RECOMMENDING INTERESTING LANDMARKS IN PHOTO SHARING SITES

Jinpeng Chen, Yu Liu, Zhenyu Wu, Ming Zou[*], *Deyi Li*[†]

Abstract: With the rapid development of location-acquisition technologies (GPS, GSM networks, etc.), more and more unstructured, geo-referenced data are accumulated on the Web. Such abundant location-based data imply, to some extent, users' interests in places, so these data can be exploited for various location-based services, such as tour recommendation. In this paper, we demonstrate that, through utilizing the location data from a popular photo sharing web site such as Flickr, we can explore interesting landmarks for recommendations. We aim to generate personalized landmark recommendations based on geo-tagged photos for each user. Meanwhile, we also try to answer such a question that when we want to go sightseeing in a large city like Beijing, where should we go? To achieve our goal, first, we present a data field clustering method (*DFCM*), which is a density-based clustering method initially developed to cluster point objects. By using *DFCM*, we can cluster a large-scale geo-tagged web photo collection into groups (or landmarks) by location. And then, we provide more friendly and comprehensive overviews for each landmark. Subsequently, we present an improved user similarity method, which not only uses the overview semantic similarity, but also considers the trajectory similarity and the landmark trajectory similarity. Finally, we propose a personalized landmark recommendation algorithm based on the improved user similarity method, and adopt a TF-IDF like strategy to produce the nontrivial landmark recommendation. Experimental results show that our proposed approach can obtain a better performance than several state-of-the-art methods.

Key words: *Landmark recommendation, Flickr, GPS trajectories, geo-tags, landmark overview*

Received: September 2, 2013 **DOI:** 10.14311/NNW.2014.24.017
Revised and accepted: May 21, 2014

1. Introduction

With the increasing availability of location-aware mobile devices (e.g., GPS-enabled portable devices), wire-less communication technologies (e.g., 3G and Wi-Fi), map

[*]Jinpeng Chen – Corresponding Author, Yu Liu, Zhenyu Wu, Ming Zou, State Key Laboratory of Software Development Environment, BeiHang University, Beijing, E-mail: {chenjinpeng,liuyu,wuzhenyu,zouming}@nlsde.buaa.edu.cn

[†]Deyi Li, Institute of Electronic System Engineering, Beijing, E-mail: lideyi@nlsde.buaa.edu.cn

services (e.g., Google Maps[1], Microsoft Bing Maps[2], and Yahoo! Maps[3]), and spatial database management systems (*DBMSs*) [11], a number of location based social networking services (*LBSNS*) have emerged in recent years, such as Loopt[4], Flickr[5], Panoramio[6] et al. People are now capturing and uploading far more photographs than ever before. In addition, LBSNS allow users to tag, rate and describe locations as they visit them, in order to aid the discovery of locations they may be interested in [10]. Therefore, billions of photos shared on websites such as Flickr serve as a growing record of our culture and environment. These photos are annotated with various forms of information including GPS coordinates, time stamps, photographer, and a wide variety of textual tags.

In this paper, we face the three challenges: (1) How to organize a large collection of photos with all those kinds of information? (2) How to model the users' dynamic behaviors based on geo-tagged photos? (3) How to generate personalized landmark recommendations based on geo-tagged photos for each user? We aim to generate a representative and comprehensive landmark overview, explore users' similarity measures which descript users' behaviors suitably, and finally implement the personalized landmark recommendation. Our work can be widely applied in location-based service, tour recommendation, travel assistance, image search optimizing and images management etc.

In order to handle the above three challenges, we present different strategies. For challenge (1), we first do some clustering, automatically detecting hot landmarks and generate an overview for each landmark. To automatically detect hot landmarks, we propose a clustering method based on data field, which is a density-based clustering method initially developed to cluster point objects. For challenge (2), we present an improved user similarity measure, which combines the semantics indicated by geo-tagged photos with travel trajectories of the user, to model the user's dynamic behaviors. For challenge (3), we put forward a collaborative filtering (*CF*) algorithm based on the improved user similarity, to generate personalized landmark recommendations for each user. We further adopt a TF-IDF like strategy in order to produce the nontrivial landmark recommendation.

Our work leverages the set of geo-tagged photos taken in Beijing from Flickr, which is a popular photo sharing web site that supports user-contributed tags and geo-referenced (or geo-tagged) photos. The location of geo-tagged photos can be automatically captured by the camera or a location aware device. The reasons we chose Flickr here are the following: Flickr, as one of the world's most popular photo sharing websites, enables users to label their friends' photos, so this brings us opportunities to understand the correlation between users and locations; Flickr contributes a tremendous volume of travel data which is a rich source of travel trajectories that can match various user preferences; Flickr is claimed to host more than one billion photos associated with tags and GPS location, which provides affluent data for analyzing geographical characteristics and generating personalized landmark recommendations based on geo-tagged photos for each user.

---

[1] http://maps.google.com.
[2] http://www.bing.com/maps.
[3] http://maps.yahoo.com.
[4] http://www.loopt.com.
[5] http://www.flickr.com.
[6] http://www.panoramio.com.

The rest parts of the paper are organized as follows. We start out discussing the related work in Section 2. In Section 3, we introduce some notations and some terms used in this paper. In Section 4, we describe the detailed process of detecting landmarks and generating landmark overviews. In Section 5, we propose a personalized landmark recommendation algorithm based on the improved user similarity method. In Section 6, we evaluate our approach based on the geo-tagged photos from Flickr. Some experiment results are also presented. Finally, we draw conclusions and offer an outlook for our future work in Section 7.

2. Related Work

Location recommendation is an important feature in location based services (LBSs). It aims to provide location suggestions that a user may be interested in [6, 7, 8, 9]. In location recommendation, there have already been a reasonable amount of researches [13, 14, 15, 22, 23, 34].

In [27], Arase et al. defined the idea of a photo trip and proposed frequent photo trip pattern mining algorithms that can detect novel trip knowledge (i.e., frequently visited city sequences and typical visit duration) from geo-tagged photo collections on the Web. In [8], Zheng et al. introduced a social networking service, called Geo-Life, which aims to understand trajectories, locations and users, and mine the correlation between users and locations on the basis of user-generated GPS trajectories. Eagle et al. [29] aimed to recognize the social pattern in daily user activity from the dataset collected by 100 users with a Bluetooth-enabled mobile phone. Takeuchi et al. [30] attempted to recommend shops to users based on their individual preferences estimated by analyzing their past location histories.

In [12], Zheng et al. raised a question that for a specific user, if she wishes to do some sightseeing or food-hunting in a large city such as Beijing, where should she go, given her previous GPS traces and other similar users' GPS histories? To solve this case, they proposed a novel approach, known as user-centered collaborative location and activity filtering ($UCLAF$), to pull many users' data together and apply collaborative filtering to find like-minded users at different locations. We also try to answer this question, but we adopt a different approach. Starting from point of geo-tagged photos, first, we start out clustering a large-scale geo-tagged web photos collection into groups by location, and then generate an overview for each landmark. Subsequently, we propose an improved user similarity method via combining overviews semantic similarity and travel trajectories similarity of users. Finally, we recommend personalized landmarks by using the improved user similarity based CF algorithm.

To model the users' similarity, in [16], Ying et al. proposed a novel approach for recommending potential friends based on users' semantic trajectories for location-based social networks. The core of their proposal is a novel trajectory similarity measurement, namely, Maximal Semantic Trajectory Pattern Similarity ($MSTP$ Similarity), which measures the semantic similarity between trajectories. Zheng et al. [7] proposed a personalized friend and location recommender for the Geographical Information Systems (GIS) on the web, as well as a framework, namely "hierarchical-graph-based similarity measurement ($HGSM$)" to uniformly model each individual's location history and effectively measure the similarity among users. In [6], Li et al. presented an approach which mines the similarity of people's trajectories based on location histories. Their proposed trajectories consist of a series of stay points representing some places where a user stays for a while and carry a particular semantic meaning. However, our presented users' similarity measure not only utilizes the similarity of people's trajectories but also considers the semantic similarity of landmark overviews.

