An Integrated Tag Recommendation Algorithm Towards Weibo User Profiling *

Deqing Yang[1], Yanghua Xiao[1], Hanghang Tong[2], Junjun Zhang[1], and Wei Wang[1]

[1] School of Computer Science, Shanghai Key Laboratory of Data Science
Fudan University, Shanghai, China
{yangdeqing,shawyh,zhangjunjun,weiwang1}@fudan.edu.cn
[2] Arizona State University, AZ, USA. hanghang.tong@gmail.com

Abstract. In this paper, we propose a tag recommendation algorithm for profiling the users in Sina Weibo. Sina Weibo has become the largest and most popular Chinese microblogging system upon which many real applications are deployed such as personalized recommendation, precise marketing, customer relationship management and etc. Although closely related, tagging users bears subtle difference from traditional tagging Web objects due to the complexity and diversity of human characteristics. To this end, we design an integrated recommendation algorithm whose unique feature lies in its *comprehensiveness* by collectively exploring the social relationships among users, the co-occurrence relationships and semantic relationships between tags. Thanks to deep comprehensiveness, our algorithm works particularly well against the two challenging problems of traditional recommender systems, i.e., *data sparsity* and *semantic redundancy*. The extensive evaluation experiments validate our algorithm's superiority over the state-of-the-art methods in terms of matching performance of the recommended tags. Moreover, our algorithm brings a broader perspective for accurately inferring missing characteristics of user profiles in social networks.
Keywords: tag recommendation, user profiling, tag propagation, Chinese knowledge graph

1 Introduction

Sina Weibo[3](Weibo in short), the largest counterpart of Twitter in China, is experiencing fast growth and becoming a world-widely used microblogging system. So far, Weibo has attracted more than 0.6 billion users in total and 5 million active users per day. The applications or services related to Weibo are creating a plenty of business opportunities since Weibo is attracting more and more users.

One of the most important services provided by Weibo is user tagging which allows a user to publish several tags to label themselves. These tags usually describe user profiles including hobby, career, education, religion and etc. Hence, Weibo tags are important for user understanding which is critical for many real industry applications, e.g., personalized recommendation, precise marketing and customer relationship management.

An effective tag recommendation algorithm is critical for Weibo. Weibo users can be divided into two groups: the groups are willing/or not to label themselves. For the

* This paper was partially supported by the National NSFC(No.61472085, 61171132, 61033010), by National Key Basic Research Program of China under No.2015CB358800, by Shanghai STCF under No.13511505302, by NSF of Jiangsu Prov. under No. BK2010280, by the National Science Foundation under Grant No. IIS1017415, by the Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053, by National Institutes of Health under the grant number R01LM011986, Region II University Transportation Center under the project number 49997-33 25. Correspondence author: Yanghua Xiao.

[3] http://weibo.com.

group willing to, an effective tag recommendation mechanism can make it easy for them to 'label' themselves. The other group is not willing to label themselves with informative tags mostly out of the privacy concerns. An effective recommendation algorithm thus is critical for the accurate characterization of these users. Despite of its importance, current tagging service only attracts 55% of Weibo users to tag themselves. The remaining users do not label themselves with any tags either due to privacy concern or inconvenient tagging service.

In general, tag recommendation for Weibo user has been rarely studied. Although many tag-based recommender systems have been proposed, they generally can not be used for tagging Weibo users due to the following reasons.

- *First, the object to be tagged is different.* In this paper, we focus on tagging Weibo users, whereas most existing tag-based recommendation systems focused on tagging Web objects, such as photos in Flickr [25] or URLs [30, 11]. In general, these systems make successful recommendations by utilizing abundant tagging activities on objects and users. However, much of these information in general is absent in Weibo setting (known as *data sparsity* problem), which poses a great challenge to accurately tag a Weibo user. Worse comes to worse, many users do not have any tag at all.
- *Second, the objective of tag recommendation is different.* Our recommendation aims to characterizing a user's *individual preference* of tags while many social tagging mechanisms were designed for *collective preference* of tags on the targeted object. Clearly, mining individual preference is different to mining collective preference since each user has his/er own unique taste. We should recommend not only diverse tags for a user but also satisfy a user's unique taste.

In this paper, we develop an effective and efficient algorithm to recommend tags for Weibo users. Although our algorithm is proposed for Weibo setting, the proposed recommendation schemes can also be imported into other social network platforms, such as Twitter and Facebook.

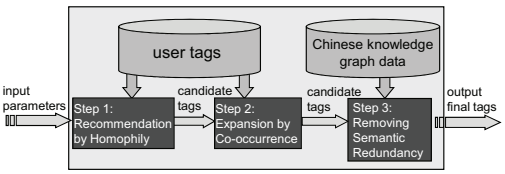


Fig. 1. Framework of tag recommendation algorithm.

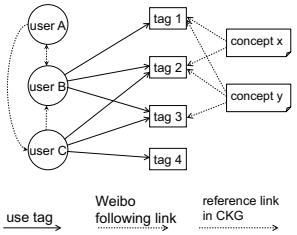


Fig. 2. The meta-graph in our recommendation framework.

1.1 Requirements

First, *the recommendation should effectively handle data sparsity*. In our scenario, nearly 45% of Weibo users have no tag. This will disable many collaborative filtering (CF in short) based recommender systems [24, 12] and co-occurrence based recommendations [25].

Second, *the recommended tags should be diversified enough to capture the* multi-facet *characteristics of a real person*. A user may publish several tags to characterize all of these aspects, e.g., education, career, hobbies, favorite idols and etc. How to recommend a set of diversified tags to a user is challenging.

Third, *the recommendation should be aware of the* semantic redundancy *in the recommended tags*. It is not suitable for real applications if too many tags are recommended. E.g., a Weibo user is restricted to use 10 tags at most. Hence a user generally

expects that the recommended tags are expressive and contain no (near-)synonyms. In contrast, it is acceptable that different users use (near-)synonyms to tag the same object [8]. Thus, in those recommender systems towards tagging objects, semantic redundancy is not an issue.

To satisfy the above requirements, in this paper we first conduct empirical studies to understand the tagging behaviors of Weibo users. Our findings reveal two effective tag recommendation mechanisms:

1. *Homophily based recommendation.* Homophily is the tendency that birds with a feather flock together [19]. It also holds on Weibo. A Weibo user tends to use the same or similar tags as his/er friends, especially when the friend is simultaneously one of his followees and followers, i.e., mutual fan.
2. *Co-occurrence based recommendation.* If a tag is deserved to be recommended, the other tags that co-occurs with it are also deserved to be recommended.

