2014 IEEE 15th International Conference on Mobile Data Management

WhereToGo: Personalized Travel Recommendation for Individuals and Groups

Long Guo, Jie Shao, Kian-Lee Tan, Yang Yang
School of Computing, National University of Singapore, Singapore
Email: {guolong, shaojie, tankl, yang.yang}@comp.nus.edu.sg

***Abstract*—With the rapid development of GPS-enabled mobile devices, huge amounts of user-contributed data with location information can be collected from the Internet. With this kind of data, one promising application is travel recommendation, which has attracted a considerable number of researches recently. However, most of the previous studies only focus on one aspect of the relations among users and locations or make a coarse linear combination of the relations. Moreover, all the existing work on travel recommendation do not consider recommendation to groups, which is an important characteristic of travelers' behavior. In this paper, we present a personalized travel recommendation system named WhereToGo. The novelty of the system is a 3R model which can unify user-location relation, user-user relation and location-location relation into a single framework and perform random walk with restart to analyze the model. We further extend our approach to provide recommendations for groups. To the best of our knowledge, this is the first work to use random walk with restart for group recommendation. We conduct a comprehensive performance evaluation using a real dataset collected from Flickr, which is one of the most popular online photo-sharing sites. Experimental results show that our approach provides significantly superior recommendation quality compared to other state-of-the-art travel recommendation approaches for both individuals and groups.**

I. Introduction

Traveling is one of the most important ways of entertainment nowadays. According to a report by the World Travel & Tourism Council, the contribution of tourism to global GDP is 9.3% in 2012. Longer-term prospects are even more positive with annual growth forecast to be 4.4% per year over the next decade to 2022[1]. With the economic growth and globalization of the world, more and more people have the money and interest to travel to other places. Meanwhile, powered by the development of GPS-enabled mobile devices and Web 2.0 technologies, many kinds of web sites with rich location information have emerged, such as the photo sharing web site (e.g., Flickr[2]) and location-based service web site (e.g., Foursquare[3]). Large-scale geo-tagged photos and check-ins become available from these web sites, which record people's movement behavior and thus are valuable resources to be analyzed for location recommendation. Compared with check-ins, geo-tagged photos are relatively more appropriate for travel recommendation (as it is observed that they better reflect travelers' preference, while check-ins often reflect local residences' preference [1]). Thus, in this work we aim at recommending personalized tourist attractions to travelers using large-scale geo-tagged photos collected from Flickr.

Location recommendation for travelers based on user-contributed data has attracted much research interest recently. However, most previous studies focus on one aspect of the relations among users and locations, or make a coarse linear combination of the relations. Moreover, all the work on travel recommendation make recommendations for individual users. Nevertheless, in reality people also often travel together, such as a couple or several friends. In this situation, considering only one user's preference is not appropriate for a group whose members have not only common tastes but also conflicting tastes. We refer to this as the *group recommendation* problem, and study how to effectively and efficiently compute results for such recommendation.

In this paper, we present a personalized travel recommendation system named WhereToGo that addresses the above shortcomings and recommends users where to go in an unfamiliar city. It builds a novel 3R model that unifies user-location relation, user-user relation and location-location relation into a single framework. In order to provide personalized recommendation for a traveler, we use random walk with restart to analyze our 3R model. Additionally, we extend the random walk with restart to support group recommendation. As a result, WhereToGo takes into account location popularity, user similarity and location similarity simultaneously and can provide recommendations with higher quality for both individuals and groups. In summary, we make the following contributions:

- We propose a novel 3R model which considers location popularity, user similarity and location similarity simultaneously and use random walk with restart to analyze this model.
- We extend our approach to support group recommendation. To the best of our knowledge, this is the first time random walk with restart is used for group recommendation.
- We develop a region-based partition technique to provide good scalability of our approach.
- We conduct a series of experiments based on a real dataset crawled from Flickr to evaluate our approach. The results show its superior performance compared with other state-of-the-art approaches.

The rest of this paper is organized as follows. We review the related literature in Section II. In Section III we present the architecture of the system. In Section IV and Section V, we introduce each module in the system. An experimental study on the different travel recommendation approaches with a Flickr dataset is reported in Section VI. Finally, Section VII gives our conclusions and future work.

[1] http://www.wttc.org/research/economic-impact-research/
[2] http://www.flickr.com/
[3] https://foursquare.com/

978-1-4799-5705-7/14 $31.00 © 2014 IEEE
DOI 10.1109/MDM.2014.12

II. Related work

A. Personalized Travel Recommendation

Personalized travel recommendation is more than just a scientific/academic exercise. In fact, it is of practical use. The mobile tour guide Cyberguide [2] was designed based on the contextual information, such as the current location of user and history of past locations. In [3], an intelligent system was introduced to provide personalized recommendations in an unfamiliar city. The system recommends tourist attractions to a user by using the Bayesian network technique and analytic hierarchy process method, and taking into account the travel behavior both of the user and of other related users. In [4], a system named TripTip was developed to recommend users the next place to visit by considering the similarity between previous visiting places and the next place. To achieve this purpose, the users need to give the place tags and the reasons for visiting the place. In [5], an enhanced Collaborative Filtering (CF) method was proposed to provide more useful results based on the current locations of users. However, most of them ask for information defined by domain experts while this kind of information is rare in practice. Moreover, such kind of information could be easily out-of-date.

Other research efforts studied location recommendation based on the user-contributed data available, such as geo-tagged photos and check-ins. In [6], Clements *et al.* provided personalized recommendation to a user based on the location similarity between tourist attractions using geo-tagged photos. In [7], Shi *et al.* studied personalized recommendation by combining user-landmark preference and landmark similarity using geo-tagged photos. However, their approach performs worse than the location popularity method and is appropriate to recommend landmarks less frequently visited. In [8], [9], Ye *et al.* recommended points of interest for users by combining the user-based CF method and geographical factors in a linear method using check-ins collected from Foursquare. However, these studies only consider one aspect of the relations among users and locations or make a coarse linear combination of some of the relations, but not fully exploit all the relations. Compared with the previous studies, our approach integrates user-location relation, user-user relation and location-location relation in a more effective way and thus can compute personalized recommendation results with higher quality.

B. Group Recommendation

Group recommendation systems are developed to support the recommendation process in activities that involve more than one person. The existing systems have been used in some other application domains such as web pages, news items, music tracks, television programs and movies. The state-of-the-art approaches in group recommendation are presented in [10], [11]. Typically, a classification of those approaches can be made from two perspectives: the type of groups considered and the way group recommendations are built.

