Neurocomputing 204 (2016) 8–16

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Flickr group recommendation using rich social media information

Cong Guo [1], Bei Li, Xinmei Tian *

Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei Anhui, China

ARTICLE INFO

Article history:
Received 9 March 2015
Received in revised form
3 July 2015
Accepted 24 August 2015
Available online 9 April 2016

Keywords:
Social Network
Group Recommendation
Collaborative Filtering

ABSTRACT

Today online social media communities have spanned the globe, browsing news from social networks almost becomes an essential part in our daily life. Groups organized by users always share something interesting. Joining groups which fit the users' tastes will help them to obtain information. However, traditional group recommendation methods usually focus on how to recommend an item to a group of users. In this paper, we study how to recommend groups to an individual user and reveal the factors which push a user to join groups. In social networks, a commonly adopted recommendation method takes advantage of the tastes of a user's trust neighbors and recommends groups which his/her neighbors have joined. It will performs poorly for the inactive users who have few trust neighbors. To overcome this problem, we try to find users' similar neighbors using tag information, which is not only from users' photos but also from their favorite photos and the common friend information. Hence we propose a group recommendation scheme utilizing users' trust neighbors and similar neighbors' tastes. We do the experiments on a real-world Flickr dataset and obtain a promising result especially for inactive users.

1. Introduction

In recent years, social networks have become an essential part in people's daily life and one of the major approaches by which people get fresh news. Social networks usually contain various media data with which their users can express themselves lively. Thus, users can communicate with others conveniently and share/find the contents they are interested in. To gather the users who share the same hobbies, social networks nowadays allow users to join the interest groups which contain rich contents for some topics. For a social website, an efficient group recommendation system should help users find their favorite groups effectively. This work will be helpful for users and advertisements. Efficient group recommendation can therefore have a positive effect on both social network members and other recommendation applications.

We take Flickr, one of the most popular photo sharing social networks to study how to recommend groups to an individual user. Flickr as one of the oldest social networks has a large number of users and is reported to have millions of new images uploaded daily. Flickr offers many services to its users. It allows users to share and self-annotate their photos. Users can follow other users to browse photos and experience others' lifestyles. Groups in Flickr are self-organized. Users can create their own interest groups and join others. The availability of rich media data helps us explore the behavior of users from different views.

In previous work, most group recommendation methods studied how to recommend an item to a group of users [1,2]. In our work, we study how to recommend groups to users. It is an interesting topic. We need to study the tastes and the social relationship of users to find out the major factors which push the users to join a group. A number of recommendation techniques have been proposed, such as user-based collaborative filtering [3,4], item-based collaborative filtering [5,6], trust-aware collaborative filtering [7,8] and matrix factorization [9]. Collaborative filtering (CF) is the most commonly used technique in social networks. Researchers have proposed a number of CF algorithms. Trust-aware CF makes the recommendation trustiness but suffers from the cold-start problem. User-based/item-based CF can find what the users like efficiently; however, the trust relation among users will be ignored. The matrix factorization method works well for the item recommendation, but is unsuitable for the binary group recommendation problem.

In this paper we propose a group recommendation model using the tastes of users' trust neighbors and similar neighbors, and make recommendation with the collaborative filtering algorisms. We take advantage of the trust-aware CF [7] and user-based CF [4] to handle the neighbors respectively.

Flickr has a direct link structure. A user can follow any other users. It is not like Facebook [10] which has an undirect link

* Corresponding author. Tel.: +86 18355102690.
E-mail addresses: gcong18@mail.ustc.edu.cn (C. Guo), afreylee@hotmail.com (B. Li), xinmei@ustc.edu.cn (X. Tian).
[1] Tel.: +86 15655170815.

http://dx.doi.org/10.1016/j.neucom.2015.08.131
0925-2312/

structure, and can well describe the trustiness among users. However, if a Flickr user A follows another user B, user A must be interested in user B's content or be familiar with B. So we can still assume that A trusts B. To find a user's trust neighbors, we explore the contact list of the user and assume all the users in the contact list to be his/her trust neighbors. Collaborative filtering is then used for group recommendation.

To find users' similar neighbors, we take advantage of users' tag information and common friend information. We try to find the content interests similarity among users from their uploaded/ favourite photos. Flickr allows its users to self-annotate their photos. Existing studies reveal that many tags provided by Flickr users are noise. There are only around 50% of those tags actually related to the photos [11]. It is because that tags annotated by different users will be different even for the same photo. Another reason is that users may have interests in something that they do not have the ability to take photos of them. For instance, a user loves the sea but he/she lives far away from it.

To overcome the disadvantages of the tags to some extent, we study the tag information and extract two kinds of content features. We order the different tags from a user's photos by frequency. The tags with high frequency will be precise enough to describe the interests of the user. To solve the various tags description problem, we crawl the tags from the photos which are in the user's favorite photo lists and also order them by frequency. The tags with high frequency from different annotators will be more general and reliable than the tags only from the owner. Thus we have two methods to describe the users' interests and can get the content similarities among users.

Another way to describe the similarity between two users is to count the common friend number. If two users always follow the same users, they may have some common interests. After employing the kernel alignment algorithm [12] to combine the similarity kernels, we use collaborative filtering for group recommendation.

We perform the group recommendation using the trust neighbors and similar neighbor's tastes. In fact, researchers find that there is a tendency that more friends will bring in more logins [13]. So we take the number of a user's followees to describe the active degree of him/her. We assign different users with different weights to make the combination. We find that users with high active degrees tend to join groups by social relationship, whereas users with low active degrees tend to join groups by interests.

In summary, this paper has the following contributions:

- We study a real-world dataset from Flickr. We find that no matter how active a user is, he/she tends to join in groups. So we can assume that browsing the contents of the interest groups is one of the main approaches which users adopt to obtain fresh news.
- We utilize users' similar neighbors tastes for group recommendation with the tag information and the common friend information. The self-annotated photos in Flickr can help us bridge the semantic gap to obtain a series of tags which present the users' interests. However only using the photos which users upload is not sufficient. So we expand the tag information from users' favorite photos and the common friend information. Combining the similarities from different views will improve the performance of our group recommendation.
- We study the relation between users and groups from two aspects. Recommendation using similar neighbors' tastes performances better on inactive users while recommendation using trust neighbors' tastes performances better on active users. We combine these two aspects and assign different users with different weights to make group recommendation.

