

Research Article

Open Access

Shivendra Tiwari* and Saroj Kaushik

Popularity estimation of interesting locations from visitor's trajectories using fuzzy inference system

DOI 10.1515/comp-2016-0002

Received July 06, 2014; accepted December 04, 2015

Abstract: Identifying the interesting places through GPS trajectory mining has been well studied based on the visitor's frequency. However, the places popularity estimation based on the trajectory analysis has not been explored yet. The limitation in the majority of the traditional popularity estimation and place user-rating based methods is that all the participants are given the same importance. In reality, it heavily depends on the visitor's category, for example, international visitors make distinct impact on popularity. The proposed method maintains a registry to keep the information about the visited users, their stay time and the travel distance from their home location. Depending on the travel nature the visitors are labeled as native, regional and tourist for each place in question. It considers the fact that the higher stay in a place is an implicit measure of the greater likings. The weighted frequency is eventually fuzzified and applied rule based fuzzy inference system (FIS) to compute popularity of the places in terms of the ratings $\in [0, 5]$. We have evaluated the proposed method using a large real road GPS trajectory of 182 users for identifying the ratings for the collected 26807 point of interests (POI) in Beijing (China).

Keywords: location based services (LBS); trajectory databases; trajectory mining; region of interest (ROI); place ratings; popularity estimation; fuzzy inference system

1 Introduction

In recent years, research on endorsement of urban Point Of-Interest (POI), such as restaurants, tourist attractions

based on social information have attracted a lot of attention. Due to the needs of effective improvements in the tour plans tour-guide applications need to be aware of the places popularity and ratings. The day-to-day life needs to find the suitable locations based on different aspects where popularity being one of the major parameters. There are many categories of the locations popularity based on their geographical size, durability and mutability behavior. A location is known as an event based popular location if it is popular based on limited period events being organized, happened naturally or held accidentally. The tourist locations or monuments are the places which are generally in constant attention and known because of their longevity and historical existence. The trending locations are mainly the news that are associated with some region, but not necessarily has an exact location to point. For example, some political decisions in a country has importance at the overall country level. The locations that are only known to the particular geographical region is known as locally popular location i.e. ATMs, schools, medical clinics, restaurants etc. On the other hand the globally popular location are known to the world. A location that is visited by some certain set of people is known as routine locations i.e. residential apartments, offices. The popularity cannot be absolutely defined solely in terms of its trend, ancientness, routine oriented, event oriented, known to a particular range of people etc. All of these categories overlap with each other in some sense. A location can neither be absolutely popular nor it can be entirely non popular. Linguistically, we call them as low-popular, not so popular, very popular etc.

Most of the point of interests (POI) are rated by their visitors to express their vote for the likings, services and niceness of the place that contributes on measuring their popularity. However, there are multiple limitations with the current rating systems. *First*, lack of digitization – the places need to be digitized before the user can caste a rating using hand-held devices or the web. There are millions of the places across the world that have not been even digitized. Only less than 0.5% place have any rating in a set of 26807 places in Beijing that we have collected using freely available latest mapping applications. *Second*, averaging – the user based manual ratings suffer with the insufficient sampling to represent the actual rating based on

*Corresponding Author: **Shivendra Tiwari:** Department of Computer Science and Engineering, Indian Institute of Technology Delhi Bharti Building, Hauz khas, New Delhi, 110016, India, E-mail: shivendra@cse.iitd.ac.in

Saroj Kaushik: Department of Computer Science and Engineering, Indian Institute of Technology Delhi Bharti Building, Hauz khas, New Delhi, 110016, India, E-mail: saroj@cse.iitd.ac.in

the average. *Third*, no consideration for the visitor's category – all the users are treated as equal whoever participates in the rating, although, the rating and popularity of a place mostly depends on the kinds of visitors i.e. a popular and global person's visit increases a place's importance and hence popularity. *Fourth*, prone to proxy rating submission – since there is no requirements of being present physically and spend time before the participation on the rating process, it is a risk for the proxy ratings submission. In order to counter the problems in the current rating systems, we propose to use trajectory mining methodology to identify and estimate the popularity for the locations.

The increasing availability of GPS-enabled devices brings us a large amount of GPS trajectories representing people's location histories. The pervasiveness of ubiquitous technologies guarantees that there will be an increasing availability of large amounts of data pertaining to individual trajectories, at increasing localization precision [1]. A trajectory is a sequence of sampled locations and time stamps along the route of a moving object. The analysis of such trajectory data is a critical component in a wide range of research and decision-making fields. However, it is a challenging problem to analyze and understand patterns in massive movement data, which can easily have millions of GPS point locations and trajectory segments described in [2]. Spatio-temporal patterns that succinctly show the cumulative behavior of a population of moving objects are useful abstractions to understand mobility-related phenomena [1]. Trajectory pattern represents a set of individual trajectories that share the property of visiting the same sequence of places with similar travel times. Therefore, two notions are central: (i) the regions of interest in the given space, and (ii) the typical travel time of moving objects from region to region. They are a spatio-temporal variant of the Temporally-Annotated Sequences (TAS) [3].

The general trajectory behavior include the sequence events both is spatial and temporal space. The user stays in a region and moves ahead to approach another regions known as Region of Interest (ROI). An ROI can be an individual Point of Interests (POI) at the lowest level or a group of POIs, or an administrative region such as city, district, villages etc. The ROIs are created using different methods as proposed in [4]. The ROI identification methods include pre-conceived ROI, density based and trajectory based approaches. The pre-conceived ROI approach has subjective background knowledge that is used to specify a set of point of interest, which are known as movement attractors. The density based approach creates cluster of the popular locations to create bigger regions. The trajectory pattern min-

ing algorithm is used for identifying the ROIs dynamically with the mining of sequence's temporal information.

Sometimes the GPS location is far from the actual road and hence it is important to project them on the road or on the POIs in order to carry about accurate computations. We have proposed to use map matching techniques to pre-process the GPS trajectory before actually estimating the popularity of the locations. Map Matching, is the process of projecting the GPS fixes on the road network graph $G = (V, E)$. It is classified generally into real-time and post-processing map-matching. The real-time map-matching captures the location of a traveler in the road network with a real-time feed of GPS locations. Post-processing map-matching takes GPS data recorded from a travel and matches it to the road network to trace the routes taken by travelers [5]. In this paper, we have used post-processing map matching in order to project the inaccurate trajectories on the road network.

Our contribution in this paper includes – *First*, proposed a method to estimate the popularity of the geospatial locations. We introduce the notion of "visitor's category" in order to differentiate the visiting travelers to a location. It has been used to identify if the location is a locally or globally popular. *Second*, proposed method has established a correspondence between the trajectories based popularity to the user ratings $\in [0, 5]$. *Third*, our method exposes the possibility to specify the timeline based popularity as some places are popular only in some certain seasons in the year, for example, Himalayan tourists locations are mostly closed in the winters. *Finally*, we demonstrate and evaluate the method with the real trajectory data-set GeoLife provided by Microsoft¹ and the HERE maps POIs² in Beijing (China). Apart from this section as introduction, Sections 2 and 3 talk about the related work in the area and the problem definition respectively. The data modeling and pre-processing of the trajectory databases is discussed in Section 4. Section 5, and 6 establish the visitor's registry management, estimating the popularity and experiments by implementations separately. *Finally*, the Section 7 includes the conclusion and future work.