For the first classification, according to [10] there are at least four different types of groups:

- **Established group**: the members explicitly choose to be part of a group with shared, long-term interests.
- **Occasional group**: the members do something occasionally together, such as going out for a tour, sharing a common aim in a particular moment.
- **Random group**: the members share an environment in a particular moment, without explicit interests that link them.
- **Automatically identified group**: the groups are automatically detected considering the preferences of the users and/or the resources available.

Clearly, the type of groups in our travel recommendation application falls into the occasional group and thus we do not need to use some algorithms to detect the groups automatically. We only need to consider the way group recommendations are built. Broadly speaking, there are two main strategies used for group recommendation as described below:

- **Merging profiles**: the individual members' preferences are merged into a single virtual user profile and recommendations are made to this virtual user.
- **Merging scores**: a variety of aggregation techniques have been proposed in the literature [12], [11], [13]. Compared with the above strategy, this strategy offers better flexibility and more opportunities for efficiency improvement, and is therefore the strategy we take in this paper. Basically, there are two main strategies: *average* and *least misery*. The former takes the average of group members' scores for an item, while the latter takes the lowest score among all group members. The performance of these two strategies are similar. In this paper, we choose to use the average strategy which is more appropriate for a large group. In [14], Amer-Yahia *et al.* take into account the disagreement among users as well and introduce the notion of consensus function, which consists of two components, relevance and disagreement. They show their approach can perform better in some situations. In this paper, we also consider it as the baseline approach.

C. Random Walk with Restart

Random Walk with Restart (RWR) [15] can provide a good relevance score between two nodes in a weighted graph, and is popular in many different areas within information retrieval, starting from link analysis [16] to image annotation and retrieval [17], text classification [18], click-through data analysis [19], collaborative recommendation [20], social recommendation [21] and venue recommendation [22]. RWR captures the global structure of a weighted graph and the multi-facet relationship between two nodes.

The studies of [21] and [22] are most related to our work. [21] recommends music tracks by leveraging an extended graph whose nodes represent users as well as tags and songs. However, its goal is to study the effect of social networking and social tagging in collaborative recommendation using data crawled from the social networking service of Last.fm, while both of the social relationship and social tagging do not exist in our dataset. [22] recommends new venues by combining social network and venue visit frequency data. Their user-location graph is similar with ours. However, they focus only on the social relationship between users and do not consider the location-location relation, resulting in a restricted improvement compared to other state-of-the-art methods. Instead, our focus

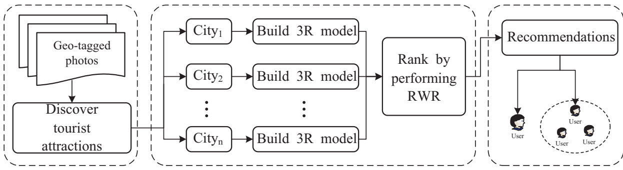
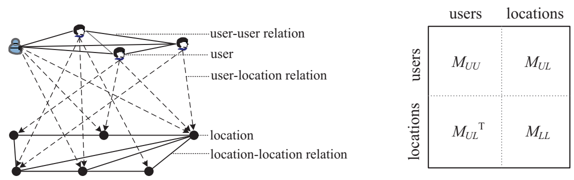


Fig. 1. System architecture of WhereToGo.



(a) Location-based social network

(b) M_{3R}

Fig. 2. An illustration of the 3R model.

is to build a novel 3R model which takes into account the location popularity, user similarity and location similarity simultaneously. In addition, our work is the first effort to use random walk with restart for group recommendation. The experiments also show that the recommendation performance can be significantly improved for both individuals and groups by performing random walk with restart on our 3R model.

III. System Architecture

In this section, we present the architecture of our travel recommendation system WhereToGo. WhereToGo is developed based on the fact that geo-tagged photos uploaded to online photo sharing web sites are significant information resources for finding tourist attractions and mining user behaviors. Our system can be used to recommend tourist attractions to its recipient (a single user or a group) when a trip is being planned for an unvisited city.

The problem studied in this paper can be defined as follows.

Definition 1: Given a large collection of user-contributed geo-tagged photos, our goal is to develop a personalized travel recommendation system, which can provide individuals or groups with tourist attractions in a targeted city matching user preferences.

Fig. 1 depicts the modules of our WhereToGo system. First, the system automatically discovers locations that are characterized by high photo density using the geo-tagged photos. These locations are viewed as the tourist attractions since they represent a positive vote by previous visitors. In this paper, the term *location* and *tourist attraction* represent the same meaning. Second, a novel model named 3R is built for each city according to our region-based partition technique. For each targeted city, random walk with restart is performed to analyze the corresponding 3R model and recommend the top-K tourist attractions to the recipient. Our 3R model can capture the location popularity, user similarity and location similarity simultaneously and thus suggest tourist attractions that of more relevance and significance. The following two sections detail each of the modules.

IV. Tourist Attraction Extraction

Since geo-tagged photos are just some images with coordinate information, we need to extract tourist attractions from this kind of user-contributed resource. In particular, we would like to find interesting places that are characterized by high photo activity in a specific area. Discovering tourist attractions can be regarded as a clustering problem that can be solved using different clustering algorithms. In this work, we adopt a density based clustering algorithm P-DBSCAN [23] to cluster photos using their spatial proximity. P-DBSCAN is variation of the classic clustering algorithm DBSCAN [24] and is more appropriate for geo-tagged photos clustering. Given a collection of geo-tagged photos P as input, the output of P-DBSCAN is a set of locations $L = \{l_1, l_2, \ldots, l_n\}$. Each element can be represented using the form $l = (P_l, G_l)$, where P_l is a cluster of photos and G_l is the geographical coordinates to represent the centroid of locations l. G_l can be computed using the geo-tags annotated to photos in the cluster P_l by equation

$$G_{l_x} = \frac{\sum_{p_i \in P_l} p_{i_x}}{N}, G_{l_y} = \frac{\sum_{p_i \in P_l} p_{i_y}}{N} \quad (1)$$

where p_{i_x} is the longitude of photo p_i, p_{i_y} is the latitude of photo p_i, and N is the number of photos in P_l. Note that we do not use P-DBSCAN for particular reason. The clustering algorithm is orthogonal with respect to our work and some other optimized clustering algorithms can be used to enhance the accuracy of tourist attraction extraction.