In [17], Zheng et al. modelled individuals' location histories with a tree-based hierarchical graph (*TBHG*). Based on the *TBHG*, they proposed a *HITS*(Hypertext Induced Topic Search)-based inference model which is used to recommend the tourism hot spots that are popular and highly recommended by the experienced users. In [18], Zheng et al. provided a collaborative filtering (*CF*) approach based on collective matrix factorization to take information sources from users as inputs and train a location and activity recommender. The *GM-FCF* system [19] directly made location aware recommendations to users using a novel combination of social relations and geographic information. In [20], Cao et al. put forward a framework that encompasses new techniques for extracting semantically meaningful geographical locations from the proliferation of *GPS* data, and for the ranking of these locations according to their significance. In [25], Papadimitriou et al. proposed a Geo-social recommender system which is capable of recommending friends, locations and activities. They used a tensor, which is updated by incremental tensor approaches, as new users, locations, or activities are being inserted into the system. In [26], Clements et al. pointed out that a user's favourite landmarks in a previously unvisited city can be predicted by re-ranking the most popular locations of the users with similar travel preference. Their results indicated that statistical improvement over all users is hard to achieve, but for users with a clear travel preference very accurate predictions can be made. In [21], Shi et al. provided a novel category-regularized matrix factorization approach (*CRMF*) to recommend landmarks to individual users based on both user-landmark preference information and category-based landmark similarity. Our aim is similar to [21], which focus on personalized landmark recommendation based on geo–tagged photos, but our recommendation method is different. In this work, we regard CRMF as one of the baselines.

3. Preliminary

In this section, we will clarify some notations and terms used in this paper.

3.1 Notations

There are three basic elements in our dataset: photos, tags and users. We define a set of photos as $F = \{f\}$, where $f$is a tuple (i_f, l_f, u_f) containing a unique photo *ID* i_f, a place l_f, which is also a tuple $(longi_f, lat_f)$ representing photo's capture location coordinates and u_f represents a Flickr user who uploaded the photo f. Set of tags is symbolized as $A = \{a\}$. We use a to denote a tag, A_f to denote a set of tags associated to the photo$f$. So set of all tags can also be denoted as $A = \cup_{f \in F} A_f$. A subset of photos associated with a specific tag can be defined as $F_a = \{f \in F | a \in A_f\}$, and given c is a cluster, set $F_c = \{f \in F | a \in c\}$ means a set of photos with a specific cluster of tags, and $F_{c,a} = F_a \cap F_c$ denotes photos with a tag a in a cluster c. Set of users is denoted by $U = \{u_f\}$. Same way, $Ua, c = U_a \cap U_c$ denotes users set who use a tag a in a cluster c.

3.2 Definitions

Definition 1 GPS Point (or GPS Coordinate): A GPS point p is a four-tuple: $\langle x, y, t, m\rangle$, where x and y are Euclidean coordinates, t is the timestamp and m is the times a user has visited this GPS point (we can call it mass or score of the GPS point). It is depicted in the left part of Fig. 1.

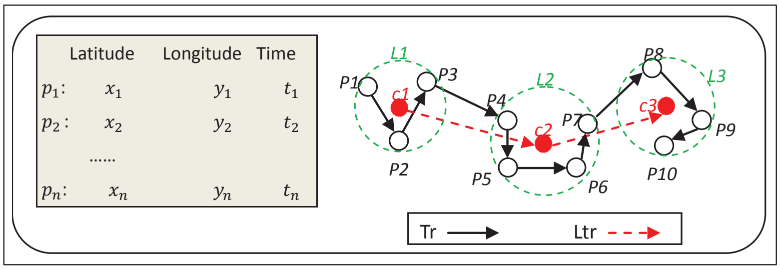


Fig. 1 *GPS points, a trajectory and a landmark trajectory.*

Definition 2 Landmark (or Hot Spot): A landmark is a geographical region which is obtained using *DFCM*. A landmark L consists of a group of consecutive GPS points $P = \{p_1, p_2, \ldots, p_n\}$. Formally, conditioned by P, ta, tl, lm and ov, a virtual landmark center $\widetilde{c} = (x, y, ta, tl, lm, ov)$, where

$$\widetilde{c}.x = \sum_{i=1}^{n} p_i.x/|P| \tag{1}$$

$$\widetilde{c}.y = \sum_{i=1}^{n} p_i.y/|P| \tag{2}$$

respectively stands for the average latitude and longitude of the collection P, $ta = p_1.t$ and $tl = p_n.t$ represent user's arrival and leaving time on L, $lm = \sum_{i=1}^{n} p_i.m$ represents the times a user has visited this landmark (we can call it mass or score of the landmark) and ov represents overviews of this landmark L. It is shown in the right part of the Fig. 1.

Definition 3 Trajectory: As shown solid arrow line in the right part of Fig. 1, a trajectory with a score Tr of a user is a sequence of GPS points based on a certain threshold ΔT. Thus, $Tr = p_1 \rightarrow p_2 \rightarrow \ldots \rightarrow p_n$, where $p_i \in P$,$p_{i+1}.t > p_i.t$, $p_{i+1}.t - p_i.t < \Delta T(1 \leq i \leq n)$ and the score is the times a user has passed this trajectory.

Definition 4 Landmark Trajectory: As shown dotted arrow line in the right part of Fig. 1, a landmark trajectory with a score Ltr of a user is a sequence of landmarks based on a certain threshold ΔT. Thus, $Ltr = L_1 \rightarrow L_2 \rightarrow \ldots \rightarrow L_m$, where $L_i \in P$, $L_{i+1}.t > L_i.t$, $L_{i+1}.t - L_i.t < \Delta T (1 \leq i \leq m)$ and the score is the times a user has passed this landmark trajectory.

Definition 5 M-length Trajectory: If the number of nodes in a trajectory is M, we call this trajectory M-length trajectory. As shown in the right part of Fig. 1, $p_1 \rightarrow p_2$ is a 2-length trajectory. Similarly, we can define an M-length landmark trajectory.

4. Generate Landmark Overview

In this section, we describe the detailed process of generating landmark overviews. This process is divided into three parts: detecting landmarks, scoring tags within a cluster, and mining topics within a landmark.

4.1 Detect Landmarks

We intuit that interesting landmarks (or hot spots) attract more visitors, thus there are more geo-tagged photos taken in it. In order to better detect landmarks (or hot spots) of users' interests in the geographic space, we can cluster geo-tagged photos. But cluster results are influenced by the granularity of the location. For instance, the larger the extent of a place, the longer the distance the activities could occur in [1]. To accommodate variable granularity, we propose a data field clustering method, which is a density-based clustering method initially developed to cluster point objects. We consider the Mean Shift (MS) [2], which has been shown effective for spatial data clustering in previous work [3, 4], as a baseline method in order to verify the effectiveness of our proposed algorithm.

We elaborate the notion of data field [32] prior to introducing the data field clustering method.

4.1.1 Data Field

There are many fields in physics, such as gravitational field, electric field and magnetic field et al. All of these fields describe the law of interaction of particles. Inspired by the knowledge of physical fields, we introduce the interaction of particles and the concept of "field" into the data space. Given a dataset containing n objects in space $\Omega \subseteq \mathrm{R}^{\mathrm{P}}$, i.e., $D = \{x_1, x_2, \ldots, x_n\}$, where $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$, $i = 1, 2, \ldots n$. Each data object can be considered as a mass point or nucleon with a certain field around it and the interaction of all data objects will form a data field through the space.

Because Gaussian function has good mathematic properties, in this work we adopt Gaussian function to define the potential at any point x as,

$$\phi(x) = \sum\nolimits_{i=1}^{n} \phi_i(x) = \sum\nolimits_{i=1}^{n} \left(m_{\times} e^{-\left(\frac{||x - x_i||}{\sigma}\right)^2} \right) \tag{3}$$

where $||x - x_i||$ is the distance between object x_i and x, m_i is the mass of object x_i, and σ is the influence factor that indicates the range of interaction.

In this work, we assume each data object x is supposed to be equal in the mass (that is, each data object x possesses the same influence over the space) and meets a normalization condition $\sum_{i=1}^{n} m_i = 1$. Thus a simplified potential function can be given as,

$$\phi(x) = \sum\nolimits_{i=1}^{n} \phi_i(x) = \frac{1}{n} \sum\nolimits_{i=1}^{n} e^{-\left(\frac{||x-x_i||}{\sigma}\right)^2} \tag{4}$$

where x is a GPS coordinate and $||x - x_i||$ is a Euclidean distance.

Given a data set in space, the distribution of the associated data field is primarily determined by the influence factor σonce the form of potential function is fixed. Thus, the choice of the σshould make the distribution of potential field as consistent with the underlying distribution of original data as possible. And then, we should find the best choice of the σ.

