Intervals. In *Proceedings of the 35th International ACM SIGIR Conference*, pages 165–174, 2012.