Keyword-aware Route Search for Various User Requests

Shuang Wang, Qiaqiao Zhang, Yinzhe Wang and Xueliang Zheng
Software College, Northeastern University, Shenyang, Liaoning, China
wangsh@mail.neu.edu.cn; {1763976417, 1065369206, 1019821493}@qq.com

Abstract: With the popularity of smart mobile devices and social network, there are numerous location-based data with semantic information. This paper proposes a new route search algorithm which can learn from users’ historical record to meet different requests from various users. In this algorithm, point matching and candidate route generation should be done firstly. Secondly, the ranking scores of candidate routes are calculated. Also, the distance of routes is considered in this part. For users’ requests, it is calculated by users’ historical record including keyword record and distance record. Keyword record represents users’ requests for keywords, and it is comprised of queried record and selected record. Distance record represents users’ tolerance of route distance. Finally, route suggestions are given to users after calculation. Meanwhile, the record is updated after users’ selections. At the end of the paper, the experiments are conducted on real location-based datasets, and the result of the experiments shows that the proposed algorithm demonstrates good performance for different users’ requests.

1. Introduction
With more and more location-based data on the Internet, route search has been playing a very important role.

On the study of route search and recommendation, the simplest way is searching routes through spatial information and returning top-k routes with the shortest distance[1]. Moreover, there are some more complex algorithms that search routes through spatial and semantic information, and also return top-k routes with the shortest distance[2]. Some route search algorithms leverage the information of spatial, semantic and even temporal information[3]. Almost all the algorithms retrieve the top-k shortest routes which satisfy the users’ queries. However, users do not always need the shortest route. Sometimes users prefer the routes which contain more points meeting their tastes. In other words, people prefer a personalized route to a short route at some time.

To meet different requests from various users, this paper proposes a new route search algorithm, which returns the top-k routes with the highest ranking scores. It learns from users’ historical record to satisfy their requests. This historical record is the extra information which is used in the algorithm.

The rest of the paper is organized as follows. The framework overview of the algorithm is presented in Section II. In Section III, the algorithm of route exploration is introduced. Scoring and ranking by users’ record is defined in Section IV. The experimental results are presented in Section V. Section VI concludes this paper.

2. Framework Overview
In this section, the proposed algorithm is described as framework in detail. As can be seen in Figure 1, the framework is comprised of the offline module and the online module.

2.1. Offline module
The offline module comprised of two modules: Data set and users’ record.

In data set, there are three kinds of information of points. They are spatial information (longitude and latitude), keywords and rating score. These three kinds of information are used by point matching and scoring and ranking in online module.

![Figure 1. Overview framework of the proposed algorithm](image)

2.2. Online module

The online module is used for the query of one user, and the proposed route search procedure is presented in this part. Given the user’s query, which contains query location (longitude and latitude) and query keywords, the first step is point matching according to the query and data set in offline module. The next step is generating candidate routes with the matched points from the last step. Afterwards, the combination of data set and users’ record in offline module is used to do scoring and ranking for the candidate routes. Then the route suggestions which is ranked are given to the user to select. Finally, the user’s selection can be gotten and the user’s record in offline module is updated according to the user’s selection.

3. Route Exploration

In this section, the basic definition is presented firstly. The procedure of route search is introduced in detail including point matching and candidate route generation. In addition, scoring and ranking and suggestions and record updating are presented briefly, and they are presented in detail in the next section.

3.1. Basic Definitions

- **Definition 1: POI(Point of Interest).** POI \( p = (p.\lambda, p.\psi, p.s) \), is comprised of spatial information \( p.\lambda \) (including longitude and latitude), keywords \( p.\psi \) and score. \( p.\psi = k_1, k_2, \ldots, k_i, \ldots, k_{|p|} \), is the keywords set, which describe the features of the POI, Where \( k_i \) is the \( i \)th keyword and \( |p| \)is the number of keywords. The rating score \( p.s \) is marked by public which represents satisfaction of people to the POI and can be retrieved by the websites like yelp.