In order to minimize the uncertainty, Shannon entropy principle is used as Equation 5 to optimize the influence factor σ.

$$\min H = \min_{\sigma} \left(- \sum\nolimits_{i=1}^{n} \frac{\phi_i}{Z} \log \left(\frac{\phi_i}{Z} \right) \right) \tag{5}$$

where $Z = \sum_{i=1}^{n} \phi_i$ is a normalized factor.

4.1.2 Clustering Method Based On Data Field

According to the data field theory and its definition, we propose the data field clustering method (*DFCM*, in Algorithm 1) which clusters the points in the data space based on the strength of interaction of objects.

Algorithm 1: *DFCM*
Input: $D = \{x_1, x_2, \ldots, x_n\}$, sample number n_{sample}, noise threshold ξ
Output: the hierarchical partition $\{\Pi_1, \Pi_2, \ldots, \Pi_k\}$
Steps:
Select n_{sample} samples randomly to construct the sample data set *SampleSet*.
//Optimization of the influence factor σ
σ = *Optimal_Sigma*(*SampleSet*)
//Apply grid partition on the space and construct an index tree
Map = *CreatMap(D,* σ)
//Search in the topological critical points
CriticalPoints = *Search_CriticalPoints(Map,* σ)
Set *MaxPoints* as the set of local maximum points and *SadPoints* as the set of saddle points.
//Initially divide the data according to the set of local maximum points
Π_1 = *Initialization_Partition(Map, D, MaxPoints,* σ, ξ)
//Combine the initial clusters iteratively according to the set of saddle points
$[\Pi_1, \Pi_2, ...\Pi_k]$ = *Saddle_Merge* (*Map*,Π_1, *MaxPoints, SadPoint,* σ, ξ)

The idea of this clustering method is to optimally select the influence factor σ for the generation of potential field distribution first. Thereafter, the data objects contained in each equipotential line/surface are treated as a natural cluster, and the nested structures consisting of different equipotential lines/surfaces are treated as the cluster spectrum. Thus, the clustering at different hierarchies is realized.

The local maximum points could be regarded as "virtual field sources", and all the data objects are convergent by self-organization due to the attraction by their own "virtual field sources". Thus, the local maximum points can be regarded as cluster centers, and the initial partition is formed. To obtain the clustering at different hierarchies, the initial clusters are combined based on regular saddle point iteration.

In order to find local maximum points and saddle points in the potential field distribution, the algorithm first searches all the critical points satisfying $\nabla\phi\left(x\right)=0$. Thereafter, it classifies the critical points according to the eigenvalues of the Hessse matrix $\nabla^2\phi(x)$. For a given critical point x, let $l_1 < l_2 \ldots < l_d$ be the d eigenvalues of the Hesse matrix, where $d >= 2$ is the dimension of the space. If $l_d < 0$, x is the local maximum point in the potential field distribution; if $l_1 > 0$, x is a local minimum point in the potential field distribution; if $l_1, l_2, \ldots, l_d \neq 0$ and the number of positive eigenvalues and the number of negative eigenvalues are both bigger than 1, the point x is the saddle point in the potential field.

There are two parameters in the Algorithm 1: random sample number n_{sample} and noise threshold ξ. n_{sample} is usually set as $n_{sample} = [\alpha * n], \alpha > 0.05$, while ξ is applied to judge whether the initial clusters are meaningful. Because the final clustering result is determined by the regular saddle points in the potential field, ξ, which is in a relatively stable domain, will not affect the clustering result. If noise data is not included in the data set D, the value domain of ξ will be $\left[0, \min\limits_{x\in MaxPoints} \phi\left(x\right)\right]$, and usually let $\xi = 0$. Let $||D_{noise}||$ be the size of noise data set, if the noise data is included in D, then $\bar{c} \leq \xi \leq \min_{x\in\bar{A}} \varphi(x)$, where $\bar{A}$ is the set of local maximum points generated by the non-noisy data, and $\bar{c}$ is equivalent to the potential value generated by $||D_{noise}||$ noise data with uniform distribution, i.e. $\bar{c} = \varphi_{D_{noise}}(x)$.

In this work, we can get the landmarks using *DFCM*. In addition, our proposed *DFCM* is a nonparametric method which does not require specifying the number of clusters, and does not assume the shape of the clusters.

4.2 Score Tags within Clusters

Once the previous clustering work has been done, we regard each cluster and its central point as a landmark and a landmark center respectively. Next, we will score the tags within a cluster.

When we try to detect representative tags for landmarks within a city, we should take measures to cope with noise tags. The number of photos with inaccurate coordinates cannot be ignored. It is really common that users could just drag the photo originally taken in a landmark like the Great Wall, to the Tiananmen Square when use Flickr map interface to record photo's capture coordinates, for Tiananmen is recognized as Beijing's symbol and some users don't care the detailed place information as long as the city is right. Another issue, in our Beijing photos dataset, some tags like "Beijing", "Asia", "China", appear much frequently across many landmarks, obviously there is no sense to select them as representative tags.

Trying to resolve the above problems and make a reasonable score for a tag, we get some heuristics from a simple analysis on the dataset:

- Tags that occur in a concentrated area (and do not occur often outside that area) are more representative than the ones that occur diffusely over a large region.
- The more users that used a tag in an area there are, the more representative the tag is for that area.

Given those heuristics, we believe a TF-IDF like method should be effective and reasonable for scoring tags. The term frequency for a given tag a in a cluster c can be denoted as:

$$tf\,(a,c) = |\,F_{c,a}|\;/\;|\,F_c| \tag{6}$$

and IDF part can be denoted as:

$$idf\,(a) = \log\left(|C|\;/\sum_{c\in C} I\,(c,a)\right) \tag{7}$$

where $C = \{c\}$ is the set of clusters, and function I is defined as:

$$I(a,c) = \begin{cases} 1, & if \quad a \in c \\ 0, & if \quad a \notin c \end{cases} \tag{8}$$

According to heuristics, we also take the tag's geographical distribution into account, so we introduce a new factor MTF, which is defined as:

$$mtf\,(a,c) = \sum_{c\in C} tf(c,a)/\,|C| \tag{9}$$

We believe the higher the ratio $tf\,(c,a)\,/mtf\,(c,a)$ is, the more representative the tag a is. We also take factor $uf(a,c) = |U_{c,a}|$ into our score function, which was first proposed in work [31] and has been tested effective to cope with the noise issues (e.g., in some scenarios, data should be contaminated if a single user took a large number of photos in one location and labelled them with the same tag) and we get our final score function for tag a in cluster $c$as:

$$tag\_score(a,c) = \frac{(tf(a,c)\cdot idf(a)\cdot uf(a,c))}{mtf(a,c)} \tag{10}$$

4.3 Topics Mining Within Landmarks

Once we have scored the tags within landmark, the top N score tags can be seen as the landmark's representative tags. But to be comprehensive we also want to show connections among tags. Next, we turn to the task of mining the topic groups among tags. It has been proposed that folksonomies contain nested groups of tags related to common topics in work [28].

Here, we utilize a clustering method based on tags similarity directed graph for topics mining. First we give our definitions for tag similarity directed graph.

In a directed graph G, nodes represent tags, edge weights represent strength of similarity, and strength of similarity is based on the number of tag co-occurrences (both two tags occur in one photo). The weight of edge from a_1 to a_2 was defined as $sim(a_1, a_2) = |F_{a_1,a_2}|/|F_{a_2}|$ and the weight of edge from a_2 to a_1 was defined as $sim(a_2, a_1) = |F_{a_1,a_2}|/|F_{a_1}|$. Next we give the cluster algorithm.

In algorithm 2, line 2 utilizes the equation (10) for computing tags' score, the higher score tag will be more prior to mine its related downstream neighbors. The algorithm's time is mainly spent in computing similarity between two nodes, and mining topic group is linear time costing. The time complexity is $O(n^2)$, where n is the number of unique tags.