Armed with these findings, we propose a tag recommendation algorithm to generate informative and personalized tags for profiling Weibo users. Our algorithm is an integrated algorithm consisting of three major steps. Each step aims to address one of the above requirements. Fig. 1 illustrates the our algorithm's framework.

1. *Step 1: Recommendation by Homophily.* We recommend to a user with the most *frequent* and *informative* tags from the tags used by his/er friends. We import TF-IDF scheme to remove those frequent but less informative tags. We use this step to solve the data sparsity problem.
2. *Step 2: Expansion by Co-occurrence.* We use co-occurrence based scheme to enrich the recommended tag list so that the final tag list is diverse enough.
3. *Step 3: Removing Semantic Redundancy.* We construct a Chinese knowledge graph (CKG in short) from online Chinese encyclopedias. Then, we map Weibo user tags into CKG entities so that we can measure the semantic similarity of tags. Next, we use an ESA-based (explicit semantic analysis) [7] metric to remove the synonyms or near-synonyms from the recommended tag list. This step satisfies the third requirement.

Fig. 2 shows the entities and their relationships in our tag recommendation algorithm. We use this figure to illustrate our recommendation mechanism. In Step 1, we recommend to user A with tag 2 and tag 3 that are mostly used by user B and C because A follows B and C (which suggests that they have similar tag preferences). In Step 2, tag 1 and tag 4 are also recommended because they are co-used with tag 2 and tag 3. In Step 3, we remove either tag 1 or tag 2 because both concept x and y in CKG refer to them implying their redundant semantics. The detailed mechanisms will be introduced in the subsequent sections.

1.2 Contributions and Organization

In summary, the main contributions of this paper include:

1. *Empirical Findings.* We conducted extensive empirical studies to show statistical user tagging behaviors and unveil effective recommendation schemes for tagging Weibo users.
2. *Effective Algorithm.* We proposed an integrated algorithm to recommend a set of tags to Weibo users towards personalized and informative user profiling.
3. *Evaluations.* We conducted extensive evaluations to justify the effectiveness of our recommendation algorithm. The results show that our algorithm is useful in enriching user profiles as well as inferring the missing characteristics of Weibo users.

The rest of this paper is organized as follows. We first display our empirical results in Section 2 which are the basis of our recommendation algorithm. In Section 3, we elaborate the detailed procedure of tag recommendation algorithm. In Section 4, we present our experiments for evaluating algorithm performance. We survey the related works in Section 5 and conclude our paper in Section 6.

2 Empirical Study

In this section, we conduct empirical studies on the collective tagging behaviors of Weibo users. The empirical findings construct the basis of our tag recommendation algorithm. We first introduce our dataset.

Dataset: We first randomly selected 3,000 Weibo users as seeds, then crawled their followers and followees. Thus there are more than 2.1 million users and 875,186 unique user tags in total. Besides tags, the following relationships between the users were fetched. All data were crawled before Oct. 2013. The statistics show that only 55.01% of these users, i.e., about 1.15 million users have at least one tag.

2.1 Homophily in Tagging Behavior

Homophily is a tendency that an interaction between similar people occurs with a higher probability than among dissimilar people [19]. Homoplily was shown to be a universal phenomenon across a variety of social media platforms such as Twitter [28]. More specifically, the Twitter users following reciprocally (mutual fans) tend to share topical interests, have similar geographic and popularity [15]. Thus, an interesting question arises: *do close social relationships in Weibo also imply similar profiles or tags?* To answer this question, we first distinguish three important types of social relationships among Weibo users: following (follower), followed (followee) and following reciprocally (mutual fan)[4]. Next, we will empirically study the effects of these three relationships on tag similarity. At first, we define two types of tags for a Weibo user u.

Definition 1 (Real Tags). If u originally labels him/erself with some tags, these tags are referred to as u's *real tags* and denoted by RT_u.

Definition 2 (Collective Tags). The tags that are most frequently used by u's friends are referred to as u's *collective tags* and denoted by CT_u.

To find the tags in CT_u, we define a score function $tf(t)$ to quantify the likelihood that tag t belongs to CT_u. The $tf(t)$ function is defined as

$$tf(t) = \frac{r(t)}{\sum\limits_{t' \in T(Neg(u))} r(t')} \tag{1}$$

where $Neg(u)$ is u's friend group and $r(t)$ is the number of users in $Neg(u)$ who have used tag t. $T(Neg(u))$ represents the tag set used by the users in $Neg(u)$. We denote the score function as tf because it is equivalent to the term frequency in document retrieval. The larger the $tf(t)$ is, the more likely the tag t belongs to CT_u. If $|CT_u|$ is limited to k, we select the top-k tags from $T(Neg(u))$ according to $tf(t)$ value. In the following text, we refer to $tf(t)$ as the *frequency* based tag ranking score.

[4] In this paper, we often refer to these three social relationships in Weibo as friend.

Metrics of Evaluation: To justify the homophily in Weibo Tagging behavior, we compare CT_u with RT_u for those users having real tags. If the matching of CT_u and RT_u is more evident than the matching of RT_u and a random tag set, the homophily in tagging behavior is evident. In this paper, we use the following three metrics to evaluate matching performance of generated/recommended tags to a user's real tags (ground truth).
Precision (P@k): It is defined as the proportion of top-k recommended tags that are matched to the ground truth (i.e., they are in real tag set), averaged over all samples.
Mean Average Precision (MAP@k): It is the mean of the *average precision score* (AP) of top-k recommended tags for all samples. AP is defined as

$$AP@k = \frac{\sum_{i=1}^{k}(P(i) \times rel(i))}{H} \tag{2}$$

where $rel(i)$ is an indicator function equaling 1 if the i-th tag is matched, 0 otherwise. $P(i)$ is the matched proportion of top-i tags and H is total number of matched tags in all top-k tags.
Normailzed Discounted Cumulative Gain (nDCG@k): It is a famous metric to measure relevance level of search results to the query in IR systems [14]. For top-k recommended tags, the nDCG score can be calculated as

$$nDCG = \frac{1}{Z}\sum_{i=1}^{k}\frac{2^{rel(i)}-1}{log_2(i+1)} \tag{3}$$

where $rel(i)$ is the same as Eq. 2 and Z is the normalized factor. Compared with MAP, nDCG is more sensitive to rank position of recommended tags. In general, a user pays less attention to the tags listed behind, hence nDCG is better to evaluate recommendation performance.

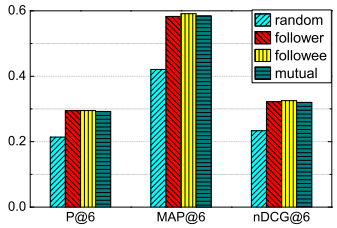


Fig. 3. Matching performance of CT_u to RT_u show that tag similarity is more evident for social friends than general users.

