Intention Oriented Itinerary Recommendation by Bridging Physical Trajectories and Online Social Networks

Xiangxu Meng
College of Computer Science
National University of Defense Technology
Changsha, China
aiguokk@gmail.com

Xinye Lin
College of Computer Science
National University of Defense Technology
Changsha, China
lxytcmn@gmail.com

Xiaodong Wang
College of Computer Science
National University of Defense Technology
Changsha, China
xdwang@nudt.edu.cn

ABSTRACT

Compared with traditional itinerary planning, intention oriented itinerary recommendation can provide more flexible activity planning without the user pre-determined destinations and is specially helpful for those strangers in unfamiliar environment. Rank and classification of points of interest (POI) from location based social networks (LBSN) are used to indicate different user intentions. Mining on physical trajectories of vehicles can provide exact civil traffic information for path planning. In this paper, a POI category-based itinerary recommendation framework combining physical trajectories with LBSN is proposed. Specifically, a Voronoi graph based GPS trajectory analysis method is proposed to build traffic information networks, and an ant colony algorithm for multi-object optimization is also implemented to find the most appropriate itineraries. We conduct experiments on datasets from FourSquare and GeoLife project. A test on satisfaction of recommended items is also performed. Results show that the satisfaction reaches 80% in average.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications; I.2.8 [**Problem Solving, Control Methods, and Search**]: Scheduling

General Terms

Algorithms, Measurement, Experimentation.

Keywords

itinerary planning, trajectory mining, location-based system, multi-level categories

1. INTRODUCTION

According to the daily experience, the nature of travel arrangement is to look for a group of geographical locations which are connected with most convenient traffic paths available and best meet the personal demands. As described in reference [2], the users usually hope to find a set of objects to meet various needs. For example, a traveler may have the following needs – shopping, dining, accommodation, sightseeing. These needs can only be met by a set of different geographic locations. Of course, there are many kinds of criteria to evaluate the final choice. Some people want the best service, and others pursue the lowest cost. On the initial stage of itinerary planning, the users usually only have a general intention. The final decision will be made after collecting enough relevant information. During the collection process, they may ask a friend for recommendation, do research on some Travel Forums, or analyze traffic data, etc.

When someone plans to travel to a new place, his initial intention is usually not very clear. In order to analyze such initial intentions, we put out questionnaires to a group of students in National University of Defense Technology (lies in Changsha, Hunan, China, about 1500km away from Beijing). The questionnaire includes the following three questions:

1. How much do you know about Beijing? (A. Have been there many times, very familiar. B. Only been there 1-5 times, not very familiar. C. Never been there, not familiar at all.)
2. Please list your arrangement there for a one-day trip?
3. If someone is willing to help you to arrange your trip, what demands will you have?

Example 1: The answer of a volunteer who belongs to C class (Never been there, not familiar at all):

In the morning, he wants to have breakfast in a fast-food restaurant and visits some historical interests afterwards. At noon, he would like to have lunch in a restaurant with local taste. As for afternoon, he is willing to walk besides a lake in a park. Last but not least, he needs buy some souvenirs for friends and then go to a party in a bar which should be convenient to get to. Before going to all these places, he hopes to get as much information as possible, and an accurate prospect on both time and money cost on the traffic.

Result analysis (20 valid questionnaires in total are collected):

Question 1: Numbers in parentheses represent the number of volunteers belongs to each class: A. (4) B. (8) C. (8)

UrbComp'12 ,August 12, 2012. Beijing, China

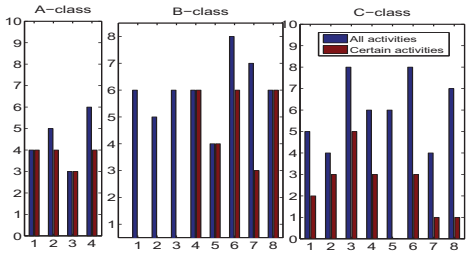

Figure 1: Results of questionnaire.

Question 2: For the convenience of analysis, the arrangements proposed by the users are divided into two categories: 1. Certain activities: activities that have locations and transportation requirements, for example, visiting the Imperial Palace in the morning. 2. General activities: General activities: activities that only have general intentions (without particular location requirement), for example, having Sichuan cuisine for lunch. As shown in Fig. 1, according to different choices made in question 1, users are divided into 3 classes, A, B and C. Users who are familiar with Beijing (A-class) are more likely to arrange certain activities. In contrast,most of the users of B and C class only have a general intention. Especially for the C-class users, the portion of general activities is higher than fifty percents. For more, 4 out of the 20 users do not have any precise preference of traveling locations.

Question 3: Five demands that have the highest occurrence rates are (the number in the parentheses indicates the number of users):1. unique/famous places(7); 2. least traffic time cost(5); 3.precise location description and best route(5); 4. avoidance of rush hour of travelers (4); 5. lower cost (4).

