International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 – 5161

Available at http://inpressco.com/category/ijcet

Research Article

A Query-based Travel Route Recommendation using Locationbased Social Networks

Ms. Sonali N. Parab and Dr. Santaji. K. Shinde

Department of Computer Engineering VPKBIET, Baramati Pune, India

Received 10 Nov 2020, Accepted 10 Dec 2020, Available online 01 Feb 2021, **Special Issue-8 (Feb 2021)**

Abstract

Travel route recommendation is broadly studied area in case of location based services. Millions of check-in records are generating all over the world through location-based social networks (LBSN), which can provide essential information about places visited by users and their traveling experiences through tag information. This information can be useful in suggesting travel routes to users to facilitate trip planning. Earlier route recommendation frameworks mainly focused on improving route optimality in case of budget, attractiveness and effectiveness but simultaneously have some limitations, which we have mentioned in this paper. The proposed system deals with planning a trip as per user preferences by recommending a proper route. Here, the user specifies his preferences via keywords. It aims to give reminder to the user about his/her schedule on trip in real time. The proposed system uses two algorithms namely, Dijkstra's and Haversine algorithm. The earlier one used to find out shortest distance between the places while the latter one used to find out the distance between two points on a sphere. The system implementation results in an android application, which aims to plan a trip as per details entered by the user and notifies user about his/her schedule in real time.

Keywords: *Trip planning; route recommendation; tag information; Location-based social network; point-of-interest recommendation*

Introduction

Typically, people tend to click photos while traveling and share their travel experiences through check-in data and geo- tagged photos. Various location-based social networking sites allow users to share their whereabouts with their friends. Nowadays, almost everyone has a GPS-enabled smartphone, which is changing the way people interact with the web by using locations as contexts. The problem of recommending travel routes to travelers is a broadly studied area. Until now, the number of approaches have suggested for points-of interest (POI) recommendation and route planning. The very first step for a traveler while exploring new places is travel route planning. Most people interested in traveling through the most popular travel path so that they can able to visit as many famous places as possible. To facilitate trip planning, the footprints of tourists at memorable destinations, i.e., the geo- tagged photos can be used to discover the travel paths to a specific location. We can help people discovering attractive and popular places as well as events when they are traveling out of town, once we understand their preferences.

It has observed that people tend to visit famous places, every place has its own proper visiting time, and people might follow their friend's footsteps. For example, users usually visit a restaurant during lunch hours and visit a pub at night. His friend might recommend the restaurant or pub. Hence, we can say that there is a geographical, temporal, and social impact on people's choices to visit any locations. The geographical influence suggests that the chance of a user visiting POI will be higher if this POI is closer to the users previously visited POIs [1]. People tend to explore POIs near their current positions or the ones that they have visited before. So, POIs visited by users often form spatial clusters, i.e., people tend to check in around several centers (e.g., "home" and "office") [2]. The temporal influence states that users tend to visit different places in different time slots, and in the same time slot, users tend to visit the same places periodically [3] [4]. The social influence here means that we take suggestions from our friends while finalizing any places to visit and their opinions have an impact on our decisions to visit any sites as compared to strange people [5]. However, identifying a preferred route is a significant problem that finds applications in map services. Hence we required to assign a score to each path as per its popularity and effectiveness.

Nowadays, with the exponential increase in the use, demand, and significance of online social media and mobile devices, it is possible to collect a large volumes of user check-in data and their experiences of traveling [6] [7]. It can be beneficial for users when they are planning a trip to multiple places of interest in an unfamiliar city by providing similar routes traveled by other people for reference [8]. We hope that people could find some exciting destinations and plan an interesting journey based on multiple user's experiences. Table 1 shows the comparison between the existing travel route recommendation models on the basis of parameters like algorithm, performance, output, limitations etc.