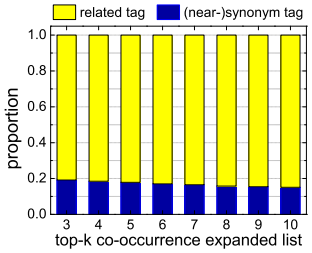


Fig. 4. The proportion of (near-)synonyms in top-k expanded list. Some expanded tags are (near-)synonyms of the parent tags, but most of them are complementary in semantic.

Results: Next, we show our empirical results which in general justify that *the users in Weibo who have close social relationships with each other tend to share similar tags*. Since the mean tag number of a Weibo user is 5.69 by our statistics, we only list the results of $|CT_u|$=6 (k=6) in Fig. 3 due to space limitation. We got consistent results under other sizes of CT_u. For comparison, we also compare RT_u with a random tag set. We randomly selected some users from the universal user set and used the most frequent tags of these users as the random tag set. The figure displays that under all metrics, random tag set have the worst matching performance and the collective tags from followees have the best performance. These results imply that *homophily is effective in tagging behaviors of Weibo users*. That is, Weibo friends tend to share similar tags. These results also justify the rationality of homophily-based tag recommendation, which is used as the basic scheme in our tag recommendation algorithm.

2.2 Co-occurrence in Tagging Behavior

From our dataset, we found that many Weibo users have more than one real tag. It inspires us to use tag cooccurrence for tag recommendation. That means, if two tags t_1 and t_2 co-occur with each other in many persons' real tag lists and t_1 has been recommended to a user, then t_2 also deserves to be recommended to this user. Tag co-occurrence was shown to be an effective mechanism for tag recommendation for photos in Flickr [25]. Next, we first give the ranking scheme of tag t' that co-occurs with t, then we justify the co-occurrence based tag recommendation for Weibo users by empirical studies.

Ranking: For a tag t recommended to a user, we first need to measure *the extent to which we recommend another tag t' that co-occurs with t to the user.* We may directly measure it by t''s co-occurrence frequency with t, denoted as $tf_t(t')$. Thus, the direct implementation of co-occurrence based tag recommendation is recommending tag t' with largest $tf_t(t')$ if t is recommended. The direct solution clearly favors those general tags with high occurrence frequency, such as 'music' and 'movie'. We need to suppress them to select informative tags. We import an idf factor to reflect this requirement. As in [10, 27], idf factor generally is defined as

$$idf(t') = log\frac{M - n(t') + 0.5}{n(t') + 0.5} \tag{4}$$

where $n(t')$ is the frequency of tag t''s co-occurrence with t. M is the user number of universal user set. Then, similar to TF-IDF in IR systems, we define a tf-idf score to measure the extent to which tag t' co-occurs with t as

$$s_t(t') = tf_t(t') \times idf(t') \tag{5}$$

Given this score function, we can enrich a tag list by homophily based recommendation.

Table 1. Co-occurrence tags ranked by tf-idf score.

machine learning	**tour**	**advertisement**
data mining	food	media
NLP	movie	marketing
recommender sys.	fashion	communication
information retrieval	music	design
computer vision	listen to music	photography
pattern recognition	80s	Internet
A.I.	freedom	innovation
big data	travel	movie
search engine	photography	art
Internet	indoorsy	fashion

Results: Next, we justify the co-occurrence expansion by case studies on three typical tags 'machine learning', 'tour' and 'advertisement', which are called as *parent tags* of their co-occurring tags. In Table 1, we list the top-10 tags ranked by $s_t(t')$ that co-occurs with the three parent tags. These tags are the candidates to enrich a recommended tag list and called as *expanded tags*. From the table, we can see that most expanded tags are semantically related but different from their parent tags. All these related tags often tend to be co-used by users, e.g., 'machine learning' is very related to 'data mining' and 'A.I.', 'design'ing an 'advertisement' needs 'innovation'. It is desirable to recommend these semantically different but related co-occurring tags so that the recommended tags are fully informative and expressive to characterize a user.

Moreover, we can also find some synonyms or near-synonym from the expanded tags. For example, in Table 1, 'travel' is very semantically close to 'tour', so does 'media' to 'communication'. We next quantify the extent to which synonyms occur in the expanded list. To do this, we first selected 1000 most frequently used tags as the parent tags. For each of them, we summarized the proportion of the (near-)synonym tags that occur in its expanded tag list. We will introduce our approach to distinguish (near-)synonym tags in Sec. 3.3. We reported the average proportions over all parent tags under different top-k expanded tags. The results are shown in Fig. 4 where (near-)synonym tags account for 15%$\sim$20% in the expanded list. It shows that most of expanded tags are meaningful. On the other hand, it also implies that we still need to remove the semantic redundancy caused by the (near-)synonyms. This problem can be solved by our CKG (Chinese knowledge graph) based approach that will be discussed in the next section.

3 Tag Recommendation Algorithm

In this section, we elaborate our tag recommendation algorithm which contains three major steps, as shown in Alg. 1. For a user u, our algorithm generates k recommended tags ordered by a ranking score. In the first step (line 2), we generate candidate tags by homophily based recommendation scheme. In the second step (line 5 to 9), we expand the tag list by the co-occurrence based recommendation scheme. In the third step (line 11 to 18), we remove all semantically redundant tags by a CKG based method.

3.1 Step 1: Recommendation by Homophily

According to the empirical results of Sec. 2.1, i.e., close social relationships imply similar tags, we can profile a Weibo user by his/er collective tags. This strategy can solve the data sparsity problem of Weibo tags. Recall Eq. 1, we directly collect the tags from u's friends, i.e., the direct neighbors of u, to constitute CT_u. This naive approach has two weaknesses. First, it will fail if no direct neighbors have real tags. Second, it does not take into account the intimacy between two friends. Next, we will improve it by taking into account indirect neighbors' information and user intimacies. We use tag propagation to materialize the effects of these factors. To better explain our algorithm, we first give some preliminary definitions.

Definition 3 (Weibo Influence Graph). The Weibo influence graph $G(V, E, w)$ is an edge-weighted directed graph, where V is user set and E is influence edge set. Each directed edge $e_{u\to v}$ indicates the social influence from user u to user v. Furthermore, we assign a weight w_{uv} to this edge to quantifies the extent to which u can influence v through it. In general, a followee has much more influence on his/er follower than the vice versa that is indicated by Fig. 3. Hence, for a better interpretation of our algorithm, we assume that only followee can influence his/er followers resulting in tag propagation from followees to followers only. Specifically, if and only if user v follows u, there is an edge $e_{u\to v}$ in the influence graph. We further set w_{uv} as the frequency that v retweets u in a given period[5].