After the analysis above, it is reasonable to summarize the following features of a common user's itinerary planning for a new place.

1. The description of the demanded destination is usually a category but not a exact geographical location, such as "a famous attraction" , "a well-known snack bar" , etc.

2. Users demand precise information of each place they want to go such as exact geographical location, the traffic route, the shortest and average taxi time between any two locations.

3. Users are interested in knowing other people's review about the places.

4. Some of the users' demands have multiple optimizing objects, which are clearly expressed and possible to be mathematically described, such as least time cost on the traffic; some are very vague and hard to be described mathematically, such as the most famous location available.

It is very difficult for the computer to implement the aforementioned search or recommendation, because the object description is not clear, and the information in need is too extensive. Even a human guide will find it hard to give such recommendations, because it is impossible for him/her to know clearly about all the restaurants, hotels, attractions of a city, and the best traffic route and time among them. Besides, because the GIS-based shortest route search algorithm doesn't consider the real-time traffic change and the rush hours (weekends, people going to and off work), it is also very hard for the route search algorithm to figure out the best travel route and least time cost. To resolve such a problem and help the users to make the best decision, we jointly take in use of different users historical trajectories and information from social networks to provide a category based itinerary planning service. Luckily, both the current social networks and trajectory mining technologies can present effective support to such a service.

Currently, most mobile devices are able to position, based on the networks or GPS. This feature makes it very easy to collect the trajectory data of mobile users. In a much larger scale, the traffic control and city planning departments of a city are also collecting the trajectory data of quantities of vehicles and mobile devices. Some large enterprises are also doing so to carry on relative research. The T-drive project [11, 12]of Microsoft is an example. Based on the fact that the taxi drivers have the richest knowledge of the road systems of a city, researchers provide a real-time navigating service by mining the driving logs of a large number of taxis.

The location based social networks, such as Jiepang [1] and Foursquare, support the users with tagging, rating and reviewing places they have been to. These user-generated data have embraced the daily experience of millions of users, and are very valuable for the recommendation of itinerary planning.

All these technologies provide us the chance of realizing accurate itinerary recommendation. To our best knowledge, none commercial entities have yet provided itinerary planning recommendation service based on general demands. This paper will focus on the realization of general itinerary recommendation.

2. MODEL AND FRAMEWORK

2.1 Model of itinerary planning

It is easy to understand that an activity must be related to a location. Therefore, an activity can be denoted as a tuple with three elements:

$$A :=< P, T, condition >$$

P (Place) denotes the geographical position, which indicates an exact location, such as the Imperial Palace, the Olympic Park, etc.

T (Time) denotes the time when the activity takes place.

C (Condition) can be a constraint on any aspect of an activity. For example, it could be a constraint of having less cost than $\$100/person$, or traveling by taxi, etc.

It is important to notice that, the places are normally hierarchically organized. According to the analysis result from the questionnaires, the users usually won't give an exact depiction of the demanded destination. Instead, users are more likely to give a category such as bar, shopping mall, etc. In order to solve this problem, this paper takes use of the categorization of different locations of Foursquare. A category (C) is a set of different location points:

$$C := \{P_1, P_1, P_1, \ldots, P_n\}$$

[1] JiePang is one of the biggest location based social networks in China, www.jiepang.com.

The categories are hierarchical organized. Places belong to categories.

$$C_i \subset C_j, P_i \in C_j$$

A *path* indicates the connecting route between to places, i is the description, such as the time cost.

$$Path :=< P_s, P_e, i >$$

A trip is a time-ordered description of several activities and the path belong them.

$$Trip :=< A_1, Path_1, A_2, Path_2, \ldots, A_n >$$

In this paper, the interrogation mark (?) is used to indicate any ONE place, the asterisk mark (*) is used to indicate all the places in a category. By using the two marks, the user's vague demands of itinerary planning could be described as follows:

$$Need := \{ \begin{array}{l} <?C_1, T_1, condition_1 >, \\ <?C_2, T_2, condition_2 >, \ldots, \\ <?C_n, T_n, condition_n >>: Optimizer\} \end{array}$$

optimizer denotes the optimization goal, such as the shortest travel time or hot spots of all POIs. According to the model, example 1 can be described as follow:

$$\begin{array}{l} <?FastFoodRestaurant, Workday \odot 8:00, Taxi >, \\ <?HistoricSite, Workday \odot 9:00, Taxi >, \\ <?ChineseRestaurant, Workday \odot 12:00, Taxi >, \\ \{ <?Plaza, Workday \odot 13:00, Taxi >, \\ <?Mall, Workday \odot 18:00, Taxi >, \\ <?WineBar, Workday \odot 20:00, Taxi >, \\ : [MinTravelTime, AllPopular]\} \end{array}$$

2.2 Traffic information networks

To find the best trip, we need to build a cost function; the cost could be the time cost, the service evaluation, or the total distance along the trip. If we want to find the least time cost, the information of driving time between any two places would be necessary. Thus, we build up the semantic traffic map G of a city based on the taxis' trajectories:

$$G := \{P, Path\}$$

P is the set of all the geographical places; Path is the set of all the existing edges between these locations. To build such a map G, we need to get information of all the places and path between them. The detailed method of building the semantic traffic map is presented in Section 3.