V. 3R Model for Individual/Group Recommendation

After the tourist attractions are discovered, we need to recommend the most relevant and significant ones to the travelers based on their preferences. In this section, we introduce a novel 3R model which unifies location popularity, user similarity and location similarity into a single framework. First, we introduce the 3R model from three aspects: user-location relation, user-user relation and location-location relation. Next, we analyze the 3R model to provide personalized recommendation using random walk with restart. Then we present how to extend our RWR approach for group recommendation. Finally, we discuss the techniques to tackle the scalability issue of our RWR approach.

A. 3R Model

In general, a Location-based Social Network (LBSN) has two kind of entities, user and location, and three kind of relations, user-location relation, user-user relation and location-location relation, as can be seen in Fig. 2(a). As we noted in the Section II-A, almost all the previous recommendation methods only consider a part of the relations, such as user-item based CF which considers only user-user relation and item-based CF which considers only location-location relation, or make a coarse linear combination of the three relations, ignoring the whole network. In this work, we propose a novel three relation model named 3R which simulates the LBSN framework and captures user-location relation, user-user relation and location-location relation simultaneously.

User-Location Relation. There exists a typical behavior between a user and a location in an LBSN, i.e., a user visits a location. In traditional recommendation systems, this behavior can be represented using an explicit rating by the user for the item, while in an LBSN, users usually do not rate for the locations they visited. We can only use an implicit rating, i.e., the visiting frequency, to represent this behavior. The visiting frequency to a location by a user should not be the number of photos taken at this location by this user, since a user may take many photos, which can increase the importance of this location as though there are few people visiting this location. Thus, we determine a visit based on the time attribute of the photos and several photos can belong to only one visit.

Definition 2 (User-Location Relation): The relation of user and location is represented by a weighted bipartite graph $G_{UL} = (U, L, E_{UL}, W_{UL})$, where U is a set of nodes that represent the users, L is a set of nodes that represent the locations, E_{UL} is a set of edges that represent visits, and the edge weights W_{UL} describe the number of visits to a location by a user, i.e., the visiting frequency.

An $m \times n$ adjacency matrix M_{UL} is built for G_{UL} with m users and n locations. Formally, $M_{UL} = [v_{i,j}], 0 \leq i \leq m, 0 \leq j \leq n$, where $v_{i,j}$ represents the visiting frequency of the i^{th} user for the j^{th} location.

User-User Relation. The user-user relation has been used widely in personalized recommendation systems. Two most typical applications are the user-based collaborative filtering that considers user similarity and the social-based collaborative filtering that considers social friendship. In this work, since geo-tagged photos are obtained from Flickr where the social friendship is inaccessible, we only consider the similarity between users. In addition, it has been shown that in general users with social friendships only share less than 10% commonly visited locations, as reported in [8], [25]. Actually, friends still reflect significantly different preferences and social tie cannot reflect the similarity of visiting behavior among users [9].

Definition 3 (User-User Relation): The relation of users is represented by a weighted undirected graph $G_{UU} = (U, E_{UU}, W_{UU})$, where U is a set of nodes that represent the users, E_{UU} is a set of edges that represent the user similarity, and the edge weights W_{UU} describe how similar two users are.

An $m \times m$ adjacency matrix M_{UU} is built for G_{UU} with m users. Formally, $M_{UU} = [u_{i,j}], 0 \leq i \leq m, 0 \leq j \leq m$, where $u_{i,j}$ represents the similarity between the i^{th} user and the j^{th} user. $u_{i,j}$ can be calculated using cosine similarity between the i^{th} user and the j^{th} user as follows:

$$sim_{u_i,u_j} = \frac{\sum_{l_k \in L} r_{i,k} \cdot r_{j,k}}{\sqrt{\sum_{l_k \in L} {r^2}_{i,k}} \sqrt{\sum_{l_k \in L} {r^2}_{j,k}}} \tag{2}$$

where $r_{i,k}$ represents the frequency of the user u_i visiting the location l_k.

Location-Location Relation. The location-location relation is also an important factor in an LBSN. The most typical usage is the item-based collaborative filtering which considers the similarity between locations. Recently, some researchers have turned to exploit the geographical influence of locations, i.e., the distance between every pair of locations visited by the same user, to model all users' visiting behaviors. On the one hand, the work [9] assumes that the distance of visited locations follows a power-law distribution. On the other hand, the work [26] assumes that the distance between visited locations and their centers (the most popular POIs) follows a multi-center Gaussian distribution. Both have shown that the geographical influence can improve location recommendation quality. However, both of these two distributions are derived from a check-in dataset where most of the POIs are local venues within one area. We argue that location popularity plays a dominant role for tourist attractions. Travelers often visit tourist attractions according to their popularity rather than their distances in a city. Due to the limitation of our dataset (i.e., people do not submit all of the photos for every place they visited to Flickr), we cannot demonstrate this assumption. We can only show that the locations a traveler visited distribute randomly for the geo-tagged photos. As a result, we only consider location similarity for the location-location relation in our 3R model. Note that if needed, the geographical influence can be integrated into our 3R model flexibly using a linear combination with the location similarity.

Definition 4 (Location-Location Relation): The relation of locations is represented by a weighted undirected graph $G_{LL} = (L, E_{LL}, W_{LL})$, where L is a set of nodes that represent the locations, E_{LL} is a set of edges that represent the location similarity, and the edge weights W_{LL} describe how similar two locations are.

An $n \times n$ adjacency matrix M_{LL} is built for G_{LL} with n locations. Formally, $M_{LL} = [l_{i,j}], 0 \leq i \leq n, 0 \leq j \leq n$, where $l_{i,j}$ represents the similarity between the i^{th} location and the j^{th} location. $l_{i,j}$ can be calculated using cosine similarity between the i^{th} location and the j^{th} location as follows:

$$sim_{l_i,l_j} = \frac{\sum_{u_k \in U} r_{k,i} \cdot r_{k,j}}{\sqrt{\sum_{u_k \in U} {r^2}_{k,i}} \sqrt{\sum_{u_k \in U} {r^2}_{k,j}}} \tag{3}$$

Unified 3R Model. Now, we unify the user-location relation, user-user relation and location-location relation into a single graph which models the location popularity, user similarity and location similarity simultaneously. We call this novel model 3R.

Definition 5 (3R model): For an LBSN with m users and n locations, its relations among users and locations can be represented using a weighted undirected graph G_{3R} which consists of three subgraph, G_{UL}, G_{UU}, G_{LL} .

Naturally, the adjacency matrices M_{UL}, M_{UU} and M_{LL} should be integrated into one matrix M_{3R} to represent the G_{3R}, which is depicted in Fig. 2(b). The weight of each edge represents the transition probability between two end nodes. Thus, we also normalize the weight of the edges belonging to G_{UL} by the maximum number of visits for a location.