This paper has published on International Conference on Security, Pattern Analysis, and Cybernetics(ICSPAC) 2014. Comparing with our previous work, we further study the factors which push users to join groups. We use collaborative filtering algorism with users' similar neighbors and trust neighbors' tastes respectively. When discovering users' similar neighbors, we keep using the tag information. Furthermore we extent another information that is common friend information to enrich the concept for us to discover users' similar neighbors. Now similar users will not only have content interests. When combining the user-base CF and trust-aware CF recommendation lists, we split the users into three parts, e.g inactive users, median active users and active users. Different from the past method which directly give them general weights for combination, we set different weights for different users. More experiments have been done to analysis and illuminate the factors which push users to join groups.

The rest of the paper is organized as follows. Section 2 presents the related work of group recommendation. Section 3 introduces our group recommendation model. Section 4 presents the experiment results, followed by the conclusion in Section 5.

2. Related work

Trust-aware CF is a commonly used technique for recommendation [7,8]. It solves the problem by using the trust networks among users. To evaluate the trust values among users, lots of useful information such as age and occupation can be used. In social networks, it can easily obtain the trust value from a user A to a user B. The value will be 1 if B is in the contact list of A or 0 if not. Finally we count the number of users both in user A's contact list and in group G's member list to present the relation of A to G. This trust-aware CF is commonly adopted in social networks to recommend friends and groups. However, it has a drawback that will perform badly for cold-start users. If a user only has a small number of followees in his/her contact list, the algorithm can not recommend groups which fit the user's interests.

For item recommendation, user-based collaborative filtering (CF) [3,4] is the most commonly used technique in social networks. Usually, we can obtain ratings from users on items and calculate the similarity among users. Whereas on group recommendation, we can only get a binary value which indicates whether a user joins a group or not. The binary value is not precise enough to estimate the similarity among users, so we cannot adopt the same technique used in item recommendation. To solve this problem, Zhuang et al. took full advantage of the Flickr heterogeneous data, estimated the similarity among users from six views, and then used the user-based CF for group recommendation [14]. Matrix factorization is another technique for item recommendation, and it tries to find the latent space of items and users [9]. However, in group recommendation, it suffers from the same problem which is we can only get a binary value to describe the relation of a user and a group. So the no-score matrix makes the technique work not well in group recommendation.

On the other hand, researchers tried to find the users' interests from the contents of photos. Luckily, the self-annotated photos in Flickr can help us bridge the semantic gap to obtain a series of tags which present the users' interests. In fact, users are willing to provide this semantic context through manual annotations to make them better accessible for the general public [15]. Many works focused on analyzing the topic modal of groups [16,17]. Also some researchers tried to connect users to groups through tags using tensor decomposition [18]. However, existing studies reveal that many tags provided by Flickr users are noise. There are only around 50% of those tags actually related to the photos [11]. Also even for the same object, different users will use different tags. As a result, it is hard to find out the true interests of users.

Collaborative filtering(CF) is a commonly adopted recommendation technique in social networks, and researchers have proposed a number of CF algorithms. The algorithms are mainly divided into two categories, i.e. model-based CF and memory-based CF. Model-based CF is developed using data mining and machine learning algorithms to find patterns based on training data, such as matrix factorization [9], latent semantic models [19] and bayesian network [20]. Memory-based CF crawls the users' ratings or behavior histories, finds out their neighbours and recommends what their neighbours like. Our model exploits the users' historical records and adopts the memory-based CF algorithms.

3. Group recommendation modeling

In this section, we present our group recommendation model. We choose the tags abstracted from the uploaded/ favourite photos and count the common friend number among users to represent similarity tastes among users and use user-based CF for recommendation. Then we use the trust-ware CF for contact information and construct a simple model to combine the two different recommendation results. The group recommendation model architecture is presented in Fig. 1.

3.1. User's similarity by tag features

To recommend groups to users, we should know what kinds of topics that will attract users. However, it is hard to find the semantic information only from photos. Luckily, the self-annotated tags of photos in Flickr can help us bridge the semantic gap to some extent. In fact, users are willing to provide this semantic context through manual annotations to make them better accessible for the general public [15]. So we can abstract the tags from users' uploaded photos to find what they are interested in. We can use Flickrapi to directly obtain the top 100 tags with highest frequency conducted by a user.

Since Flickr allows users to self-annotate photos, users can describe the same photo using different tags. This makes the tags imprecise. To solve this problem, we expand the tag information. By intuition, it is more complicated for a user to upload a photo than to mark a photo as favorite. We crawl the tags from users' favorite photo lists and choose the top 100 tags with highest frequency for each user.

However, some tags are so common such as "Nikon" and "Canon" that these tags will not present users' interests correctly. So we use traditional tf-idf method [21] to get the weights of tags and use a tag vector z to present the tag information of a user u. Thus for the tag list which abstracts from a user's uploaded photos, we use a tag vector z_s to represent it. And for the tag list which abstracts from a user's favourite photos, we use another tag vector z_f to represent it.

To find the users' neighbours who share the same topic contents, we should calculate the similarity among users. As a base case, we consider that a users is maximally similar to himself/ hersel. So we adopt the normalized linear kernel to measure the similarity between u_i and u_j.

$$S_{i,j} = \frac{z_i^T z_j}{\sqrt{z_i^T z_i}\sqrt{z_j^T z_j}} \tag{1}$$

Since we have two kinds of descriptors to represent the tag information (z_s, z_f), we can get four similarity matrixes using Eq. (1). We denote $S^{(1)}$ to be the matrix in which both u_i and u_j use the z_s tag descriptor, $S^{(2)}$ to be the matrix in which both u_i and u_j use the z_f tag descriptor, $S^{(sf)}$ to be the matrix in which u_i uses z_s tag descriptor and u_j uses z_f tag descriptor, and finally $S^{(fs)}$ to be the matrix in which u_i uses z_f tag descriptor and u_j uses z_s tag descriptor. In fact $S^{(fs)}$ is the transposed matrix of $S^{(sf)}$. To make it easy for the latter kernel alignment algorithm, we define $S^{(3)}$ to be the average of $S^{(sf)}$ and $S^{(fs)}$, i.e. $S^{(3)} = \left(S^{(sf)} + S^{(fs)}\right)/2$. An example of the similarity between two users is presented in Fig. 2.