2.3 Framework of Joint itinerary planning

This paper access information of geographical locations from social networks, and build up semantic traffic map by mining the GPS trajectories, as shown in fig.2. According to different cost functions, the system will present different trip recommendations for itinerary planning with detailed explanations respectively. The explanation includes valuable information for the user, such as the popularity of the places and the driving time between adjacent locations in the trip. The whole system consists of three repositories:

1. Repositories of location points: We crawling all location points, which are provided by different users and evaluation by millions of users, of the target city from location based social networks. This collaborative evaluated information can more objectively indicate the popular degree of location points;

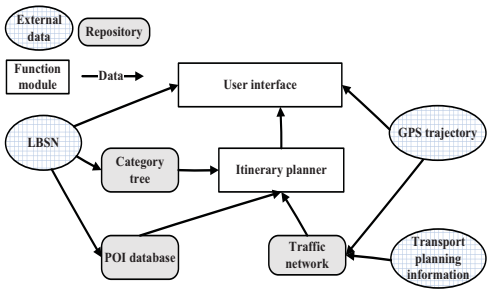

Figure 2: Framework of joint itinerary planning.

2. Categories of location points: It saves classification information of places. In this paper, we use hierarchy categories provided by foursquare;

3. Traffic information networks: It preserves the shortest travel time and the best path between any two arbitrary semantic location points of a city. We get semantic locations, which paid close attention by users, from public transit agencies, and mining taxi history data to obtain accurate shortest path information (section 3.2).

What's more, itinerary planner is responsible for recommending reasonable itineraries based on the information above, according to the user's requirements. The UI module calls the online map system (such as Google Map [2]) to visual display itineraries, and supports users to modify requirements interactively.

3. VORONOI GRAPH BASED TRAFFIC INFORMATION NETWORKS

Traffic departments of many countries and cities are collecting trajectories of public transport vehicles. These GPS data have high frequency and large time span, hence the data are very difficult to process. Besides, these trajectory data only contain the physical space and time stamps, so they cannot provide any semantic information [9], which the users can understand better and usually concern much more. To convert the physical trajectory into feasible semantic knowledge, there are two main obstacles: 1. How to determine the physical location point that the user understands best and knows most. 2. How to get the shortest path and driving statistics between any two location points. In the following part of this section, we propose the semantic position determining algorithm based on public transport stops, and the semantic traffic map building algorithm base on Voronoi diagram. These two algorithms make it possible to get the shortest time and optimized path between any two places.

3.1 Voronoi based semantic point building

Every city contains thousands or even more geographical locations, it is very hard for the residents to memorize all of them, not to say a stranger. In tradition, the most used method to describe some geographical location is to combine the "traffic network" and the landmarks. For example, No.163 of Haidian West Road, Beijing; or 400m east

[2]http://maps.google.com/

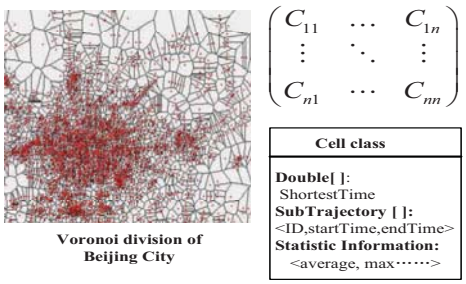

Figure 3: Voronoi based semantic point.

to the south gate of the Olympic Park. Fortunately, when planning a stop or station for public transport, the related departments always take the population, physical distance, traffic situation and other factors into consideration. And these stops are commonly named with semantic marks that the residents know well. Concerning the planning principle of these stops, it is reasonable to assume that for any important location A of a city, there is always a stop B, so that the time cost from B to A is smaller than a given threshold, for example, 10 minutes. Based on the assumption above, this paper tries to build up a semantic map of a city, based on the stops of public transportation. As shown in Fig.3, the whole city area is decomposed using Voronoi diagram, taking the stops as seeds. Each cell has a unique id, and is the named after the seed stop within it. Thus each cell generates a semantic point with constant id and name(***SemiPoint***). Voronoi diagram insures that, for any geographical point(GPS-Point,***Point***), there is a corresponding nearest semantic point (in our case, a stop). So, a GPS trajectory can be described by a semantic trajectory(Semi-trajactory,***SemiList***), which is a serial of ***SemiPoints***.The information of all the stops can be accessed on the traffic department's website. Voronoi diagram is built by the classic plane sweep algorithm [3].