B. Random Walk with Restart

After building the 3R model, the problem of ranking locations for a specific user is reduced to the problem of proximity evaluation between the user node and the location nodes. Random walk with restart (RWR) provides a good relevance score between two nodes in a weighted graph and thus is very appropriate to be used to analyze our 3R model. Random walk with restart is defined as Equation 4.

$$\vec{r}_{i+1} = c\widetilde{M}\vec{r}_i + (1-c)\,\vec{e} \quad (4)$$

Consider a query user[4] u_m that starts from node m. u_m iteratively travels to her neighborhood with the probability that is proportional to their edge weights. Additionally, at each step, u_m has some probability c to return to the node m. Here $\vec{r}_i$ is a column vector whose element $\vec{r}_i\,[x]$ denotes the probability that u_m is at node x at step i. $\vec{e}$ is an indicator vector with the element corresponding to the starting node set to 1 and other elements set to 0. $\widetilde{M}$ is a column normalized adjacency matrix of M_{3R} as illustrated in Fig. 2(b), where each element $M_{i,j}$ gives the probability of j being the next state given that the current state is i. The relevance score of node n with respect to the starting node m is defined as the steady-state probability $r_{m,n}$ that u_m will finally stay at node n. A popular approach to solve random walk with restart is the iterative method which iterates Equation 4 until convergence. Finally, we can rank the tourist attractions for the query user by their relevance scores, and recommend the top-K tourist attractions to the user.

The intuition of performing RWR on the 3R model is that for a specific traveler u, she will first follow the other travelers who share a high user similarity with her to some tourist attractions with a high probability. This is the intuition of user-based collaborative filtering. Then for each tourist attraction l, u will also visit other tourist attractions which are similar with l with a high probability. This is the intuition of item-based collaborative filtering. In other words, u performs a random walk on the 3R model. The advantage of this method is we can find both the popular locations and the relevant locations for the traveler. The former locations can be found since usually there are many links linked to them, while the latter locations can be found since they are linked to by the travelers who are similar to the specific traveler or by the other relevant locations. In summary, our 3R model can take into account location popularity, user similarity and location similarity simultaneously.

C. RWR for Group Recommendation

In reality, people often travel together, such as a couple or several friends. This fact drives us to extend our approach to recommend tourist attractions not only to individuals but also groups.

A state-of-the-art average strategy is to first rank the tourist attractions for each member in the group separately and then merge the relevance scores for each tourist attraction. Then, the top-K tourist attractions can be recommended to the group. Our RWR approach can be easily adapted to this strategy. We preserve a ranking vector $\vec{r}_u$ for each user u where the i^{th} item of $\vec{r}_u$ represents the relevance score of location i to user u. The size of $\vec{r}_u$ is the number of tourist attractions. Then for each group G, we compute a ranking vector $\vec{r}_g$ for this group, where $\vec{r}_g(j) = \sum_{u \in G} \vec{r}_u(j)$. Finally, recommendations can be provided based on r_g.

However, one drawback of this method is that RWR needs to be run for each member in a group. Suppose the group size is large, this is potentially very costly, especially for RWR which needs matrix multiplication. Thus, we propose another approach denoted as $\mathrm{RWR_g}$ which only needs to run RWR once no matter how large a group is.

At the beginning of the algorithm, instead of setting one element of $\vec{e}$ corresponding to a query user to 1, we set all the related elements (i.e., each related element corresponds to one of the members in the query group) to 1 and set other elements to 0. Then we can just perform RWR on the 3R model as before. Finally, we can recommend the top-K tourist attractions to the group according to their relevance scores.

Proposition 1 (Effectiveness & Efficiency of $\mathrm{RWR_g}$*):* $\mathrm{RWR_g}$ gives the same results as the average strategy while saves $(N-1)\cdot T$ running time compared with the average strategy, where N is the number of members in a group and T is the running time of the average strategy.

Proof: Suppose there are two members u_1 and u_2 in the group.

The average strategy works as follows.

$$\boldsymbol{R}' = c\boldsymbol{M}\boldsymbol{R}' + (1-c)\boldsymbol{E}' \text{ for } u_1$$
$$\boldsymbol{R}'' = c\boldsymbol{M}\boldsymbol{R}'' + (1-c)\boldsymbol{E}'' \text{ for } u_2$$
$$\hat{\boldsymbol{R}} = \boldsymbol{R}' + \boldsymbol{R}''$$

$\mathrm{RWR_g}$ works as follows.

$$\boldsymbol{R} = c\boldsymbol{M}\boldsymbol{R} + (1-c)\boldsymbol{E}$$

where $\boldsymbol{E} = \boldsymbol{E}' + \boldsymbol{E}''$.

We show that $\hat{\boldsymbol{R}} = \boldsymbol{R}$. For each step,

$$\begin{aligned}\hat{\boldsymbol{R}} &= \boldsymbol{R}' + \boldsymbol{R}'' \\ &= c\boldsymbol{M}\boldsymbol{R}' + (1-c)\boldsymbol{E}' + c\boldsymbol{M}\boldsymbol{R}'' + (1-c)\boldsymbol{E}'' \\ &= c\boldsymbol{M}(\boldsymbol{R}' + \boldsymbol{R}'') + (1-c)(\boldsymbol{E}' + \boldsymbol{E}'') \\ &= c\boldsymbol{M}\boldsymbol{R} + (1-c)\boldsymbol{E} \\ &= \boldsymbol{R}\end{aligned}$$

This equation remains valid until the steady state is reached. ■

Actually, the above proof is an extension of the following theorem.

Theorem 1 (Linearity of RWR [27]): For any preference vectors $\vec{r}_1$, $\vec{r}_2$ and positive constants w_1, w_2 with $w_1 + w_2 = 1$, the following equality holds:

$$RWR(w_1 \cdot \vec{r}_1 + w_2 \cdot \vec{r}_2) = w_1 \cdot RWR(\vec{r}_1) + w_2 \cdot RWR(\vec{r}_2)$$

Here we demonstrate that without the restriction of $w_1 + w_2 = 1$, Theorem 1 also holds.

[4]The delivery model of a recommendation system can be either ‘pull’ or ‘push’. Our system not only serves the users that make explicit requests for travel recommendation, but may also actively push suggestions.

