Proceedings of the Sixth International AAAI Conference on Weblogs and Social Media

Composing Traveling Paths from Location-Based Services

Hsun-Ping Hsieh, Cheng-Te Li

Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei, Taiwan
d98944006@csie ntu.edu.tw, d98944005@csie ntu.edu.tw

Abstract

With the emergence of location-based services, such as Foursquare and Gowalla, users are allowed to easily perform check-in actions anywhere and anytime. The location-based check-in not only enables personal geospatial journeys but also serves as a kind of fine-grained source for trip planning. In this work, we aim to collectively compose traveling paths by leveraging the check-in data through mining the moving behaviors of users. A novel system, TP-Comp, is developed. To compose travel paths, TP-Comp not only allows users to specify starting/end and/or must-go locations, but also provides the flexibility of the time constraint requirement (i.e., the expected duration of the trip). By considering a sequence of check-in points as a traveling path, we mine the frequent sequences with some ranking mechanism to achieve the goal. Our TP-Comp targets at travelers who are unfamiliar to the objective area/city and have time limitation in the trip.

Introduction

Nowadays, one of the major characteristics of location-based services (LBS), such as Foursquare[1] and Gowalla[2], is to construct personal geospatial journeys by check-in actions. With smart phones, users can easily perform check-in actions to transmit the geographical information of current locations and the timestamps to certain location-based services. By leveraging the check-in records of users, this work develops an intelligent system, TP-Comp, to compose popular traveling paths for travelers or backpackers who are unfamiliar with some areas. While existing travel planning systems consider either the shortest geodesic distance or the shortest time period to recommend traveling locations, we alternatively consider the frequent check-in sequences as traveling paths derived from collective visiting tracks of past users. In addition, the check-in data allows the flexibility of time constraint requirement. That says, we enable users to enter the expected total time of the objective travel. In short, we targets at travelers who are not only unfamiliar with the objective areas, but also have time limitations in their trips.

There are real-world scenarios needing the proposed composer of traveling paths: (a) a user is the first time to a city and wonders to have a one-day trip route starting from her hotel. (b) In a city, a backpacker wants to visit the landmarks which are popularly travel by her friends, and thus need to have a trip plan connecting the desired locations. (c) A knowledge worker is under a business travel and wants to have a route with some must-go spots. (d) A taxi driver desires to recommend a visiting path to his customers along the direction to the destinations.

Our goal is to compose travel paths, which consists of sequences of check-in locations, for travelers with time requirement. We formulate a k-TP (i.e., top-k traveling paths) problem to achieve the proposed composer. The central idea is to represent a traveling path as a sequence of check-in points and to mine the top-k time-based frequent subsequences. Given (a) a database of user check-in records, (b) the starting and/or ending location points, and (c) the time constraint of this trip, our goal is to return a ranked list of top k frequent traveling paths satisfying the query requirements.

Here we give an illustration. Assume it is the first time for a traveler to visit New York City. He has two must-go attractions, Times Square (TS) and Metropolitan Museum of Art (MMA). He considers MMA as the first attraction to go and TS as the final destination for his 10-hour trip. A recommended traveling path, MMA $\xrightarrow{6hr}$ KFC restaurant $\xrightarrow{1hr}$ Macy's store $\xrightarrow{1.5hr}$ H&M store $\xrightarrow{1hr}$ 34 St-Herald Sq $\xrightarrow{15min}$ Times Sq-42St $\xrightarrow{2min}$ TS, is one possible result for his requirement.

Existing works to discover traveling paths are limited. The approaches of mining GPS trajectories to provide route recommendation are discussed in (Yuan et al. 2010), (Zheng et al. 2009). Yuan et al. infer fastest routes from historical trajectories on the road network. Based on the users' interested activity, Zheng et al recommend the travel sequence from GPS trajectory database. Arase et al. mine travelers' frequent trip patterns from photo database (Arase et al. 2010). To the best of our knowledge, we are the first to tackle the composition of frequent traveling paths with time constraints using location-based check-in data.

System Framework and Method

We collect a large-scaled check-in data from Gowalla, which contains 6,442,890 check-ins and 950,327 friendships from Feb. 2009 to Oct. 2010. We give the system overview of TP-Comp in Figure 1.

[1] https://foursquare.com/
[2] http://gowalla.com/

Preprocessing. For each user, we transform his check-in records into sequences of locations. We also associate the time span between two consecutive locations by averaging the time duration between two check-in places for all people. A database of traveling paths is then built.