Based on the Weibo influence graph, we further define *social influence* which characterizes the intimacy between two Weibo users. It is similar to the influence proposed by Mashiach et al. for optimizing PageRank algorithm [2].

Definition 4 (Social Influence). For a directed path $p = (u_0, u_1, ..., u_r)$ in G, the *social influence* along p from u_0 to u_r equals to

[5] The frequency of mention (@username) and comment can also be used to quantify the influence weight between Weibo users. Our experimental results show that the selection of weighting scheme does not affect the performance of our algorithm.

Algorithm 1 Tag recommendation algorithm with three steps.

Input: a Weibo user u; parameter k, q, λ, α;
Output: recommended tag list;

```
C ← φ;
compute S_u; //Step1: recommendation by homophily.
i ← 1;
while |C| < k do
    k' ← k × i; // begin Step 2: expansion by co-occurrence.
    C ← C ⋃ {top-k' tags ranked by s(t)};
    for each tag t in C in the descending order of s(t) do
        C ← C ⋃ {top-q tags ranked by s_t(t_i)};
    end for
    set all newly added tags' parents;
    Rank tags in C by ŝ(t) defined in Eq. 10; //begin Step 3.
    for each tag t in C in the descending order of ŝ(t) do
        for each tag t' ordered after t do
            if sim(t, t') ≥ α then
                remove t' from C;
            end if
        end for
    end for
    i ← i + 1;
end while
return the top-k tags in C;
```

$$si(p) = \prod_{i=0}^{r-1} \frac{w_{u_i u_{i+1}}}{\sum_{u:u \to u_{i+1}} w_{u u_{i+1}}} \tag{6}$$

where u is u_{i+1}'s in-neighbor in G. Let $P_r(v, u)$ be the set of all paths of length r from v to u, thus the social influence of v on u at radius r is

$$si_r(v, u) = \sum_{p \in P_r(v,u)} si(p). \tag{7}$$

Furthermore, we define $si_0(u, u) = 1$ and $si_0(v, u) = 0$ for all $v \neq u$. Then, the total social influence of v on u is $si(v, u) = \sum_{r=0}^{\infty} si_r(v, u)$.

Computation: Suppose there are overall N tags in G, the first step of our algorithm aims to calculate a *tag score vector* $\boldsymbol{\mathcal{S}}_u = [s(1), ..., s(N)] \in \mathbb{R}^N$ for a user u, in which $s(j)$ $(1 \leq j \leq N)$ quantifies the extent to which tag j can profile u, i.e., the ranking score of candidate tag j. To consider the influence of indirect neighbors, we let the tags of indirect neighbors propagate along the path in the influence graph. Intuitively, *if a user v has a more significant influence on u (i.e., larger $si(v, u)$), u will be more tending to use v's tags to profile him/erself*. To reflect these facts, we define:

$$\boldsymbol{\mathcal{S}}_u = \sum_{v \in V} si(v, u) \boldsymbol{\mathcal{T}}_v = \sum_{v \in V} \sum_{j=0}^{r} si_j(v, u) \boldsymbol{\mathcal{T}}_v \tag{8}$$

where $\boldsymbol{\mathcal{T}}_v \in \mathbb{R}^N$ is v's real tag distribution vector and its entry $t_j = 1/n (1 \leq j \leq N)$ if user v originally uses tag j, otherwise $t_j = 0$. n is the number of user v's real tags and $\sum t_j = 1$. Refer to Eq. 6 and Eq. 7, we can recursively compute the social influence of user v on user u at radius r as

Algorithm 2 Step1: Computing u's tag score vector $\boldsymbol{\mathcal{S}}_u$.

Input: u, r;
Output: $\boldsymbol{\mathcal{S}}_u$;
1: $\boldsymbol{\mathcal{S}}_u \leftarrow \phi$;
2: $layer_0 \leftarrow u$;
3: $si_0(u,u) \leftarrow 1$;
4: **if** u has origin tags **then**
5: $\quad \boldsymbol{\mathcal{S}}_u \leftarrow \boldsymbol{\mathcal{T}}_u$;
6: **end if**
7: **for** i=1 to r **do**
8: $\quad layer_i \leftarrow$ {all in-neighbors of the nodes in $layer_{i-1}$};
9: $\quad$ **for** $\forall v \in layer_i$ **do**
10: $\quad\quad$ **if** v has real tags **then**
11: $\quad\quad\quad$ **for** each v's out-neighbor x **do**
12: $\quad\quad\quad\quad si_i(v,u) \leftarrow \sum_{x:v\to x} \frac{w_{vx}}{\sum_{v':v'\to x} w_{v'x}} si_{i-1}(x,u)$;
13: $\quad\quad\quad$ **end for**
14: $\quad\quad\quad \boldsymbol{\mathcal{S}}_u \leftarrow \boldsymbol{\mathcal{S}}_u + si_i(v,u) \times \boldsymbol{\mathcal{T}}_v$;
15: $\quad\quad$ **end if**
16: $\quad$ **end for**
17: **end for**
18: **return** $\boldsymbol{\mathcal{S}}_u$;

$$si_r(v,u) = \sum_{x:v\to x} \frac{w_{vx}}{\sum_{v':v'\to x} w_{v'x}} si_{r-1}(x,u) \tag{9}$$

where x is v's out-neighbor who has a path of $r-1$ length to u at least, and v' is x's in-neighbor. That is, the social influence of v on u at radius r equals to the weighted average influence of v's out-neighbors on u at radius $r-1$. This implies that we can compute $\boldsymbol{\mathcal{S}}_u$ iteratively as shown in Alg. 2. The computation starts from u. In the i-th iteration (line 7 to 17), for each user v that is i steps away from u and have real tags, we calculates its social influence on u by summing up the weighted social influences on u of each v's out-neighbor x at radius $i-1$.

Optimization: Next, we optimize above computation from two aspects.

1. *Setting A Shorter r.* Obviously, the computation cost of Eq. 8 is unbearable if r is big. Refer to the observations on Twitter that more than 95% of information diffusion is less than the scope of 2 hops from the origin [15], we can set $r \leq 2$ in the real applications. We will present how to learn this upper bound of r in the experiment section.