Table1 Comparison of Existing Travel Route Recommendation Models

Name of the model	Algorithm	Output	Performance Evaluation	Limitations
1. Pattern Aware Skyline Travel Route (PASTR)	Block Nested Loop Algorithm	Diverse routes with Interest, Time, and Matched score	1) High visiting time threshold (θ) leads to high precision but low recall 2) The hit rate, i.e. aggregating query points to existing trajectories of the model is about 69%.	It has not considered sociogeographical influence on recommendation
2. Keyword-aware Skyline Travel Route Recommendation (KSTR)	Route Reconstruction Algorithm	Routes associated with Average score	1) The precision of keyword extraction is reasonably high and unaffected by no. of K. 2) KSTR model offers the lowest edit distance and highest prediction accuracy	It is difficult for users to select a proper route, as full skyline routes are generated as a result
3. Recommend trajectory with POI ranking and transition (Rank + Markov model)	Rank + Markov Algorithm	Routes with POI preferences and POI to POI transitions	1) This algorithm outperforms its variants having only transition information	The proper visiting time of POI is unclear.
4. Keyword-aware Representative Travel Route Recommendation (KRTR)	Candidate Route Generation and Travel Route Exploration Algorithm	Travel routes considering attractiveness of POI, Proper visiting time of POI and Social influence on users.	1) KRTR framework outperforms the baseline algorithms in terms of effectiveness and efficiency.	Due to the real-time requirements for online systems, it has higher computational cost.

This paper deals with planning a trip as per user preferences. In order to achieve this Dijkstras and Haversine algorithms are used.

Literature Survey

Recently, some studies have done on Location-based services due to its wide applications such as POI recommendation, trip planning and traffic forecasting. In paper [9], large-scale check-in data from Gowalla has used for experiments to exploit the visiting time information of locations to recommend routes for users. Some documents have used the knowledge extracted from large-scale check-in data to recommend time-sensitive trip routes, consisting of a sequence of locations with associated time stamps. Existing works collaboratively studied the geographical, temporal, and social influence on the user's selection of POI and traveling route. It has been perceived that users tend to visit nearby places (Geographical influence), they like to visit different places in different time slots (temporal influence), and they prefer to visit the same places which are visited by their friends (Social influence). A graph named Geographical-Temporal Influences Aware Graph (GTAG) has been constructed using user's check-in records to exploit geographical and temporal influences [10]. To capture the social influence considered the impact of friends on users while selecting the travel routes, visiting places and POIs. The problem of approximate keyword searches in the massive semantic trajectories has studied in paper [11]. Approximate keyword query of the semantic trajectory (AKQST) supposed to return k trajectories that contain the most relevant keywords to the entered query and yield the least travel effort in the meantime.

Trip planning is the primary step for people who want to explore new places with having some preferences. To develop a collaborative recommendation model, which combines the popularity and convenience of travel routes is one of the challenges of online trip planning. Several works constructed personalized routes according to user queries. User queries may comprise of a starting point, destination point, preferences, and span of trip. In paper [12], Pattern Aware Skyline Travel Route (PASTR) framework has suggested in which interface was provided through which the user specify query points and ranges as rectangle via Google map. It focused on providing locations on the basis of scores. Keyword- aware Skyline Travel Route (KSTR) used score functions for the three features, namely the attractiveness of POIs, the visiting time of POIs, and the geographical-social influence [13]. KSTR further adapted the representative skyline search instead of the traditional top-k route recommendation system. Moreover, to provide a diverse set of travel routes it claimed that more features of Places of Interests (POIs) should be extracted from user's check-in data. Another approach for route recommendation in which a probabilistic model was proposed to combine the results of POI ranking and the POI to POI transitions in paper [14]. This approach was feature-driven and learned from past behavior without having to design specialized treatment for spatial, temporal, or social information. Rank + Markov path algorithm jointly optimized point preferences and routes. While Keyword-aware Representative Travel Route Recommendation (KRTR) framework generated Travel routes considering attractiveness of POI, Proper visiting time of POI and Social influence on users [15].