For each stop point, we treat it separately as start and destination to build up the semantic traffic map, which is stored in form of adjacent matrix(***Time-Matrix[Stop Num][Stop Num]***). An element of this matrix describes the traffic statistics between two semantic points. The specific method to generate these traffic statistics is determined by the administrator according to different demands, which is implemented by different statistic methods(***StatisticFunction()***). For example, 7 days a week, calculate the shortest time and best path of every 2 hours. Considering that there're only 7 days data in the demo system, we only calculate the shortest time and best path between any two semantic points for rush hours on weekdays and workdays.

3.2 Run through model based traffic information networks building

After decomposing the city map, we need to calculate the statistics between any two cells. In traditional GIS system, the best path is simply the shortest path which could easily be found, and the shortest time could be estimated according to the Euclidean distance. However, in the real life, traffic restrictions, one-way road, dynamic traffic flows, and many other factors together generate a very complicate situation, in which the shortest path usually is not the one with smallest time cost. Furthermore, there isn't any model or algorithm yet that can find the path with shortest travel time both precisely and independently.

Algorithm 1 Voronoi-based Time Matrix Building

Input: Trajectory:***traj***; Voronoi-Map: ***map***
Output: Time-Matrix[CellNum][CellNum]: ***matrix***
1: ***Point p=traj.next()***;//Fetch a new GPS point
2: ***SemiList tmpList***=new ***SemiList()***;
3: **while** (***P***!=null) **do**
4: ***SemiPoint Sn=M.semi(P)***; //Get the semi-point based on physical location
5: **if** (***Sn*** is different from ***temList.last()***) **then**
6: **for** each ***SemiPoint Si*** ∈ ***temList*** **do**
7: ***Matrix[Si.ID][Sn.ID]=StatisticFuction()***;
8: **end for**
9: ***tmpList.add(semiPoint)***;
10: **end if**
11: ***p=traj.next()***;
12: **end while**
13: **return** ***matrix***;

The GPS trajectory contains a detailed log of the vehicle's travel history. Each GPS point includes the latitude and longitude coordinates, as well as the corresponding time stamp. Taking use of these data we can directly get the travel path and time cost between two locations. To match the semantic city map, the cell-ID of each GPS is allocated to it as the Semantic Point. After that, we have mapped millions of POI points to tens of thousands of stops, largely decrease the storage and computing cost. Considering how the semantic city map is generated, we can confirm that the timing error, which were introduced by such a mapping strategy, is no more than the time cost of vehicle passing through a cell. Take Beijing as an example, the travel time of buses between any two stops in Beijing is strictly less than 10 minutes; that means the final found shortest time cost will include an error less than 10 minutes, which is acceptable in the scenario of itinerary planning. Practically, the difficulty is that, there are not enough trips (a trip, as defined before, is a complete trajectory from start to the end) for any pair of Semantic Points to calculate the statistical driving time. Therefore, we propose that, for all the cells passed by, even they are not the start or end point for the path, we still treat them as statistic source. Thus, if we denote a trip as $A_s - A_e$, then after we take into consider the pass-by cells, it can be denoted as:

$$< A_s, A_1, A_2, A_3, A_4, \ldots, A_e >$$

As Algorithm 1 shows, when calculating the path statistics between cell A_1 and A_3, the sub-trip $A_1 - A_2 - A_3$ of $A_s - A_e$, which runs through both A_1 and A_3, is also taken part into statistics. Especially, this method can find potential shortest paths lying behind a longer trip. After entering a new cell, the algorithm will traverse over all the historical semantic points in the trip, store the travelling time between each historical semantic point and the new semantic point, and update the statistics. (line 5-8)

4. CATEGORY BASED ITINERARY RECOMMENDATION ALGORITHM

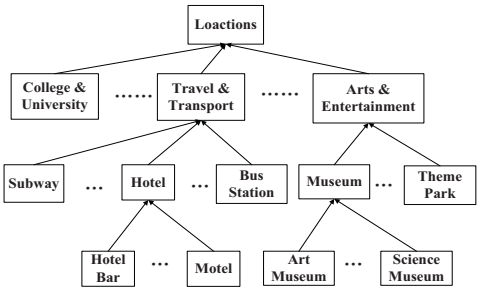

Figure 4: Hierarchy of the location points

4.1 Hierarchy of the location points

There are many hierarchical categorization methods of locations. This paper takes in use the categorization of Foursquare, one of the earliest and largest LBSN websites today. Fig. 4 partly shows the hierarchy of this categorization in the form of a tree. As shown in Algorithm 2, when a user gives his/her demanded location category, all the locations belong to that category and the corresponding subcategories are taken into consideration.