2. *Suppressing General Tags.* Similar to co-occurrence tag expansion, we should suppress the tags that are too generally used by all users in order to find the specific and informative tags. Therefore, we also import an idf factor matrix D into Eq. 8. That is replacing $\boldsymbol{\mathcal{T}}_v$ with $\boldsymbol{\mathcal{T}}_v D$, where $D = diag[d_1, ..., d_N]$ is an $N \times N$ diagonal matrix and each non-zero entry $d_j (1 \leq j \leq N)$ is defined as Eq. 4. After $\boldsymbol{\mathcal{S}}_u$ is computed, we rank all tags according to $s(j)$ and then select the top-k tags as the candidate set, namely C, that will be fed as the input of Step 2. k is the number of tags to profile a user.

3.2 Step 2: Expansion by Co-occurrence

We have shown in Sec. 2.2 that co-occurrence is also an important tag recommendation mechanism. Therefore, we use this mechanism to enrich the recommended tags. The input of this step is the ranked tag list C generated in Step 1. The output is a new ranked list consisting of C and other expanded tags.

In Step 2, for each tag $t \in C$, in order to generate its *expansion list*, we select the top-q co-occurring tags, namely t_i, according to $s_t(t_i)$ value (refer to Eq. 5). If a

co-occurring tag t_i can be found in more than one expansion list, t_i will only join the expansion list of the tag t having the maximal $s(t)$. We refer to such t as t_i's *parent tag*, namely $p(t_i)$. Thus, for each t_i, $p(t_i)$ is unique. If an expanded tag has existed in C, we just ignore it. As a result, at most $k \times q$ new tags can be discovered. Let C' be the new candidate tag list after expansion. Thus, $C' - C$ contains all newly expanded tags.
Re-ranking: After we generated the new recommendation tag set C', we need to re-rank each member of C'. The key of the new ranking is to ensure that the tags in $C' - C$ can fairly compete with those tags in C. To meet this requirement, we define a new ranking score $\hat{s}(t_i)$ for each tag $t_i \in C'$:

$$\hat{s}(t_i) = \begin{cases} s(t_i) & t_i \in C; \\ \lambda \times s(p(t_i)) \times \frac{s_{p(t_i)}(t_i)}{Z} & \text{otherwise} \end{cases} \tag{10}$$

where $\lambda \in (0, 1)$ is a damping parameter, Z is used for normalization and set as the maximal $s_t(t_i)$ of all t_is that co-occur with t. If t_i is one of the original tag found in Step 1 (i.e., $t_i \in C$), we directly use the $s(t_i)$ as its new score. Otherwise, we inherit the score from $p(t_i)$'s ranking score $s(p(t_i))$ generated in Step 1 and use λ and $\frac{s_{p(t_i)}(t_i)}{Z}$ as two multiplicators to suppress it ($s_{p(t_i)}(t_i)$ is also defined according to Eq. 5). Since $p(t_i)$ is unique, $\hat{s}(t_i)$ is well defined.

The rationality of the new score is two-fold:
1. $\hat{s}(t_i)$ should be smaller than $s(p(t_i))$. The definition can ensure this because $\lambda \in (0, 1)$ and $\frac{s_{p(t_i)}(t_i)}{Z} \in (0, 1]$. On the other hand, to ensure t_i is competitive enough, we usually set $\lambda \geq 0.5$.
2. For any two tags t_i, t_j in one tag t's expansion list, $\hat{s}(t_i) < \hat{s}(t_j)$ should hold if $s_t(t_i) < s_t(t_j)$. It is not difficult to prove that $\hat{s}(t_i)$ satisfies the requirement.

3.3 Step 3: Removing Semantic Redundancy

As pointed out in [8], users often tag the same resource with different terms for their various habits or recognition. Similarly, Weibo users may use different terms to express the same or close semantics. As a result, many synonyms or near-synonyms tend to exist in Weibo tags. For example, tag 'tour' and 'travel' are both widely used in Weibo. Thus, the candidate tag set may have some tags of the same or similar semantics. These tags are redundant and should be avoided due to space limitation of a Weibo user's tags. For this purpose, we first construct a *Chinese Knowledge Graph* (CKG in short) and then use an *Explicit Semantic Analysis* (ESA in short) [7] based model to represent a tag's semantics through the concepts in CKG.

The CKG is a big graph constituted by millions of concepts and entities extracted from online encyclopedias such as Baike[6]. Each concept can be classified into one or more categories and there exists a unique Web article to explain it. In each Web article, there are many hyperlinks referring to other concepts, namely *reference concept*. These hyperlinks constitute the edges of CKG (refer to Fig. 2). A concept can be referred to by more than one article. As well, a concept can also be referred to more than once in an article. Thus, for a reference concept, we can use the concepts whose articles refer to it, to represent its semantics. The number of referring also allows us to calculate a tf-idf score to select expressive concepts.

Based on above idea, we can quantify the semantics of a Weibo tag by first mapping it into Baike concepts, i.e., the concepts in CKG. Specifically, given a tag a and a Baike concept b, we map a to b if $s_a = s_b$ or s_b is the maximal substring of s_a, where s_a and s_b are the name strings of a and b, respectively. Under this mapping scheme, we can find an appropriate Baike concept for 88.7% of Weibo tags.

[6] http://baike.baidu.com

According to ESA, two tags are considered semantically related if their mapped concepts are co-referred to in the same article pages of CKG. The more such articles can be found, the more semantically related the two tags are. Based on it, we first formalize a tag's semantic representation as follows. Suppose CKG has L concepts in total, the semantic interpretation of a tag i can then be represented by a *concept vector* defined as $\boldsymbol{\mathcal{C}} \in \mathbb{R}^L$ of which each entry, namely c_j, represents the semantic relatedness of concept j to tag i. c_j can be calculated as the tf-idf score of tag i in concept j's article. We notice that many concepts in CKG are quite general and cover a wide range of topics. These concepts in general have less semantic descriptiveness on a tag than those specific concepts. Hence we need to suppress these general tags. Intuitively, the concepts belonging to more categories are more general than the concepts belonging to less categories. Consequently, we further define

$$c_j = \frac{ts_j(i)}{|cat(j)|} \tag{11}$$

to punish general tags, where $ts_j(i)$ is the tf-idf score and $cat(j)$ is concept j's category set.

Then, given two tags i and j, we can measure their semantic similarity by computing the cosine similarity of $\boldsymbol{\mathcal{C}}_i$ and $\boldsymbol{\mathcal{C}}_j$, i.e., $sim(i,j) = cosine(\boldsymbol{\mathcal{C}}_i, \boldsymbol{\mathcal{C}}_j)$. According to the definition of concept vector, the larger the $sim(i,j)$ is, the more possible that i and j are (near-)synonyms. The detailed procedure of removing semantically redundant tags is shown in Alg. 1. We first sort the tags in C by the descending order of $\hat{s}(t)$ value. For each tag t in the ordered list, we start an inner loop to scan each tag t' ordered after t. If $sim(t,t')$ is larger than a threshold α, we remove t' from C. Finally, if C contains more than k tags, we just return the top-k tags ranked by $\hat{s}(\cdot)$ function (refer to Eq. 10).