Proposed Methodology

The proposed system aims to recommend the route to facilitate trip planning. In this system user interact with the system by doing the registration and after successful registration, he or she can access the trip planning system by log into the system. Map interface is used to display the output. This system allows user to schedule his/her trip and sends notification to the user in real time.

A. System Architecture

Figure 1 describes the architecture of the proposed system.

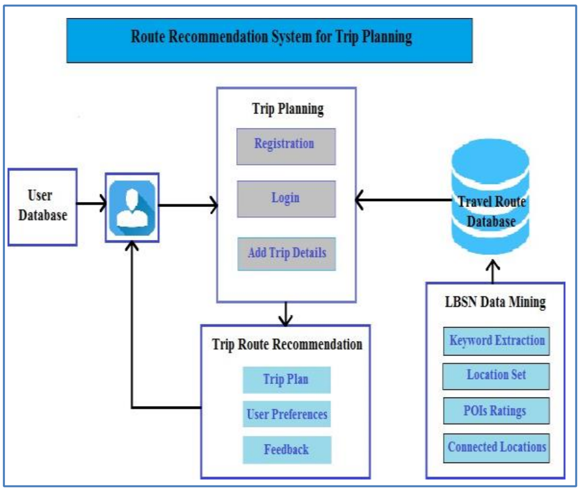

Figure1: System Architecture

In order to build the system, some data mining concepts such as keyword extraction and map to render the result are used. Extracted information may include keywords describing particular locations, various point-of-interests, proper time to visit specific location and reviews of users who has already visited the places. This information is stored into travel route database. Travel route database consists of location set, POIs such as hotels with ratings, petrol pumps, hospitals etc. To access the system, new user needs to register and then only he/she can enter the trip details like set of locations, schedule timings and preferences.

The user database has made up of registered users who have accessed the system. It consists of necessary user information such as username, password, email-id, mobile number, and stores username and password created by user during registration.

The trip planning application is nothing but the interface through which the user interacts with the route recommendation system by registration and then login. After successful login, he or she can search places and plan trip.

Travel route database is a repository in which location information, connected locations, and points-of interests such as hotels, shops with their ratings is stored. Users want some preferences while planning the trip to a particular destination such as petrol pumps, restaurants, parks, etc. Hence, the proposed system considers all this information during suggesting the trip plan.

LBSN dataset is nothing but traveling histories of users to particular locations collected form Facebook API. Here, we collect information such as POI ratings, reviews of users to particular places, etc. From the dataset, only active user's data has considered for further processing, because most of the users are quite inactive in sharing their whereabouts. The mining of LBSN data involves,

1) Keyword Extraction:

It is the process of extracting semantic meaning of the keywords and assigns a matched score to describe the degree of connection between keywords and routes [15]. There are three types of keywords extracted from LBSN dataset namely, *a) Geo-specific Keywords:*

Some tags describe specific location and represents its spatial nature. To quantify the geo-specificity of a tag, an external database identifies geo-terms in the overall tag set and then the tag distribution on the map rates the identified geo-term. The

geo-specificity (GS) score of a tag (w) is given as follows,

$$GS(w) \propto \quad GDA(w). \quad \exp(-GeoVar(w)) \tag{1}$$

$GeoVar$ (w) is the (latitude, longitude) set including a tag w.

b) Temporal Keywords:

Some tags describe time interval representing its temporal nature. To quantify the temporal-specificity of a tag, time distribution on a tag rates the identified temporal-terms. The temporal-specificity (TS) score of a tag w is given as follows,

$(w) \propto e(-timeVar(w))$ (2)

$timeVar$ (w) is the creation time of check-ins including a tag w.

c) Attribute Keywords:

Some tags are frequently associated with a POI describing for what that POI famous for, such tags can be defined as attribute keywords. An attribute score (AT) of a tag w can be defined as follows,

$$AT(W) \propto max_{p \in L} \frac{pf(I_p, w) * uf(I_p, t)}{rf(L, w)} \quad (3)$$

Here, I_p is the check-in set of p. Check-in frequency (pf), user-frequency (uf), and POI frequency (rf) are the three frequencies used in above equation. While L is the set of all POIs.

