

---

# Constructing Trip Routes with User Preference from Location 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

**Abstract**

This paper presents a novel trip route construction using location check-in data. Given a set of preference lists of locations from users, we aim to coordinate their preferred locations to visit and construct a route which not only satisfying user preferences as many as possible, but also being popular and reasonable. We formulate such preferred route construction as an optimization problem. We solve it efficiently and effectively by devising some greedy methods. Experiments on Gowalla large-scale check-in data show the promising effectiveness and efficiency of our methods.

**Author Keywords**

Preferred route construction; trip recommendation

**ACM Classification Keywords**

H.2.8 [Database Applications]: Data Mining

**General Terms**

Algorithms, Design, Experimentation

**Introduction**

Location-based Services (LBS), such as Foursquare and Gowalla, allow users easily performing check-in actions that pin the geographical information of current locations and timestamps on their personal pages. Such rapid accumulative check-in records enable us to not only collectively represent the real-world human geographical activities, but also provide the potential for location-based applications. One of the interesting applications is to compose the travel routes because the check-in records implicitly reveal how people move around an area with rich

---

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

Copyright is held by the author/owner(s).

*UbiComp'13*, September 8–12, 2013, Zurich, Switzerland

ACM 978-1-4503-2215-7/13/09.

<http://dx.doi.org/10.1145/2494091.2494155>

**Problem Definition.** Preferred Route Construction (PRC): Given a set  $S$  of preference lists of locations,  $S = \{PL(u_1), PL(u_2), \dots, PL(u_m)\}$  which is specified by users, the location check-in records, and a parameter  $k$  representing the desired number of locations in the composed route, the goal is to find a length- $k$  sequence of location as the trip route, denoted by  $r^*$ , which not only covers the locations required by  $S$  as many as possible, but also maximum the route goodness  $f(r^*)$ .

**Proposition.** The PRC problem is NP-complete. We can prove this proposition by a reduction from the *maximum coverage* problem. Due to the page limit, we skip the proof here.

spatial and temporal information, including longitude, latitude, and check-in timestamp. Existing works [1][6] analyze, mine, and recommend traveling paths from GPS trajectories, geo-tagged photos, and check-in data. In addition, some studies [3][4] consider past geographical activities of users to provide personalized trip planning. Besides, the recent works by Z. Chen et al. [2] and L.-Y. Wei et al. [7] aim to recommend popular and attractive trajectories. H.-P. Hsieh [5] et al. emphasize that the recommended routes should be time-sensitive to have better user experience. To the best of our knowledge, our work is the first attempt to recommend trip routes by coordinating the preference lists from multiple users.

In this paper, instead of generating trip routes for a single user, we solve a novel problem, constructing the trip routes from the preferred locations of a group of individuals. Consider the scenario that some individuals desire to go sightseeing together for the economic concern or a graduation trip, and each person has his/her own preferred locations to visit, how to intelligently coordinate the preference lists of them? How to construct a popular and reasonable trip route that satisfies the preference requirement?

We develop an intelligent system to construct preferred routes for group sightseeing. Given (a) user check-in records in a certain LBS, (b) the preference lists of some individuals, and (c) the number of desired locations in this trip (i.e., route length), we claim that a good route should (a) satisfy the location preferences of all the individuals in the group as much as possible, and (b) be essentially popular and reasonable according to the traveling routes in existing check-in records. By modeling the user preference and the route popularity into an objective function, we solve the problem by formulating it as an optimal route search. Since finding the optimal routes can be proved to be NP-complete, we devise several greedy solutions to tackle the problem.

## PROBLEM DEFINITION

Let  $A = \{a_1, a_2, \dots, a_n\}$  denote the collection of **locations**. First, we define the *popularity*( $a_i$ ) as the number of location check-in at  $a_i$ . Second, we define a **route** as  $r = \langle a_1, a_2, \dots, a_k \rangle$ , which is a sequence of locations. Third, for each route  $r$ , we define its **route support**  $RS(r)$ , which represents the number of people who had ever travelled along the locations in route  $r$ . Third, we define a **preference list** of an user  $u$ , denoted by  $PL(u)$ , as a ranked list of locations. Locations at the top-ranked positions are the more preferred ones.

Our goal is to construct a route which not only satisfies the preferences of users but also has high route popularity. We formulate the proposed route construction as an optimization problem, which consists of two parts. First, we model how a route satisfies the preference lists of users. We sum up the satisfaction value of each individual according to the preference list. We define *User Satisfaction* for route  $r$  by

$$US(r) = \sum_{u_i \in U} \sum_{j=1}^{|PL(u_i)|} \frac{|PL_{max}| - HIT(PL(u_i)_j, r) + 1}{|PL(u_i)|}$$

where  $U$  is the set of users in this trip,  $|PL(u_i)|$  is the number of preferred locations in  $u_i$ 's preference list, and  $|PL_{max}|$  is the largest number of preferred locations among all preference lists. The function  $HIT(PL(u_i)_j, r)$  returns the ranked position where the  $j^{\text{th}}$  of  $PL(u_i)$  is hit in route  $r$ . Higher the  $US(r)$  scores indicate more satisfactory of the route with respect to users. We also define the *Route Popularity*:

$$RP(r) = \sum_{n=1}^{|r|} \text{norm} \left( \sum_{(i,j), i-j=n-1, i>0, j>0} RS(a_i a_{i+1} \dots a_j) \right)$$

where  $\text{norm}()$  is a normalization function  $[0, 1]$ . For example, suppose there has a route  $r_1 = \langle a_{k1}, a_{k2}, a_{k3} \rangle$ ,  $RP(r_1) = \text{norm}(RS(a_1) + RS(a_2) + RS(a_3)) + \text{norm}(RS(a_1 a_2) + RS(a_2 a_3)) + \text{norm}(RS(a_1 a_2 a_3))$ . Routes and their sub-routes traversed more frequently will be given higher scores. The *route popularity* considers the location popularity in terms of both the order relation and the sequential

frequency. The final objective function is defined as  $f(r) = \alpha \times US(r) + (1-\alpha) \times RP(r)$ , where the parameter  $\alpha$  allows people putting different weights between user satisfaction and route popularity. The routes with higher  $f(r)$  values are preferred to be recommended. The formal problem definition and the NP-complete proposition are given in the left margin of the previous page.

**METHODOLOGIES**

To solve the *PRC* problem, enumerating all possible routes is very time-consuming and cannot perform real-time response. Therefore, we propose some pattern-based and greedy strategies to efficiently approximate the optimal solution.

*Pattern-based Approach*

Since we are required to construct the, preferred, popular, and reasonable route, we propose to mine the pattern of location sequence from the check-in data, because existing route patterns could be presumed to be good solution. The action can be achieved in the **offline** stage. We consider the mined frequent subsequences as the possible routes to return. Such pattern-based approach can be regarded as heuristically maximizing the route popularity. And then as the set of preference lists is given, we can **online** calculate the score of each pattern. The pattern with higher scores will be returned. The advantage of the pattern-based approach is three-fold: (1) allowing the offline processing of pattern mining to boost the time efficiency, (2) modeling the nature that people prefer to visit popular locations, and (3) enabling top-*k* route recommendation. However, the running time of scoring patterns is sometime unacceptable if too many patterns are mined.

*Greedy Search*

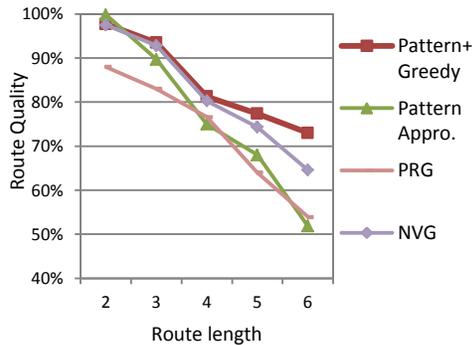
The greedy method adds locations incrementally. By starting from the location which has the maximum number of being check-in and is one of the user commonly preferred one, each time we find a neighbor location  $a_x$  such that appending  $a_x$  to the end of the current composed route can have the

maximum increase for the route goodness objective function. Such process is executed until the desired number of location is achieved. The greedy method efficiently reduces the search space. In real-world cases, it can provide the dynamic recommendation. For example, it can help backpackers decide the next location to leave for. To recommending next locations, in the following, we present a variety of greedy methods, which of each possess it own physical meaning and application scenario. .

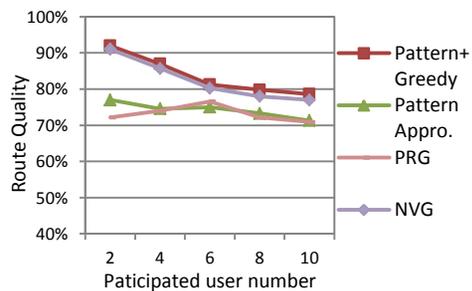
**Preference-based Greedy (PRG)** search assumes that users in the group are all familiar with the travel area, so considers only their preference. It iteratively picks the next location which can lead to the maximum score increase for the route popularity function  $RP(r)$ , until the number of desired locations is achieved.

**NaiVe Greedy (NVG)** search considers purely the route goodness function  $f(r)$ . While choosing the next location, we attempt to append each of the neighboring location to the current sub-route, calculate the route goodness  $f(r)$ , and choose the location with the highest goodness score. Thus we can expect that the route generated from *NVG* can get higher route goodness than above methods. However, in our experiments, the execution time of *NVG* is a bit inefficient since too much candidate locations in each round should be considered.