3.4 Parameter Learning

There are several parameters in our tag recommendation algorithm, i.e., r in Step 1, q and λ in Step 2 and α in Step 3. In this subsection, we introduce how to set the best parameter values.

To find the best α, we used the synsets in Cilin[7] (a popular Chinese synonym database) as positive samples and manually labeled non-synonym paris as the negative sample. We use these samples as the training dataset to train a binary classification model. Then we found that α=0.007 is the most effective threshold for distinguishing (near-)synonyms.

Next, we introduce how to learn the best value for q, r and λ. We first introduce how to evaluate the goodness of a recommended tag set. For a user with real tags, we can take his/er real tags as the ground truth. We can compare the recommended tags to the ground truth for the evaluation. In Sec. 2.1, we use the results of *exact match* for the comparison. But it is too strict for tag recommendation. For example, it is reasonable to recommend 'tour' to a user with a tag of 'travel' although the two tags are lexically different. To relax the match, we use the aforementioned cosine similarity between concept vectors to measure the match between two tag sets.

More formally, suppose our algorithm of the parameter setting θ recommends u with a tag set, namely $T(u,\theta)$. Let $\boldsymbol{\mathcal{C}}_u(\theta)$ be the concept vector of $T(u,\theta)$'s. According to Eq. 11, each entry of $\boldsymbol{\mathcal{C}}_u(\theta)$, namely c_j, can be defined as

$$c_j = \sum_{t \in T(u,\theta)} \frac{ts_j(t)}{|cat(j)|} \times \hat{s}(t,\theta) \tag{12}$$

where t is a tag in $T(u,\theta)$ and $\hat{s}(t,\theta)$ is t's score derived by our algorithm under the setting θ. For computing u's real tag set RT_u's concept vector, namely $\bar{\boldsymbol{\mathcal{C}}}_u$, we set

[7] http://www.datatang.com/data/42306/

$\hat{s}(t) = 1/|RT_u|$ because we can not acquire u's extent to which s/he prefers to a real tag. Then, we propose an objective function $\mathcal{F}$ to measure the semantic similarity between $T(u, \theta)$ and RT_u as

$$\mathcal{F}(u, \theta) = sim(T(u, \theta), RT_u) = cosine(\boldsymbol{\mathcal{C}}_u(\theta), \bar{\boldsymbol{\mathcal{C}}}_u)$$

.Thus, the best parameter setting (including q, r and λ) should be

$$\theta^* = \arg\max_{\theta \in \Theta} \mathbb{E}(\mathcal{F}(u, \theta)) = \arg\max_{\theta \in \Theta} \frac{\sum_{u \in U} \mathcal{F}(u, \theta)}{|U|} \tag{13}$$

. U is the training user set consisting of the seed users having real tags in our Weibo dataset. Finally, we found that $q = 1, r = 2, \lambda = 0.5$ are the best θ in our tag recommendation algorithm.

4 Evaluation

In this section, we evaluate the performance of our tag recommendation algorithm through the comparisons with some state-of-the-art methods. We not only present the match performance of our recommendation algorithm, but also display the effectiveness of the recommended tags on inferring user profiles.

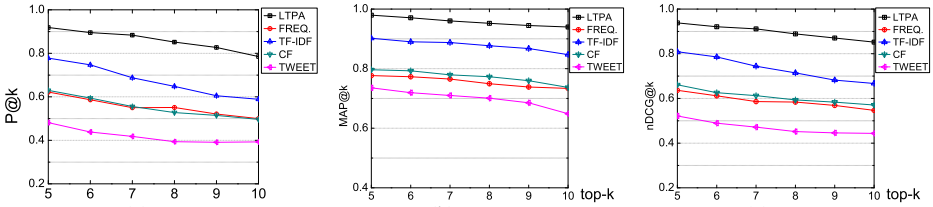
4.1 Experimental Settings

We first introduce the evaluation method and the competitors of our algorithm.
Human Assessments: One direct way to assess the recommended tags is comparing them with the real tags since the real tags are each Weibo user's preferences. However, nearly half of Weibo users have no real tags. So we have to resort to human assessments for evaluating the recommended tags. Specifically, we inquired each test Weibo users whether s/he will accept the recommended tags. Each user can select an option of yes, no and $unknown$ for a tag. We only take the tag of yes as matched tag.
Baselines:
1. FREQ.: The first baseline is a naive method because it selects the recommended tags merely by ranking the frequency of candidate tags used by a user's followees, i.e., collective tags.
2. TF-IDF: This baseline recommends the tags according to the TF-IDF scheme.
3. CF: The CF approach has been proposed in [25] to recommend tags for a Flickr image based on tag co-occurrence mining. That is, for a user with real tags, we recommend to him/her with some tags that are co-used with his/er own tags by many other users. In fact, this method can be viewed as an item-based collaborative filtering approach when we regard a tag as an item and the tags co-used by a user as similar or related items. Clearly, this recommendation method can not be applied for the users without real tags.
4. TWEET: This approach is a content-based recommendation scheme which has been widely used in previous recommender systems [6, 9]. This approach extracts some keywords from a user's tweets as the recommended tags since a user's tweets are direct indicators of users interests or preferences.

In our algorithm, the tags are generated from local neighbors within radius 2 (r=2). Hence, we name our algorithm as *Local Tag Propagation Algorithm* (LTPA in short). Besides r, the parameters q and λ of our algorithm were also set as the best values tuned by corresponding learning models (see Sec. 3.4) in the experiments.

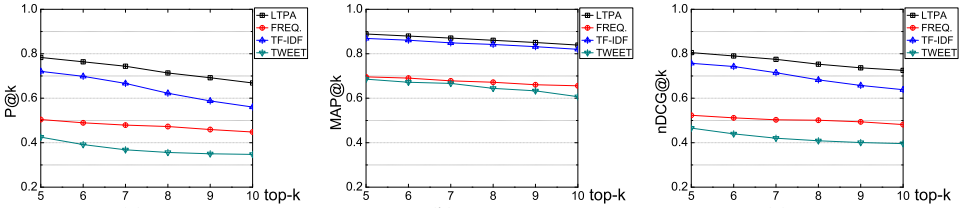
4.2 Effectiveness

We justify our algorithm's effectiveness from two aspects. We first present the global match performance of our tag recommendation algorithm by comparing to the baselines. Then we justify the effectiveness of each step of our algorithm.



