TripRec: Recommending Trip Routes from Large Scale Check-in Data

Hsun-Ping Hsieh
Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei, Taiwan
d98944006@csie.ntu.edu.tw

Cheng-Te Li
Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei, Taiwan
d98944005@csie.ntu.edu.tw

Shou-De Lin
Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei, Taiwan
sdlin@csie.ntu.edu.tw

ABSTRACT

With location-based services, such as Foursquare and Gowalla, users can easily perform check-in actions anywhere and anytime. Such check-in data not only enables personal geospatial journeys but also serves as a fine-grained source for trip planning. In this work, we aim to collectively recommend trip routes by leveraging a large-scaled check-in data through mining the moving behaviors of users. A novel recommendation system, *TripRec*, is proposed to allow users to pecify starting/end and must-go locations. It further provides the flexibility to satisfy certain time constraint (i.e., the expected duration of the trip). By considering a sequence of check-in points as a route, we mine the frequent sequences with some ranking mechanism to achieve the goal. Our TripRec targets at travelers who are unfamiliar to the objective area/city and have time constraints in the trip.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications – *Data mining*.

General Terms

Algorithms, Performance, Design.

Keywords

Route recommendation, trip planning, subsequence mining, check-in data.

1. INTRODUCTION

Location-based services (LBS), such as Foursquare[1] and Gowalla[2], allow the construction of personal geospatial journeys through check-in actions. With smart phones, users can easily perform check-in actions that transmit the geographical information of current location and timestamp to certain location-based services. Leveraging the check-in records of users, this paper aims to develop an intelligent system, TripRec, to recommend trip routes for travelers or backpackers who are unfamiliar with some given area. While most of the existing trip planning systems [3][6] consider either the shortest geodesic distance or the shortest time period to recommend routes, we recommend the frequent check-in sequences as trip routes derived from collective travel tracks of past users. We also allow users to specify the expected time they want to spend for the trip.

The central idea is to represent a route as a sequence of check-in points and to mine the top-k time-based frequent subsequences. Given (a) user check-in records, (b) the starting and/or ending location points, and (c) the time constraint of this trip, our goal is to recommend a ranked list of k frequent routes satisfying the query requirements.

Related Works. Most of the existing works discover trip routes using the GPS trajectory records [2][7]. Yuan et al. [7] infer fastest routes from historical trajectories on the road network. Based on the user-interested activities, Zheng et al. [8] recommend the travel sequences from GPS trajectory database. Though Arase et al. [1] mine travelers' frequent trip patterns from geo-tagged photos, they provide no spatio-temporal user querying. To the best of our knowledge, we are the first to tackle the recommendation of frequent trip routes with time constraints using check-in data.

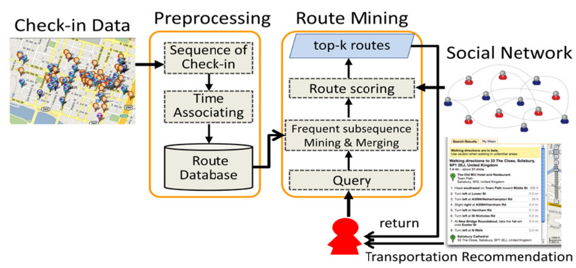

Figure 1. System Overview.

2. METHODOLOGY

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 TripRec in Figure 1.

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

Frequent Subsequence Mining. The goal is to mine frequent subsequence from the route database. For a query containing some locations with a time constraint, we first retrieve the sequences with query locations. Then we devise a frequent subsequence mining method to find the frequent routes 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 routes.

We adopt similar concepts from [5] 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.

The idea of closed pattern mining was introduced by Pasquier et al. [4], to improve the efficiency for large-scaled data. A close frequent subsequence is a sequence with no super-sequences of the same frequency in the database. A deliberate closed subsequence discovery can eliminate frequent but redundant subsequences so that the efficiency is boost and the memory usage is reduced. We proposed three pruning strategies to achieve such goal. The first strategy is *same projected database removal*. If P_1 is a super-sequence of P_2 and both share the same projected database, P_2 needs not to be grown because the subsequence generate from P_2 will not be closed. The second strategy is *forward checking scheme*.

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

WWW 2012 Companion, April 16 –April 20, 2012, Lyon, France.
ACM 978-1-4503-1230-1/12/04.

A pattern P is not closed if there is a frequent subsequence e in P's projected database, whose support is equal to P's support. The third strategy is *backward checking scheme*. A subsequence P needs not to be grown if there is a frequent location e before P, whose support is equal to P's support. Thus, every pattern generated from P is contained by the subsequence generated from concatenating P and e and both subsequences have the same support. By applying the three strategies, the closed frequent subsequences can be efficiently mined.

Since we aim to recommend the routes based on the query starting/end locations, the mined subsequences can be further pruned accordingly. Besides, a subsequence is ruled out if its total traveling time exceeds the time constraint specified by the traveler.