3.2. User's similarity by common friend information

Another way to estimate the similarity among users is to count the common friend number. If two users always follow the same users, they may have some common interests. So we construct a common friend kernel to describe the similarity among users from another view.

$$F_{i,j} = \#user\ u_i\ and\ u_j\ both\ followed \tag{2}$$

where $F_{i,i}$ is the number of users whom user u_i has followed.

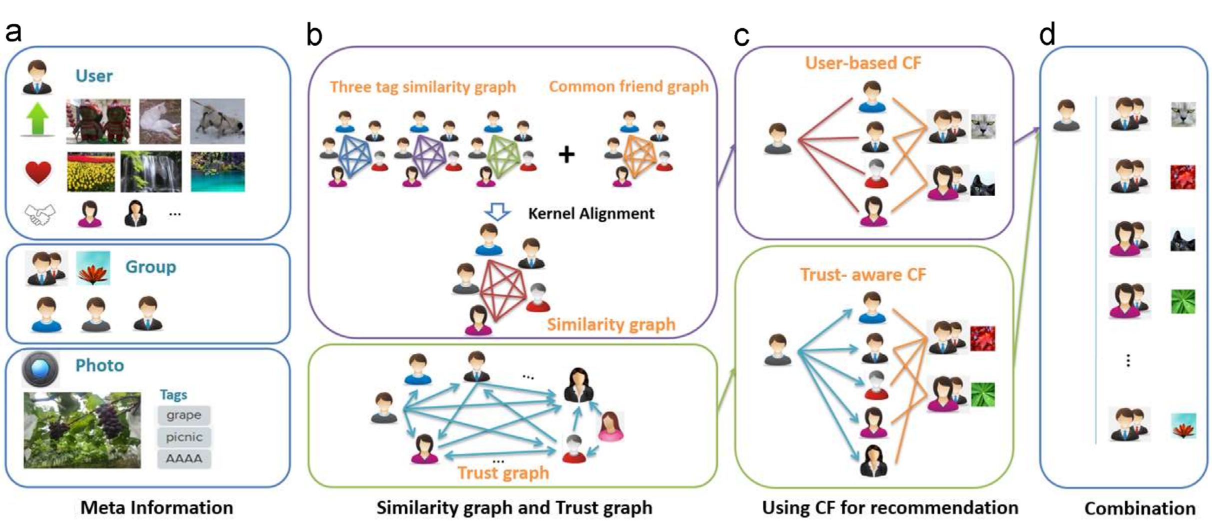

Fig. 1. The structure of our group recommendation model. (a) shows the basic information we use for our group recommendation. We build the relation graphs among users in (b). In (c), we use collaborative filtering for the similarity graph and trust graph respectively. Then combining the two recommendations, we get the final recommendation list as shown in (d).

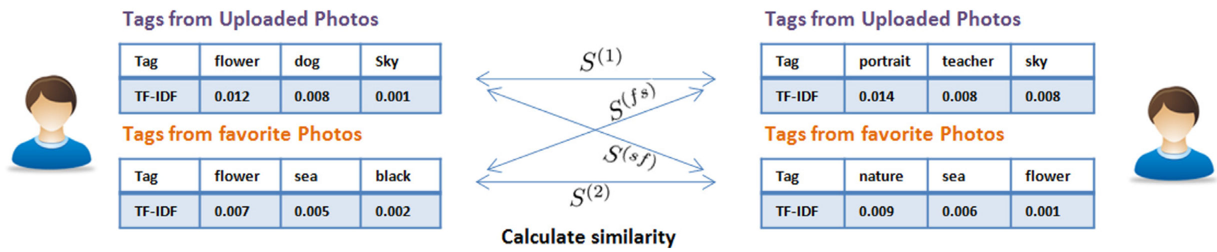

Fig. 2. An example of the tag similarities between two users.

To make the format of F matrix has the same format as that of S matrix, we normalize it by Eq. (3).

$$S^{(4)}_{i,j} = \frac{F_{i,j}}{\sqrt{F_{i,i}}\sqrt{F_{j,j}}} \tag{3}$$

3.3. Kernel alignment

In Sections 3.1 and 3.2, we define four tag similarity matrixes which describe the similarities among users from different views. To find the best way to combine them, we use the kernel alignment algorithm [12] which is a linear combination of multiple kernels to measure the final similarity kernel.

$$K(u_i, u_j) = \sum_{t=1}^{N_k} \theta_t K_t(u_i, u_j) \tag{4}$$

where K_t is the t-th kernel which describes the similarity among users, and N_k is number of kernels.

Some naive kernel combination techniques do not consider the redundancy among the kernels, the kernels are tackled independently. In this section, we present a kernel-based learning technique considering the redundancy among the kernels. Firstly we need to give a target matrix Y which describes the existing similarity among users. And then we adopt the kernel alignment algorithm [12] to calculate the weight vector θ for combination.

In group recommendation, the target matrix should maximally describe the similarity and can represent joining groups statement of users. We define G to be the common interest groups matrix which measures the number of interest groups that both users join.

$$G_{i,j} = \#group\ u_i\ and\ u_j\ joined \tag{5}$$

When $i{=}j$, the value is the group number that u_i has joined.