Algorithm 2: Clustering on directed tag graph

Input: directed tag graph G, similarity threshold τ
Output: clusters set C
1: Initialize clusters set $C = \emptyset$, set $U = \{u|u \in G.nodes\}$
2: Initialize array $centers = sorted(U)$ in descending order of $tag\_score(u)$
3: Remove the edges with weight less than τ
4: For$c$ in centers do
5:     If c is not visited do
6:         $v = v \cup c$
7:         While $v! = \emptyset$ do
8:             $n = v.pop()$
9:             If n is not visited do
10:                 Find n's downstream neighbors set $\{n_d\}$ and update $v = v \cup \{n_d\}$
11:             End if
12:         End while
13:         Update $C = C \cup v, v = \emptyset$
14:     End if
15: End for

4.4 Generate Landmark Overviews

Given the results of the clustering step, we rank the clusters according to how well they represent the various tags within a landmark. For each cluster, we use the highest score of its member tags as the cluster's score. We select tags with score above a threshold from the top N score clusters as topics to display, and hide the lower-ranked clusters.

After text overview is determined, the overview photos generating is just a retrieval work. For each topic, we just retrieve photos whose tags are appeared in the topic tags set. The photos are ranked according to the numbers of its own tags appeared in corresponding topic's tags set. Fig. 2 shows the text and photos overview of Landmark Tsinghua. In this case, top three topics in this landmark were selected to display.

As shown in the Tab. I, it also presents an example of a landmark (Tsinghua) overview. The first entity of each row represents a ranking topic of an overview, and the following entities represents ranking sub-topics.

Ranking Overview	Topic	Sub-topic		
		1	2	3
1	tsinghua university	students	classroom	auditorium
2	old summer palace	qing dynasty	lotus	imperial gardens
3	peking university	weiming lake	baya pagoda	campus

Tab. I *An overview of landmark Tsinghua, three topics and three sub-topics are selected for displaying.*

Fig. 2 *Overviews of landmark Tsinghua, three topics are selected for displaying and red italic tags are with scores at top 20.*

5. Recommend Interesting Landmarks

In this section, we introduce our process of recommending interesting landmarks in detail. First, we model the user's similarity, which not only uses the overview semantic similarity, but also considers the trajectory similarity and the landmark trajectory similarity. And then, based on the user's similarity, we present a novel landmark recommendation algorithm.

5.1 User Similarity Exploration

In this section, we detail the processes of user similarity exploration, including location history extraction, overview semantic similarity extraction, trajectory similarity extraction and landmark trajectory similarity extraction.

5.1.1 Location History Extraction

We construct two location histories: users' trajectories and users' landmark trajectories.

With users' travel experiences and their interests in locations, we can calculate a classical score for each GPS point (or each landmark) and each trajectory (or each landmark trajectory) within the given geospatial region. The classical score of each GPS point (or each landmark) is regarded as the times users have visited this GPS point (or landmark). Also, the classical score of each trajectory (or landmark trajectory) is regarded as the times users have passed this trajectory (or landmark trajectory).

As shown in the Fig. 3, we present two graphs: the graph of GPS points in the left part of the Fig. 3 and the graph of landmarks in the right part of the Fig. 3. These two graphs contain all trajectories and all landmark trajectories of each user respectively. For the graph of GPS points in the left part of the Fig. 3, the graph nodes ($p1, p2, p3, p4$ and $p5$) stand for GPS points, and the graph edges denote users' trajectories among GPS points. Take a 2-length trajectory ($p_1 \rightarrow p_2$) as an example, the number shown on the nodes (6 and 4) and the edge (3) represents the score of the GPS point and the trajectory respectively. For the graph of landmarks, we have the similar analysis.

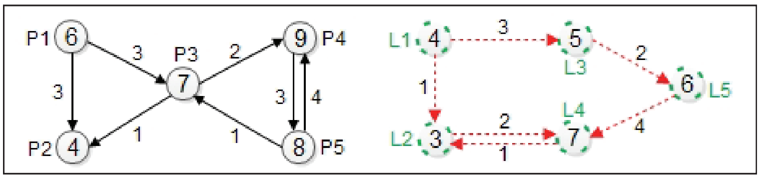


Fig. 3 *Trajectory graph and landmark trajectory graph.*

For similarity normalization (in Section 5.1.2), we construct user-GPS point matrix UP and user-landmark matrix UL. The row of UP (or UL) represents the GPS-point (or landmark). The column of UP (or UL) represents the user. Each entry in the UP (or UL) is rating how many times a user has visited the GPS-point (or landmark).

5.1.2 Overview Semantic Similarity

In Section 4.4, we introduce an overview which consistes of topics and sub-topics. For calculating the overview semantic similarity, we construct user-topic matrix UT and user-sub-topic matrix UST. Each entry in the UT or UST denotes whether a user add tags to pictures with this topic or sub-topics. Here, 1 denotes this user uses it, while 0 expresses this user does not use this topic or sub-topic.

Based on these two matrixes, we can learn users' profiles utilizing user similarities, which are reflected via considering whether users add the same tags (topics or sub-topics). In other words, if two users are more similar, they are likely to add more similar tags. Here, we can obtain two user similarities: user-topic similarity ($sim\_T(u_i, u_j)$) and user-sub-topic similarity ($sim\_ST(u_i, u_j)$). Note that, we use cosine similarity to compute $sim\_T(u_i, u_j)$ and $sim\_ST(u_i, u_j)$. The overview semantic similarity is formalized as:

$$sim\_view\,(u_i, u_j) = \beta_1 sim\_T(u_i, u_j) + \beta_2 sim\_ST(u_i, u_j) \tag{11}$$

where $\beta_1, \beta_2 > 0$ and $\beta_1 + \beta_2 = 1$,$u_i \in U, u_j \in U$and $U = \{u_1, u_2, \ldots, u_n\}$ is a set of users.

5.1.3 Trajectory Similarity

In Section 5.1.1, the trajectory graph and the matrix *UP* have been constructed. We adopt the similar sequence matching method proposed by Li Q. [6] in order to find the similar trajectories for each user-pair. The retrieved similar trajectories are used to calculate an overall similarity score for each user-pair. When computing the score, we take into account two factors: the length of a similar trajectory with weight (that is, M-length trajectory) and the mass of each node in this trajectory. So the score an M-length trajectory obtains can be formulated as:

$$score\_Mlt = M * \sum_{j=1}^{M-1} Tr.score * sim\_mass(u_i, u_j) \tag{12}$$

where $M(M > 1)$ is defined in Definition 5, $Tr.score$ is the summation of each score in the similar trajectory, u_i and u_j are defined in Section 5.1.2 and $sim\_mass(u_i, u_j)$ is regarded as cosine similarity based on matrix *UP*.

For similarity normalization, the equation (12) can be formulated as:

$$score\_Mlt = M * \sum_{j=1}^{M-1} Tr.score * sim\_mass(u_i, u_j) / \left(\sum_{u_i \in U} Tr.score * \sum_{u_i \in U} p.m \right) \tag{13}$$

where $\sum_{u_i \in U} Tr.score$ is the summation of scores given by all users for this similar trajectory, $\sum_{u_i \in U} p.m$ is the summation of times that all users have visited this GPS point.

As shown in equation (14), the trajectory similarity of two users is measured based on all the similar M-length trajectories. Here, n is the number of similar trajectories in two users.$score\_Mlt_i$ is the score of the i-th similar trajectory, which can be calculated according to equation (14). N_1 and N_2 denotes the number of the GPS points of the two users respectively.

$$sim\_Tra = \frac{1}{N_1 N_2} \sum_{i=1}^{n} score\_Mlt_i \tag{14}$$

5.1.4 Landmark Trajectory Similarity

Like trajectory similarity in Section 5.1.3, the landmark trajectory similarity of two users can be formulated as:

$$sim\_LTra = \frac{1}{N_1 N_2} \sum_{i=1}^{n} score\_Mllt_i \tag{15}$$

Note that, the difference between equation (14) and equation (15) is that we need replace trajectory and GPS point with landmark trajectory and landmark respectively.

5.2 Landmark Recommendation

Incorporating the three similarity measures, we propose an improved user similarity method (fusion similarity), which is demonstrated in equation (16).

$$sim\_Fusion\left(u_i, u_j\right) = \gamma_1 sim\_view\left(u_i, u_j\right) + \gamma_2 sim\_Tra + \gamma_3 sim\_LTra \quad (16)$$

where $\gamma_1, \gamma_2, \gamma_3 > 0$ and $\gamma_1 + \gamma_2 + \gamma_3 = 1$. Although more sophisticated functions can be designed, the weighted summation of these three similarity measures is simple and intuitive.