D. Scalability of RWR Approach

There are two ways to perform RWR on the 3R model for each user or group, i.e., offline and online. For the offline method, we precompute Equation 4 for each user offline. Then for each user, we preserve a ranking vector $\overrightarrow{r}_u$. For an individual user, the recommendation can be provided to the user directly using her $\overrightarrow{r}_u$. For a group, the recommendation can be provided using the average strategy method illustrated in Section V-C. For the online method, we compute Equation 4 for each user or group using RWR or $\mathrm{RWR_g}$ respectively on the fly when the user or group queries the system. The advantage of offline method is that it is very efficient and can afford a system with large number of users. The disadvantage is that a ranking vector needs to be preserved for each user which incurs a high storage cost. The online method can save the storage cost. However, it is not efficient for a large system.

Fast random walk with restart. When we use the online method, the traditional iterative method cannot satisfy the real-time response requirement. Fortunately, there exists some efficient solutions which can reduce online processing time significantly. In [28], a method called NB_Lin is used to solve Equation 4. It uses a low-rank approximation and is much more efficient than the iterative method. In [29], a method called K-dash is used to efficiently find top-K relevant nodes in RWR. It exploits an estimation-based approach to prune unlikely nodes and unlike [28], it computes exact proximities of nodes. With these implementation optimizations, we can afford a system with rather large dataset and satisfy the real-time response requirement.

Region-based partition. No matter the offline method or the online method is adopted, a good recommendation system should tackle the scalability problem. If the dimension of the 3R model is very large, it is costly to perform matrix multiplication for both offline and online method. Thus, there should be some methods to reduce the number of users and locations in the 3R model.

For our application, it is impractical and unnecessary to consider all the worldwide locations and users in a single 3R model. Moreover, we need to update the 3R model at regular intervals, which will be costly if we have only one single 3R model. We make our approach scalable from two aspects. On the one hand, we partition the locations based on the region they locate and maintain a 3R model for each region. In this way, the number of locations in a 3R model can be dramatically reduced. Usually, the region can be chosen as a city, since travelers often treat a city as a unit. However, it should be noted that the performance would be different if we perform RWR on a sub-3R model for a city, since some relations will be lost compared with a global 3R model for worldwide locations. For example, consider the case that there is a location l outside the targeted city that has been visited by a query user. If some locations inside the targeted city are similar with l, these locations will also be visited by the user with a high probability based on the intuition of item-based CF if we use a global 3R model. We compare the global 3R model and sub-3R model in the experimental evaluation. Against the expectation, the performance of sub-3R model is better than that of the global 3R model, which shows that our region-partition technique not only is more scalable but also appears to be more effective (see experiment for further explanation). On the other hand, we also need to reduce the number of users in a 3R model. Note that for a region such as a city, the 3R model must contain the users who have visited the city since these users contribute to the recommendation performance. For the other users who have not visited the city, we can partition them based on their home region and maintain a 3R model for each home region. The home information of the users can be obtained using the method introduced in [30]. The reason we partition the users based on their home region is that we want to support group recommendation better if we use the online method. In general, the members in a group often come from the same home region, such as a city. Thus, we can perform $\mathrm{RWR_g}$ directly on the corresponding 3R model. If the members in a group come from different home regions which can also happen occasionally, we can first calculate the relevance ranking score of each location in a targeted city for each home region, and then average the relevance ranking scores. If we use the offline method, the users who have not visited the targeted city can be partitioned randomly. In this way, the number of users in a 3R model can be dramatically reduced.

In summary, the region-based partition technique has two advantages. First, the number of locations and users in a 3R model can be dramatically reduced which makes our system more scalable. Second, we can only update those 3R models with an obvious change due to the increased visiting to the targeted region.

VI. Experiments

A. Setup

Dataset. We collected metadata of geo-tagged photos that were taken in five different cities of USA between January 16, 2011 and January 17, 2013 from Flickr using its publicly available API. According to our region-based partition technique, we set the location region as a city. We first chose several major cities in USA. For each city, geo-tagged photos taken within 20km from its center were crawled. The characteristics of the dataset are shown in Table I(a).

After that, we used P-DBSCAN (see Section IV) to extract the tourist attractions within each city. In this clustering algorithm, there are two important parameters: $MinOwners$ which represents the minimum number of neighbors of a specific location, and ϵ which represents the neighborhood radius. The number of tourist attractions detected is different with different $MinOwners$ and ϵ. For a smaller $MinOwners$ and larger ϵ, there will be more tourist attractions detected, since the density of the cluster can be smaller. In order to evaluate whether different parameters can affect the performance of our method, we set two different sets of parameters (which are also used in [23]): (i) $MinOwners$=150, ϵ=30, and (ii) $MinOwners$=50, ϵ=50. The characteristics of the results are shown in Table I(b). As can be seen, more locations are detected with (ii).

Evaluation of individual recommendations. First, we test how our model performs when used to make recommendations for individuals, compared to other baseline approaches. For evaluation, we used the standard approach to divide the dataset into two parts: the training cities and the test city. We select

TABLE I. CHARACTERISTICS OF THE FLICK DATASET.

(a) original dataset

City	#photos	#users
NewYorkCity	277,151	7,046
WashingtonDC	163,101	3,507
Chicago	381,147	14,517
LosAngeles	165,278	5,452
Miami	94,603	7,288

(b) processed dataset

City	#locations	#users	#locations/user	#visits/location
NewYorkCity	232/702	3,011/4,321	4.8/6.2	76.3/38.2
WashingtonDC	145/618	1,050/2,118	6.7/7.3	52.5/25.8
Chicago	329/1,124	4,892/6,432	6.4/8.1	95.2/50.3
LosAngeles	102/498	956/1,882	4.3/5.8	45.3/28.4
Miami	52/344	1,132/1,980	2.8/3.2	71.4/35.8

100 users that have visited the test city and at least one other training city as the test users. Note that, we only consider those users who have visited at least 5 locations in the training cities to ensure that the test users have provided a decent amount of preference information. Then we filter out the tourist attractions the test users have visited in the test city (making the test users seem never been to the test city). Our aim is to find as many of these tourist attractions as possible in the test city for the test users.

Evaluation of group recommendations. Next, we test whether our model can provide quality recommendations for groups. We generated some user groups of varying sizes and similarities by sampling the test city dataset using the method similar with [12]. We consider two categories of groups which are common when people travel together, *random groups* and *similar groups*. The former group represents people who are not close, which will lead to a group with few common tastes, such as random people traveling on package tour. The latter group represents people with common tastes, such as a couple or several good friends. Within each category of groups, we consider groups containing three and six travelers. Similar groups are defined as those containing users with user-to-user similarity larger than 0.25. In our dataset, 24% of all the possible user pairs have similarity larger than the threshold of 0.25. Random groups are generated without considering any restriction on the user-to-user similarity. This leads to a smaller overall inner group similarity. Using the above method, we generate 100 groups for random groups and similar groups, with average similarity of the users 0.12 and 0.47, respectively.