¹ GeoLife GPS Trajectories – Microsoft Research: <http://research.microsoft.com/en-us/downloads/b16d359d-d164-469e-9fd4-daa38f2b2e13/default.aspx>

² HERE – Maps for Life: <http://here.com/>. Last accessed on June 13th, 2014.

2 Related work

There have different methods been developed for trajectory and movement analysis. In general, most trajectory analysis methods involve the two steps i.e. simplify and generalize each trajectory; and compare and group trajectories to find general patterns. The simplification or generalization of trajectories involves several different aspects, i.e. the complex routes or geometric shapes of trajectories may need simplification; or they can further be partitioned into sub-trajectories and subsequent analysis can focus on sub-trajectories [2]. Lammerena et al in 2010 presented the results of user's usability test, and the tester's experiences through a location-based, partly route-dependent application. It allows the user to record, store and upload their own experiences using multimedia contents. The stored data can be shared with others when they visit the same locations. Results of the test show the technology acceptance of the respondents, impact on their experiencing of the environment and their personal behavior [6].

Yan in [7] proposed semantic trajectory analysis based on the statistical computation and semantic concepts. It involves three major perspectives, i.e. trajectory modeling, trajectory computing, and trajectory pattern discovery. They surveyed three types of modeling requirements for comprehensively explaining trajectories, in terms of geometric knowledge, geographical knowledge, and application domain knowledge. Zenger in [8] proposed a new framework for trajectory-based POI recommendation. The method constructs a k -truncated generalized suffix tree to represent a historical trajectory database, and use it to execute exact matching recommendation queries. Two variants are developed, allowing for the execution of fuzzy matching and order-flexible queries.

Kang in [9] suggested method to mine the spatio-temporal pattern in the trajectory data – it first finds meaningful spatio-temporal regions and extracts frequent spatio-temporal patterns based on a prefix-projection approach from the sequences of these regions. They experimentally analyzes that the proposed method improves mining performance and derive more intuitive patterns. Lee in [10, 11] proposed a trajectory clustering method based on the partition and group framework. They established the importance of discovering the common sub-trajectories in many applications, especially if we have regions of special interest for analysis. The new framework partitions a trajectory into a set of line segments, and then, group's similar line segments together into a cluster. The primary advantage of the framework is to discover common sub-trajectories from a trajectory database. Based

on the partition-and-group framework, they developed a trajectory clustering algorithm based on partitioning and grouping in order to discover the common sub-trajectories from a trajectory database.

Patel in [12] utilized stay duration and region association information available in trajectory data during feature generation. The features are generated using spatial distribution, duration and region association information of trajectories. Two types of features, region rules and path rules, are generated from trajectories for classification. Region rules consider the spatial distribution of trajectories, the time spent (duration) by the trajectories in the region and the association information with other regions. Path rules differentiate objects based on their traveling patterns and speed. Efficient algorithms are devised to obtain region rules and path rules. Based on the discovered rule, trajectory classification model is built to predict the class label of new trajectory. However, Dalumpines in [5] offered post processing based approach for the map matching algorithm in order to project the inaccurate GPS trajectories on the road network using the geometric, buffer, and network functions in a GIS software. The algorithm also generated relevant route attributes such as travel time, travel distance, and number of left and right turns that serve as explanatory variables in route choice models.

A hierarchical graph based method for mining the interesting locations and travel sequences from GPS trajectory databases has been discussed in [13]. They model multiple individual's location histories with a tree-based hierarchical graph (TBHG). They further defined an inference model, which regarded an individual's access on a location as a directed link from the user to that location. The model infers the interest of a location by taking into account some factors i.e. users travel experience; mutual reinforcement relationship between travel experience and location interest. The method eventually mined the classical travel sequences among locations considering the interests of these locations and travel experiences. Later in 2010 authors suggested supervised learning based approach to infer people's motion modes from their GPS logs in [14]. They also introduced a social networking service, called GeoLife¹, which aims to understand trajectories, locations and users, and mine the correlation between users and locations in terms of user-generated GPS trajectories. GeoLife offers three key applications scenarios: sharing life experiences based on GPS trajectories; generic travel recommendations, e.g., the top interesting locations, travel sequences among locations and travel experts in a given region; and personalized location recommendation [14]. We have used trajectory database from the GeoLife project for the experiments in this paper. We have

used fuzzy logic to fuzzify the values of places attributes that can be used by the rule based fuzzy inference system to estimate the popularity of the locations. Zadeh introduced Fuzzy set in 1965 [15] to represent data and information processing, non-statistical uncertainty.

3 Motivation and challenges

The recommendations of the places have been one of the major features of the tourist and mapping solutions. They address the problem of filtering information that is likely of interest to individual users. Typically, these systems consider the ratings for the locations by the similar profile visitors. However, the most of the places are un-rated. The averaging based rating systems suffer from the insufficient sampling and proxy rating submission issues. It motivates us to discover an alternate solution that does not require active participation of the users. The next generation rating method should consider the user's activities anonymously and compute the ratings from the implicit user liking indications. The ratings of the broader regions (i.e. ROIs) are completely unavailable. Estimating the popularity for the broader regions such as villages, forest reserves etc. is another source of motivation. Averaging, median or mode are the probable solutions known so far for estimating ROI's popularity. Hence the regions rating also carry the shortcomings of the location ratings. Our method can be used to record new places in the remote areas with the minimum information of reverse-geocoded address, geo-coordinates and the popularity. It can be easily extended to add POI category and other details using different POI collection methods that is out of the scope for our work.

The major challenges in the trajectory based popularity estimation include – disconnected and broken trajectories, routine visits of the places, uneven stay time at different places, heterogeneous shapes and sizes of the regions etc. Some of the challenges are summarized as below:

1. **Inaccurate trajectories:** The points are sometimes inconsistent, off the road and unequal sampled. The trajectory segments need to be accurate in order to compute the travel distances. The locations where there is no digitized map available, it cannot be map matched. The trajectories are not sufficiently sampled and hence they require adding the new points based on their speed and interpolation among the consecutive GPS points.
2. **Disconnected and broken trajectories:** The devices run out of batteries and therefore the trajectories are broken. The navigation systems are normally closed

after reaching the destination. It is difficult to keep the GPS loggings ongoing all the time for any logger application. We might need to keep the trajectories joined once they resume the movement in case the travel restarts from the last ended location.

3. **Heterogeneous nature of the geospatial regions:** They are of different shapes, sizes, geographic conditions that leads to change the trajectory pattern. These differences makes it difficult to identify the correct shape and the size of the interesting regions.
4. **Category of the interesting regions:** There are places that could be routine for some certain visitors; however at the same time it may be a tourist visit for others. For example, a historical monument is visited by its management staff regularly; however, the other visitors do not visit so often. The stay-time is uneven for each kind of visitors. It is also different for the different places depending on the categories and nature of the place itself.
5. **Identification of user categories:** The popularity of the place also depends on 'who' visits the underlying place i.e. visitors from the same locality as compared to the visitors from across the world. Defining the correct user category is difficult as it depends on how accurately the trajectory travel pattern is identified.
6. **Verification of popularity result accuracy:** The available trajectories do not cover all the visitors of the places, for example, the people who do not use such tracking devices, cannot participate in this method. The devices are also kept switched off when the users are traveling to some private places. Therefore, it is difficult to verify and prove the accuracy of the trajectory based popularity estimation.

