Trip Similarity Computation for Context-Aware Travel Recommendation Exploiting Geotagged Photos

Zhenxing Xu
supervised by: Ling Chen *, Abdul Majid, Mingqi Lv, Gencai Chen
College of Computer Science, Zhejiang University, Hangzhou, P.R. China
* lingchen@zju.edu.cn

***Abstract*—The popularity of GPS-enabled digital cameras, smart phones, and photo sharing web sites, e.g. Flickr and Panoramio, has led to huge volumes of community-contributed geotagged photos (CCGPs) available on the Internet, which could be regarded as digital footprints of photo takers. In this paper, we propose a method to make context-aware and trip similarity based travel recommendations by mining CCGPs. We obtain user-specific travel preferences from the travel history of user in one city, and use these to recommend tourist locations in another city. The season and weather context are considered during the mining and the recommendation processes. The similarity of users is computed by the modified longest common subsequence and a user-location graph is built from their travel histories in one city, which is then exploited to make travel recommendations. Our method is evaluated on a Flickr dataset, which contains photos taken in four cities of China. Experimental results show the effectiveness of the proposed method.**

I. Introduction

With the popularity of smart phones and GPS-enabled digital cameras, people can casually take photos with geotags, and share these photos on websites, e.g. Flickr and Panoramio. The number of geotagged photos in Flickr has exceeded 200 million by 2012 and is still growing at a rapid pace. These community-contributed geotagged photos (CCGPs) on the Web are publically available and cover most of the lands in the world. The information of CCGPs can be utilized by many applications, e.g. searching, advertising, annotation, and recommendation.

Due to the intrinsic relationship with everyday lives of people, CCGPs based travel recommendation has recently drawn the attention of the research community. These researches can be roughly divided into two kinds: context-free and context-aware recommendation methods. Context-free methods usually extract travel locations and sequences from CCGPs first, and then cluster, index, and recommend these travel locations. Popescu *et al*. focused on the query with temporal constraints in terms of duration of the trip [1]. Lu *et al*. proposed an interactive tourist recommendation approach which took into account a number of factors, e.g. duration of the trip and traveling cost, to help the tourist for trip planning [2],[3]. Arase *et al*. focused on the detection of frequent trip patterns of people [4]. However, these works only consider the duration of the trip, traveling cost, and trip theme, not consider the context information of users when making recommendations for users.

Context-aware travel recommendations take the context information of users into account, so these methods can provide more accurate recommendation results. Kurashima *et al.* incorporated present location information and preference of user into the probabilistic behavior model to make recommendations[5],[6]. Cheng *et al.* proposed a probabilistic travel recommendation model which used people attributions for mobile travel recommendation and route planning [7],[8]. Majid *et al.* considered weather context for travel recommendations of user [9],[10]. These works described above show promising recommendation results based on different contexts. However, when making recommendations, these works either do not take the similarity of users into consideration, or just employ a simple similarity computation.

To counter the problems mentioned above, in this paper, we consider not only context information (e.g. season and weather) that influence the locations where users wish to visit, but also the similarities among users based on their traveling trips, using the modified longest common subsequence algorithm, which considers the order of travel locations of a user. This method can find the top *n* most similarity users and mine the preference of users.

The main contributions of this paper are:

(1) Propose a travel recommendation method based on the similarity of travel trips of users and the context history in which a location has been visited. Considering the similarity of trips of users can be leveraged to find the top *n* most similarity users and mining the preferences of users.

(2) Use the popularity of locations in different contexts as profile matching criteria to filter the travel locations, and rank the locations using the collaborative filtering method for travel recommendations.

(3) Conduct the experiments on CCGPs data set covering four major cities in China, and show that our method has the potential to improve the performance of personalized travel location recommendation.

The remainder of this paper is structured as follows. In the next section, we give preliminaries and problem definition. Section 3-6 introduces the framework and details of our method. Section 7 reports the experimental results. Section 8 concludes and discusses future research directions.