4.2 Least-time itinerary planning recommendation

This subsection proposes the itinerary planning recommendation algorithm that uses a cost function considering time cost, aiming at giving recommendations with least time cost.

4.2.1 BaseLine: Single-object optimization

We implement the Baseline recommendation algorithm with three steps. The algorithm enumerates all the possible locations, and traverses all the feasible trips and tries to find the best one to recommend. In previous section, we have received needs of user(***needs***),built traffic graph matrix(***matrix***) with each elements present cost from two semi-points(***Cost*** gotten by ***CostFunction()***),gotten category tree from Foursquare(CategoryTree,***ct***)and built the index of POIs(POIsIndex,***PI***)with ***CategorySet*** to describe which categories one POI belongs. As shown in Algorithm 2, in the first step (line 3),the k most popular geographical locations are given to each activity instance as candidates; in the second step (line 5-8), pick one from the k locations for each activity instance, and find the path connecting them by checking the semantic traffic map; then a recommendation item is generated; in the third step (line 10), rank the set of all recommendation items according to the cost of each item, and generate the final recommendation list.

4.2.2 Challenges of the Baseline algorithm

The baseline algorithm is simple and instinctive, but faces the following challenges:

1. Combinatorial explosion: the algorithm needs to traverse all the possible combinations of all the activities' candidate locations, which is at the level of k^{dim} (dim is the number of activities). When k increases, the computing cost will increase immensely. For example, when dim is 6 and k is set to 10, the time cost of the baseline algorithm reaches 2328 ms, which cannot meet the real-time needs of online service.

Algorithm 2 Category-based Activity Scheduling

Input: ***needs***; ***ct***; ***PI***; ***matrix***;***k***
Output: Tuple(POI[n], Path[n-1],Cost):***scheduling***
1: **for all** need ***needs[i]*** **do**
2: ***CategorySet cs[i]=ct.subCategory(needs[i].category)***;
3: get top-k famous POIs ***POIs[i][k]*** belongs to ***cs[i]***;
4: **end for**
5: **for all** composition ***POI[n]*** built by geting one elements from each column of ***POIs[i][k]*** **do**
6: **for all** $i < n-1$ **do**
7: build ***Path[i]*** using ***POI[i],POI[i+1]***;
8: ***cost=CostFunction(matrix.pathCost[i])*** ;
9: **end for**
10: ***scheduling.add(POI[n],Path[n-1],cost)***;
11: **end for**
12: ***scheduling.sortBy(cost)***;
13: return ***scheduling***;

2. Over-fitting of single object: when optimization object is to get least time cost and k is set to a large value (which means more candidate locations), the algorithm may be over-fit to the single object, causing that the popularity of locations are ignored, and the final recommendation will be a group of locations geographically assembling in a small area.

Thus we conclude that the itinerary planning recommendation problem should be treated as a multi-objective optimization problem, which do not strictly require the optimized result; the approximate optimization result is acceptable. In the following section, we propose a multi-objective optimization algorithm, which aims at finding approximate optimizations in a short time.

4.3 Ant Colony Optimization (ACA) based multi-objective itinerary planning

Multi-objective optimization problem is very common in scientific research and engineering practice. In general, it has a final object which is composed of several objects. This kind of optimization is usually high-dimensional and with large scale, and need to consider the weight allocation for different sub-objects. However, we cannot precisely design the weight allocations for different sub-objects of the problem in this paper. For example, it's hard to determine which of the two, the travel time and popularity of the locations, is more important.

Fortunately, ACA, Simulated Annealing, Genetic Algorithm and Neural Network and other intelligent algorithms have been developed and applied extensively. This paper proposed a modified version of ACA to solve the multi-objective optimization of itinerary planning recommendation. The modified ACA mainly focus on the weight allocation for different sub-objects. Furthermore, it is convenient to adjust the number of ants and running times according to the system load, improving the service of the system.

4.3.1 Build up the ant colony system

ACA is a bionic algorithm [4]; its main idea is to simulate the food seeking behavior of the ants. It initially put a large number of ants that randomly wander in the search space, once an ant found the "food", it put some pheromone along the path to increase its attraction to other ants; be-

sides the pheromone evaporates with time goes on. Such a positive feedback strategy leads the algorithm to a global optimization. ACA is intrinsically parallel, which makes it to be easily programmed in parallel. This paper introduces in ACA and makes the following modification: keep the global optimization object unchanged, i.e. finding the shortest path; introduces in a heuristic rule, that merges the popularity of locations into the pheromone updating phase, making the ants tend to choose the locations with higher popularity. Activity scheduling is denoted as a ordered list with elements from n non-intersecting sets.

$$path = \langle S_1, S_2, S_3, \dots S_n \rangle$$

P_{ij} is the $j-th$ POI of $i-th$ set. Here we support every sets include K POIs. Every POI in a set existing a edge, $E(P_{ij}, P_{(i+1)l})$, between all POIs of neighborhood sets.
In ant colony system, we use follow annotations:

$A^t(k)$ —describe the state of the kth ant in t times visit.