When testing the group recommendation performance, we adopt two different methods. For a small group, we regard the tourist attractions in the test city visited by all of the members in the group as the ground-truth locations, and our aim is to test whether the recommendation algorithms can find these locations well. However, for a large group, it is impractical to use the same method, since it is hard to find some locations visited by all of the group members. Thus, we regard the tourist attractions visited by at least half of the group members as the ground-truth locations. In other words, we employ the majority voting strategy. If a tourist attraction is favored by most of the members in a group, it can be viewed as a good place.

B. Performance Metrics

In general, a tourist attraction recommendation algorithm computes a ranking score for each candidate location (i.e., tourist attraction of the targeted city) and returns N highest ranked tourist attractions as results. To evaluate the recommendation quality, it is important to find out how many tourist attractions actually visited by the user in the test data are discovered by the recommendation algorithm. Note that unlike the traditional recommendation systems (such as those for movie or book recommendation applications) where ranking of the results is sensitive, travel recommendation system pays more attention to the coverage of the results, as travelers do not have explicit ratings for the locations. For this purpose, we employ two standard metrics, *precision* and *recall*:

- Precision is defined as the ratio of the discovered tourist attractions to the N recommended tourist attractions, which is represented as *precision@N*.
- Recall is defined as the ratio of the discovered tourist attractions to the tourist attractions that have been visited by the user in the test set, which is represented as *recall@N*.

In our experiments, we test the performance when N=5,10.

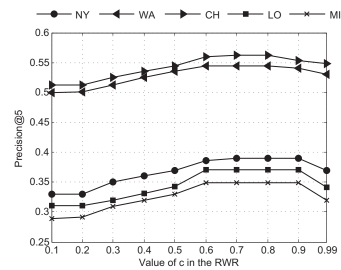
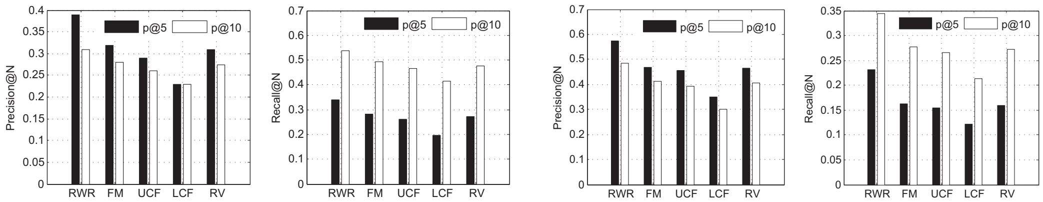


Fig. 3. P@5 when varing c parameter in RWR.

C. Recommendations for Individuals

1) Evaluated Approaches: The recommendation algorithms implemented in our experiments are listed below.

- **Rank by Visits** (**RV**): it ranks the tourist attractions based on the location popularity, i.e., the most visited tourist attractions in a give city. Notice that for tourism related applications, this is a hard-to-beat baseline.
- **User-based CF** (**UCF**): it ranks the tourist attractions based on the user similarity.
- **Item-based CF** (**LCF**): it ranks the tourist attractions based on the location similarity.
- **Fusion Method** (**FM**): it ranks the tourist attractions by using a simple combination of RV, UCF and LCF, i.e., $(1-\alpha-\beta)\cdot RV+\alpha\cdot UCF+\beta\cdot LCF$, which has been used by [9].
- **Random Walk with Restart** (**RWR**): it performs random walk with restart on the 3R model. We built the 3R model using the tourist attractions in the test city and the users who have visited the test city. Furthermore, we have evaluated what the best value for the damping parameter c is in RWR. Figure 3 shows the precision@5 when varying the parameter c. As can be seen, the value of c which falls between 0.6 and 0.9 corresponds to a better result



(a) $MinOwners$=150, ϵ=30 (b) $MinOwners$=150, ϵ=30 (c) $MinOwners$=50, ϵ=50 (d) $MinOwners$=50, ϵ=50

Fig. 4. Comparision of different recommendation algorithms for individuals.

TABLE II. AVERAGE PRECISION@5 FOR OTHER CITIES.

City	RWR	FM	UCF	LCF	RV
WashingtonDC	0.545	0.471	0.445	0.300	0.464
Chicago	0.562	0.508	0.492	0.342	0.500
LosAngeles	0.371	0.305	0.291	0.213	0.297
Miami	0.349	0.264	0.250	0.184	0.258

TABLE III. AVERAGE RECALL@5 FOR OTHER CITIES.

City	RWR	FM	UCF	LCF	RV
WashingtonDC	0.420	0.347	0.335	0.225	0.346
Chicago	0.421	0.372	0.354	0.238	0.370
LosAngeles	0.331	0.265	0.253	0.198	0.262
Miami	0.318	0.233	0.220	0.176	0.229

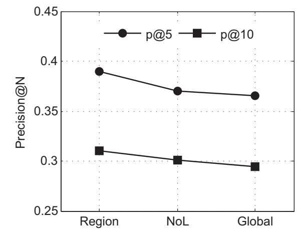


Fig. 5. Comparison of average precision@N.

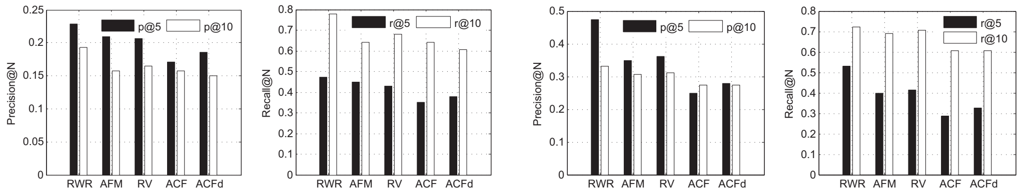
independently from the dataset on which RWR is applied. Similar observations can also be made for other N values in $precision@N$. Thus, in our experiments, the parameter c of RWR is set to 0.75 which can produce a good enough result.