Based on the fusion similarity, we propose a landmark-based *CF* algorithm. In our presented algorithm, to derive the landmark recommendation for a target user u, usually only k most-similar neighbors are selected (U_k). When predicting a score of a given user u for a landmark L, the weighted sum of the other users is computed by:

$$\overline{\overline{score\left(u, L\right)}} = \overline{score_u} + \frac{\sum_{v \in U_k} sim\_Fusion\left(u_i, u_j\right)\left(score\left(v, L\right) - \overline{score_v}\right)}{\sum_{v \in U_k} sim\_Fusion\left(u_i, u_j\right)} \quad (17)$$

where $\overline{score_u}$ and $\overline{score_v}$ represents the average times a user u and $v$have visited L respectively and $score\left(v, L\right)$ represents the times a user v has visited L.

6. Experiments

6.1 Data Set

The photos for our experiments were collected from the datasets of geo-tagged photos available on Flickr using the site's public API[7]. These photos we crawled meet such requirements: they were taken in Beijing and the upload time is between 4th, January, 2005 and 10th, February, 2012. These photos collection contains 533,594 unique photos associated with 2,760,614 textual tags and taken by 16,196 unique users.

Because Flickr allows users to geotag their photographs using a map interface, photos could be geotagged incorrectly, or inappropriately, or assigned a misleading accuracy level. And photos are also likely to be assigned text tags that are irrelevant to the location. Naturally, there is no guarantee that photos taken in Tiananmen, for example, will all have the Tiananmen tag, or any other tag relevant to the location. To make the dataset reliably extracted and used, we adopt the visualization method [31] to deal with issues of errors and noise in the Flickr data.

[7]http://www.flickr.com/services/api/flickr.photos.search.htm.

6.2 Evaluation Metrics

Our recommendation algorithm computes a ranking score for each candidate landmark (i.e., those a user has not visited) and returns the top-K highest ranked landmarks as recommendation to a target user. To evaluate the prediction accuracy, we focus on how many locations previously removed in the preprocessing step re-appear in the recommended results. Therefore, we apply four popular performance metrics, namely Mean Average Precision (*MAP*), *Precision@K*, *Recall@K* and *nDCG* (normalized Discounted Cumulative Gain) [24], to capture the performance of our proposed algorithm.

Precision@K is the ratio of recovered locations to the K recommended locations. *Recall@K* is the ratio of recovered locations to the set of locations deleted in preprocessing. We divide the locations into two sets: the test set T_u and the top-K set R_u. Locations that appear in both sets are members of the hit set. *Precision* and *Recall* is defined as follows:

$$Precision = \frac{\textit{size of hit set}}{\textit{size of top\_Nset}} = \frac{|T_u \cap R_u|}{K} \tag{18}$$

$$Recall = \frac{\textit{size of hit set}}{\textit{size of test set}} = \frac{|T_u \cap R_u|}{|T_u|} \tag{19}$$

MAP is the most frequently used for the summary measure of a ranked retrieval run. In this work, *MAP* stands for the mean of the precision score after each landmark is recommended.

nDCG, which is commonly used in information retrieval to measure the search engine's performance, computes the relative-to-the-ideal performance. The higher *nDCG* value is, the better a ranking results list is. In particular, *nDCG*$[K]$, or referred as *nDCG@K*, measures the relevance of top K results:

$$nDCG@K = \frac{DCG@K}{IDCG@K} \tag{20}$$

$$DCG@K = rel_1 + \sum_{i=2}^{k} \frac{rel_i}{\log_2 i} \tag{21}$$

where *nDCG@K* measures the relevance of top K results, *IDCG@K* is the *DCG@K* value of ideal ranking list. rel_i is a relevance value.

Note that, when computing *MAP* and *Precision@K*, only the landmarks with score above 1.5 are considered as significant.

6.3 Settings and Parameter Selection

6.3.1 Settings

We discovered landmarks in the dataset, using the method described above in Section 4.1. We set the bandwidth as 0.02 for Meanshift and the initial points are selected randomly. For *DFCM*, we set σ as 0.036. As a result, Meanshift returns 655 landmark clusters, and *DFCM* returns 670 landmark clusters. For each location cluster, representative tags are determined by scoring frequent tags within the cluster. For the tags chosen by the system, we retain the information about the tag and the clusters where the tag scored well.

After the landmark clustering step, we generate the trajectories and landmark trajectories according to the visiting order of the locations. We extract the frequent sequential patterns by leveraging the PrefixSpan algorithm [28] and treat them as trajectory or landmark trajectories patterns. Given a set of sequences, sequential pattern mining algorithm will find all the sequential patterns whose frequencies are no smaller than the minimum support. The frequency of a pattern is defined as the number of sequences that are derived by subsuming the pattern. In this work, we set the minimum support threshold as 2 to collect as many trajectory or landmark trajectory patterns, namely a 2-length trajectory or a 2-length landmark trajectory. As a result, we construct the trajectory dataset that consists of 23,448 raw trajectories and the landmark trajectory dataset that consists of 6,694 raw landmark trajectories.

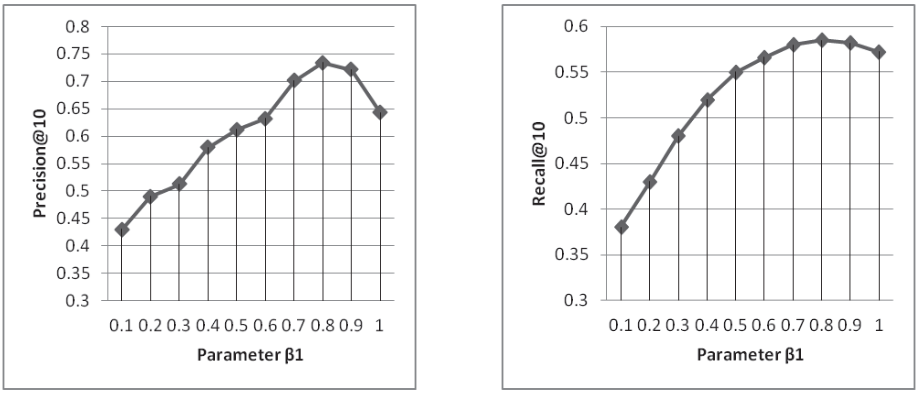
The dataset we have used in our experiments are split into training and test sets. In this data set, we have used the 80-20% rating splits in the data set distribution and have performed 10-fold cross validation. Additionally, we have further extracted validation data from the training data to optimize the parameters $\beta_1, \beta_2, \gamma_1, \gamma_2, \gamma_3$ and k (the neighborhood size). We have varied the neighborhood size from 10-60 by an interval of 10 and the other five parameters from 0 to 1 by an interval of 0.1. Using the validation data, we have found the best β_1 to be 0.8, β_2 to be 0.2, γ_1 to be 0.2, γ_2 to be 0.3, γ_3 to be 0.5 and k to be 20.

6.3.2 Parameter Selection

In this section, we will introduce the process of parameter selection in detail. We use the validation data to investigate the impact of the parameters of our proposed method, discuss their role in recommendation, and determine the parameter settings that we use for the experiments.

First, we investigate the impact of parameters β_1, β_2 in our proposed method (refer to Equation 11). Here, β_1 and β_2 control the overview semantic similarity and play a role of balancing the influence between user-topic similarity and user-sub-topic similarity. In order to get optimal values of β_1 and β_2, we fix other four parameters (let γ_1 be 0.3, γ_2 be 0.2, γ_3 be 0.5 and k be 10) and vary β_1 from 0 to 1 by an interval of 0.1. Fig. 4 plots the *Precision* and *Recall* against β_1 with β_1 in the range $[0, 1]$. From Fig. 4(a), we can observe that the optimal recommendation performance (*Precision*) is achieved when β_1=0.8, (β_2=0.2). Meanwhile, from Fig. 4(b), we can get the similar result. It indicates that the user-topic similarity plays a more important role than the user-sub-topic similarity.