$D(P_{ij}, P_{(i+1)l})$—describe the minimal delay between two POIs.

$I^t(P_{ij}, P_{(i+1)l})$—describe the information between two POIs of t times visit.

$P_k^t(P_{ij}, P_{(i+1)l})$—describe the probility of the kth ant transfer form P_{ij} to $P_{(i+1)l}$.

$H(P_{ij})$—describe the famous value of a POI.

4.3.2 Itinerary planning based on modified ACA

The procedure of modified ACA.

1. Initialization, randomly put each ant on locations of the first set. (Algorithm 3, line 1-4)
2. The ants traverse the n sets in order, the transfer probability between two cities is as below. (Algorithm 3, line 6-10)

$$P_k^t(P_{ij}, P_{(i+1)l}) = \frac{I^t(P_{ij}, P_{(i+1)l}) H(P_{(i+1)l})}{D(P_{ij}, P_{(i+1)l})} \quad (1)$$

3. After all the ants have completed their travel, calculate the total cost of each path, save the one with the smallest cost. In the meantime, evaluate the current state and determine whether the ending condition is satisfied. If it is satisfied, return the best path; else, update the pheromone of each edge. When the ant completes a round, the pheromone of each edge is changed according to the following equation. (Algorithm 3, line 11-18)

$$\Delta I_k^t(P_{ij}, P_{(i+1)l}) = \begin{cases} \frac{Q}{L_t^k} & if A^t(k) cover E(P_{ij}, P_{(i+1)l}) \\ aa & else \end{cases} \quad (2)$$

$$I^{t+1}(P_{ij}, P_{(i+1)l}) = I^t(P_{ij}, P_{(i+1)l})(1-\gamma) + \sum_{k=1}^{m} \Delta I_k^t(P_{ij}, P_{(i+1)l}) \quad (3)$$

4. Re-put all the ants, start a new period. (Algorithm 3, line 19)

According to the procedure described above, the pseudo code of the modified ACA is given below.

Algorithm 3 Ant Colony based Activity Scheduling

```
1: for all edge, ant do
2:    I^1(P_ij, P_(i+1)l) = I_initial;
3:    Place ant on a randomly choose POI of S_1 ;
4: end for
5: Let Path_minDelay be the best path and L_min its delay;
6: for t = 1 to t_max do
7:    for k = 1 to m do
8:       Build tour Path_k^t by applying n-1 times the following step;
9:       Choose next POI with probability computed by Formula (1);
10:   end for
11:   for k = 1 to m do
12:      Compute the delay L_k^t of Path_k^t produced by ant k;
13:   end for
14:   if an faster path found then
15:      Update the Path_minDelay and L_min;
16:   end if
17:   for all edge do
18:      Update the I^(t+1)(P_ij, P_(i+1)l) by Formula (3);
19:      Replace ant on a randomly choose POI of S_1 ;
20:   end for
21: end for
22: Return the Path_minDelay and L_min;
```

5. DEMO SYSTEM AND EVALUATION

Foursquare is the largest location based social network, which has more than 20 million active users. We have built an experiment system based on Foursquare and T-drive [12] dataset for Beijing. Information about the experiment platform and dataset is listed as below:

- System configuration: Our experiments are conducted on a quad-core 2.27GHz Intel i3 CPU with 2G RAM and a 320G 5400 RPM disk driver. Disk page size is 4KB (4096Byte). OS is Ubuntu 11.04 (Linux 2.6.16).
- Public transportation stops data: We collected 10684 bus stops from the Beijing public transport network, which covers all the urban and suburb area.
- Foursquare dataset: We collected 30,784 effective POIs and category information by calling open API provided by Foursquare Company. Every POI has a variety of information, such as total "sign in" times, number of historical visitors, user submitted reviews, website address and so on. In order to find all the candidate POIs the belongs to the same categorie, we built Inverted index [7] for every category and sort each POI list by "sign in" times.
- GPS trajectory dataset: Real dataset is provided by the T-drive project of Microsoft Research Asia, which includes all the trajectories of 10357 Taxis from 2008-02-02 to 2008-02-08 in Beijing.