II. Problem Definition

Before we formally define the problem, we give definitions of some basic concepts and terms.

Definition 1. (*Geotagged photo*) A geotagged photo p can be defined as $p = (id, t, g, X, u)$ containing a photo's unique identification, id; its geotags, g; its time-stamp, t; and the identification of the user who contributed the photo, u. Each photo p can be annotated with a set of textual tags, X. Geotags g of photo p is the coordinates of the geographical region where photo p was taken.

Definition 2. (*User trip*) A user trip is a sequence of locations visited by a user according to temporal order and the difference between the visiting time of two locations in the sequence is not greater than a $trip_{dur}$ threshold. It can be denoted as $T = \{l_1, l_2, \ldots, l_n\}$, where $l_i.t > l_{i-1}.t$ and $l_i.t - l_{i-1}.t < trip_{dur}$.

Definition 3. (*Context-aware query*).A context-aware query Q is defined as $Q = (u_a, s, w, d)$, where u_a is a target user, s is the season information, w is the weather information, and d is the target city for u_a. The output of query is a personalized list of locations for target city determined by u_a.

Our research problem can be formulated as, given a collection of CCGPs $P = \{P_1, P_2, P_3, ..., P_n\}$, where P_u (u = 1, ..., n) is the set of photos contributed by user u, how to locate and summarize tourist locations and tourist trips, and then build the travel histories of users to obtain the travel preferences to answer context-aware personalized queries. By exploiting travel histories of users, we try to find the top n most similarity users and the preference of a user, then to recommend best fit tourist locations for the user. Figure.1 depicts the modules comprising our tourist locations recommendation framework. Locations are discovered by utilizing spatial proximity of photos. The profiles of locations are built to describe the contexts in which they are visited. A measure is defined to identify similar users in previously visited cities and to aggregate the opinions of these users. For making recommendations, we first filter the locations based on contextual constraints, and then rank locations by combining the opinions of similar users and the significance score of a location in the target city.

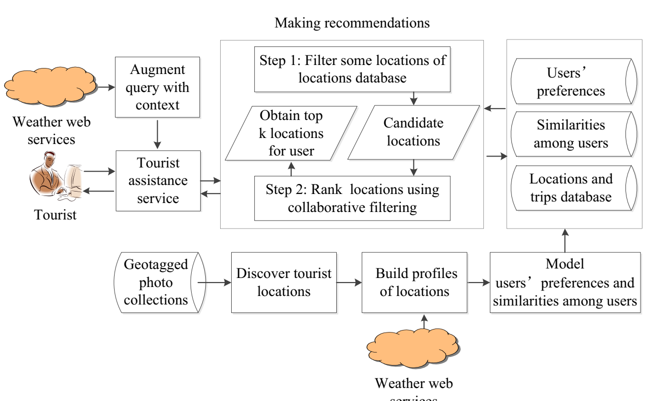

Fig. 1. The architecture of the proposed recommendation method

III. Discovering Tourist Locations

Discovering tourist locations can be viewed as a clustering problem of identifying highly photographed locations. Clustering algorithms such as k-mean and mean shift have been utilized to cluster photos using associated geotags for the identification of tourist locations. However, density based clustering algorithms, e.g. DBSCAN [11], have several advantages over other types of clustering algorithms: (1) Require minimum domain knowledge to determine the input parameters and can identify clusters with arbitrary shape; (2) Can filter outliers and have good efficiency on large-scale data. DBSCAN algorithm needs only two parameters: ε and *minPts* (the minimum number of points required to form a cluster). DBSCAN algorithm randomly chooses an object and forms a range search with radius ε and iteratively discovers subsequent density reachable objects to make the cluster. DBSCAN algorithm works with generic points having a unified density threshold for all clusters; however, the locations extracted by clustering the given collection of photos can have varying sizes and densities. In order to address this problem, Kisilevich et al. proposed P-DBSCAN [12]. They extended the definition of directly density reachable by adding an adaptive density technique. In P-DBSCAN, an object O is directly density reachable from another object O' if it is not farther away than a given density radius ε, and the ratio of surrounding objects between O and O' must be less than a density ratio w.