4 Problem definition

Given a set of locations $\mathbf{L} = \{L_1, L_2, L_3, \dots, L_n\}$ where L_i is a point or a polygon; and a trajectory database $\mathbf{T} = \{T_1, T_2, T_3, \dots, T_m\}$ where T_i is a ordered set of geospatial coordinates along with the time stamps. The objective is to estimate the popularity $\mathbf{P} = \{P_1, P_2, P_3, \dots, P_n\}$ corresponding to the each given location, where $P_i \in [0, 5]$. There are various attributes which indicate the popularity and the niceness of a location such as user's visit frequency, crowd strength, stay time and so on. We start defining the problem statement with the definition of some terminologies, which will eventually be used in the process of popularity estimation.

Definition 1. Location or Place: A location L is a geo-spatial point (or a polygon in case it is a region) with a of shape coordinates and non-spatial properties i.e. $L = \langle \langle p_1, p_2, p_3, \dots, p_n \rangle, C, I \rangle$, where $p_i = \langle x_i, y_i, z_i \rangle$ is the spatial coordinate, C is category of the place and I is the time interval when it is allowed to access the location.

Definition 2. Trajectory: A trajectory or a spatio-temporal sequence is an order $T = \{T_1, T_2, T_3 \dots T_m\}$, where, $T_i = \langle \langle p_1, t_1 \rangle, \langle p_2, t_2 \rangle, \langle p_3, t_3 \rangle \dots \langle p_k, t_k \rangle \rangle$; $p_i = \langle x_i, y_i, z_i \rangle$; $t_i \{i = 0 \dots k\}$ is a time-stamp, for all $\forall_{0 \leq i < k} t_i < t_{i+1}$; and $p_i \in R^3$.

Definition 3. Moving Trajectory: A trajectory T or an adjoining maximal sub-trajectory of T are called moving trajectories if they meet any of the four conditions:

- It is between the two successive halts of T
- It is between the start point of T , and the initial stop of T
- It is between the last stop and the last point of T
- If T has no stops (then trajectory T itself is a moving trajectory)

In other words, the individual points of a trajectory, which do not belong to a stay region, it therefore belong to a motion region known as moving trajectory. Unlike stay region, the motion trajectories have no minimal time threshold, they may interconnect or not an aspirant stop.

Definition 4. User Check-in: A user u in a location L is said to be checked-in at a time t_k if u stays until t_m at the same location so that $Diff(t_m, t_k) \geq \partial_{temporal}$, and $Dist(p_m, p_k) \leq \partial_{spatial}$. Here, p_m and p_k are the spatial coordinates at time t_m and t_k respectively; $\partial_{temporal}$ is the given temporal threshold and $\partial_{spatial}$ is spatial threshold.

Definition 5. User Checkout: A user u from a location L is said to be checked-out at a time t_r and position p_r , if u was already checked-in at the same location at time t_k such that, $p_k, p_r \in L$ and $p_{r+\phi} \notin L, \forall t_r \geq t_k$. Where $p_{r+\phi}$ is a coordinate at a time $t_{r+\phi}$ subsequently after the checkout time and position.

Definition 6. Noise Checkout: A user u from a location L is said to have a noise checkout at a time t_n and coordinate p_n , if it was already checked-in at the same location at time t_k , and $p_{n-i}, p_{n+j} \in L$ and $p_n \notin L \wedge Diff(t_{n+i}, t_{n-j}) \leq \partial_{noise}$.

Here, t_{n+i} , and t_{n-j} are the timestamps for the GPS positions p_{n+i} , and p_{n-j} respectively; ∂_{noise} is a threshold for the noise checkout time interval. It means that if the user has inaccurate GPS signals or moves out of the POI

premises for a small interval and comes back within a period, it is still assumed as checked-in.

Definition 7. Home Location: The home location H_u for the user u is a geospatial region within the given trajectory database T which has been visited more than any other identified locations in set L . In other words, $H_u = \{\ell | \forall \ell, \ell' \in L_u \wedge L_u \subseteq L \wedge frequ(\ell) \geq argmax_{\ell'} frequ(\ell')\}$; where L_u is the set of locations visited by the user u and $frequ(\ell)$ is the visitor's frequency in the underlying place for the user u .

Definition 8. Visitor's registry: A visitor's registry v is a set of tuples V of the visitor's attributes. More formally, $v = \{V | V = \langle U_{id}, H_u, T_{arrival}, T_{departure} \rangle \wedge \forall T_{arrival} < T_{departure} \wedge |T_{departure} - T_{arrival}| \geq \partial_{temporal}\} \wedge H_u \in R^3$.

Here, U_{id} is given unique identification for the users. H_u is the home location for the user visiting the place p_i . The check-in process takes the user's trajectory road network graph and set of POIs, and the region sets visited by the user.

Definition 9. Trajectory Based Popularity: Given a trajectory database T and a set of locations with their polygonal shapes L . Popularity $P = \{P_1, P_2, P_3, \dots, P_n\}$, $P_i \in [0, 5]$ corresponding to the each given location. The popularity of a location is a real number which is computed based on the visitor's registry attributes like user category and stay time, and visit frequency.

In this paper, we have used fuzzy rule based inference system to categories the user based on their travel distance. The stay time at a location is also used as a factor of popularity – it has been observed that the mostly people spend more time in their favorite places. The stay time has been categorize as low, medium and high based on the fuzzy membership functions for the different categories which finally influence the ratings computation.

5 Proposed solution

The proposed solution includes mainly three major steps i.e. pre-processing of the raw trajectories, user check-in and popularity estimation. Figure 1 shows the high-level component and flow of the trajectory data processing the popularity estimation. The pre-processing module involves improving the trajectory to make it usable. The pre-processing step however can be skipped in case the underlying trajectory has high quality GPS sequences. The user check-in step includes the stay region/place determination

and recording the stay duration in the individual places, known as visitor's registry. Based on the visitor's log registry a home location is identified from all the visited locations by each user. The popularity estimation step is responsible for fuzzifying the user check-in attributes w.r.t. the individual places and finally computing the rating as discussed in the further subsections.

5.1 Data modeling and pre-processing

In this section, the pre-processing of the trajectory data is carried out based on the existing map matching and interpolation techniques [2, 5]. The trajectories and given set of POIs are stored in the spatial grid data structures after pre-processing. The individual user trajectories do not store the complete details of the road attributes; however they have only the reference to the links in order to maintain the re-usability and smaller memory usage. The pre-processing module involves improving the trajectory to make it usable – this step however can be skipped in case the underlying trajectory has high quality GPS sequences.

5.1.1 Map matching and road network generation

The process of matching the trajectory geo-coordinates with the existing road networks is known as map matching. However, there are still regions (mostly in the country side rural areas) where the roads, polygons, POIs are yet to be digitized and hence it is not possible to directly use the map matching algorithms for creating the trajectory graphs. We need to explicitly define the methods to detect the intersections and nodes of the graph. The line segment intersection algorithms are useful on such formulations. The process of extracting the map from the trajectory databases is named as map generation. We have discussed the map generation in short that is inspired from the work in [5].