2) Location set:

Location set has created by manually recording user's check- in data of vising specific location. The goal is to extract a check-in triple, <*who, where, when*> from the LBSN dataset. From the check-in data who and when are clear from the user ID and the timestamp and *where* specifies the visited location.

3) POIs Ratings:

Locations and ratings of various POIs such as hotels, petrol pumps, parks, shopping malls etc. are stored in the travel database. These ratings will be considered while recommending a point of interest.

4) Connected Locations:

If there exists a proper route from one location to other, then these two locations can be considered as connected locations. Hence such route segments are extracted from LBSN dataset and stored into travel route database.

Trip route recommendation is nothing but the output of the system. As soon as user enters trip details, keyword matching process is done to find the entered locations. Keyword matching measure can be defined by using three keyword scores as follows,

$$KM(p, K) = \sum_{w \in K} tf\_idf\ (w, p).\ (GS(w) + TS(w) + AT(w)) \quad (4)$$

Here, tf is the frequency of tag w in POI and idf is the number of POIs with the tag w.

To find out the proper route Dijkstra's and Haversine algorithms are used. Detailed description of these algorithms has given in next section. A personalized trip plan will be returned to the user as output.

B. *Algorithms*

In proposed system, Dijkstra's algorithm find out shortest distance between two places. While Haversine algorithm find out the distance between two points on a sphere. With the help of these two algorithms, the proposed system can able to recommend a proper route and nearby POIs as per user interest.

1. Dijkstra's Algorithm:

This algorithm takes source location of user as input. Let the node from which trip started be the initial node. While destination place is considered as target node for example 'Y'. Let the distance of node 'y' be the distance from the initial node to 'Y'. Dijkstra's algorithm will assign some initial distance values and will try to improve them gradually.

Description of Dijkstra's algorithm has given below,

Step 1- Assign to every node a tentative distance value;
Dist [Initial_node] = 0;
Dist [v] = ∞;

Step 2- Current_node = Initial_node ;
Mark all other nodes unvisited. Create a set of all the unvisited nodes called the *unvisited set (Q)*.

Step 3- For the current node, consider all of its neighbors and calculate their *tentative* distances. Compare the newly calculated *tentative* distance to the current assigned value and assign the smaller one.

Step 4- When we are done considering all of the neighbors of the current node, mark the current node as visited and remove it from the *unvisited set(Q)*.

Step 5- If the destination node has been marked visited or if the smallest tentative distance among the nodes in the *unvisited set(Q)* is infinity then stop. The algorithm has finished.

Step 6- Otherwise, select the unvisited node that is marked with the smallest tentative distance, set it as the new "current node", and go back to step 3.

2. Haversine Algorithm

In the proposed system Haversine algorithm has used to find out the distance between two points on a sphere. The Haversine algorithm is used in electronics and other applications such as navigation. The Haversine formula determines the great-circle distance between two points on a sphere given their longitudes and latitudes. Haversine algorithm to calculate the distance from target point to origin point has given as follows,

Step 1- R is the radius of earth in meters.
Lat_O = latitude of origin point
$Long_O$ = longitude of origin point
Lat_T = latitude of target point
$Long_T$ = longitude of target point

Step 2- Difference in latitude = Lat_O - Lat_T
Difference in longitude = $Long_O$ - $Long_T$

Step 3- Φ = Difference in latitude in radians
Λ = Difference in longitude in radians
O = Lat_O in radians
T = Lat_T in radians

Step 4- A= sin (Φ/2) * sin (Φ/2) + cos (O) *cos (T) * sin (Λ/2)* sin (Λ/2)

Step 5- B = min (1, sqrt (A))
Distance = 2 * R * B

Result And Discussions

A. System Implementation:
The proposed system is implemented on windows 10 OS having Intel I3 processor with 20 GB memory requirement. The system is an android application developed by using Android studio and Java programming language. While Xampp, MySql database is used to store information in database.
1. Database used:
We require three databases namely,
i. LocationPoints-
To store information about attractive places in as well as information about POIs such as hotels, hospitals, and petrol pumps etc. In order to start the navigation Google map service is used to locate these places.
ii. TripDetails-
This database stores information about trip schedule of all users who have planned the trip.
iii. UserInfo-
This database store login details and other necessary information of registered users.