Table 1 lists the recommendation results, computed by the baseline algorithm and modified ACA with k=3 and k=6 respectively, for example 1. The user interface is as shown in

Table 1: Itinerary recommendations for Example 1 in Beijing City

Algorithms	Activity scheduling	Time	Evaluation
k=3 (Baseline)	McDonald's(143)− >(6 Mins)Forbidden City(2914)− >(9 Mins) Beijing Roast Duck Restaurant (720) − >(5 Mins)Jianwai SOHO(277) − >(4 Mins)Sanlitun Village(5688) − >(5 Mins)CJW The Place(167)	25 Mins	8.1
k=6 (Baseline)	McDonald's(143)− >(3 Mins)East Gate: Temple of Heaven(193)− >(3 Mins) Duck de Chine(287) − >(3 Mins)Jianwai SOHO(277) − >(2 Mins)Silk Street Market(1017)− >(2 Mins)CJW The Place(167)	11 Mins	7.4
k=3 (Ant colony)	McDonald's(143)− >(6 Mins)Forbidden City(2914)− >(9 Mins) Beijing Roast Duck Restaurant(720) − >(11 Mins)Tian'anmen Square(4908) − >(4 Mins)Joy City(1933)− >(2 Mins)Cepe(94)	32 Mins	8.9
k=6 (Ant colony)	McDonald's(143)− >(6 Mins)Forbidden City(2914)− >(9 Mins) Beijing Roast Duck Restaurant(720) − >(5 Mins)Jianwai SOHO(277) − >(0 Mins)The Place(1798)− >(5 Mins)Enoterra(338)	25 Mins	8.2

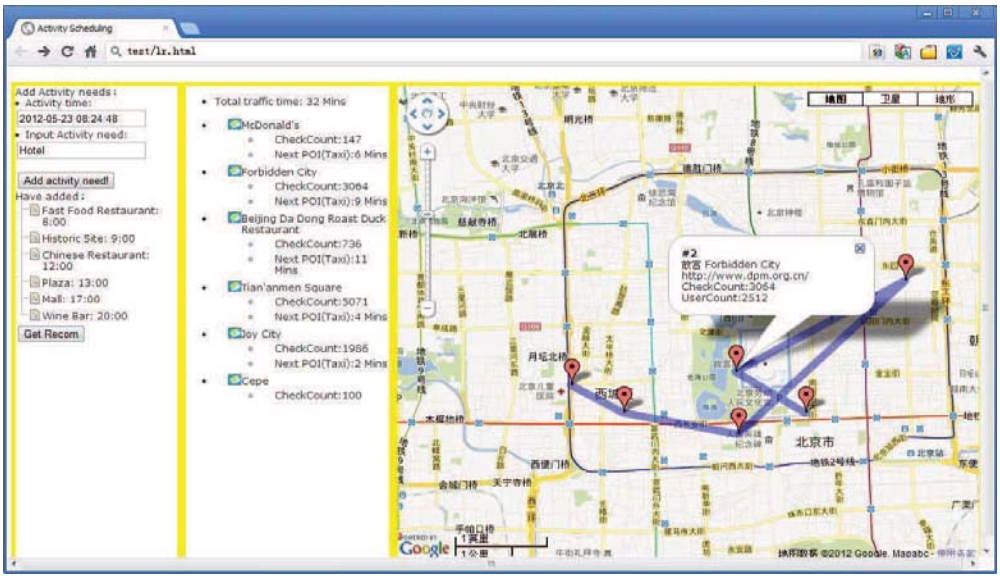

Figure 5: User interface.

Fig. 5. The recommendation result includes the location of each activity and the popularity indicated by "sign in" number (shown after POI name). What's more, users can click the link to browse homepage of every POI and information on foursquare website, which contains reviews and evaluation of other users. At the same time, recommendation results are drawn in GIS map and the shortest driving time between two activities has been given (the number in brackets, in minutes). From the above information, user can get a full understanding of the recommended itinerary.

The evaluation value in table 1 is the average value of all the evaluations (a value within 1-10 is used to indicate their degree of satisfaction) made by the 14 volunteers belong to class A and B in the questionnaire. The satisfaction degree of all the four recommendations are above 7, indicating that the results have good user acceptance and important reference value. At the same time, we find that recommended results of ant colony algorithm, though with larger time cost, are better accepted by users than the baseline algorithm. Something interesting is that, when there are more candidate locations (larger k) taken for recommending, the satisfaction degree decreases. It shows that, compares to several minutes' additional cost, the users care more about the popularity. Next step, we will provide online services, which is more sensitive to the recommendation algorithm's performance. In the next subsection, the time cost of the two algorithms are evaluated.

6. RELATED WORKS

This section lists the related studies related to our work and points out their difference with our work:

- GeoLife project of Microsoft Research Asia collects 165 users' trajectory data from 2008 to 2010. They have developed many interesting applications, such as the traffic navigation system [13], and studied the relations between frequently visited locations and the users. Their discovery can answer questions such as "What's the most popular tourist attractions of Beijing city?", "where a tourist, who has gone to the Imperial Palace, will be?", "which is the most frequent route

to a certain place?" and so on[15]. They also provides smart itinerary recommendation services,which use simplified query composed of start point, end point and duration to get a complete set of itinerary is automatically generated based on real user-generated GPS trajectories[10]. At the same time,they also implements a personalized recommendation that provides an individual with locations matching her travel preferences[14].However,they have not provided a intention oriented itinerary recommendation service.