5.1.1.1 Map Matching

The map matching, can be viewed as a problem of "matching", i.e., finding similarity between two graphs. It is characterized by two objectives – identify the link traversed by the traveler and find the actual location of the noisy GPS fix within that link. The post-processing map matching is process of matching the trajectory database offline when the user has already completed his travel with the road network map. Road network map and GPS data are often enough for post processing map matching. The kind

and nature of these data inputs and the purpose of the algorithm largely influence the development of the map-matching procedures. The shortest path algorithm can be appropriately used for post-processing map-matching. Map matching plays an important role on putting the user either on the road or in a region based on the users location. It is possible that the user stays at the POIs for some time and hence there might not be the road at all. In such cases, the map matching algorithm has to put the user on the POI so that it can be concluded that the user indeed visited the POI. The stay point finding method includes task of map matching in case there is inaccuracy; however it also finds the stay regions where the traveler has spent some time. The input to the map marching is the road network, trajectory database; however the output is the accurate GPS sequence that guarantees the point being exactly on the road or in some geographical location (if there is a stay). The map matching algorithm is defined by the definition 10.

Definition 10. Map Matching: It is a process of finding a correspondence R_c between vertices of trajectory T and location on street network G such that the two matched sets V and R_c , minimize an objective criteria C_o . The criteria C_o is to match the points in T to the nearest link using only the geometric relationships when the point is the first GPS point or the distance between the previous point P_{t-1} and the present point (P_t) is too long. When the distance between P_{t-1} and P_t is under a particular threshold $\partial_{distance}$, evaluate the proximity and the orientation between the reference line which connects these two points and the segment (S_{t-1}) to which the previous GPS point (P_{t-1}) was matched. The point P_t is matched to S_{t-1} when the evaluation criteria is met. When the evaluation criteria is not met, segments which are directly connected to S_{t-1} is evaluated through the same process [16].

5.1.1.2 Map generation

In the scenarios when the digitized geographical map is not available in the area, we need to generate the map based on the all the trajectory paths in that area. Map generation is a process of creating the road network based on the trajectory databases. Bearing in mind about the essential imprecision in the LDT's geo-coordinate determination, a spherical region is applied to cumulative geo-coordinate and to derive smaller number of representative coordinates. The size of the spherical region is derived based on the imprecision in the LDT. The process includes two steps as given below:

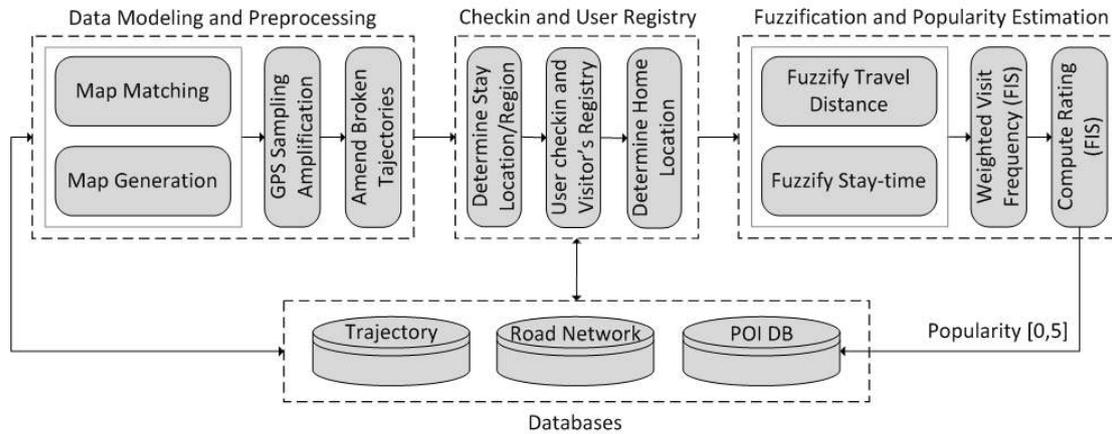


Figure 1: Overall component diagram for popularity/rating estimation.

- *First, identify the road segment* – The objective is to find out the roads segments automatically by extracting representative geo-coordinates. A moving-window is placed on each geo-coordinate, whose position is modified to the average of all geo-coordinates within the spherical region. The process brings the coordinate nearer to the road median. If geo-coordinates do not have other points within a predefined threshold distance, it is considered as real position and hence it does not get modified.
- *Second, reduce redundancy* – Select a smaller set of new positions as the representatives of the original geo-coordinates in order to decrease redundancy and the size. If there is no any other geo-coordinate within a threshold distance to a position p_i , then p_i represents itself.

The representative point based map generation method is inspired by the graph based trajectory analysis method in [2].

5.1.2 Sampling amplification

Geospatial coordinates jointly can divulge the road network as we discussed in the earlier subsection. However, the trajectory also needs to include enough sampling of the points. If the GPS is retrieved in a long time interval, then it can be heavily inaccurate while computing the other attributes such as travel speed, distance, stay time etc. Therefore, we need to insert additional sampling of the points in order to make the trajectory well sampled. The trajectory interpolation is a method of estimating such points within the trajectory line segments. The challenge in the interpolation is that this is not a linear as a straight-

line segment needs to be interpolated to shape the turns and the curves of the roads. The interpolation improves the resolution and accuracy. This method assumes that the trajectory either has highly accurate GPS fixes or it has already been map matched.

5.1.3 Amend broken trajectories

Some of the trajectories are disconnected and broken due to the devices run out of batteries, or the applications are shut down after reaching the destinations. Therefore, the trajectory recorder treat those segments as different trajectory sequences. From the trajectory analysis point of view, these sequences are separate and hence, they would lead to inaccurate results because of the missing information between end of the original trajectory and the start of the new one. Amending the trajectories might not be able to recover the information completely; however, it can connect them if they meet some certain criterion. The algorithm below describes the procedure of amending the trajectories – AmendTraject algorithm sorts the trajectories for individual users based on the time stamp of the first GPS points in the individual trajectories. Now ever trajectory for each user is traversed. If the time difference of the last point of the first trajectory t_i and the first point of the second trajectory t_{i+1} is greater than the threshold $\partial_{temporal}$ then they are not merged. Similarly, trajectories are skipped if their connecting nodes (i.e. last point of t_i and first point of t_{i+1}) are far away than the spatial threshold. In case both the thresholds are satisfied; the trajectories are merged and the original trajectories now point to the merged one so that the newly generated trajectory can be evaluated if the later trajectories can be merged further.

Data: Trajectory database for all the users.

Result: Trajectory database with merged trajectories for individual users.

```

while  $user_{u_i} \in U$  do
  Sort trajectories set  $T_{u_i}$  for user  $u_i$  by their start
  time stamp of the first GPS fix.;
  while trajectories  $t_i \in T_{u_i}$  do
    if
       $|t_{i+1}.firstPoint.time - t_i.endPoint.time| >$ 
       $\partial_{temporal}$  then
        Continue;
      else if
         $Dist(t_i.endPoint, t_{i+1}.firstPoint) > \partial_{spatial}$ 
        then
          //last point of  $t_i$  refers to first point in
           $t_{i+1}$ .;
           $t_{merged} = merge(t_i, t_{i+1})$ ;
        else
          //both trajectories now removed;
           $T_{u_i}.remove(t_i)$ ;  $T_{u_i}.remove(t_{i+1})$ ;
          //insert the merged trajectory into the
          trajectory set;
           $T_{u_i}.insert(i, t_{merged})$ ;
    end
  end

```

Algorithm 1: AmendTraject: Merge two or more disconnected trajectories that fulfill thresholds.