(a) P@k (b) MAP@k (c) nDCG

Fig. 5. Human assessment results of the recommended tags to the test users having real tags.



(a) P@k (b) MAP@k (c) nDCG

Fig. 6. Human assessment results of the recommended tags to all the test users.

Global Performance: From the 3000 seed users in our dataset, we randomly selected 500 users as the test users in our experiments in which the spam users were excluded. Then, we designed two groups of experiments to recommend tags to these test users. In the first group, we compared all recommendation algorithms on the 268 test users having real tags since CF can only work on the users with tags. In the second group, we compared all competitors except for CF on all 500 test users. The human assessment results are shown in Fig. 5 and Fig. 6, respectively. The results show that all algorithms perform the best when recommending top-5 tags. It proves that the algorithms can rank the best tags to a top position. The results also reveal that LTPA performs the best in all cases. The superiority of LTPA and TF-IDF over FREQ. justifies that the effectiveness of idf factor to discover informative and personalized tags for profiling a user. We will illustrate it by case studies in the next subsection. TWEET almost performs the worst in all cases, implying that *the keywords directly extracted from tweets are generally not appropriate for user profiling*. Further investigation on the tweet content reveals that, most tweet keywords are *colloquialisms* or *person name* of friends and newsmakers. For example, 'Diaos' is a new Internet vocabulary and is widely used by Chinese youngsters. These words produced due to the oral and informal language style in short tweets (less than 140 characters) can not accurately and completely characterize a user.

Effectiveness of Each Step: In Sec. 2.1, we have justified the rationality of homophily based recommendation. Next, we present the recommendation results after we add Step 2 and Step 3 incrementally into our algorithm to justify the co-occurrence based expansion and removing semantic redundancy. Since our algorithm has the best performance when k=5, we only evaluated our algorithm by recommending top-5 tags to the test users.

Step 2: To justify Step 2, we investigated the expanded tags generated by our algorithm consisting of Step1 and Step 2. We found that 75.11% of the expanded tags are newly discovered tags. In average, about 35.37% of these newly expanded tags were labeled as matched by the volunteers. These results imply that co-occurrence based expansion is necessary and effective in enriching the recommended tag list.

Step 3: Then, we ran the whole algorithm consisting of the three steps. We found that 14.55% of the tags after Step 2 were identified as (near-)synonyms of the previous tags.

By surveying the volunteers' acceptance about the removed tags, we found that 74.7% of these (near-)synonyms identified by Step 3 are really redundant. These results justify the effectiveness of removing semantic redundancy.

Table 2. Inference accuracy in four profile categories.

category	accuracy
location	94.64%
occupation	76.47%
education	95.24%
religion	99.21%

Table 3. Case studies to justify inference performance of the recommended tags.

user	algorithm	tag list
userA	real tags	music, fashion
	CF	movie, food, listen to music, tour, 80s
	TWEET	Jehovah, Miss HongKong, beauty, child, good man
	FREQ.	Christian, food, movie, 80s, tour
	TF-IDF	Christian, Bible, Emmanuel, micro fashion, tide
	LTPA	Christian, Bible, faith, God's baby girl, God's child
userB	TWEET	Shantou (a Chinese city), WeChat, Internet, Shantou people, girl
	FREQ.	tour, food, movie, Internet, music
	TF-IDF	machine learning, Internet, data mining, Fudan University, technology
	LTPA	machine learning, IT, Internet, Fudan University, data mining

4.3 Inference of User Profiles

Many users are reluctant to publish their profiles, i.e., location, professions and religion, possibly due to the privacy concern. Hence, accurately inferring user profiles is very important for better understanding the users who have no tags or no informative tags. For the users who do not introduce themselves completely, the recommended tags can be used to infer the absent user characteristics. Identifying user profile characteristics can contribute to many real applications such as maintaining social cliques, search for target user and etc. To test the performance of inferring user profiles, we ran our algorithm on the test users to recommend top-5 tags. Then, we filtered out the test users whose recommended tags contain profile information and evaluated inference accuracy by inquiring the users. Table 2 lists the inference accuracy of tags generated by our algorithm w.r.t. four basic profile information: location, profession, education and religion. The results verify that our algorithm is effective on inferring user profiles.

Case Studies: Finally, we give two case studies to highlight our algorithm's effectiveness on recommending personalized and informative tags to enrich a user's profile.

Case 1: User A in Table 3 is a test user who has real tags. We can see that user A's real tags uncover nothing about her religion. CF can not recommend any tags indicating her religion either. In TWEET and FREQ., there is only one word, i.e., 'Jehovah' and 'Christian', implying user A's religion (Christianism). In contrast, TF-IDF and LTPA can recommend more than one tag that apparently reveal user A's religion. It is because these two algorithms can find more personalized and informative tags through idf factor. By investigating user A's tweets, we confirmed that she is really a Christian.

Case 2: Another test user B has no real tags. As a result, CF can not be applied on this user. From Table 3, we find that FREQ. only reveals general interests of youngsters. TWEET can only find keywords about his hometown ('Shantou'). In contrast, TF-IDF and LTPA can recommend more personalized and informative tags. From these tags we can confidently infer that user B is a university student (the university name is anonymous for blind review) who is interested in machine learning and data mining. In fact, user B is a student volunteer in the data mining laboratory of Fudan University.

5 Related Work

Tag Recommendation and Social Recommender: Most previous works of tag recommendation were employed on a triplet basis, i.e., user, tag and resource [11, 29, 18, 13], instead of tagging a user. Xu et al. [29] proposed a set of criteria for a high quality tag. Based on these criteria, they further proposed a collaborative tag suggestion algorithm to discover the high-quality tags. Song et al. [26] recommended tags for a document according to the mutual information between words, documents and tags. In addition, Sigurbjornsson et al. [25] presented some recommendation strategies based on tag co-occurrence. Liu et al. [17] introduced a tag ranking scheme to automatically rank the tags associated with a given image according to tag relevance to the image content. All these methods were designed on the premise that each tagged object already has tags resulting in vulnerability to data sparsity towards tagging Weibo users. Similar to Step 1 in our algorithm, many scholars tried to improve recommendation performance by exploiting social context. These systems are generally called *social recommender* [3]. Ma et al. [18] used social relationships to solve the cold start problem of CF, but they mainly focused on rating objects instead of persons. Ben-Shimon et al. [4] explicitly quantified user similarity by computing their distances in the social graph without considering personality. Quijano-Sanchez et al. [22] resorted to a TKI survey upon users to acquire personality values which is not feasible to real on-line applications. Hotho et al. [13] also used a PageRank-based model to rank tags but they did not consider the semantic redundancy of tags.
Tag Semantics: One of the prerequisites to study the user tagging behavior is understanding the semantic of tags [1]. In general, to understand tag semantics, tags should be mapped into a thesauri or a knowledge base. E.g., mapping Flickr tags [25] and Del.icio.us tags [5] into WordNet, or mapping tags into Wikipedia categories by the content of tag-associated objects [21]. Moreover, some meta graphs are also constructed for understanding tags, such as a tag graph encoding co-occurrence relationships among tags [30, 17]. Given that the low tag coverage of WordNet, we resort to Wikipedia-like encyclopedia, i.e., CKG in this paper. Furthermore, we improve ESA [7] by taking into account the categories of CKG concepts to improve the precision of a tag's semantic interpretation.
User Profile Inference: Sadilek et al. [23] presented a system to infer user locations and social ties between users. Mislove et al. [20] tried to infer user profile based on the open characteristics of a fraction of users. These mechanisms are not as flexible as our approach because they only work under the assumption that characteristics of some users have been uncovered in advance. The authors in [16] proposed an influence based model to infer home locations of Twitter users. Although their work also resorted to social relationships for an accurate inference, their model can only be used to infer location and needs expensive analysis of tremendous tweets. In contrast, our solution mainly depends on crawling and analyzing tags that are less costly than processing on tweets.