Our next experiment in this section investigates the impact of the trade-off parameter $\gamma_1, \gamma_2, \gamma_3$ in the proposed method (refer to Equation 16). γ_1, γ_2 and γ_3 control the influence of fusion similarity, including overview semantic similarity, trajectory similarity and landmark trajectory similarity. We fix β_1 and β_2 to the optimal values determined during the first experiment (β_1=0.8,β_2=0.2) and set k to be 10. At the same time, we vary the value of $\gamma_1, \gamma_2, \gamma_3$ from 0 to 1 by an interval of 0.1 in order to observe its influence on the ability of the proposed method. As can be seen in Tab. II and Tab. III, the optimal recommendation performance



(a) Precision (b) Recall

Fig. 4 *Impact of β_1 on recommendation performance.*

γ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.1	0.551	0.562	0.582	0.665	0.549	0.535	0.521	0.512	0.508
0.2	0.557	0.636	0.734	0.524	0.521	0.518	0.508	0.504	-
0.3	0.564	**0.753**	0.519	0.512	0.509	0.497	0.501	-	-
0.4	0.613	0.515	0.509	0.488	0.479	0.456	-	-	-
0.5	0.505	0.501	0.483	0.457	0.435	-	-	-	-
0.6	0.498	0.475	0.453	0.421	-	-	-		-
0.7	0.462	0.431	0.403	-	-	-	-		-
0.8	0.425	0.402	-	-	-	-	-		-
0.9	0.308	-	-	-	-	-	-		-

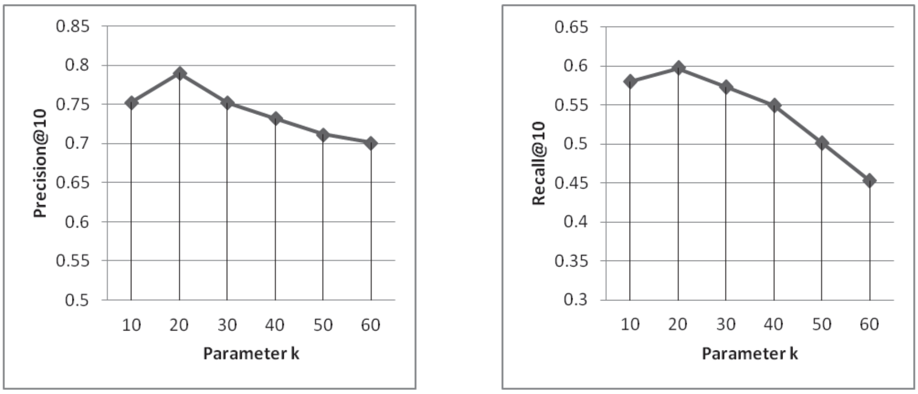
Tab. II *Impact of $\gamma_1, \gamma_2 (\gamma_1{+}\gamma_2{=}1)$ on recommendation performance (Precision@10). Here, the first row represents γ_1, the first column represents γ_2 and "–" represents this pattern does not exist. Each item in the table represents the value of Precision.*

(*Precision* and *Recall*) can be achieved when γ_1=0.2 (γ_2 = 0.3, γ_3=0.5). At the same time, from Tab. II and Tab. III, we can observe that the recommendation performance ($\gamma_1 < \gamma_2$ and γ_3=0.5) is better than the recommendation performance ($\gamma_1 > \gamma_2$ and γ_3=0.5), which indicates that the trajectory similarity is more important than the overview semantic similarity. Also, we can see that the recommendation performance ($\gamma_1 + \gamma_2 < 0.5$) is better than the recommendation performance ($\gamma_1 + \gamma_2 > 0.5$). It indicates the landmark trajectory similarity plays the most important role among the three similarities.

As a final point in this section, we further examine the impact of parameter k in the proposed method (refer to Equation 17). k controls the neighborhood size affecting the performance of our proposed method. We also fix other five paprameters ((let β_1 to be 0.8, β_2 to be 0.2, γ_1 be 0.2, γ_2 be 0.3, γ_3 be 0.5,). In Fig. 5, x axis represents the neighborhood size, from 10 to 60, and *y axis* refers to

γ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.1	0.446	0.457	0.498	0.512	0.435	0.428	0.421	0.407	0.396
0.2	0.438	0.483	0.583	0.431	0.424	0.419	0.401	0.391	-
0.3	0.428	**0.597**	0.423	0.420	0.417	0.398	0.387	-	-
0.4	0.476	0.419	0.417	0.412	0.389	0.385	-	-	-
0.5	0.415	0.411	0.395	0.387	0.376	-	-	-	-
0.6	0.402	0.388	0.382	0.371	-	-	-	-	-
0.7	0.378	0.375	0.364	-	-	-	-	-	-
0.8	0.367	0.356	-	-	-	-	-	-	-
0.9	0.351	-	-	-	-	-	-	-	-

Tab. III *Impact of $\gamma_1, \gamma_2 (\gamma_1{+}\gamma_2{=}1)$ on recommendation performance (Recall@10). Here, the first row represents γ_1, the first column represents γ_2 and "–" represents this pattern does not exist. Each item in the table represents the value of Recall.*



(a) Precision (b) Recall

Fig. 5 *Impact of k on recommendation performance..*

Precision (Recall). From Fig. 5(a), we can see that the optimal recommendation performance (*Precision*) is achieved when $k = 20$. Also, we can obtain the similar result from Fig. 5(b).

6.4 Experimental Results

In this work, we employ five baselines for comparison: MeanShift (*MS*) + sim_Fusion (*sF* $(\gamma_1 = 0.2, \gamma_2 = 0.3, \gamma_3 = 0.5)$), *DFCM* + *sF* $(\gamma_1 = 1)$ (or *DFCM+sV*), *DFCM+ sF*$(\gamma_2 = 1)$(or *DFCM+sT*), *DFCM+ sF*$(\gamma_3 = 1)$ (or *DFCM+sL*) and *CRMF*. Here, *sF* represents the fusion similarity, *sim_Fusion*. *sV*, *sT* and *sL* represents the overview semantic similarity (*sim_View*), the trajectory similarity (*sim_Tra*) and the landmark trajectory similarity (*sim_LTra*), respectively. In order to simplify the expression, *sF* $(\gamma_1 = 0.2, \gamma_2 = 0.3, \gamma_3 = 0.5)$ is expressed as *sF*.

Fig. 6 shows the comparision of *MAP* between our approach (*DFCM+sF*) and baselines. Our proposed method (*DFCM+sF*) is slightly superior to (*MS+sF*) and significantly superior to other methods. Using *nDCG*, Fig. 7 further differentiates our approach from baselines. Obviously, (*DFCM+sF*) leads the performance in both *nDCG@10* and *nDCG@20* among these methods. Moreover, (*DFCM+sF*) better improves the performance compared with (*MS+sF*) and *CRMF*. The possible reasons for these are: 1) *CRMF* only captures the similarity of category-based and user-landmark preference (i.e., trajectory similarity); 2) (*DFCM+sF*) not only uses the overview semantic similarity, but also considers the trajectory similarity and the landmark trajectory similarity; 3) (*DFCM+sV*), (*DFCM+sT*) and (*DFCM+sL*) only consider one of the three similarities.

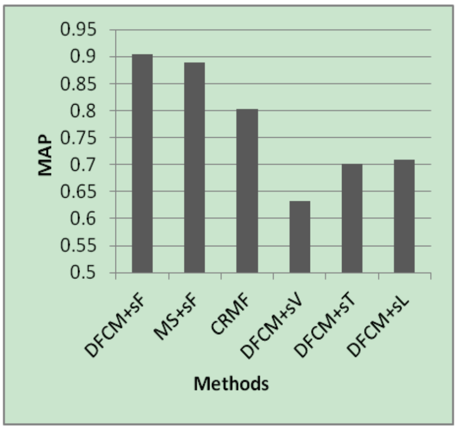


Fig. 6 *Comparision of MAP among different methods.*

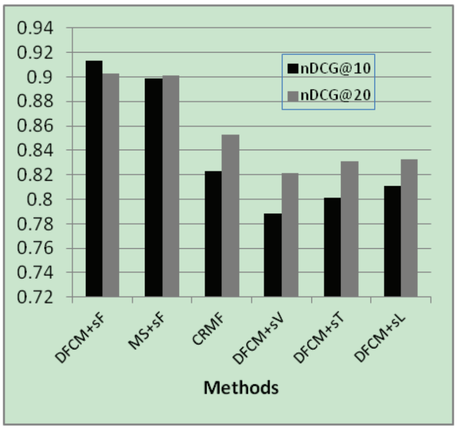


Fig. 7 *Comparision of nDCG among different methods.*

In Fig. 8, the horizontal axis plots the number of recommended landmarks (K), and the vertical axis plots the value of precision. From Figure 8, we can observe all methods show the same type of sensitivity, that is, with the number of recommended landmarks increasing, precision of the six algorithms is trending downward. However, our proposed method (*DFCM+sF*) slightly outperforms (*MS+sF*) and significantly outperforms other methods. The explanation for this is similar to ones for Fig. 6 and Fig. 7. Note that, when the number of the recommended landmarks reaches 10, the optimal value of precision is obtained.