To make the format of matrix G has the same format as that of matrix S, we define the target matrix Y as one transformed from G by Eq. (6).

$$Y_{i,j} = \frac{G_{i,j}}{\sqrt{G_{i,i}}\sqrt{G_{j,j}}} \tag{6}$$

Definition 1 (*Centering kernels.*). Let K be a kernel function defined over $m \times m$, then the centering kernels equation is defined as:

$$[K_c]_{i,j} = K_{i,j} - \frac{1}{m}\sum_{i=1}^{m} K_{i,j} - \frac{1}{m}\sum_{j=1}^{m} K_{i,j} + \frac{1}{m^2}\sum_{i,j=1}^{m} K_{i,j} \tag{7}$$

Definition 2 (*Kernel Alignment.*). Let K and Y be two kernel functions defined over $m \times m$ such that $0 < E[K_c^2] < +\infty$ and $0 < E[Y] < +\infty$, and then the alignment between K and Y is defined as:

$$\rho(K, Y) = \frac{E[trKY]}{\sqrt{E[trKK]}\sqrt{E[trYY]}} \tag{8}$$

The algorithm is based on the notion of centering in the feature space. So the similarity matrixes should be centered by Eq. (7). We aim to find a linear combination to make the users fit their neighbors' tastes, i.e. $K = \Sigma_{i=1}^{N_k} \theta_i K_i$. The following theorem guarantees that the optimal solution can be computed efficiently.

Theorem 1. *The optimal solution θ^* can be obtained to solve the following quadratic program:*

$$\theta^* = \operatorname*{argmin}_{\theta \ge 0} \theta^T M \theta - 2\theta^T a \tag{9}$$

where a is the vector $[trK_1Y, \ldots, trK_{N_k}Y]^T$ and M is matrix $[M]_{kl} = trK_kK_l$.

Finally, we can get the combination weight vector θ, where $0 \le \theta \le 1$ and $\Sigma_t \|\theta_t\| = 1$.

3.4. User-based CF

Collaborative filtering is the most commonly used technique in recommendation domain. Researchers have modeled a number of collaborative filtering approaches. Collaborative filtering aims to recommend items to a user based on the tastes of the user's neighbors. For user-based CF, we need to estimate the similarity between users and then we can use the similarity matrix which we get above to replace it.

User-based CF algorism predicts the votes of the active user (indicated with a subscript a) based on some partial information from the active user and a set of weights calculated from the user database. Then it assumes that the predicted vote of the active user for item j, $p_{a,j}$ is a weighted sum of the votes of the other users [4].

$$w_{a,u} = \frac{\Sigma_{i=1}^{m}(r_{a,i} - \overline{r_a}) \times (r_{u,i} - \overline{r_u})}{\sigma_a \times \sigma_u} \tag{10}$$

$$p_{a,j} = \overline{r_a} + \frac{\Sigma_{u=1}^{n}(r_{u,i} - \overline{r_u} \times w_{a,u})}{\Sigma_{u=1}^{n} w_{a,u}} \tag{11}$$

r is a rating that a user assigns to an item, n is the number of neighbors and $w_{a,u}$ is the similarity weight between the active user a and the neighbor u.

The ratings in Eq. (10) will highly present the users' interests. But for group recommendation, usually we only have the binary values indicated if a user joins a group. Such a binary values will be so inaccuracy. Luckily, we can use other method to replace it. As mentioned above, we have get the similarity matrix among users from different views. So in the first step of user-based CF, we use the combination kernel to replace the w matrix.

In the second step, we should recommend items which users' neighbours love. We follow the user-based CF idea and simplify the Eqs. (11) and (12) to make it fit the group recommendation.

$$p^{(1)}_{a,j} = \sum_{u=1}^{n} \delta_{u,j} \times w_{a,u} \tag{12}$$

where $\delta_{u,j} \in \{0,1\}$ indicates whether u has joined group j.

3.5. Trust-aware CF

Trust-aware CF [7] is similar to user-based CF. The only difference is that trust-aware CF uses the trust networks between users instead of estimating the similarities. In the user-based CF, the recommend system tries to find the active user a's similar users and ignores the trust relation among them. In fact same to the real world, users will trust their acquaintances more than the strangers. Especially for some websites, recommendation based on trust network will perform better. For example, in the eBay.com marketplace site users can create “fake” auctions [22] and for many social networks, recommend systems use the users' contact information to recommend friends to them.

In the social networks, one's behavior can well predict the trust value among users. We define $f_{a,u}$ as whether an active user a follows another user u (1 if follow; 0 if not) and normalize it by the number of the users' followees to estimate the trust value among users.

$$T_{a,u} = \frac{f_{a,u}}{|U_a|} \tag{13}$$

U_a is the set of users whom user a follows.

Using $T_{a,u}$ to replace the $w_{a,u}$ in Eq. (12) and defining n as the number of users whom user a follows, we can transform the trust-aware CF into another form.

$$p^{(2)}_{a,j} = \frac{|U_a \cap Gm_j|}{|U_a|} \tag{14}$$

Gm_j is the set of users who join group j.

3.6. Combination of user-based CF and trust aware CF

User-based CF and trust aware CF use the relation among users from different views. The user-based CF presented above focuses on the similar neighbors' tastes. The users have much more topics to share with their neighbors. As a result, the predicted vote of the active user for a group will be highly possible to be accepted. Trust-aware CF recommends groups works more socially than user-based CF and focuses on the social relationship of users to find their trust neighbors instead of similar neighbors.

The two algorithms are complementary. Trusted users are good predictors. But the algorithm suffers from a problem, it will not work well for cold-start users. However cold-start users occupy a large portion of the users in most social networks. While for a fresh user, the first thing he/she does is to upload and browse photos. The behavior records produced by the fresh users will help us find their hobbies quickly. Also in fact, the active users and cold-start users have different behaviour patterns when they choose to join groups.

We try to combine the two different recommendations with a simple linear method. The late integration fusion weight parameter is empirically selected by exhaustive search and determined when the integrated predictions achieve the best performance on the training set.

$$p_{a,j} = \alpha \times p^{(1)}_{a,j} + (1-\alpha) \times p^{(2)}_{a,j} \tag{15}$$

We define the active degree of a user to be the number of his/her followees. Users with different active degrees will join groups with different patterns. In fact, user-based CF will perform better on inactive users since they have few followees, while trust-aware CF will work better on the active users since they are more social and easy to be effected by others. So we can divide the users into three parts: inactive users, medium active users and active users. Giving them different α to combine the two aspects will make our group recommendation perform better.

4. Experiment

We evaluate the proposed model on the same dataset as [14] comprised of multimodal information of 16,346 users from Flickr. To find a big component, we start from a random user as seed and expand the crawling according to users' contact lists in a breadth-first search manner. We stop at 5,000 users.