6 Conclusion

Motivated by many real applications built upon user profiles, we dedicate our efforts in this paper to tag recommendation for Weibo users. We conducted extensive empirical studies to unveil effective tag recommendation scheme based on which we proposed an integrated tag recommendation algorithm consisting of three steps, i.e., tag recommendation based on local tag propagation, tag expansion by co-occurrence and CKG-based elimination of semantically redundant tags. Extensive experiments validate that our algorithm can recommend more personalized and informative tags for profiling Weibo users than the state-of-the-art baselines.

References

1. Ames, M., Naaman, M.: Why we tag, motivations for annotation in mobile and online media. In: Proc. of CHI (2007)
2. Bar-Yossef, Z., Mashiach, L.T.: Local approximation of pagerank and reverse pagerank. In: Proc. of CIKM (2008)
3. BELLOGIN, A., CANTADOR, I., DIEZ, F., CASTELLS, P., CHAVARRIAGA, E.: An empirical comparison of social, collaborative filtering, and hybrid recommenders. ACM Transactions on Intelligent Systems and Technology 4 (2013)
4. Ben-Shimon, D., Tsikinovsky, A., Rokach, L., Meisles, A., Shani, G., Naamani, L.: Recommender system from personal social networks. In: Proc. of AWIC (2007)
5. Cattuto, C., Benz, D., Hotho, A., Stumme, G.: Semantic grounding of tag relatedness in social bookmarking systems (2008)
6. Chen, J., Geyer, W., Dugan, C., Muller, M., Guy, I.: Make new friends but keep the old, recommending people on social networking sites. In: Proc. of CHI (2009)
7. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using wikipedia-based explicit semantic analysis. In: Proc. of IJCAI (2007)
8. Gupta, M., Li, R., Yin, Z., Han, J.: Survey on social tagging techniques. In: Proc. of SIGKDD (2010)
9. Hannon, J., Bennett, M., Smyth, B.: Recommending twitter users to follow using content and collaborative filtering approaches. In: Proc. of RecSys (2010)
10. Hassanzadeh, O., Consens, M.: Linked movie data base. In: Proc. of LDOW (2009)
11. Heymann, P., Ramage, D., Garcia-Molina, H.: Social tag prediction. In: Proc. of SIGIR (2008)
12. Hofmann, T.: Latent semantic models for collaborative filtering. ACM Transactions on Information Systems 2, 89-115 (2004)
13. Hotho, A., Jschke, R., Schmitz, C., Stumme, G.: Information retrieval in folksonomies: Search and ranking. LNCS 4011, 411-426 (2006)
14. Jarvelin, K., Kekalainen, J.: Cumulated gain-based evaluation of ir techniques. ACM Transactions on Information Systems 20, 422-446 (2002)
15. Kwak, H., Lee, C., Park, H., Moon, S.: What is twitter, a social network or a news media? In: Proc. of WWW (2010)
16. Li, R., , Wang, S., Deng, H., Wang, R., Chang, K.C.C.: Towards social user profiling: Unified and discriminative influence model for inferring home locations. In: Proc. of SIGKDD (2012)
17. Liu, D., Hua, X.S., Yang, L., Wang, M., Zhang, H.J.: Tag ranking. In: Proc. of WWW (2009)
18. Ma, H., Yang, H., Lyu, M.R., King, I.: Sorec: Social recommendation using probabilistic matrix factorization. In: Proc. of CIKM (2008)
19. McPherson, M., Smith-Lovin, L., Cook, J.: Birds of a feather: Homophily in social networks. Annual Review of Sociology 27, 415 - 445 (2001)
20. Mislove, A., Viswanath, B., Gummadi, K.P., Druschel, P.: You are who you know: Inferring user profiles in online social networks. In: Proc. of WSDM (2010)
21. Overell, S., Sigurbjornsson, B., van Zwol, R.: Classifying tags using open content resources. In: Proc. of WSDM (2009)
22. Quijano-Sanchez, L., Recio-Garcia, J.A., Diaz-Agudo, B., Jimenez-Diaz, G.: Social factors in group recommender systems. ACM Trans. on Intelligent Systems and Technology 4 (2013)
23. Sadilek, A., Kautz, H., Bigham, J.P.: Finding your friends and following them to where you are. In: Proc. of WSDM (2012)
24. Schafer, J., Konstan, J., Riedi, J.: Recommender systems in e-commerce. In: Proc. of EC (1999)
25. Sigurbjornsson, B., van Zwol, R.: Flickr tag recommendation based on collective knowledge. In: Proc. of WWW (2008)
26. Song, Y., Zhuang, Z., Li, H., Zhao, Q., Li, J., Lee, W.C., Giles, C.L.: Real-time automatic tag recommendation. In: Proc. of SIGIR (2008)
27. Wang, J., Hong, L., Davison, B.D.: Tag recommendation using keywords and association rules. In: Proc. of RSDC (2009)
28. Weng, J., Lim, E.P., Jiang, J., He, Q.: Twitterrank: finding topic-sensitive influential twitterers. In: Proc. of WSDM (2010)

29. Xu, Z., Fu, Y., Mao, J., Su, D.: Towards the semantic web: Collaborative tag suggestions. In: Proc. of Collaborative Web Tagging Workshop in WWW (2006)
30. Zhou, T.C., Ma, H., Lyu, M.R., King, I.: Userrec: A user recommendation framework in social tagging systems. In: Proc. of AAAI (2010)