2. Results
Final output of the system will be a route joining source, halt point and destination. But before using the system, user first needs to go through registration process, verification and login. New user first needs to register by filling the information specified in the registration form like name, password email, address etc. During the registration process, mobile number and email address are verified via sending the secret key to the respective mobile number and email address. After registration user now can log into the system by using his login credentials. System then notifies the user about success or failure of login. Successful login will redirect the user to the screen where he can add the trip details. It will ask user to enter the source, halt station, destination, and other required information like time for lunch, rating of hotels etc. Figure 2 shows the final route when user starts the navigation. It shows source, halt point and destination.

Figure 2: Screenshot of Trip route

When the journey starts, system will notify the user about the popular places and his schedule regarding trip. Figure 3 shows how the system notify attractive places to the user.

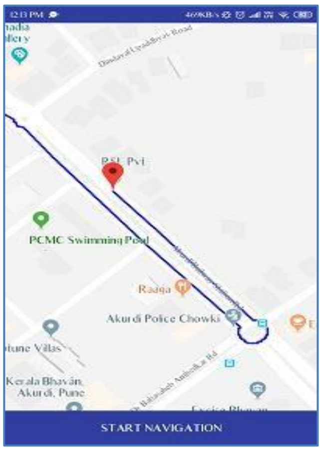

Figure 3: Screenshot of recommended places

In this way, after entering all the specified details, the user will be navigated to the map interface where he/she can see the route and the system will notify attractive places and timing of activities in real time.
The accuracy of proposed system is depends on the keyword matching accuracy and route prediction accuracy.
Keyword matching accuracy: It is the process of evaluating quality of the extracted keywords. For that precision of keyword is measured by ranking the tags by using scores. Also it is necessary to check performance of check-in extraction. We can use the evaluation measures like precision, recall, and F1 score as follows,

$$Precision = \frac{RP \cap Cp}{C_p} \quad (5)$$

$$Recall = \frac{Rp \cap Cp}{R_p} \quad (6)$$

$$F1 = \frac{2 * Precision * Recall}{Precision * recall} \quad (7)$$

Here, C_p is the set of check-in photos at location p while R_p is the relevant check-in data at p labeled manually.
Route prediction accuracy can be measured by using following goodness functions described given below,

1. Edit Distance:
The edit distance can be defined as the distance between two sequences in terms of the minimum number of edit operations required to transform one sequence into the other. The edit operations are

namely, insert into a sequence, delete from a sequence, and replace one landmark with another.

2. Geographical Region Cover Ratio:
The test route and recommended route can both be bounded by a geographical box. Geographical region cover ratio can be defined as the ratio of the overlapped region to the testing route region.

Conclusion

Travel route recommendation based on check-in data collected from LBSNs is an extensively studied area. This paper gives information about different route recommendation frameworks that have been implemented the extracted knowledge from the LBSN dataset to recommend diverse travel routes. Although collecting data from location based social networks is not problematic process but mining that huge volume of data is a tedious task. By studying the existing systems, we can say that the route recommendation models have evolved with time by using different scoring techniques used to rank the routes. The proposed system uses extracted knowledge from the LBSN dataset and creates route recommendation system. According to user preferences, a trip plan will be suggested as output. In order to achieve the proper route prediction and recommendation Dijkstra's and Haversine algorithm are used.