In our work, P-DBSCAN algorithm is adopted to cluster photos to discover tourist locations using the geotags of photos. After this step, we obtain a set of tourist locations $L = \{l_1, l_2, \ldots, l_n\}$. Each element $l = \{P_l, g_l\}$, where P_l is a group of geographically clustered photos, and g_l are geographical coordinates to represent the centroid of cluster P_l, which are computed from a group of geotags associated with the photos in the cluster P_l.

IV. Profiling Tourist Locations

After we discover the tourist locations, we formulate the profiles of locations and build a database of locations. In this part, we mainly focus on mining of context of locations which is useful for travel recommendation [13],[14]. Algorithm-1 illustrates the method for the profiling of locations.

First step is to identify visits made by different users from photos taken by them on these locations (lines 1-12). For each location, we sort photos of each user according to the taken time of photos. We get visit v from a photo p taken by a user u at a location l at time t. Note that a user u can take more than one photo in a visit at a location. Therefore, if the duration between the time-stamps of two photos taken by a user at the same location is less than visit duration threshold $visit_{thr}$, we consider that both photos belong to a same visit. We utilize the median of time-stamps associated with photos as the visit time t of the visit v.

Second step is to build the history of contexts in which locations have been visited (line 13). The time-stamp enables us to derive the season information and to retrieve weather context from weather services.

The last step is to find the popular context of each location from the history of contexts derived from visits made to respective location (line 20). We consider season and weather context concepts with highest frequency as popular context of location l.

After these steps, we build a locations database LDB=$\{l_1, l_2, \dots, l_n\}$, where each location $l_i = \{V_{li}, pop(s), pop(w)\}$, V_{li} are visits made to location l_i by different users, $pop(s)$ is the frequent season context of location, and $pop(w)$ is the frequent weather context.

Algorithm 1 Profiling locations.

Input : $L=\{l\}$ set of locations where $l = (P_l, g_l)$
Output: LDB = Database of locations with updated profiles
1. For all locations $l = (P_l, g_l) \in L$ do
2. Create list V_l
3. Create list of users U_{pl} from P_l and sort photos $P_{ul} \in P_l$ taken by each user $u \in U_{pl}$ according to photo taken time $p.t$
4. For all user $u \in U_{pl}$ do
5. Create list T_v
6. For all $p \in P_{ul}$ do
7. If $p_i.t - p_{i-1}.t < visit_{thr}$ then
8. Add $p.t$ to T_v
9. Else
10. $v \leftarrow$ New(visit)
11. $v.t \leftarrow$ Median(T_v)
12. $v.w \leftarrow$Retrieve-from-weather-DB($v.t$)
13. Abstract($v.t$, $v.w$)
14. Add v to V_l
15. Clear P_v
16. Add p to P_v
17. End if
18. End for
19. End for
20. $l.pop(w, s) \leftarrow$Popular(V_l)
21. Add l to LDB
22. End for

V. Modelling The Preferences And Similarities Of Users

A. Building User-Location Matrix

To obtain the preferences of users U in a set of locations L, we exploit the links (set of visits V) between users U and locations L to build a weighted undirected graph $G_{UL} = (U; L; E_{UL}; W_{UL})$, where U and L are nodes to represent users and locations, respectively. E_{UL} and W_{UL} are sets of edges and edge weights between U and L to represent visits of users and the number of visits to particular locations.