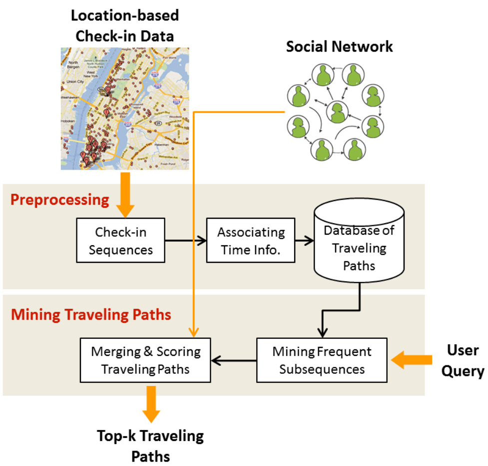

Figure 1. System Overview.

Frequent Subsequence Mining. The goal is to mine frequent subsequence from the database of traveling paths. For a query containing some locations with a time constraint, our TP-Comp system first retrieves the sequences with query locations. Then we devise a frequent subsequence mining method to find the frequent paths of locations. We measure the importance of a subsequence by calculating its *support* value, which is the number of sequence containing such subsequence. A subsequence P is *frequent* if its support is not less than *minsup*, where *minsup* is a user-specified minimum support threshold. In our system, we prefer to set *minsup* = 1% to ensure that the system contains enough traveling paths. We adopt similar concepts from (Wang et al. 2007) to mine the frequent subsequence. The mining process is recursively performed in a depth-first search manner until no more frequent subsequences can be found.

Since we aim to compose the traveling paths based on the query starting and/or ending locations, the mined subsequences can be further pruned accordingly. Besides, we also remove subsequences violating the time constraint. A subsequence is ruled out if its total traveling time exceeds the time constraint specified by the traveler.

Since the number of query locations could be larger than the length of mined traveling paths, we might need to merge the two or more separate subsequences to obtain traveling paths satisfying query requirements. We merge two traveling paths if they share at least one location. Here we use an example to elaborate the idea. Assuming that there are two frequent subsequences, $P_A = <L_{A1}, L_{A2}, L_{A3}, L_{A4}>$ and $P_B = <L_{B1}, L_{B2}, L_{B3}, L_{B4}, L_{B5}, L_{B6}>$, are mined, where $L_{A2}= L_{B3}$. If the location L_{A1} is the starting location, and the L_{B6} is the destination, we merge P_A and P_B to generate a new path $P_C = <L_{A1}, L_{A2}, L_{B4}, L_{B5}, L_{B6}>$. Besides, we average the support values of original routes to be the support of the merged traveling path.

To return a ranked list of top-k traveling paths, we provide three ranking criteria. The first one is the support value, which determines the popularity of mined subsequences. The second one is the time constraint. If the total time of a mined traveling path is much shorter than time constraint, the remaining time could be wasted. We rank traveling paths based on the remaining time (the less, the better). The third one is the social effect. Users may prefer those mined subsequences that had ever visited by their friends. Therefore, for a mined traveling path, if the sum of visit times for its locations is larger, it will tend to have a higher rank. Such three ranking criteria can be combined to have better quality of traveling paths.

System Interface

The system interface of TP-Comp is shown in Figure 2. Travelers are allowed to input some desired locations or choose popular attractions we suggest, as well as the expected travel time duration. The generation top-k traveling paths are shown in the right panel. If the traveler has Gowalla ID, the system can use his own social circle to derive the results. Furthermore, TP-Comp can also recommend the transportation mode between locations, which is developed by integrating with Google Map API.

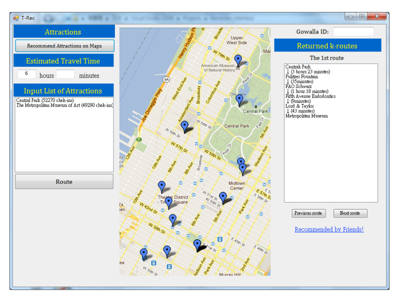

Figure 2. The interface of TP-Comp system.

References

Arase,Y., Xie, X., Hara, T., and Nishio, S.. Mining people's trips from large scale geo-tagged photos. In ACM MM, pages 133-142, 2010.

Chen, Z., Shen, H. T., and Zhou, X.. Discovering popular routes from trajectories. In ICDE, pages 900-911, 2011.

Wang J., Han J., and Li C. Frequent closed sequence mining without candidate maintenance. IEEE Transactions on Knowledge and Data Engineering TKDE, 19(8), 2007.

Yuan, J., Zheng, Y., Zhang, C., Xie, W., Xie, X., Sun, G., and Huang, Y.. T-drive: driving directions based on taxi trajectories. In ACM GIS, pages 99-108, 2010.

Zheng, Y., Zhang, L., Xie X, and Ma, W.-Y. Mining interesting locations and travel sequences from GPS trajectories. In WWW, pages 791-800, 2009.