In Fig. 9, the horizontal axis plots the number of recommended landmarks (K), and the vertical axis plots the value of recall. From Fig. 9, we can notice that, with the number of recommended landmarks increasing, recall of the six algorithms is trending upward. However, our proposed method (*DFCM+sF*) slightly outperforms (*MS+sF*) and significantly exceeds other methods. The explanation for this is similar to ones for Fig. 6 and Fig. 7. Note that, when the number of the recommended landmarks reaches 25, the optimal value of recall is obtained.

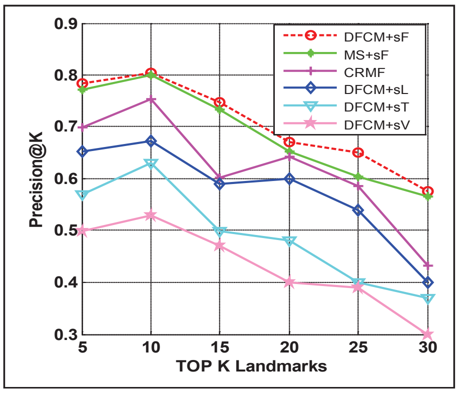


Fig. 8 *Precision changing over number of recommended landmarks.*

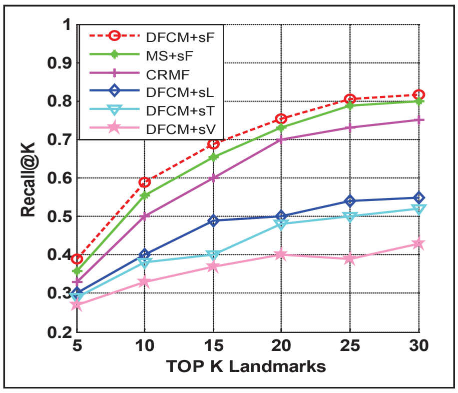


Fig. 9 *Recall changing over the number of recommended landmarks.*

6.5 Discussion

So far, we have got a Top-K list for each user, but we still have an important issue, how to produce nontrivial landmark recommendations, to be addressed. For example, recommending Great Wall in Beijing is probably right for any tourist, but such a recommendation is not meaningful, since the tourists know Great Wall even without recommendations. There are some important previous works done to solve the problem of nontrivial landmark recommendation. In [33], Shi et al. proposed a novel approach called WMF-CR, incorporating weighted matrix factorization and category-based regularization. This approach exploits geo-tagged images from an on-line community for the purpose of personalized landmark recommendation. This research motivates us to work on how to produce nontrivial landmark recommendations. We adopt a recommendation re-ranking approach to solve the problem of nontrivial landmark recommendations.

6.5.1 Recommendation Re-ranking

In this section, we will introduce our recommendation re-ranking approaches. We have two assumptions: the more users that used a tag in a landmark there are(or the more users that visited a landmark there are), the more popular the landmark is for most users; further, a trivial coefficient takes the number of users visiting a landmark, divided by the total number of users in the dataset. The bigger this trivial coefficient about a landmark is, the more trivial this landmark is. Based on these two assumptions, we use TF-IDF like strategy for re-ranking the candidate items. Here, the landmark (L) frequency in the dataset can be denoted as:

$$tf(Li)=Li.lm \Big/ \sum\nolimits_{Li \in L} Li.lm \tag{22}$$

and IDF part can be denoted as:

$$idf(ui, Li) = log(|U| / \sum\nolimits_{ui \in U} I(ui, Li)) \tag{23}$$

where $Li.lm$ represents the times a user has visited this landmark (refer to Definition 2), L is a set of landmarks, $|U|$ represents the total number of users in the dataset, $\sum_{Li \in L} Li.lm$ is a normalized factor and function $I(ui, Li)$ is defined as:

$$I(ui, Li) = \begin{cases} 1, & if \quad ui \quad visited \quad Li \\ 0, & else \end{cases} \tag{24}$$

Finally, we get our recommendation re-ranking TF-IDF like strategy to rank our recommendation list. The TF-IDF ranking strategy can be defined as:

$$Rerank\_Score(ui, Li) = tf(Li)Idf(ui, Li) \tag{25}$$

Here, the *Rerank_Score* value increases proportionally to the number of times a landmark is visited by users, but is offset by the frequency of the landmark in the dataset, which helps to control for the fact that some landmarks are generally more common than others.

6.5.2 Experimental Results

After recommendation re-ranking, we get some results of nontrivial landmark recommendation. In this section, we will show the comparisons between our *DFCM+sF* and *DFCM+sF+Rerank_Score* considering nontrivial landmark recommendation.

In Fig. 10 and Fig. 11, it can be seen that *DFCM+sF+Rerank_Score* works at the aspect of nontrivial landmark recommendations. From Fig. 10 and Fig. 11, we can notice that, *DFCM + sF + Rerank_Score* averagely improves *Precision@K* and *Recall@K* by 1.24% and 1.01% on the Flickr dataset, respectively. The possible reason for these is that *DFCM+sF+Rerank_Score* assigns a punished weight to the common landmarks and puts the nontrivial landmarks before the common landmarks.

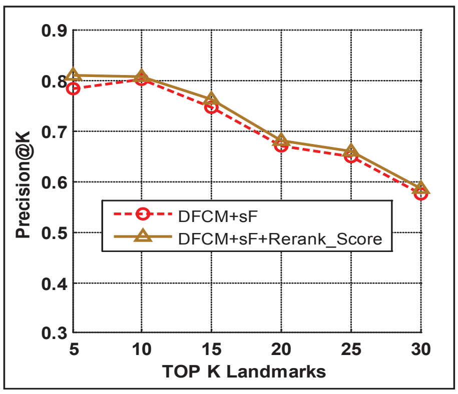


Fig. 10 *Precision changing over the number of recommended landmarks.*

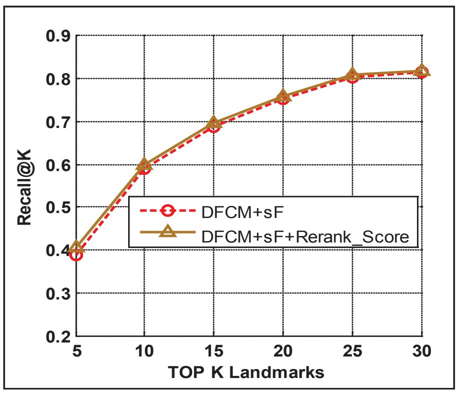


Fig. 11 *Recall changing over the number of recommended landmarks.*

We further show an example of nontrivial landmark recommendations for five users in Tab. IV. From Tab. IV, we can see that we can get different results based on different methods for different users. For user 1, *Ds* method mainly gets some common recommendation results except *Tsinghua* and *Nationallibrary*, but *DsR* method gets more nontrivial recommendation results. We can speculate that User 1 may be interested in study or is likely to be a student from the results based on *DsR*. For User 2, two methods both recommend "798 art dist", in addition, *DsR* recommends other two landmarks, namely, "Art museum", "CAFA". We can also speculate User 2 may be interested in art or is likely to be an artist. Compared with *Ds*, *DsR* can mine more landmarks that users are interested in, by filtering some common landmarks. Meanwhile, we can have similar analysis for other three users.