5.2 Determine stay regions

Stay region determination process takes three inputs i.e. road network graph $G = (V, E)$; user GPS trajectory database $T = \{T_1, T_2, T_3 \dots T_m\}$, where $T_i = \{ \langle x_1, y_1, t_1 \rangle, \langle x_2, y_2, t_2 \rangle, \langle x_3, y_3, t_3 \rangle \dots \langle x_k, y_k, t_k \rangle \}$; and the places database in question. It generates a list of regions $\mathcal{R} = \{R_1, R_2, R_3, \dots R_k\}$ where user spent some time off the road network for more than a minimum threshold. Figure 2 shows as example of the stay region computation based on the trajectory point sequences. The black point in the center of the stay region denotes the center point of the stay region that works as a reference points for the travel distance computation purposes.

The extraction of a stay point depends on two scale parameters, a time threshold $\partial_{temporal}$ and a distance threshold $\partial_{spatial}$. For the points $\{p_5, p_6, p_7 \dots p_{17}\}$ demonstrated in Figure 2, a single stay point s (black point at the center of rectangular region) is regarded as a virtual location characterized by a group of consecutive GPS points $P = \{p_m, p_{m+1} \dots p_n\}$, where $\forall_{m < i \leq n}, Dist(p_m, p_n) \leq \partial_{spatial}$ and $|p_n.T - p_m.T| \geq \partial_{temporal}$. Formally, conditioned by P , $\partial_{spatial}$ and $\partial_{temporal}$, a stay point $s = (A_{R_i}, Latitude, Longitude, T_{arrival}, T_{departure})$, where,

$$s.Latitude = \frac{\sum_{i=m}^n p_i.Latitude}{|P|} \quad (1)$$

$$s.Longitude = \frac{\sum_{i=m}^n p_i.Longitude}{|P|} \quad (2)$$

The region $s.A_{R_i}$ is the rectangle with center as $(s.Latitude, s.Longitude)$. Equation (1) and (2) respectively stand for the average latitude and longitude of the collection P . However, $s.T_{arrival} = p_m.T$ and $s.T_{departure} = p_n.T$ represent a user's arrival and departure times on stay point s included in rectangular region. These stay points occur where an individual remains stationary exceeding a time threshold $\partial_{temporal}$. In most cases, this status happens when people enter a building and lose satellite signal over a time interval until coming back outdoors. The other situation is when a user wanders around within a certain geo-spatial range for a period. In most cases, this situation occurs when people travel outdoors and are attracted by the surrounding environment. As compared to a raw GPS point, each stay point carries a particular semantic meaning, such as the shopping malls we accessed and the restaurants we visited, etc.

Definition 11. ROI Neighborhood: The neighborhood of a spatial point is the whole region it falls in, i.e., two points

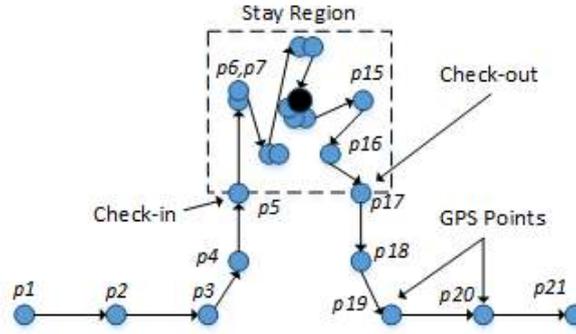


Figure 2: Demonstration of the stay-region computation from the GPS trajectories.

are considered similar iff they fall in the same region.

$$N_R(x, y, z) = \begin{cases} A & \text{if } A \in R \wedge (x, y, z) \in A \\ \Phi & \text{Otherwise} \end{cases} . \quad (3)$$

Here, we assume to receive as input a set R of disjoint spatial regions – each representing a place that is relevant for our analysis – which is used to define a neighborhood function. The neighborhood of a spatial point is the whole region it falls in, i.e., two points are considered similar iff they fall in the same region. All points that are not covered by the regions in R have an empty neighborhood, meaning that they are not similar to any point. The result is that points disregarded by R will be treated as the part of road network rather than a part of the ROIs.

5.3 User checking and visitor's log registry

As shown in component diagram in Figure 1, the second major steps of the proposed system is to identify the stay regions and the POIs where user spent some time more than the threshold value. It also maintains the visitor's registry in the individual places, keeping the record of their home location, checking-in and checking-out time etc. The stay points are the regions where the user has spent some time off-the-road. When user stays on the road, it could be due to heavy traffic and hence it needs to be carefully avoided. Making assumptions about the movement of objects out of the observation points means to provide a model for such movement. In general an object stays inside a region R_i for a time interval $I = [t_1, t_2]$, instead of a single instance t , and therefore it is obvious to associate the time interval with the region R_i . A time-stamp should be chosen following some criteria that correctly models the kind of events described in the resulting temporal user registry:

1. If the trajectory starts at time t from a point already inside a region R_i , i.e. the user has already checked-in

into the region, therefore the check-in time should be recorded as t .

2. Take entering times of the trajectory for each region, and associate it with the region name. The check-out time is recorded once the user moves out.
3. In case, the user exits for a small interval ∂ and re-enters into the region, it is simply considered a noise checkout and avoided from being recorded as a check-out event.
4. A region is defined as in Definition 12; however, the neighborhood property is defined in definition 11. The decision of point's belongingness in a region R_i is decided by ROI neighborhood definition in definition 11.

Definition 12. Region of Interest (ROI): Given the trajectory T of a user; the spatial and temporal thresholds $\partial_{spatial}$ and $\partial_{temporal}$; an ROI is defined as an enclosing area of a maximal sequence S of the trajectory T where points remain within a spatial area A_{R_i} for a more than a certain period of time. More precisely: $S = \{ \langle p_m, p_n, \dots, p_k \rangle \mid 0 < m \leq k \leq n \wedge \forall_{m \leq i < j \leq k} Dist(p_m, p_k) \leq \partial_{spatial} \wedge Diff(p_m, p_k) \geq \partial_{temporal} \wedge \forall_{m \leq i < j \leq k} p_i \in A_{R_i} \}$. Where, A_{R_i} is the rectangular polygon containing ROI. The $Dist()$ is an Euclidean distance function and $Diff()$ is the time difference in the temporal coordinates. The ROI can also be a POI with a small region size i.e. a restaurant also has a polygonal shape of its premise.

As shown in algorithm "User-check-in", the trajectories are iterated one by one. Each trajectory has an associated userid with it. All the shape points in the trajectories are looped through in order to check if they belong to any POI in the given POI database L . If a point is a noise check-out, then it is simply ignored. In case the shape point now does not belong to the currently checked-in POI, then the currentPOI is updated with the user checkout in case the user stayed enough time at the POI (i.e. `hasUserStayedEnough()` call determines if the user stayed for more than

the threshold in the same place). The user checkout time is taken from the previous point in the sequence. In case the stay time in the POI is below than the threshold, then the current entry of the user is removed from the POI using `removeRecentVisitor()` call. Further, if the user was not checked-in at any POI before; but now suddenly falling within a POI ℓ , then a visitor entry is created in the POI ℓ with the current shape's time stamp as arrival time in the POI premises.

5.4 Home location and user categorization

A home location for a user is a location that is visited and spent time there on a regular basis. The home location can be implemented by considering the time of check-in(s) and the visit pattern, for example, the user comes home and spends night at a location regularly. An approximate home location determination is enough for our purpose, as we only need a notion if the user is a tourist or a native. The home location H_u for the user u is a geospatial region within the given trajectory database T which has been visited more than any other identified locations in set L . In other words, $H_u = \{\ell | \forall \ell', \ell' \in L_u \wedge L_u \subseteq L \wedge \text{freq}_u(\ell) \geq \text{argmax}_{\ell'} \text{freq}_u(\ell')\}$; where L_u is the set of locations visited by the user u and $\text{freq}_u(\ell)$ is the visitor's frequency in the underlying place for the user u . For simplicity, we have considered the highest visited place as home location of the user. It can be a temporary house, hostel or a hotel where the user is residing.