2) Experimental Results: Fig. 4 shows the precision@N and recall@N of each algorithm when New York City is chosen as the test city. As can be seen, the popularity-based baseline RV produces better results than the other two personalized methods UCF and LCF. This is due to the sparsity of the data. In addition, travelers often go to the popular tourist attractions of the targeted city. We also note that item-based collaborative filtering LCF provides results much worse than the other methods, since travelers only provide photos for small subset of the tourist attractions. Thus, location similarity is not as accurate as user similarity. Moreover, for the simple combination method FM, we tune the parameters α and β to get the best possible results. We can see that FM outperforms the other three single methods, which indicates that it is helpful to combine the single methods. However, the best result of FM appears when α=0.15 and β=0.05, which shows that location popularity plays a dominant role in FM. This fact can be demonstrated by the results in Fig. 4. As can be seen, FM only performs a bit better than RV, which suggests that a coarse linear combination of the three single methods is not effective. Instead, we clearly observe that our approach RWR performs better than all the other baseline approaches on both the metrics independent of $MinOwners$ and ϵ. This shows the effectiveness of our 3R model which considers location popularity, user similarity and location similarity simultaneously. In the following, we only compare the performance of each recommendation algorithm for $MinOwners$=150, ϵ=30. We also test the performance of the recommendation algorithms for other cities for completeness. The results are shown in Table II and Table III. As can be seen, the results are in accordance with our analysis above.

Fig. 5 shows the precision@N for the region-partition technique described in Section V-D. **Region** denotes the method that builds a 3R model for the targeted city using the locations within the city, i.e., New York City. **Global** denotes the method that builds a 3R model using the worldwide locations, i.e., including the locations within other training cities. **NoL** denotes the method that does not consider the location-location relation. We have argued that the performance of Region is different from that of Global in Section V-D. As can be seen from Fig. 5, Region outperforms NoL and Global while NoL outperforms Global. Let us consider a scenario discussed in Section V-D. Assume there is a location l_o outside the targeted city that have been visited by a query user. If some locations inside the targeted city are similar with l_o, these locations will also be visited by this user with a high probability based on the intuition of item-based CF if a global 3R model is built. In fact, Global will produce a biased result towards location similarity. However, as can be seen from Fig. 4, LCF performs much worse than UCF, which indicates that location similarity is not as accurate as user similarity. This explains why the performance of Global is not as good as Region and NoL. Different from Global, location similarity can positively contributes to the performance of Region instead. This is because user similarity plays a dominant role compared with location similarity in Region. The intuition behind Region is that a user will first visit the locations L following similar neighbors based on user similarity. Then for each location l_i belonging to L, this user will also visit the other locations similar with l_i based on location similarity. In this way, not only the user similarity plays a dominant role, but also some new locations similar with l_i can be found which will be missed if the location similarity is not considered. Thus, Region performs better than both NoL and Global.

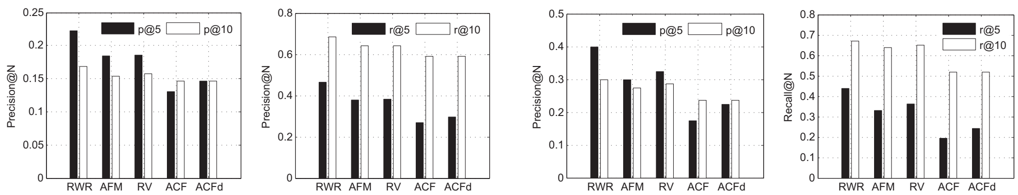
D. Recommendations for Groups

In this section, we compare the effectiveness of the group recommendation algorithms when varying the group category and the group size.



(a) small group (b) small group (c) large group (d) large group

Fig. 6. Comparision of different group recommendation algorithms for similar groups.



(a) small group (b) small group (c) large group (d) large group

Fig. 7. Comparision of different group recommendation algorithms for random groups.

1) Evaluated Approaches:

- **Average User-based CF** (**ACF**): it takes the average ranking score of each member in the group for a tourist attraction. The ranking score is obtained by using user-based CF approach.
- **Average User-based CF with disagreement** (**ACFd**): it considers both group relevance and group disagreement.
- **Average FM** (**AFM**): it takes the average ranking score of each member in the group for a tourist attraction. The ranking score is obtained by using FM approach.
- **Rank By Visits** (**RV**): it just recommends the most popular tourist attractions to the group.
- **Random Walk with Restart** (**RWR**): it sets the items of the indicator vector corresponding to the users in the group to 1 and performs random walk with restart.

We do not consider item-based CF approach since it produces results much worse than the other methods. In addition, we find that for the FM method, the results are the same when considering disagreement as well. This is because location popularity plays a dominant role in FM and the disagreement in the group can be ignored actually. Thus, we just show the results of the AFM method.

2) Experimental Results: Fig. 6 and Fig. 7 show the results of our experiments using similar groups and random groups with different group size respectively. Note that the reason the values of precision are higher for the large group is that we put a relaxed limitation on the candidate tourist attractions for large group because of the sparsity of the dataset. Basically, RWR, ACF and ACFd perform worse for random groups than similar groups. This is because people have more diverse tastes in the random groups which makes it hard to predict the tourist attractions that can satisfy all of the people's preferences. This can also explain why both collaborative filtering methods ACF and ACFd perform worse even than RV that just recommends travelers the most popular tourist attractions. Moreover, ACF and ACFd performs worse than RV even for the similar group where users' preferences are very similar, which indicates that collaborative filtering is not appropriate for travel group recommendation. ACFd does not always outperform ACF, which is accordance with the results shown in [14]. For AFM, its performance closely relates to the performance of RV since location popularity plays a dominant role in FM. However, different from the situation where it performs a bit better than RV for individuals, it performs worse than RV for both similar group and random group. This is because the effectiveness of AFM is influenced by the diverse taste in a group. Once again, it is clear that the RWR approach based on the 3R model outperforms the other baseline approaches with a significant improvement not only for similar groups but also for random groups. More importantly, RWR is the only approach that outperforms RV under all circumstances. A possible explanation for the superiority of RWR is that it captures the global structure of an LBSN, and we can combine the location popularity, user similarity and location similarity in a more effective way with our the 3R model. This is beneficial not only for individuals but also for groups.

E. Discussion

Collaborative filtering has shown its success in the traditional recommendation systems such as book recommendation and movie recommendation. However, we show that it is not appropriate for travel recommendation. The reason is threefold. First, unlike traditional recommendation systems where people have an explicit rating for the objects, travel recommendation systems only have an implicit rating such as visits for the locations which cannot represent travelers' preferences so well. Second, although many kinds of user-contributed resources have been studied to provide travel recommendation, all of them are faced with the data sparsity problem. Last but not least, object popularity plays a more important role in the travel recommendation systems, since travelers often visit the most popular places in an unfamiliar city. However, we cannot just use location popularity as the sole measure, which is not appropriate for a personalized recommendation system, let alone group recommendation where people's preferences are much more complicated. The main advantage of our approach is to unify the location popularity, user similarity and location similarity in a single model, which makes it perform much better than only considering one relation. Also, with the help of random walk with restart, the three factors are combined in an effective way to contribute to the personalized recommendation together. Moreover, when used for group recommendation, RWR is more effective compared with other ranking algorithms and is the only method that outperforms RV which considers location popularity only.