We conduct a basic data analysis on those 5,000 users. We calculate the number of users whose number of followees falls into a certain region. Fig. 3 shows the statistical histogram. We set the width of the bins to be 25. As Fig. 3 shows, the tendency of the histogram fits the asymptotic power-Law distribution and also has the heavy-tail property. So we can use it to study the real-world social network.

In fact, researchers find that there is a tendency that more friends will bring in more logins [13]. So we use the number of followees of a user to describe the active degree of him/her. We firstly divide the 5,000 users into ten parts according to their followees' numbers by ascending ordering. Each part contains 500 users. Then we calculate the average uploaded photo number, favourite photo number, group number and contacts number of the users in each part. Fig. 4 shows the statistical results. From Fig. 4, we can find:

- Regardless of the active degree, users will upload lots of photos. So even for a user who has few followees, we can mine what he/she likes from the uploaded photos.
- The uploaded photo number and the favourite photo number of a user will increase when the active degree of the user increases. This is easy to explain. More friends will bring in more logins and also will bring more behavior records. However when a user is not enough social, he/she tends to use the Flickr as a storage space. But these users are still willing to join groups. The 0-10% part of the users have an average number of groups about 47.3 while contacts about 14.3. This phenomenon reveals that for these inactive users, group recommendation will be more useful than friend recommendation.
- Though uploaded photo number and the favourite photo number have the same tendency, the rapid of the tendency is quite different. The favourite photo number increases more faster than the uploaded photo number. In social networks,

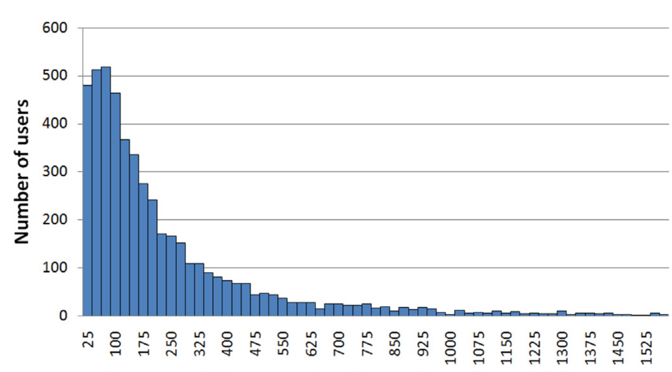

Fig. 3. The statistical histogram using the number of the 5,000 users' followees. The width of each bin is set to be 25.

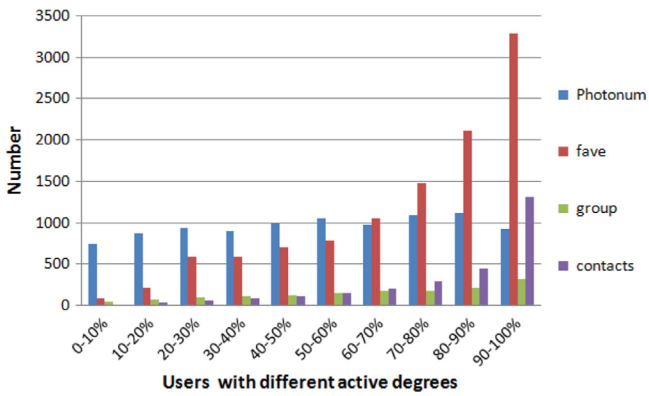

Fig. 4. The average upload photo number, favourite photo number and group number of users with different active degrees.

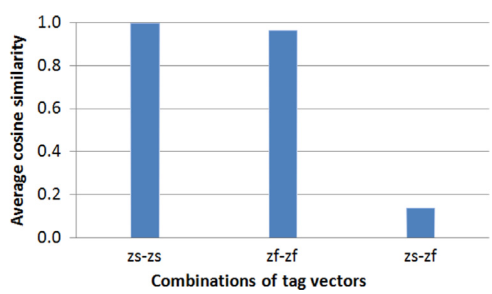

Fig. 5. The average similarity with different combinations of tag information among the 5,000 users dataset.

marking a photo as favourite is quite a simple task than uploading a photo.

4.1. Similarities of users

To construct the two tag vectors z_s and z_f presented in Section 2, we crawl the users' profiles. In fact, Flickrapi provides the top 100 tags with highest frequency for each user. Unfortunately, we could not obtain those 100 tags from users' favorite photos directly. We solve this problem by crawling Flickr online. We expand the tag information by downloading the metadata of photos from the users' favorite photo lists and abstracting the tags of each photo. After doing some statistics, we also obtain the top 100 tags with highest frequency. We use the traditional tf-idf method to get the weights of tags to build the z_s tag vector. Same to the method of deriving z_s, we obtain the z_f vector.

We evaluate the similarity among users following the method presented in Section 2 and get the matrixes: S^1, S^2, S^3 and S^4. To prove that the z_s and z_f vectors are quite different even for the same user, we calculate the average similarity values which use the different combinations of z_s and z_f vectors in the 5,000 users dataset. The results of different combination are presented in Fig. 5. Moreover, we calculate the average similarities among different active users by ascending ordering, the results are presented in Fig. 6.

As Fig. 5 shows, there are only three combinations because the combination of z_s and z_f is the same to the combination of z_f and z_s. The former two bins' values are not equal to 1 because some users have not uploaded any photos or marked any favorite photos. The average similarity using z_s and z_f descriptors is 0.14. This indicates that the compositions of z_s and z_f are different but still have some in common. One reason for this phenomenon is that the tags are self-annotated by users. There may be different tags annotated by different users even for the same photo. Another reason is that

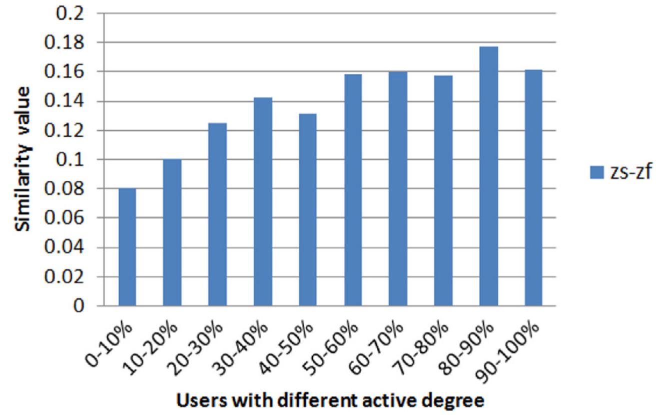

Fig. 6. The average similarity between z_s and z_f among different active users.