The visiting pattern such as leaving the place in the morning and coming back to the same location in the evening in a regular pattern, then it is marked as the 'home' location. The POIs under evaluation are stored in the spatial grid so that the algorithm only search the appropriate grids while looking for the check-in locations. Those points are evaluated for the POI check-in only if the user has spent more than a specified minimum time. The POIs are searched for a given point pi using the adjacent grid based search (AGBS) method described in [17]. Once the 'home' location and the visited POIs are identified, the distance from the nearest home location to the POI is computed and the visitor's registry is maintained in the individual POIs. This process needs to be executed for each user in the trajectory database.

It is important to note that we also consider the change of the home location in case user stays for a certain amount of time in the same location and keeps visiting it. For example, if user moves to other city, then it should be marked as home location after a threshold time $\partial_{\text{home-duration}}$. Additionally, there is a notion of multiple home locations as

the user might move to different region, city, or country after some certain time. Since after spending few months in a city, she is no longer a tourist; therefore, she is marked as regional; and subsequently, a native resident. We consider the closest home location for the user categorization purposes. The locations in the visitor's log registry are pruned based on the label it receives i.e. the routine location for a particular user has been removed from evaluation and added into the routine location category for further references. A location is a routine POI if the same user visits a place in the repeated manner. We remove such entries from the visitor register if a user falls in routine category for that particular place.

Figure 3 shows the process for user categorization using the fuzzy inference engine. We have used a rule based fuzzy system that takes input as distance from an individual POI to the 'home' location of the visitor; and then it categorizes the user as native, regional or tourist. The user category is further used in the weighted rating procedure in order to estimate the popularity of a POI.

5.5 Popularity estimation

The popularity of a location is a fuzzy measurement. The popularity is measured based on the different attributes such as visiting frequency, visitor category, and stay time. The process includes three steps i.e. fuzzify the visitors registry attributes, compute the weighted frequency based on the basic attributes using rule based fuzzy inference system (FIS); and again, establish the rule based FIS to compute popularity using the weighted frequency. An FIS is a system that uses fuzzy set theory to map inputs features to outputs classes [18]. Having all the visitor's information ready with the location database visitor's registry, it is easy to apply the fuzzy rule based approach to estimate the popularity of the underlying locations. The method takes an input of the 'place database' and the output is the estimated popularity for each individual places.

Since not all the attributes have the same importance, the analytics hierarchy process can be used to find the weights for the attributes; however, the rule based fuzzy inference systems are another way around. These attributes are broken down to different linguistic fuzzy variables. The membership function of a fuzzy set is a generalization of the indicator function in classical sets. In fuzzy logic, it represents the degree of truth as an extension of valuation. A fuzzy membership function are used for each fuzzy variable and then calculates the membership value for each variable in selected domain. The fuzzification of the individual attributes i.e. visiting frequency,

Data: Trajectory database τ , POI/regions database L ; and the road network database G .

Result: The visitor's log registry $v = \{V | V = \langle U_{id}, H_u, T_{arrival}, T_{departure} \rangle\}$. The field home location H_u is empty that will be filled as part of the next step.

```

//Initialization;
currentPOI = NULL;
Uid = NULL;
while trajectoriest ∈ T do
    Uid = UserId(ti);
    while shape-point sj ∈ ti do
        if isNotNoiseCheckout(sj) then
            | Continue;
        else if currentPOI = NULL AND sj ∉ currentPOI then
            | //Is it a checkout?;
            | Tdeparture = TimeStamp(sj-1);
            | if currentPOI.hasUserStayedEnough(Uid, Tdeparture) then
            | | //update checkout time; currentPOI.updateVisitor(Uid, Tdeparture);
            | | currentPOI = NULL;
            | else
            | | //not a valid check-in, remove visitor;
            | | currentPOI.removeRecentVisitor(Uid);
        else if currentPOI == NULL AND sj ∈ ℓ AND ℓ ∈ L then
            | //Is it check-in?;
            | currentPOI = ℓ;
            | Tarrival = TimeStamp(sj);
            | //add visitor entry in v with check-in time;
            | v.addVisitor(Uid, Tarrival);
    end
end

```

Algorithm 2: User-check-in: Determines the check-in time, and checkout time for the individual locations and individual users participating in the given trajectory.

visitor category, and stay time as shown in Figure 3, returns the fuzzy variables that are further used for the popularity estimation. We have mainly used the Z-shape membership function [18] for the negative side (placed left in the membership graph) of the fuzzy variables; however the S-Shape membership function is used for the highly positive (placed right in the membership graph). The moderate fuzzy variable are derived using a Guassian membership function. The fuzzification process includes multiple steps including the weighting of the different category user's visits. The global traveler's visits are considered more significant in terms of popularity than that of the local visitor. Here $\Sigma L_i(f_w)$ represents the aggregation of all the weighted frequencies f_w for each visitor's entry in the location L_i .

First, user categorization – the first step is to categorize the visitors in a POI based on their travel distance from their home locations. As shown in Figure 3 the visitors are categorized as native, regional and tourist. Clearly, the more distance visitors travel away from their home location, the more they influence in the places popularity and ratings.

Second, stay time – the fuzzy linguistic terms low, medium and high have been used for the stay time categorization. It also uses similar membership functions i.e. Z-shape, Guassian and S-shape to represent the low, medium and high stay time. The notion is that the more time a visitor spends in a location the better for the popularity scores.

Third, weighted frequency – the visitor's visit frequency is normally the number of times the visitor enters into a place and stays for more than a defined time threshold. However, in the proposed system, the user category (computed based on the travel distance) and the time duration contribution influences the visit frequency value known as weighted frequency. The weighted frequency is computed for each individual visitor's registry record. Clearly, the $\Sigma f \leq f_w$. Where f and f_w are normal frequency and weighted frequencies for the visitor's record. The weighted frequency is computed based on the rule based FIS as shown in Figure 4.

Once the weighted frequency f_w is computed it is one step away to compute the final place rating. The locations in the location database τ is pruned based on the label it receives i.e. the routine location for a particular user has been removed from evaluation and added into the routine location category for further references. Using the rule based fuzzy inference systems to estimate the membership of the location into the popularity fuzzy variable i.e. low, medium and high. Figure 5 shows the FIS setup for computing the popularity label (or rating after defuzzification).

6 Implementation and evaluation

We have used the GPS trajectory data-set that was collected in Microsoft Research Asia's GeoLife project by 182 users in a period of over three years. A GPS trajectory of this data-set is represented by a sequence of time-stamped points. This data-set contains 17,621 trajectories with a total distance of about 1.2 million kilometers and a total duration of 48, 000+ hours. These trajectories were recorded by different GPS loggers and GPS-phones, and have a variety of sampling rates. 91 percent of the trajectories are logged in a dense representation, e.g. every 1 5 seconds or every 5 10 meters per point.