**Pattern+Greedy** search. Since the *NVG* method work inefficiently but effectively, we resort to find the balance between *NVG* and the pattern-based method. We first use the pattern-based approach to offline mine the candidate locations (i.e., the next locations to be selected) for reducing the search space. And then we use *NVG* to compose routes with higher  $f(r)$  scores from the mined patterns. In details, we first adopt the offline pattern-based approach to iteratively extend the frequent length- $k'$  sub-route pattern  $P(k' \geq 1)$  at level  $k$  (i.e., length  $k'$  sub-route) to get its frequent length- $(k'+1)$  sub-route patterns in a depth-first search manner. Then we construct a pattern tree to combine the mined



**Figure 1:** Route Quality by varying the route length.



**Figure 2:** Route Quality by varying the number of users.

**Table 1.** Criteria and questions for user study.

Criteria	Question
Popularity (LP)	Do you think the recommended is popular?
Visiting Order (VO)	Is the visiting order in the route suitable/ acceptable?
Preference Satisfaction (PS)	Do you think the recommended route is good enough to fairly satisfy the every member in groups?
Response Time (RT)	Is the response time acceptable during traveling?
Overall Acceptance (OA)	If you are a traveler, do you want to adopt this route?

patterns, in which each node represents a frequent location, and the level of the tree represents the length of a route. Each node (except for the root) has exactly one parent and it can be traversed starting with the root node, to obtain a certain route. After constructing the pattern tree, we greedily traverse the tree to construct the final route with the maximum  $f(r)$  score. This method only takes into account the mined connected sub-routes in the tree when considering the next location. For example, assuming the user is currently at location  $B$  and asks where to go next, and also assuming we have length-2 sub-route patterns,  $BE$ ,  $BA$  and  $BD$ . In such example we consider only the locations  $E$ ,  $A$  and  $D$  as the next one to visit.

### EXPERIMENTAL RESULTS

We use large-scaled check-in data from Gowalla, which contains 6,442,890 check-ins from Feb. 2009 to Oct. 2010. We vary the desired location number (i.e., route length) and the number of participated users to show the effectiveness for different methods. Assume our method generates route  $r$  and the optimal route is  $r^*$ , which can be derived using the brute-force manner. We define the *Route Quality*  $RQ(r) = f(r)/f(r^*)$  as the effectiveness measure. Routes with higher *route quality* and more reasonable running time are considered good ones. We also conduct a user study to ensure our method is convincing.

**Effectiveness vs. Route length.** We choose top-50 popular attractions as candidate locations. The participated users can choose any preferred locations from this pool. The number of participated users is 6 and the length of each preference list is limited to be desired length. We randomly generated 1,000 queries (each query contains a set of preference lists), and compute the route quality. We set the parameter  $\alpha$  in our goodness function as 0.5. Figure 1 shows Pattern+Greedy can achieve better quality, followed by *NVG* for almost scenarios. The pattern-based approach performs well when the route length is small but does poorly for the other cases.

The drop rate of Pattern+Greedy is also the least one while increasing the route length since the pattern tree can effectively avoid our greedy method falling into the bad local maximum.

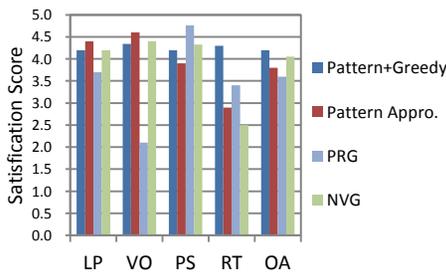
**Effectiveness vs. Number of Participated Users.** By setting the route length as 4, the results is shown in Figure 2, we can observe that Pattern+Greedy method still outperforms other methods. In general, the more users participate, the more diverse the preferences in this trip become.

### User study

We conduct a user study to test if our Pattern+Greedy and other methods can output rational, useful and acceptable results. We invite 50 persons to participate the user study. The evaluation criteria are listed in Table 1. For each user, we ask him/her to give the 0~5 score to each criterion for each constructed route. Higher score represents better satisfaction for such criteria. The route length ranges from 3 to 7. Each participator will provide  $5(\text{criteria}) * 5(\text{routes}) * 4(\text{methods}) = 100$  scores. Finally we report the average value for each criterion in Figure 3. Pattern+Greedy produces scores greater than 4 for all the criteria. The pattern-based approach and *NVG* can achieve high scores in the most criteria but do not work well for the response time. Other methods are not suitable to cover the popular factor and do not satisfy the preferences given by users.

### References

- [1] J. Bao et al. Location -based and Preference-Aware Recommendation Using Sparse Geo-Social Networking Data. In ACM GIS 2012.
- [2] Z. Chen et al. Discovering Popular Routes from Trajectories. In IEEE ICDE 2011.
- [3] Z. Chen et al. Searching Trajectories by Locations: An Efficiency Study. In ACM SIGMOD 2010.
- [4] Y. Ge et al. Cost-aware travel tour recommendation. In ACM KDD 2011.
- [5] H.-P. Hsieh et al. Recommending Time-Sensitive Routes by Exploiting Large-Scale Check-in Data. In ACM UrbComp 2012.
- [6] Q. Liu et al. Personalized Travel Package Recommendation. In IEEE ICDM 2011.
- [7] L.-Y. Wei et al. Constructing Popular Routes from Uncertain Trajectories. In ACM KDD 2012.



**Figure 3:** Results of subjective evaluation for four methods.