Acknowledgement

This paper would not be completed without the encouragement and guidance of Dr. S. K. Shinde, Guide of ME Dissertation work, and Mr. M. D. Shelar, Co-guide of ME Dissertation work. We especially thankful to Dr. C. S. Kulkarni, ME Coordinator and all the professors of the computer engineering department of VPKBIET, Baramati for their valuable advice and support to work on travel route recommendation using Location-based social network.

References

[1]. W. Wang, H. Yin, L. Chen, Y. Sun, S. Sadiq, and X. Zhou, GeoSAGE: A geographical sparse additive generative model for spatial item recommendation,| in Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2015, pp. 1255–1264.

[2]. H. Yin, X. Zhou, Y. Shao, H. Wang, and S. Sadiq, -Joint modeling of user check-in behaviors for point-of-interest recommendation,| in Proc. 24th ACM Int. Conf. Inf. Knowl. Manage., 2015, pp. 1631– 1640.

[3]. H. Wang, Z. Li, and W.-C. Lee, -PGT: Measuring mobility relationship using personal, global and temporal factors,| in Proc. IEEE Int. Conf. Data Mining, 2014, pp. 570–579.

[4]. Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann, -Who, where, when and what: Discover spatio-temporal topics for twitter users,| in Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2013, pp. 605–613.

[5]. K. Zheng, S. Shang, N. J. Yuan, and Y. Yang, -Towards efficient search for activity trajectories,| in Proc. IEEE 29th Int. Conf. Data

[6]. Eng., 2013, pp. 230–241.

[7]. H.-P. Hsieh and C.-T. Li, -Mining and planning time-aware routes from check-in data,| in Proc. 23rd ACM Int. Conf. Conf. Inf. Knowl. Manage., 2014, pp. 481–490.

[8]. M.-F. Chiang, Y.-H. Lin, W.-C. Peng, and P. S. Yu, -Inferring distant-time location in low-sampling-rate trajectories,| in Proc. 19th

[9]. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2013, pp.

[10]. 1454–1457.

[11]. H. Yin, B. Cui, Y. Sun, Z. Hu, and L. Chen, -LCARS: A spatial item recommender system,| ACM Trans. Inf. Syst., vol. 32, no. 3, 2014, Art. no. 11.

[12]. Y.-T. Wen, P.-R. Lei, W.-C. Peng, and X.-F. Zhou, -Exploring social influence on location-based social networks,| in Proc. IEEE Int. Conf. Data Mining, 2014, pp. 1043–1048.

[13]. Q. Yuan, G. Cong, and A. Sun, -Graph-based point-of-interest recommendation with geographical and temporal influences,| in

[14]. Proc. 23rd ACM Int. Conf. Conf. Inf. Knowl. Manage., 2014, pp. 659– 668.

[15]. Zheng, B., N. J. Yuan, K. Zheng, X. Xie, S. Sadiq, and X. Zhou, Approximate keyword search in semantic trajectory database,| Proc. IEEE 31st Int. Conf. Data Eng., pp. 975–986. 2015.

[16]. W. T. Hsu, Y. T. Wen, L. Y. Wei, and W. C. Peng, -Skyline travel routes: Exploring skyline for trip planning,| in Proc. IEEE 15th Int. Conf. Mobile Data Manage., 2014, pp. 31–36.

[17]. Y.-T. Wen, K.-J. Cho, W.-C. Peng, J. Yeo, and S.-W. Hwang, KSTR: Keyword-aware skyline travel route recommendation,| in Proc. IEEE Int. Conf. Data Mining, 2015, pp. 449–458.

[18]. D. Chen, C. S. Ong, and L. Xie, -Learning points and routes to recommend trajectories,| in Proc. 25th ACM Int. Conf. Inf. Knowl. Manage., 2016, pp. 2227–2232.

[19]. Wen, Y.-T, J. Yeo, W.-C. Peng, and S.-W. Hwang, -Efficient Keyword-Aware Representative Travel Route Recommendation,| IEEE Transaction on Knowledge and data engineering, Vol. 29, No. 8. 2017.