Users	*Methods*	*Landmarks*				
User 1	Ds	Great Wall	Tiananmen	Forbidden City	Tsing hua	National-library
	DsR	Tsinghua	National-library	Guozizjian	Zhongguancui	Junbo
User 2	Ds	Summer-Palace	798 art dist	Beihai	Sanlitun	Tuanjiehu
	DsR	Art museum	798 art dist	Sanlitun	CAFA	Houhai
User 3	Ds	Tiananmen	Gugong	Oldsummer-palace	Shimatai	Xiangshan
	DsR	Shidu	Qinglongxia	Tanzhe-Temple	Phoenix-ridge	Shimatai
User 4	Ds	Xidan	Sanvillage	Xinjiekou	Wudoukou	Nanluogu-xiang
	DsR	Huguoshi	Jiumen	Guijie	Nanluogu-xiang	Wangfujin
User 5	Ds	Tuanjiehu	Tiananmen	BLCU	BFSU	Houhai
	DsR	San village	Wudaokou	Wangjing	Houhai	Panjiayuan

Tab. IV *Results of landmark recommendation based on two proposed methods. Here, Ds represents DFCM + sF, DsR represents DFCM + sF+Rerank_Score.*

7. Conclusions and Future Work

In this paper, facing the proposed three challenges, we put forward the corresponding solutions. First, we present a data field clustering method, whose performance is slightly superior to Mean Shift method. And then, we provide more friendly and comprehensive overviews for each landmark. Subsequently, we present an improved user similarity method, which not only utilizes the overview semantic similarity, but also considers the trajectory similarity and the landmark trajectory similarity. Finally, we propose a personalized landmark recommendation algorithm based on the improved user similarity method and adopt a TF-IDF like strategy in order to produce the nontrivial landmark recommendation. Experimental results show that the proposed method can provide reasonable and high quality personalized landmark recommendations.

In the future, we intend to extend our work in the following three directions. First, we will attempt to model the users' dynamic behaviors using more useful features, such as the landmark popularity etc. Second, we will try to propose a new method which is used to generate landmark overviews. Third, we will spread our work to other domains, such as music, book etc.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments. This work is supported by the National Natural Science Foundation of China under Grant No. 61202238, No. 61300006 and Fundamental Research Funds for the Central Universities under Grant No. YWF-13-T-RSC-077.

References

[1] Deng D. P., Chuang T. R., Lemmens R.: Conceptualization of Place via Spatial Clustering and Co-occurrence Analysis. Proceedings of the International Workshop on Location Based Social Networks, 2009, pp. 49-55.

[2] Fukunaga K., Hostetler L.: The estimation of the gradient of a density function, with applications in pattern recognition. IEEE Trans. Information Theory, 21, 1, pp. 32-40, 1975.

[3] Cao L., Luo J., Gallagher A., Jin X., Han J., Huang T. S.: A worldwide tourism recommendation system based on geotagged web photos. In: ICASSP, 2010, pp. 2274-2277.

[4] Crandall D. J., Backstrom L., Huttenlocher D., Kleinberg J.: Mapping the world's photos. In: Proceedings of the 18th international conference on World wide web Conference. WWW'09. New York, NY, USA: ACM, 2009, pp. 761-770.

[5] Hao Q., Cai R., et al.: Generating Location Overviews with Images and Tags by Mining User-Generated Travelogues. In: ACM MM, 2009.

[6] Li Q., Zheng Y., Xie X., Chen Y., Liu W., Ma W.: Mining user similarity based on location history. In: Proc. of the 16th Intl. Conf. on Advances in geographic info. system, ACM Press, Santa Ana, USA, 2008, pp. 298-307.

[7] Zheng Y., Zhang L., Xie X.: Recommending friends and locations based on individual location history. ACM Trans. on the Web, 2011.

[8] Zheng Y., Xie X., Ma W.: GeoLife: A collaborative social networking service among user, location and trajectory. IEEE Data Engineering Bulletin, 2010, pp. 32-40.

[9] Horozov T., Narasimhan N., Vasudevan V.: Using Location for Personalized POI Recommendations in Mobile Environments. In Proc. of the International Symposium on Applications on Internet, 2006, pp. 124-129.

[10] Quercia D., Lathia N., Calabrese F., Di Lorenzo G., Crowcroft J.: Recommending Social Events from Mobile Phone Location Data. In: IEEE ICDM, Sydney, Australia, 2010.

[11] Chow C.-Y., Bao J., Mokbel M. F.: Towards location-based social networking services. In Proceedings of the ACM SIGSPATIAL International Workshop on Location Based Social Networks, 2010.

[12] Zheng V. W., Cao B., Zheng Y., Xie X., Yang Q.: Collaborative filtering meets mobile recommendation: A user-centered approach. In: AAAI, 2010.

[13] Gao Y., Tang J., Hong R., Dai Q., Chua T. S., Jain R.: W2Go: A Travel Guidance System by Automatic Landmark Ranking. In: ACM Multimedia, 2010.

[14] Lu X., Wang C., Yang J.-M., Pang Y., Zhang L.: Photo2trip: Generating travel routes from geo-tagged photos for trip planning. In: ACM Multimedia, 2010.

[15] Choudhury M. D., Feldman M., Amer-Yahia S., Golbandi N., Lempel R., Yu C.: Automatic construction of travel itineraries using social breadcrumbs. In: HT, 2010, pp. 35-44.

[16] Jia-Ching Ying J., Hsueh-Chan Lu E., Wang-Chien Lee, Tz-Chiao Weng, S. Tseng V.: Mining user similarity from semantic trajectories. In: LBSN, 2010, pp. 19-26.

[17] Zheng Y., Zhang L., Xie X. and Ma W.: Mining interesting locations and travel sequences from GPS trajectories. In: Proc. of the 18th Intl. Conf. on World Wide Web, ACM Press: 791-800, Madrid Spain, 2009.

[18] Zheng V. W., Zheng Y., Xie X., Yang Q.: Collaborative location and activity recommendations with gps history data. In: WWW, 2010, pp. 1029-1038.

[19] Ye M., Yin P., Lee W.-C.: Location Recommendation in Location-based Social Networks. In: GIS, 2010, pp. 458-46.

[20] Cao X., Cong G., Jensen C.: Mining Significant Semantic Locations From GPS Data. In: VLDB, 2010, pp. 1009-1020.

[21] Shi Y., Serdyukov P., Hanjalic A., Larson M.: Personalized landmark recommendation based on geotags from photo sharing sites. ICWSM '1, AAAI, 2011, pp. 622-625.

[22] Levandoski J. J., Sarwat M., Eldawy A., Mokbel M. F.: LARS: a location-aware recommender system. Int. Conf. on Data Engineering, Washington D.C., USA, 2012.

[23] Leung K. W.-T., Lee D. L., Lee W.-C.: CLR: A Collaborative Location Recommendation Framework based on Co-Clustering. In: Proceedings of SIGIR, 2011, pp. 305-314.

[24] Manning C. D., Raghavan P., Schtze H.: Introduction to Information Retrieval. Cambridge University Press, 2008.

[25] Papadimitriou A., Symeonidis P., Manolopoulos Y.: Geo-social recommendations [C], ACM Recommender Systems 2011 (RecSys) Workshop on Personalization in Mobile Applications, 2011.

[26] Clements M., Serdyukov P., de Vries A. P., et al.: Using flickr geotags to predict user travel behaviour [C], Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval. ACM, 2010, pp. 851-852.

[27] Arase Y., Xie X., Hara T., Nishio S.: Mining people's trips from large scale geo-tagged photos. In MM'10, 2010, pp. 133-142.

[28] Heymann P., Garcia-Molina H..: Collaborative Creation of Communal Hierarchical Taxonomies in Social Tagging Systems. Technical report, Technical Report 2006-10, Computer Science Department, 2006.

[29] Eagle N., Pentland A.: Reality mining: sensing complex social systems [J]. Personal and ubiquitous computing, 10, 4, 2006, pp. 255-268.

[30] Takeuchi Y., Sugimoto M.: An outdoor recommendation system based on user location history[C]//Proceedings of the 1st International Workshop on Personalized Context Modeling and Management for UbiComp Applications, 2005, pp. 91-100.

[31] Ahern S., Naaman M., Nair R., Hui J.: World explorer: visualizing aggregate data from unstructured text in geo-referenced collections. In: Proceedings of the 7th ACM/ IEEE-CS joint conference on Digital libraries, ser. JCDL'07. New York, NY, USA: ACM, 2007, pp. 1-10.

[32] Gan Wenyan, Li Deyi, Wang Jianmin: An Hierarchical Clustering Method Based on Data Field. ACTA ELECTRON-ICA SINICA, Vol. 34, No. 2, 2006.

[33] Shi Y., Serdyukov P., Hanjalic A., et al.: Nontrivial landmark recommendation using geotagged photos [J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2013, 4, 3, 47.

[34] Chen J., Wu Z., Gao H., et al.: Recommending Interesting Landmarks Based on Geo-tags from Photo Sharing Sites [M], Web Information Systems EngineeringWISE 2013. Springer Berlin Heidelberg, 2013, pp. 151-159.