- The research of [5]tries to find out trajectory pattern and analyze the life style of people. The work associates each of the social network users with a specific geographic location to build a spatial social network graph. Their main contribution is that they proposed a serial of operators to the query social networks and spatial networks together, and have implemented them based on both relational and graph databases. The work has inspired us to combine spatial and social networks for exploring new application and technology.

- The classic scene of spatial keyword query is to give a physical location and a set of keywords to find a single object which best matches the input keywords with minimal distance [1, 6]. The research of [2] expands spatial keyword search, proposes and realizes the search for a set of objects, which together matches the input keywords with minimum space distance among all the objects. The applications' scenarios of this kind of query are restricted for not fully using all kinds of background knowledge from other information sources. Our work can use collective intelligence to find out more useful locations and give a reasonable travel path for the recommended itinerary.

- The work of [8] supports the trajectory retrieval using k query points. This work requires the user to input k accurate location points, and then finds out trajectories pass through or is nearest to all the input points. However, it doesn't support category based queries and can not guarantee that the result trajectory is shortest or with minimum travel time.

7. CONCLUSION

By jointing the data from social network with physical trajectories, and take usage of hierarchical categories of geographical locations, this paper realizes recommendation for itinerary planning. Based on the sematic traffic information contained in the historical trajectories, the recommendation algorithm gives the best path along multiple points, which satisfies the user's demands better than the shortest path searching algorithm. Meanwhile, the recommendations are presented with related reviews about the recommended locations on social network, effectively help the user to understand the final recommendations. Therefore, the recommending results are meaningful for the users to plan their itinerary.

8. ACKNOWLEDGMENTS

Thank Dr. Xing Xie from Microsoft Research Asia (MSRA) who gives me a lot of guidance and his team, who provides me with the Taxi trajectory data of Beijing.

9. REFERENCES

[1] X. Cao, G. Cong, and C. S. Jensen. Retrieving top-k prestige-based relevant spatial web objects. *PVLDB*, 3(1):373–384, 2010.

[2] X. Cao, G. Cong, C. S. Jensen, and B. C. Ooi. Collective spatial keyword querying. In *Proceedings of ACM International Conference on Management of Data, (Athens, Greece, June 12-16,2011)*, pages 373–384, 2011.

[3] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars. *Computational Geometry: Algorithms and Applications(Third Edition)*. Springer-Verlag., Heidelberg, 2008.

[4] M. Dorigo and L. M. Gambardella. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Trans. Evolutionary Computation*, 1(1):53–66, 1997.

[5] Y. Doytsher, B. Galon, and Y. Kanza. Querying geo-social data by bridging spatial networks and social networks. In *GIS-LBSN*, pages 39–46, 2010.

[6] I. D. Felipe, V. Hristidis, and N. Rishe. Keyword search on spatial databases. In *ICDE*, pages 656–665, 2008.

[7] M. Patil, S. V. Thankachan, R. Shah, W.-K. Hon, J. S. Vitter, and S. Chandrasekaran. Inverted indexes for phrases and strings. In *SIGIR*, pages 555–564, 2011.

[8] L. A. Tang, Y. Zheng, X. Xie, J. Yuan, X. Yu, and J. Han. Retrieving k-nearest neighboring trajectories by a set of point locations. In *SSTD*, pages 223–241, 2011.

[9] Z. Yan, D. Chakraborty, C. Parent, S. Spaccapietra, and K. Aberer. Semitri: a framework for semantic annotation of heterogeneous trajectories. In *Proceedings of the 14th International Conference on Extending Database Technology, (Uppsala, Sweden, March 21-24, 2011)*, pages 259–270, 2011.

[10] H. Yoon, Y. Zheng, X. Xie, and W. Woo. Smart itinerary recommendation based on user-generated gps trajectories. In *UIC*, pages 19–34, 2010.

[11] J. Yuan, Y. Zheng, X. Xie, and G. Sun. Driving with knowledge from the physical world. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, (San Diego, CA, USA, August 21-24, 2011)*, pages 316–324, 2011.

[12] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang. T-drive: driving directions based on taxi trajectories. In *ACM-GIS, (San Jose, CA, USA,November 3-5, 2010)*, pages 99–108, 2010.

[13] Y. Zheng, Y. Chen, X. Xie, and W.-Y. Ma. Geolife2.0: A location-based social networking service. In *Mobile Data Management*, pages 357–358, 2009.

[14] Y. Zheng and X. Xie. Learning travel recommendations from user-generated gps traces. *ACM Transactions on Intelligent Systems and Technology*, 2(1):2, 2011.

[15] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma. Mining interesting locations and travel sequences from gps trajectories. In *WWW*, pages 791–800, 2009.