- **Definition 2: Query.** Users’ queries \( q = (q.\lambda, q.\psi) \), are comprised of spatial information \( q.\lambda \) (longitude and latitude) and query keywords. The spatial information is the user's current
location or started point, and keywords $q, \psi = k_1, k_2, \ldots, k_i, \ldots, k_{|q|}$, are the Poi's categories that a user wants to go. The number of query keywords may be more than one.

3.2. Point matching

Point matching is the first step of the proposed algorithm. The goal of this step is to search points which contain query keywords. These points are used in the next step candidate route generation.

Given the query from a user, the algorithm searches every point for the query location as the center, with a radius of $R$. If the point in the radius contains some keywords of query, it is inserted to the candidate point set. This searching will halt when every keyword in the query can be satisfied by the points in the candidate point set. Otherwise, $R$ can be enlarged as below and continue to search points in the range until the candidate point set contains every query keyword.

$$R \leftarrow (1 + \varphi_R) \cdot R$$  \hspace{1cm}  (1)

Where $\varphi_R$ is the adjusting parameter of search radius updating. Moreover, $R$ is updated after the user’s selecting, and it is presented later in the paper.

3.3. Candidate route generation

Assumed firstly that there are $N$ points in candidate point set after point matching. This step is to generate all the candidate routes.

As known, every keyword in query must be contained by generated routes. Nevertheless, for one keyword in query, there may be several points in candidate point set which contain this keyword in query. Here is a simple example to show how to generate all the possible routes.

There are three keywords in query as $q, \psi = (k_1, k_2, k_3)$. Candidate point set has five points, they are $A(k_1), B(k_2), C(k_3), D(k_1), E(k_1)$. A table (Table 1) below containing all the information can be easier to understand.

| $k_1$ | $k_2$ | $k_3$ |
|-------|-------|-------|
| A     | B     | D     |
| C     |       | E     |

The first row is the keywords in the query. Every column is the points which contain corresponding keyword in query. The algorithm searches all the possible routes which contain every keyword in query. In this example, the possible routes are as follows:

- $A \rightarrow B \rightarrow D$
- $A \rightarrow C \rightarrow D$
- $A \rightarrow B \rightarrow E$
- $A \rightarrow C \rightarrow E$

Finally, adjust the sequence of points in every possible route to make the route have the shortest distance. At the same time, the distance of every candidate route is calculated. It is worth noting that the distance of route is comprised of two parts, the first part is from the query location to the start of route, and the second part is from the start of route to the end of route. After adjusting, every possible route with its distance is inserted to candidate route set. At this moment, the routes in candidate route set are as follows:

- $B \rightarrow A \rightarrow D$ 3km
- $A \rightarrow D \rightarrow C$ 4km
- $A \rightarrow B \rightarrow E$ 5km
- $E \rightarrow A \rightarrow C$ 4km

4. Scoring and ranking by Users’ Record
In this section, scoring and ranking in online module is presented. Meanwhile, the method of users’ record updating is introduced in corresponding parts.

4.1. Route Score
The route suggestions are ranked by route score, and it represents the score of routes. Route score is comprised of the score of points in the route and the score of route distance. It is defined as:

\[
RS(u, r) = \varphi_{RS} \cdot \frac{1}{|q|} \sum_{p \in r} PS(u, p) + (1 - \varphi_{RS}) \cdot RDS(u, r),
\]

(2)

Where \(PS(u, p)\) is point score of point \(p\) in route \(r\) for user \(u\), \(\frac{1}{|q|} \sum_{p \in r} PS(u, p)\) is the mean value of point score in the route, \(RDS(u, r)\) is route distance score of route \(r\) for user \(u\), and \(\varphi_{RS} \in (0, 1)\) is the adjusting parameter of route score, which represents the preference of point score in route and route distance score in route score. \(PS\) and \(RDS\) are between 0 and 1, and they are introduced in detail later.

4.2. Point Score
Point score is a combination of point popularity score and keyword record score. The point popularity score is obtained by rating scores from public. Keyword record score is calculated through the record of every user. It represents every user’s preference. Point score is defined as:

\[
PS(u, p) = PPS(p) \cdot KRS(u, p),
\]

(3)

Where \(PPS(p)\) is point popularity score of point \(p\) and \(KRS(u, p)\) is keyword record score of point \(p\) for user \(u\). As can be seen in the formula of \(PS(u, p)\), point popularity score is strengthened by keyword record score through the way of multiplying. \(PPS(p)\) and \(KRS(u, p)\) are between 0 and 1, and they are introduced in the paper later.