Table 1
The weights of the tag similarity kernels.

Kernel	$S^{(1)}$	$S^{(2)}$	$S^{(3)}$	$S^{(4)}$
Weight	0.060	0.125	0.195	0.620

users may have interests in something that they do not have the ability to take photos of them.

From Fig. 6, we find that the tendency of the histogram can be divided into two parts. The tendency at first fits a linear growth and then becomes stable. So we can conclude that tags from users' uploaded photos and favourite photos are quite different but still have something in common. So combining these tag information will help us find users' similar neighbors more accuracy. Another observation is that even for an inactive users the similarity value of z_s and z_f is only half of that of active users. So even for few favourite photos, there is rich information which can indicate users' interests.

4.2. User-based CF for group recommendation

In this part, we will perform the group recommendation using the user-based CF. We abstract the top 5,000 popular groups among the 5,000 users for recommendation.

We use the kernel alignment algorithm [12] to get a combination matrix with the four similarity matrix built above, i.e $S = \Sigma_{i=1}^{4} \theta_i S^{(i)}$. Since the kernel alignment algorithm requires the kernels to satisfy the positive semi-definite (p.s.d.) property and user-based CF just finds the neighbors rather than the users themselves, we assign the diagonal values of the similarity matrixes and the target matrix Y to be 1 and make them p.s.d. The learned weights of the kernels are presented in Table 1.

After combination, we evaluate the performance of the group recommendation using the user-based CF. The neighbor number n is set to be 25 for all users. The kernels include:

- Similarity kernels: The similarity kernels defined in Section 2: $S^{(1)}$, $S^{(2)}$, $S^{(3)}$ and $S^{(4)}$.
- MKL kernel: The combination kernel using the similarity kernels $S^{(1)}$, $S^{(2)}$, $S^{(3)}$ and $S^{(4)}$ by the kernel alignment algorithm [12].

We use Normalized Discount Cumulative Gain(NDCG) [23] as the evaluation measure for our group recommendation. NDCG is used to consider the ranked position among top-k recommended list which is provided by a recommendation algorithm. It suggests that more relevant items will get higher scores than irrelevant

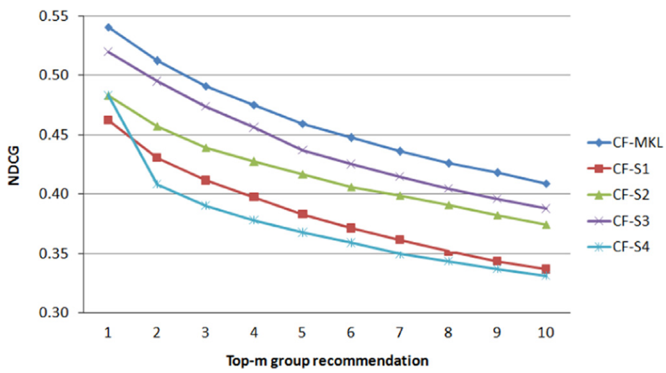

Fig. 7. The top-10 group recommendation results. The curves are the results of user-based CF using different kernels, e,g. CF-S1 is the result of the user-based CF using the $S^{(1)}$ kernel.

items and the items ranked lower will score lower since it has less value for the user. Then, *NDCG@k* is defined as follows:

$$DCG_R@k = \sum_{i=1}^{k} \frac{2^{r_{Ri}} - 1}{\log(i+1)} \tag{16}$$

$$NDCG_R@k = \frac{DCG_R@k}{DCG_{ground-trurh}@k} \tag{17}$$

r_{Ri} denotes the binary judgment (i.e., 1 for true and 0 for false). k is the length of the ranking list.

We evaluate the Top-10 group recommendation performance. The top-10 group recommendation results are shown in Fig. 7. By analyzing the curves presented in Fig. 7, we suggest that:

- The recommendation based on kernel $S^{(2)}$ works better than kernel $S^{(1)}$. This means that generally the tags from users' favorite photos can present their interests more precisely than the tags that users self-annotate.
- The recommendation based on kernel $S^{(3)}$ works better than $S^{(1)}$ and $S^{(2)}$. The combination of the two tag descriptors will convey more information.
- The CF-MKL works best among all the evaluated kernels. Such a combination method can estimate the relation among users better than the other single kernels and make it more precise to find neighbors for group recommendation.

4.3. Combination of user-based CF and trust-aware CF

In this section, we evaluate our group recommendation combining the user-based CF and trust-aware CF. After estimating the user-based CF of users above, we crawl the profiles of users and groups and then abstract the contact lists and member lists. We use Eq. (14) to get $p^{(2)}_{a,j}$.

We randomly choose 2,500 users for training purpose. We estimate our model's performance on users with different active degree. In fact, researchers find that there is a tendency that more friends will bring in more logins [13]. So we use the number of users' followees to present their active degree. We get the α with the best performances on different active degree users which the range of the users' active degree is set to be 20, and we use the average active degrees to present these regions. The results of the α is presented in Fig. 8. We divide the test users into three parts: inactive users, medium active users and active users. The active degree thresholds are 40 and 100. We choose the training users with different active degree. Finally we obtain the best α for

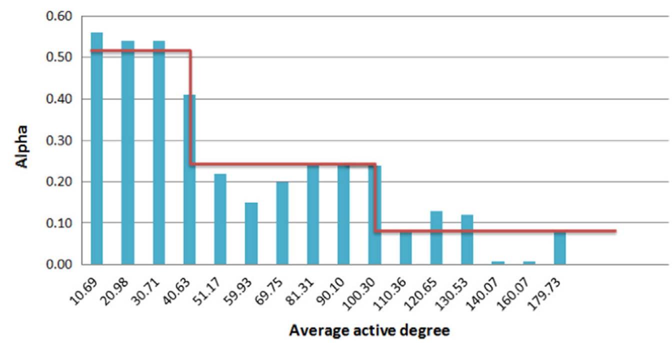

Fig. 8. The best α of different active degree users.