Given m users and n locations, we build an m by n adjacency matrix M_{UL} for graph G_{UL}. Formally, $M_{UL} = [h_{ij}]$, where h_{ij} represents how many times the ith user has visited the jth location. From matrix M_{UL}, the travel interest of user u_a can be derived as an array $R_a = <r_{a0}, r_{a1}, \dots, r_{an}>$, where r_{ai} is the implicit rating (visits made by u_a) of u_a in a location i. $S(R_a)$ is the subset of R_a, $\forall_{r_{ai}} \in S(R_a)$, $r_{ai} \neq 0$, that is, the set of locations that have been preferred (visited) by u_a. The average rating in R_a is denoted as $\overline{R_a}$. For example, if $R_1 = <2, 3, 2, 0, 3, 1, 2, 0, 1>$, $R_2 = <2, 3, 0, 3, 3, 4, 4, 2, 0>$, then $S(R_1) = <2, 3, 2, 3, 1, 2, 1>$, $S(R_2) = <2, 3, 3, 3, 4, 4, 2>$, $\overline{R_1}=2$, $\overline{R_2}=3$.

B. Building User-User Similarity Matrix

In our method, the similarities among users are calculated based on their traveling sequences (trips), which are mined from their travel histories. A trip of a user is a sequence of visited locations arranged according to temporal order. There are several steps for the process of identifying the trips:

First, the time-stamp of photos is exploited to sort the photos of a user.

Second, based on whether the time duration of two consecutive photos in the travel history of a user is longer than a threshold $trip_{dur}$, the travel history is split into user trips.

Third, since a user can have more than one photo in one tourist location in one visit. In this case, if two consecutive photos represent the same location, then only one photo is taken into consideration, and then each photo is replaced by its corresponding tourist location in LDB.

After the extraction of trips for all users, a database for trips is maintained. The database can be represented as TDB = $\{T_1, T_2, \dots, T_n\}$, where T_i represents the set of trips made by user i.

The similarities among users are calculated based on their traveling sequences (trips), and we employ the modified longest common subsequence (LCSS) to measure the similarity of two trips [15],[16]. The main idea here is that the matching of two sequences (trips) allows some locations to be unmatched within a *minimum bounding envelope*, but the order of locations in the match must remain. In other words, two sequences $T_a = \{l_{a1}, l_{a2}, \dots, l_{an}\}$ and $T_b = \{l_{b1}, l_{b2}, \dots, l_{bn}\}$ are deemed to be similar if there exist a long subsequence T'_a of T_a that can be mapped to a long subsequence T'_b of T_b. Here, T'_a (or T'_b) does not necessarily consist of consecutive locations from T_a (or T_b), but the locations in T'_a (or T'_b) must be in the same order as in T_a (or T_b)[17]. Moreover, the minimum bounding envelope allows for some noise and outliers in the sequences (trips) matching. The similarity of sequence T_a and T_b is calculated by Equation 1.

$$S'(T_a, T_b) = \frac{|T'_a|}{\max(|T_a|, |T_b|)} \quad (1)$$

where $|T_a|$ and $|T_b|$ are the number of locations in trips T_a and T_b. $|T'_a| = |T'_b|$ is the cardinality of the longest common subsequence.

We calculate the similarities among users based on their traveling sequences using Equation 2.

$$S(u_a, u_b) = \max\{S'(T_a, T_b)\} \quad (2)$$

where T_a and T_b are the trips visited by users u_a and u_b, respectively.

After this step, we build similarity matrix M_{TT} of users. Each entry in M_{TT} represents the similarity between u_a and u_b. A larger value means that both users are more similar in terms of traveling sequences.

VI. RECOMMENDATIONS

Input: a query $Q = (u_a, s, w, d)$, where u_a is a target user; s is the season information; w is the weather information; and d is the target city user u_a will visit.

Output: a list of locations in target city d that are recommended for user u_a to visit.

A query Q is processed by the following two steps:

In the first step, locations of the target city that meet the contextual constraints s and w are filtered out to form the candidate set of tourist locations L'.