5. Experiments
To evaluate the performance of the proposed algorithm, the experiments are introduced in this section. The dataset is the data of Sydney from OpenStreetMap database. OpenStreetMap is the project that creates and distributes free geographic data for the world.

The algorithms were implemented by python and all the experiments are conducted on an Intel(R) Core(TM) i5-2450M CPU @2.50GHz and 8GB memory.

5.1. Efficiency
In this subsection, a set of experiments to study the efficiency of the proposed algorithm is presented in detail. Runtime can be used to reflect the efficiency of the algorithm. Before the experiments, the adjusting parameter and learning rate are set as: \(\varphi_{RS} = \varphi_{KRS} = \varphi_{QRS} = 0.5\), \(\varphi_{LP} = \varphi_{LD} = 0.25\), \(\varphi_{OS} = 0.8\), \(\alpha_{KRS} = \alpha_{QRS} = \alpha_{OS} = 0.2\). Meanwhile, the searching range \(R\) is 1(km).

5.2. Learning Speed
To study the learning of the algorithm, a term called “learning speed” is defined. It means how many query times a candidate route may take to get the highest ranking score when the query is fixed and the user keeps selecting the same route. It represents how fast the algorithm can learn the users’ requests from record.

There are three parameters of learning rate including the learning rate of keyword selected record score, the learning rate of distance record score and the learning rate of \(R\). The default learning rate is 0.2. Meanwhile, the number of query keywords is 3 and the number of points in dataset is 17539. Before the experiments, the adjusting parameters are set as: \(\varphi_{RS} = \varphi_{KRS} = \varphi_{QRS} = 0.5\), \(\varphi_{LP} = \varphi_{LD} = 0.25\), \(\varphi_{OS} = 0.8\) and the searching range \(R\) is 1(km).
Two algorithms are compared in this subsection. They are the proposed algorithm with greedy pruning (GP), and the proposed algorithm without greedy pruning (GP).

5.2.1. Varying the learning rate of KSRS & DRS. The experiments are to study the learning speed of the algorithm with variation of $\alpha_{KSRS, DRS}$. Moreover, $\alpha_{KSRS} = \alpha_{DRS}$ in this experiment. Figure 2 shows the query times with different $\alpha_{KSRS, DRS}$ for the proposed algorithms with GP and without GP. As can be seen in Figure 2, query times decrease with higher learning rate of keyword selected record score and distance record score.

5.2.2. Varying the learning rate of $R$. The experiment is to study the learning speed of the algorithm with variation of $\alpha_R$. Figure 2 shows the query times with different $\alpha_R$ for the proposed algorithms with GP and without GP. As can be seen in the Figure 3, there is no clear tendency of the effect of learning rate of $R$. Because $R$ only influences the search range in the step of point matching.

6. Conclusion
In this paper, a new route search algorithm is presented, which can learn from users’ record to meet different requests from various users. The ranking scores of the algorithm are design to give users route suggestions, which are measured as a combination of route popularity and users’ requests. Moreover, the methods of updating from the users’ queries and selections are introduced to learn users’ requests. The results of empirical studies show the performance of efficiency and learning speed in the proposed algorithm.

Acknowledgments
(1) Keyword-Aware Route Search for Various User requests: Project 201910145212 Supported by National Training Program of Innovation and Entrepreneurship for Undergraduates.
(2) This work is supported by "the Fundamental Research Funds for the Central Universities"(N182410001)

References
[1] Li Rong Hua, Lu Qin, et al. Optimal multi-meeting-point route search[J]. IEEE Transactions on Knowledge and Data Engineering, 2015, 28(3): 770-784.
[2] Gao Cong, Hua Lu, et al. Efficient spatial keyword search in trajectory databases[J]. arXiv preprint arXiv:1205.2880, 2012.
[3] Abdel-Fatao Hamidu, Li Jiuyong, et al. Unifying spatial, temporal and semantic features for an effective GPS trajectory-based location recommendation[C]//Australasian Database Conference. Springer, Cham, 2015: 41-53.