Route Merge. Since the number of desired locations could be larger than the length of mined routes, we might need to merge the two or more separate subsequences to obtain routes satisfying query requirements. We merge two routes if they share at least one location. Taking Figure 2 as an example, assume 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, and $L_{A2}= L_{B3}$. 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 route P_C: $<L_{A1}, L_{A2}, L_{B4}, L_{B5}, L_{B6}>$. We average the support values of original routes to be the support of the merged route.

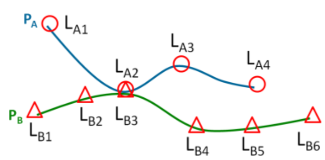

Figure 2. An example of merging subsequences.

Route Scoring. To obtain a ranked list of top k routes, we provide three ranking criteria. The first utilizes the support value, which determines the popularity of mined subsequences. The second criterion utilizes the time constraint. We rank routes based on the whether its total spending time is close to the expected travel time given by the user. The third is the social effect. Users may prefer those mined subsequences that had ever visited by their friends. Therefore, for a mined route, if the sum of visiting times for its locations is larger, it will be assigned a higher rank. These ranking criteria can further be combined to have better quality of routes.

3. TRIPREC SYSTEM AND EXPERIMENT

The system interface of TripRec is shown in Figure 3. Travelers are allowed to input some desired locations or use the popular attractions the system suggest. They can also enter the expected travel time duration. The recommended top-k trip routes are shown in the right panel. If the traveler has Gowalla ID, the system can use his own social circle for recommendation. Furthermore, TripRec also recommends the transportation mode between locations, which is developed using Google Map API.

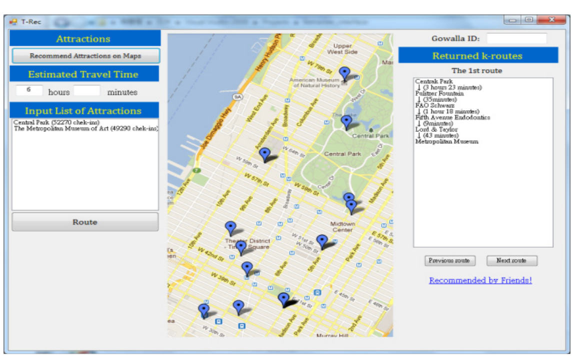

Figure 3. The system interface of TripRec.

We aim to provide a real-time recommendation based on the user query. We conduct experiments to show the response time (in seconds) using some manually-compiled queries. We randomly select one hundred attractions in New York City and combine them to form new queries. The Figure 4 presents the average run time by varying the *minsup* from 1% to 20%. We can find that as the *minsup* gets smaller, the response time of our system increases slowly. Even with low *minsup*, our system still works efficiently, especially when using pruning strategies. Such results indicate TripRec can provide efficient real-time recommendations.

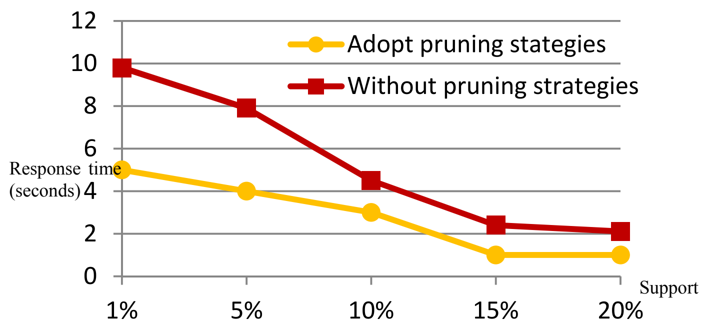

Figure 4. Time efficiency in seconds for our TripRec.

4. Conclusion

In this paper, we demo a system that recommends trip to users based on check-in data. Different from the GPA-data driven recommendation, the check-in data provides a discrete time sequence of objects that allow us to provide more accurate recommendations. TripRec has several potential usage scenario such as recommending one-day trip starting from a hotel for the first-time traveler; or recommending a series of landmarks that visited by the friends to a user.

5. ACKNOWLEDGEMENTS

This work was supported by National Science Council, National Taiwan University and Intel Corporation under Grants NSC 100-2911-I-002-001, and 10R70501.

6. REFERENCES

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

[2] Chen, Z., Shen, H. T., and Zhou, X. Discovering popular routes from trajectories. In ICDE 2011.

[3] Lu, H.C., Lin, C.Y., and Tseng, V.S., Trip-Mine: An Efficient Trip Planning Approach with Travel Time Constraints, In MDM 2011

[4] Pasquier, N., Bastide, Y., Taouil, R., Lakhal, L., Discovering frequent closed itemsets for association rules, In ICDT 1999.

[5] Wang J., Han J., and Li C. Frequent Closed Sequence Mining without Candidate Maintenance, IEEE Transactions on Knowledge and Data Engineering, 19(8), 2007.

[6] Yoon, H., Zheng, Y., Xie, X., and Woo W., Social Itinerary Recommendation from User-generated Digital Trails, Personal and Ubiquitous Computing, 2011

[7] 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, 2010.

[8] Zheng, Y., Zhang, L., Xie, X., and Ma, W.-Y. Mining interesting locations and travel sequences from gps trajectories. In WWW 2009.