This data-set recorded a broad range of user's outdoor movements, including not only life routines like go home and go to work but also some entertainments and sports activities, such as shopping, sightseeing, dining, hiking, and cycling. Although this data-set is wildly distributed in over 30 cities of China and even in some cities located in the USA and Europe, the majority of the data was created in Beijing, China. 73 users have labeled their trajectories with transportation mode, such as driving, taking a bus, riding a bike and walking¹. The total data size is approximately 1.55GB which takes around 2 hours 45 minutes to complete parsing the popularity estimation (excluding the pre-processing and sampling amplification in the trajectory database). Figure 6 show the data distribution based on the travel distance, collection duration, and the effective travel duration. We have used FuzzyLite C++ library³ in order to create the fuzzy inference engine.

We collected 26807 POIs using the Nokia's HERE Maps² application within the Beijing area. The part of the trajectory database overlaps with the POI database as the major part of the trajectory data belong to China. The total of 1477 places were visited by the trajectory users out of the underlying POI database. It is actually the number of places that overlap with the whole trajectory. The POI data has a variety of the place categories including the restaurants and other food related places being maximum as shown in Figure 7(a). But the most interesting point is that only 0.037% of the place have any ratings as shown in Figure 7(b). This is also the main motivation for us exploring towards the trajectory based popularity and rating estimation methods.

³ Fuzzylite version 4.0 – A Fuzzy Logic Control Library and Application in C++ <https://code.google.com/p/fuzzylite/>. Last accessed on June 13th, 2014

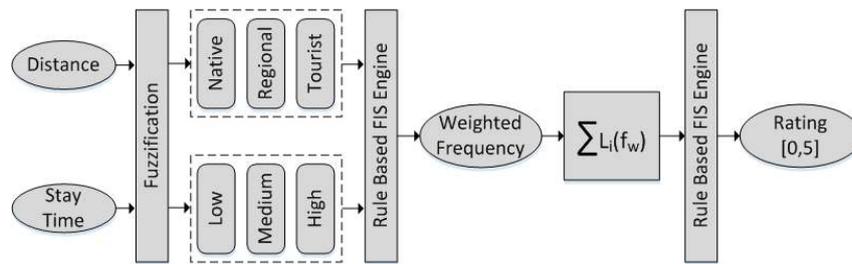


Figure 3: Block diagram of popularity estimation explaining rating computation process.

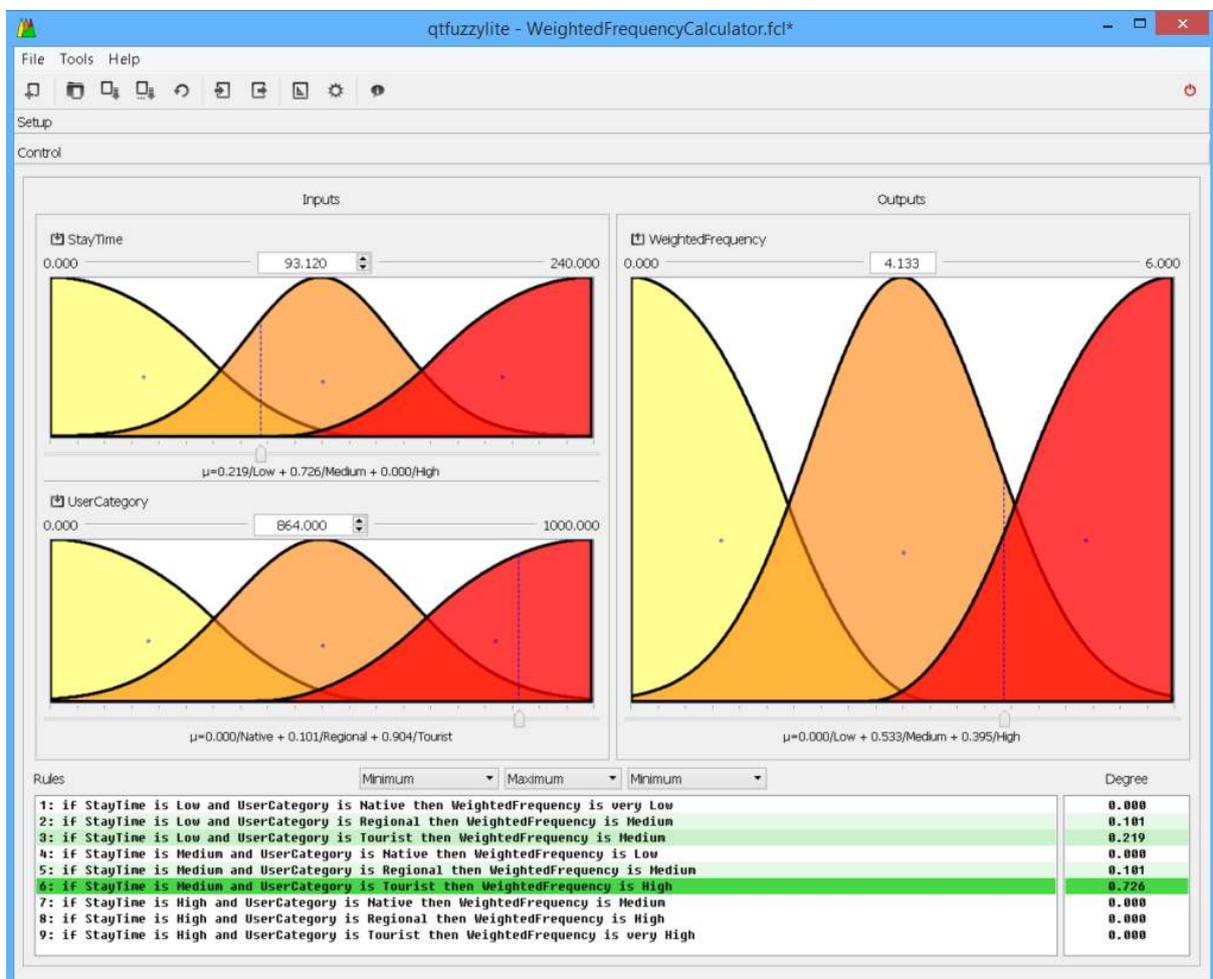


Figure 4: Fuzzification of the stay time and travel distance of the visitors and computing the weighted frequency (visitor importance).

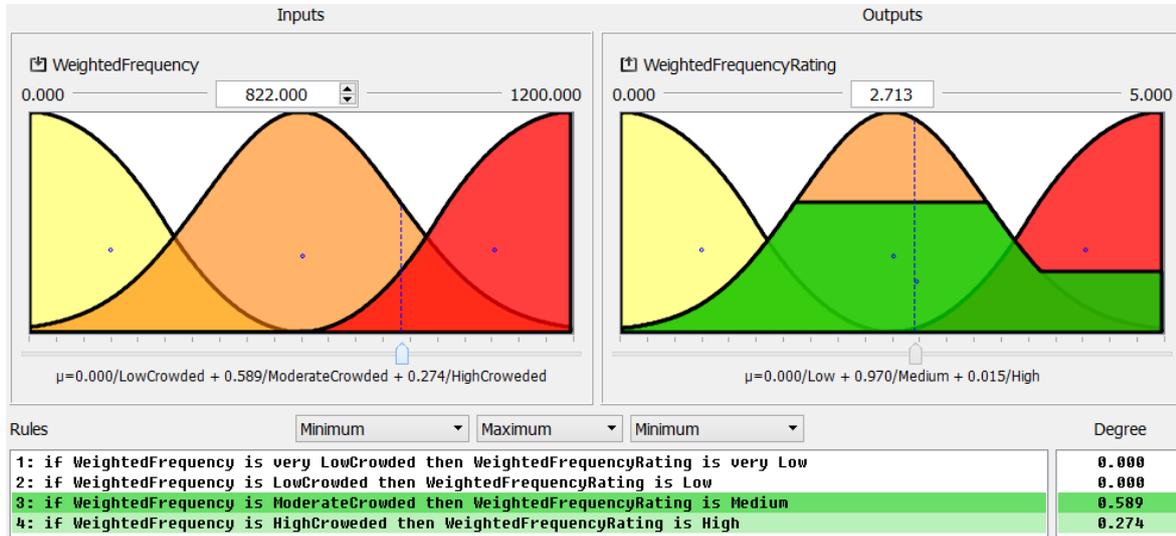


Figure 5: Fuzzification of the total weighted frequency of a place and computing the final rating based on the weighted frequency.