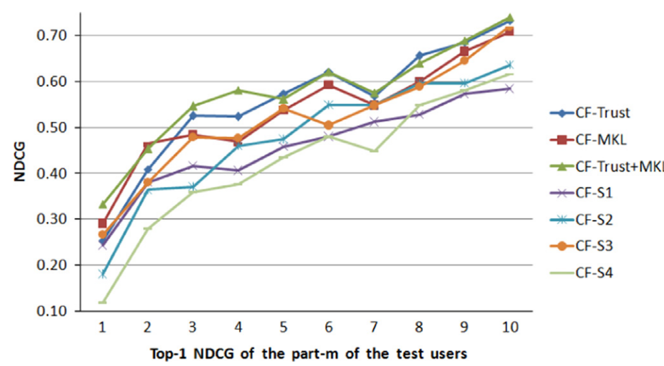

Fig. 9. The top-1 NDCG of users with different active degree by ascending ordering, each part contains 10% of testing users. We use the number of users' followees to present their active degree.

different active degree users.

$$\alpha(d) = \begin{cases} 0.54, & d \in [0, 40]. \\ 0.23, & d \in (40, 100]. \\ 0.07, & d \in (100, \infty). \end{cases} \tag{18}$$

where d is the value of active degree.

We divide the rest 2,500 testing users into ten parts according to their followees' numbers by ascending ordering and make group recommendation using the α which we obtain above. The group recommendation results of the ten parts of users at Top-1 NDCG are presented in Fig. 9.

We can see in Fig. 9 that the trust-ware CF works worse than our model among the users with a small number of followees, but it works the same when users have amount of followees. The CF-MKL performs better than trust-ware CF for users with a small number of followees. Then we choose the users who have different followees to make the group recommendation respectively. One thing to be noted is that the CF using the $S^{(2)}$ kernel works worse than using the $S^{(1)}$ for few-followee users while works better when the number of users' followees become larger. So we can infer that only when users have marked a large number of photos, the tags from marked photos can present the users' interests precisely. Moreover, our model performs better for the top 40% of the users with less than 100 followees. In fact, the proportion of the users with less than 100 followees will has a larger proportion in the social networks than our dataset. So our model will be applicable for group recommendation in social networks.

We analyze the results of the group recommendation shown in Fig. 10. The information of users in the experiment is presented in Table 2.

- Our combination model yields the best and our CF-MKL yields the second best when users are inactive. Such a promising

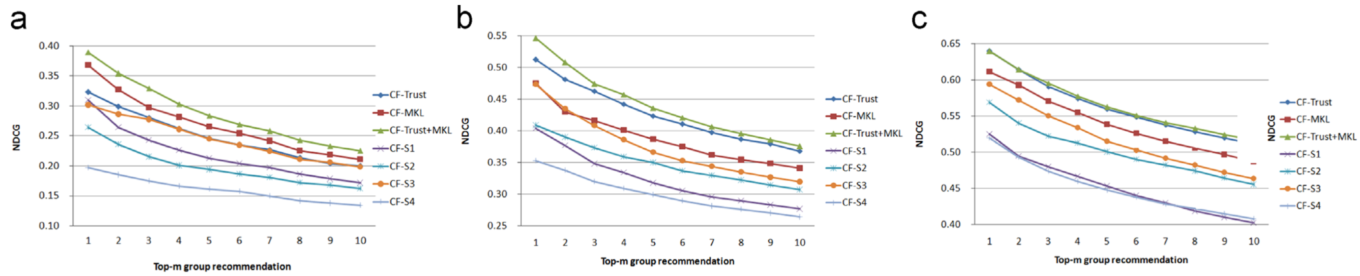

Fig. 10. Top-10 group recommendation for users who have different followees. (a) Top-10 group recommendation for users who have less than 40 followees. (b)Top-10 group recommendation for users who have more than 40 and less than 100 followees. (c)Top-10 group recommendation for users who have more than 100 followees.

Table 2
The number of testing users.

#Followee	[0 40]	(40 100]	(100 ∞)
#User	425	596	1479

result infers that using the users' similar neighbors informaiton for group recommendation is possible.

- Trust-CF works better than CF-MKL when users have a large number of followees while CF-MKL works better when users have a small number of followees. It is easy to find that the gap between trust-based CF and user-based CF gets smaller when the number of users' followees becomes larger. Users with a smaller number of followees tend to find interest groups by interests more often than social relation. When the users become more social and have more followees, they tend to join groups which their followees have joined.
- CF-S1 works better than CF-S2 when users have a small number of followees while CF-S2 works better when users have a large number of followees. This phenomenon indicates that if a user is an inactive user, he/she is not familiar with the social network and will only use some basic services(such as uploading photos). When a user becomes active, he/she will integrate himself/herself into the social network.

5. Conlusion

In this paper, we propose a group recommendation model using the users' trust neighbors and similar neighbors' tasts. We find that whether a user has a high active degree or low active degree, he/she is willing to join groups. When a fresh user joins Flickr, he/she usually uploads photos firstly. The photos can present his/her hobbies precisely. Also the user will mark photos as favourite. However only when the user has marked a large amount of photos, these photos can present the user's hobbies precisely. The two kinds of photos have their advantages and have complementary advantages. Combining them will get a promising result for group recommendation especially for the inactive users.

Another observation is that inactive users tend to join groups by interests more often than social relation. When users become more social and have more followees, they tend to join groups which their followees have joined. So we can assume that when users are fresh, they are not familiar with the social network, during this period they tend to join groups which fit their interests. However when the fresh users become social and have already joined their interest groups, they need to explore other interests. At this time, the trust neighbors' interests will help them a lot.

Our group recommendation model combines the trust-aware CF and user-based CF, and it performs well especially on inactive users. In our future work, we will attempt to adopt some multi-view learning methods [24–26] to combine various features for group recommendation.

Acknowledgement

This work is supported by the NSFC under the contract No.61201413 and No. 61572451, Youth Innovation Promotion Association CAS CX2100060016, the Fundamental Research Funds for the Central Universities No. WK2100060007 and No. WK2100060011, the Specialized Research Fund for the Doctoral Program of Higher Education No. WJ2100060003, to Dr. Xinmei Tian.