In the second step, we utilize the user-location matrix M_{UL} that represents the preferences of users and M_{TT} that represents the similarities among users to personalize the location recommendations for user u_a in the target city. From M_{TT}, we retrieve top n similar users of user u_a, who have visited the target city. Then, Equations 3-6 is used to predict the preferences of user u_a for each location l_i in L', which is based on collaborative filtering [18].

$$Score(l_i) = \overline{R_a} + k' \sum_{T_b \in T'} S(u_a, u_b) \cdot \left(r_{b_i} - \overline{R_b}\right) \tag{3}$$

$$k' = \frac{1}{|T'|} \sum_{T_b \in T'} S(u_a, u_b) \tag{4}$$

$$\overline{R_a} = \frac{1}{|S(R_a)|} \sum_{j \in S(R_a)} r_{a_j} \tag{5}$$

$$\overline{R_b} = \frac{1}{|S(R_b)|} \sum_{j \in S(R_b)} r_{b_j} \tag{6}$$

where T' is the set of similar users of user u_a. After computing the preference of user for each location l_i in L', we order the locations based on preference score and return k locations as the query result.

VII. EXPERIMENTS

We collected about 1 million geotagged photos which were taken in cities (Hangzhou, Shanghai, Beijing, and Hongkong) of China from Flickr, and the additional information of photos (e.g. taken time and user ID) were also collected. Historical weather data of these cities was collected utilizing the public API of Wunderground. To detect locations from photos, for P-DBSCAN we set *minPts* to 50 photos, ε to 50 meters, and density ratio w to 0.5. To detect visits made by users to different locations from their contributed photos, we set visit duration threshold $visit_{thr}$ to 6 hours. Additionally, we set $trip_{dur}$ to 12 hours to identify trips.

As baseline methods, we utilize: (1) Frequent Rank (FR) [19], that uses PrefixSpan to extract frequent locations and sequential patterns from SDB whose frequencies are not smaller than the given minimum support threshold; (2) Classic Rank (CLR) [20], that can score the travel locations and sequences by exploiting reinforcement relationships between users and locations from matrix M_{UL}; (3) Personalized Context-aware Rank (PCR) [10], that considers weather context information and simple user similarity computation.

Precision is utilized as performance metric, which can be defined as the percentage of correct predictions. Table 1 shows the performance of our method and baseline methods in terms of precision. m represents the number of locations visited by each user.

TABLE I
PERFORMANCE COMPARISON IN TERMS OF PRECISION

Method	FR	CLR	PCR	Our method
Precision(m<5)	0.42	0.39	0.45	0.49
Precision(5<m<10)	0.39	0.41	0.49	0.54
Precision(10<m)	0.31	0.43	0.56	0.62

From this table, it can be seen that the precisions of these methods are 0.42, 0.39, 0.45, 0.49 when the number of locations is no more than 5. CLR gives the worst result, and our method gives better results than other methods. But there is no significant increase in terms of precision among these methods. It might be because most users do not have special interest but visit locations which are popular and significant when they visit a city. When the number of locations is more than 10, it can be seen from this table, there is a significant increase for the performance of PCR and our method. These results indicate that computation of similarity of users is useful to improve the precision of recommendation for users when they visit more locations. It can be also seen that the precision of our method is 0.62. It outperforms PCR recommendation methods. These results also suggest that considering the similarity of trips and contexts (i.e. season and weather) is efficient to increase the recommendation precision.

VIII. CONCLUSIONS AND FUTURE WORKS

In this paper, we propose a travel recommendation method which mines meaningful locations from CCGPs and takes into account the similarity and context of users for travel recommendations. We mine the travel preferences of users from their travel history, and present the evaluation of our method on a Flickr dataset. Experimental results show that the method is able to predict the preferences of users in an unknown city precisely and generate better recommendations than baseline methods. These results imply that considering the similarities among users based on their traveling trips is useful for travel recommendations. In the future, we will use the attributes of photographers (e.g. gender and age) to further improve the accuracy of tourist preference mining.

REFERENCES

[1] A. Popescu, G. Grefenstette, "Deducing trip related information from flickr," In *Proceedings of the 19th International Conference on World Wide Web*, ACM, 2009, pp. 1183-1184.

[2] H. Yin, X. Lu, C. Wang, N. Yu, and L. Zhang, "Photo2trip: an interactive trip planning system based on geo-tagged photos," In *Proceedings of the International Conference on Multimedia*, ACM, 2010, pp. 1579-1582.