With the help of efficient random walk with restart algorithm and region-based partition technique described in Section V-C, our method can be implemented either offline or online and has a good scalability. In addition, another advantage of our 3R model is that we can integrate other optimized similarities among users and locations to provide better recommendation results. In this work, we only consider the classic collaborative filtering similarities.

VII. Conclusions

This research aims at building a travel recommendation system to provide personalized tourist attractions for individuals and groups. We build a novel 3R model which unifies user-location relation, user-user relation and location-location into a single framework. We then use random walk with restart to analyze our 3R model. We further extend the random walk with restart approach to support group recommendation. Experimental results show that the unified framework provides significantly superior performance than all other recommendation approaches evaluated for both individuals and groups under various settings. Besides that, our 3R model also has some other good properties. First, it can integrate other optimized similarities among users and locations to provide better recommendation results. Second, it is very flexible which can be used on other datasets where users and locations can be extracted such as datasets from Foursquare, Twitter and Wikipedia.

Acknowledgment

This work is funded by the NExT Search Centre (grant R-252-300- 001-490), which is supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Program Office.

References

[1] J. Liu, Z. Huang, L. Chen, H. T. Shen, and Z. Yan, "Discovering areas of interest with geo-tagged images and check-ins." in *ACM Multimedia*, 2012, pp. 589–598.

[2] G. D. Abowd, C. G. Atkeson, J. I. Hong, S. Long, R. Kooper, and M. Pinkerton, "Cyberguide: A mobile context-aware tour guide." *Wireless Networks*, vol. 3, no. 5, pp. 421–433, 1997.

[3] Y. Huang and L. Bian, "A bayesian network and analytic hierarchy process based personalized recommendations for tourist attractions over the internet." *Expert Syst. Appl.*, vol. 36, no. 1, pp. 933–943, 2009.

[4] J. Kim, H. Kim, and J. hee Ryu, "Triptip: a trip planning service with tag-based recommendation." in *CHI Extended Abstracts*, 2009, pp. 3467–3472.

[5] T. Horozov, N. Narasimhan, and V. Vasudevan, "Using location for personalized poi recommendations in mobile environments." in *SAINT*, 2006, pp. 124–129.

[6] M. Clements, P. Serdyukov, A. P. de Vries, and M. J. T. Reinders, "Personalised travel recommendation based on location co-occurrence." *CoRR*, vol. abs/1106.5213, 2011.

[7] Y. Shi, P. Serdyukov, A. Hanjalic, and M. Larson, "Personalized landmark recommendation based on geotags from photo sharing sites." in *ICWSM*, 2011.

[8] M. Ye, P. Yin, and W.-C. Lee, "Location recommendation for location-based social networks." in *GIS*, 2010, pp. 458–461.

[9] M. Ye, P. Yin, W.-C. Lee, and D. L. Lee, "Exploiting geographical influence for collaborative point-of-interest recommendation." in *SIGIR*, 2011, pp. 325–334.

[10] L. Boratto and S. Carta, "State-of-the-art in group recommendation and new approaches for automatic identification of groups." in *Information Retrieval and Mining in Distributed Environments*, 2011, vol. 324, pp. 1–20.

[11] A. Jameson and B. Smyth, "Recommendation to groups." in *The Adaptive Web*, 2007, pp. 596–627.

[12] L. Baltrunas, T. Makcinskas, and F. Ricci, "Group recommendations with rank aggregation and collaborative filtering." in *RecSys*, 2010, pp. 119–126.

[13] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar, "Rank aggregation methods for the web." in *WWW*, 2001, pp. 613–622.

[14] S. Amer-Yahia, S. B. Roy, A. Chawla, G. Das, and C. Yu, "Group recommendation: Semantics and efficiency." *PVLDB*, vol. 2, no. 1, pp. 754–765, 2009.

[15] L. Lovász, "Random walks on graphs: A survey." in *Combinatorics, Paul Erdös is Eighty*, 1996, vol. 2, pp. 353–397.

[16] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web." Stanford InfoLab, Technical Report 1999-66, November 1999. [Online]. Available: http://ilpubs.stanford.edu:8090/422/

[17] J. Urban and J. M. Jose, "Adaptive image retrieval using a graph model for semantic feature integration." in *Multimedia Information Retrieval*, 2006, pp. 117–126.

[18] Y. Xu, X. Yi, and C. Zhang, "A random walks method for text classification." in *SDM*, 2006.

[19] N. Craswell and M. Szummer, "Random walks on the click graph." in *SIGIR*, 2007, pp. 239–246.

[20] F. Fouss, A. Pirotte, J.-M. Renders, and M. Saerens, "Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation." *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 3, pp. 355–369, 2007.

[21] I. Konstas, V. Stathopoulos, and J. M. Jose, "On social networks and collaborative recommendation." in *SIGIR*, 2009, pp. 195–202.

[22] A. Noulas, S. Scellato, N. Lathia, and C. Mascolo, "A random walk around the city: New venue recommendation in location-based social networks," in *SocialCom '12*, 2012, pp. 144–153.

[23] S. Kisilevich, F. Mansmann, and D. A. Keim, "P-dbscan: a density based clustering algorithm for exploration and analysis of attractive areas using collections of geo-tagged photos." in *COM.Geo*, 2010.

[24] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *KDD*, 1996, pp. 226–231.

[25] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: user movement in location-based social networks." in *KDD*, 2011, pp. 1082–1090.

[26] C. Cheng, H. Yang, I. King, and M. R. Lyu, "Fused matrix factorization with geographical and social influence in location-based social networks." in *AAAI*, 2012.

[27] T. H. Haveliwala, "Topic-sensitive pagerank," in *WWW*, 2002, pp. 517–526.

[28] H. Tong, C. Faloutsos, and J.-Y. Pan, "Fast random walk with restart and its applications." in *ICDM*, 2006, pp. 613–622.

[29] Y. Fujiwara, M. Nakatsuji, M. Onizuka, and M. Kitsuregawa, "Fast and exact top-k search for random walk with restart." *PVLDB*, vol. 5, no. 5, pp. 442–453, 2012.

[30] A. Popescu and G. Grefenstette, "Mining user home location and gender from flickr tags." in *ICWSM*, 2010.