References

[1] S. Amer-Yahia, S.B. Roy, A. Chawlat, G. Das, C. Yu, Group recommendation: semantics and efficiency, Proceedings of the VLDB Endowment 2 (1) (2009) 754–765.
[2] L. Baltrunas, T. Makcinskas, F. Ricci, Group recommendations with rank aggregation and collaborative filtering, in: Proceedings of the fourth ACM conference on Recommender systems, ACM, 2010, pp. 119–126.
[3] J.L. Herlocker, J.A. Konstan, A. Borchers, J. Riedl, An algorithmic framework for performing collaborative filtering, in: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, ACM, 1999, pp. 230–237.
[4] J.S. Breese, D. Heckerman, C. Kadie, Empirical analysis of predictive algorithms for collaborative filtering, in: Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence, Morgan Kaufmann Publishers Inc., 1998, pp. 43–52.
[5] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Item-based collaborative filtering recommendation algorithms, in: Proceedings of the 10th international conference on World Wide Web, ACM, 2001, pp. 285–295.
[6] M. Deshpande, G. Karypis, Item-based top-n recommendation algorithms, ACM Transactions on Information Systems (TOIS) 22 (1) (2004) 143–177.
[7] P. Massa, P. Avesani, Trust-aware recommender systems, in: Proceedings of the 2007 ACM conference on Recommender systems, ACM, 2007, pp. 17–24.
[8] J. O'Donovan, B. Smyth, Trust in recommender systems, in: Proceedings of the 10th international conference on Intelligent user interfaces, ACM, 2005, pp. 167–174.
[9] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Computer 42 (8) (2009) 30–37.
[10] C. Dwyer, S. Hiltz, K. Passerini, Trust and privacy concern within social networking sites: A comparison of facebook and myspace, AMCIS 2007 Proceedings (2007) 339.
[11] L.S. Kennedy, S.-F. Chang, I.V. Kozintsev, To search or to label?: predicting the performance of search-based automatic image classifiers, in: Proceedings of the 8th ACM international workshop on Multimedia information retrieval, ACM, 2006, pp. 249–258.
[12] C. Cortes, M. Mohri, A. Rostamizadeh, Two-stage learning kernel algorithms, in: Proceedings of the 27th International Conference on Machine Learning (ICML-10), 2010, pp. 239–246.
[13] M. Moricz, Y. Dosbayev, M. Berlyant, Pymk: friend recommendation at myspace, in: Proceedings of the 2010 ACM SIGMOD International Conference on Management of data, ACM, 2010, pp. 999–1002.
[14] J. Zhuang, T. Mei, S.C. Hoi, X.-S. Hua, S. Li, Modeling social strength in social media community via kernel-based learning, in: Proceedings of the 19th ACM international conference on Multimedia, ACM, 2011, pp. 113–122.
[15] M. Ames, M. Naaman, Why we tag: motivations for annotation in mobile and online media, in: Proceedings of the SIGCHI conference on Human factors in computing systems, ACM, 2007, pp. 971–980.
[16] R.A. Negoescu, D. Gatica-Perez, Analyzing flickr groups, in: Proceedings of the 2008 international conference on Content-based image and video retrieval, ACM, 2008, pp. 417–426.

[17] R.-A. Negoescu, B. Adams, D. Phung, S. Venkatesh, D. Gatica-Perez, Flickr hypergroups, in: Proceedings of the 17th ACM international conference on Multimedia, ACM, 2009, pp. 813–816.
[18] N. Zheng, Q. Li, S. Liao, L. Zhang, Flickr group recommendation based on tensor decomposition, in: Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval, ACM, 2010, pp. 737–738.
[19] T. Hofmann, Latent semantic models for collaborative filtering, ACM Transactions on Information Systems (TOIS) 22 (1) (2004) 89–115.
[20] N. Friedman, D. Geiger, M. Goldszmidt, Bayesian network classifiers, Machine learning 29 (2–3) (1997) 131–163.
[21] G. Salton, M.J. McGill, Introduction to modern information retrieval.
[22] P. Massa, A survey of trust use and modeling in current real systems, Trust in E-services: Technologies, practices and challenges, Idea Group Publishing 22.
[23] K. Järvelin, J. Kekäläinen, Cumulated gain-based evaluation of ir techniques, ACM Transactions on Information Systems (TOIS) 20 (4) (2002) 422–446.
[24] J. Yu, Y. Rui, B. Chen, Exploiting click constraints and multi-view features for image re-ranking.
[25] W. Liu, D. Tao, Multiview hessian regularization for image annotation, IEEE Transactions on Image Processing 22 (7) (2013) 2676–2687.
[26] C. Xu, D. Tao, C. Xu, Large-margin multi-view information bottleneck, IEEE Trans. Pattern Anal. Mach. Intell. 36 (8) (2014) 1559–1572.

Cong Guo received the B.E. degree from the University of Science and Technology of China, Hefei, China, in 2013. He is currently working towards the PhD. degree in the CAS Key Laboratory of Technology in Geo-Spatial Information Processing and Application System, University of Science and Technology of China, Hefei, China. His research interests lie primarily in multimedia search, information retrieval, and machine learning.

Bei Li received the B.E. degree from Beihang University, Beijing in 2003 and his master degree from the University of Melbourne, Australia, in 2005. He is now working towards the Ph.D. degree in University of Science and Technology of China, Hefei China. Also he is a Research Program Manager in Microsoft Research Asia Outreach team for over 8 years. His research interest is multimedia, machine learning and education.

Xinmei Tian received the B.E. and Ph.D. degrees from the University of Science and Technology of China, Hefei, China, in 2005 and 2010, respectively. She is an Associate Professor in the CAS Key Laboratory of Technology in Geo-Spatial Information Processing and Application System, University of Science and Technology of China, Hefei, China. Her current research interests include multimedia information retrieval and machine learning. She received the Excellent Doctoral Dissertation of Chinese Academy of Sciences award in 2012 and the Nomination of National Excellent Doctoral Dissertation award in 2013.