[3] X. Lu, C. Wang, J. M. Yang, Y. Pang, and L. Zhang, "Photo2trip: generating travel routes from geo-tagged photos for trip planning," In *Proceedings of the International Conference on Multimedia*, ACM, 2010, pp. 143-152.

[4] Y. Arase, X. Xie, T. Hara, and S. Nishio, "Mining people's trips from large scale geo-tagged photos," In *Proceedings of the ACM International Conference on Information and* Multimedia, ACM, 2010, pp.133-142.

[5] T. Kurashima, T. Iwata, G. Irie, and K. Fujimura, "Travel route recommendation using geotags in photo sharing sites," In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*. ACM, 2010, pp. 579-588.

[6] T. Kurashima, T. Iwata, G. Irie, and K. Fujimura, "Travel route recommendation using geotagged photos," Knowledge and Information Systems, 2013, 37, pp. 37-60.

[7] A. J. Cheng, Y. Y. Chen, Y. T. Huang, W. H. Hsu, and H. Y. M. Liao, "Personalized travel recommendation by mining people attributes from community-contributed photos," In *Proceedings of the International Conference on Multimedia*, ACM, 2011, pp. 83-92.

[8] Y. Y. Chen, A. J. Cheng, and W. H. Hsu. "Travel Recommendation by Mining People Attributes and Travel Group Types From Community-Contributed Photos," IEEE Transactions on multimedia, 2013, 15(6): pp. 1283-1295.

[9] A. Majid, L. Chen, G. Chen, H. T. Mirza, and I. Hussain, "GoThere: travel suggestions using geotagged photos," In *Proceedings of the 21st International Conference on World Wide Web*., ACM, 2012, pp. 577-578.

[10] A. Majid, L. Chen, G. Chen, H. T. Mirza, I. Hussain, J. Woodward, "A context-aware personalized travel recommendation system based on geotagged social media data mining," International Journal of Geographical Information Science, 2013, 27(4) pp. 662-684.

[11] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, 1996, pp. 226-231.

[12] S. Kisilevich, F. Mansmann, and D. Keim, "P-DBSCAN: a density based clustering algorithm for exploration and analysis of attractive areas using collections of geo-tagged photos," In *Proceedings of the 1st International Conference and Exhibition on Computing for Geospatial Research & Application*, ACM, 2010, pp. 38.

[13] D. Lee, S. E. Park, M. Kahng, S. Lee, S. g. Lee, "Exploiting contextual information from event logs for personalized recommendation," Computer and Information Science, 2010, pp. 121-139.

[14] G. Adomavicius, A. Tuzhilin, "Context-aware recommender systems", Recommender systems handbook. Springer US, 2011, pp. 217-253.

[15] Y. T. Zheng, J. Z. Zheng, and T. S. Chua, "Mining travel patterns from geotagged photos," *ACM Transactions on Intelligent Systems and Technology*, 2012, 3 (3) pp. 56.

[16] M. Vlachos, M. Hadjieleftheriou, D. Gunopulos, and E. Keogh, "Indexing multi-dimensional time-series with support for multiple distance measures," In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, 2003, pp.216-225.

[17] G. Das, D. Gunopulos, and H. Mannila, "Finding similar time series," In *Principles of Data Mining and Knowledge Discovery*. Springer Berlin Heidelberg,1997, pp. 88-100.

[18] G. Adomavicius, and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," IEEE Transactions on Knowledge and Data Engineering, 2005, 17(6), pp. 734-749.

[19] Z. Yin, L. Cao, J. Han, J. Luo, and T. Huang, "Diversified trajectory pattern ranking in geo-tagged social media," In *Proceedings of the SIAM International Conference on Data Mining*, 2011, pp. 980-991.

[20] Y. Zheng, L. Zhang, X. Xie, and W. Y. Ma, "Mining interesting locations and travel sequences from GPS trajectories," In *Proceedings of the 19th International Conference on World Wide Web*, ACM, 2009, pp. 791-800.