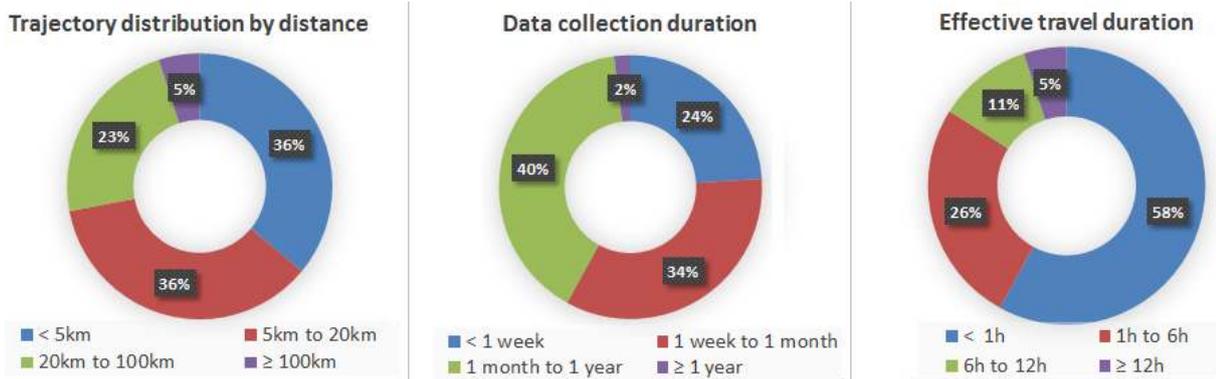


Figure 6: a) Trajectory distribution by distance, b) Data collection duration distribution, c) Effective travel duration distribution

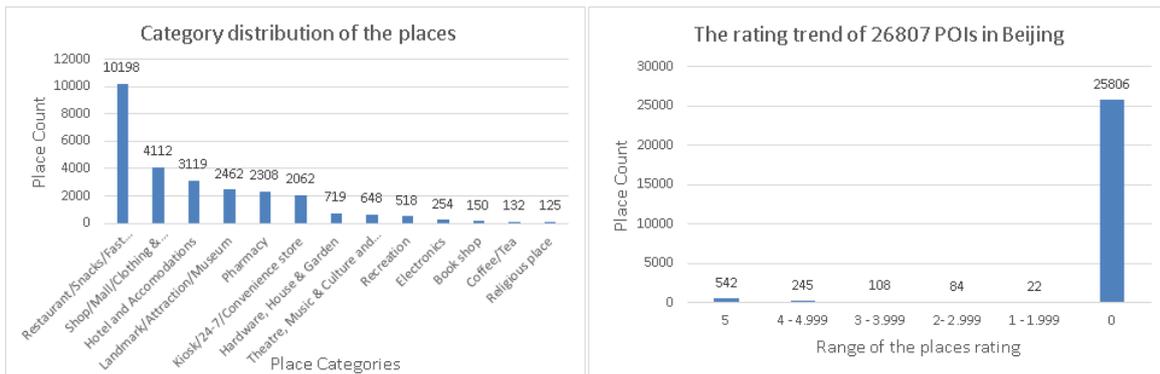


Figure 7: a) Category distribution of the places in the underlying POI database. b) Rating trend of the POI – most of the POIs are unrated and hence the 0 rating is the maximum.

Table 1: Ratings comparison of top 20 places in Beijing visited by the trajectory users.

	Place Category	Frequency	Weighted Frequency	Review Count	Manual Ratings	Frequency Ratings	Weighted Frequency Ratings
1	Hotel	490	1283.161	0	5	4.1727	4.2305
2	Hotel	307	759.8676	0	0	2.5418	2.5717
3	Restaurant	205	558.1162	0	0	2.4835	2.5
4	Shop	170	424.6227	0	0	2.3625	2.4016
5	Restaurant	131	355.2668	0	0	1.994	2.2087
6	Sights / Museums	122	330.9166	0	0	1.8704	2.1084
7	Building	133	329.8117	1	5	2.0202	2.1034
8	Restaurant	114	297.9203	0	0	1.7595	1.9493
9	Electronics	105	283.7347	0	0	1.6374	1.8769
10	Snacks/Fast food	126	253.6427	0	0	1.9262	1.7248
11	Restaurant	91	228.6495	0	0	1.4566	1.6014
12	Restaurant	90	218.7402	0	0	1.4443	1.5537
13	Hotel	99	211.2394	0	0	1.5582	1.5181
14	Mall	75	209.9281	0	0	1.2709	1.5119
15	Hotel	90	209.8834	0	5	1.4443	1.5117
16	Shop	75	207.6965	0	0	1.2709	1.5014
17	Hotel	70	206.6292	0	5	1.2185	1.4964
18	Restaurant	83	199.6854	0	0	1.3605	1.4643
19	Hotel	86	189.7606	0	5	1.3959	1.4192
20	Cinema	88	189.065	0	0	1.4199	1.4161

Jade Palace Hotel in Beijing was the most visited locations among the 1477 visited places with the total of 490 visitors visited the place. The original rating was 5; but without any reviews. It was also not clear how many people participated in the original rating process. On the other hand, the trajectory based method is more transparent and intuitive that compounds to be around the rating of 4.23 based on the user category based system as shown in Table 1. The table shows the top 20 places and their ratings comparison in terms of the original ratings, frequency based and weighted frequency based ratings etc. It is also noticed that the only one place has the review in the top 20 places. The Figure 8(a) shows the distribution of the visitor's frequency across the POIs. It is interesting to note that most of the frequently visited places belong to the restaurant and other eat & drink categories. The original user's average based rating distribution in the Figure 8(b) shows that most of the places go unrated and only some of the places have ratings other than 5 and 0. It indicates that either the places are unrated or they are rated by some favorable contributors to keep the ratings high. On the other hand, we have results the for frequency based ratings along with the weighted frequency based popularity points. It is clear that all the places have a non-zero rat-

ing which have been visited from the trajectory users. It also clearly shows that these results have much more values $\in [0, 5]$ and hence it proves a wide range of ratings depending on how much time the users spent. Our method has boosted the rating of many places as shown in Figure 9.

7 Discussion and conclusion

Proposed popularity estimation method gives more importance to the travelers who take long way to visit and spends longer in the places premise. Biggest advantage of this method is that it does not require active participation of the users and hence it overcomes from the proxy submission threat in the manual rating methods. We have compared popularity computed by frequency count and weighted frequency count methods. Frequency count based method is the commonly used to find popularity of POI in the literature. It is found that popularity computed by weighted frequency count is better than and comparable with popularity computed by frequency count method. It is to be noted that comparison of obtained results with

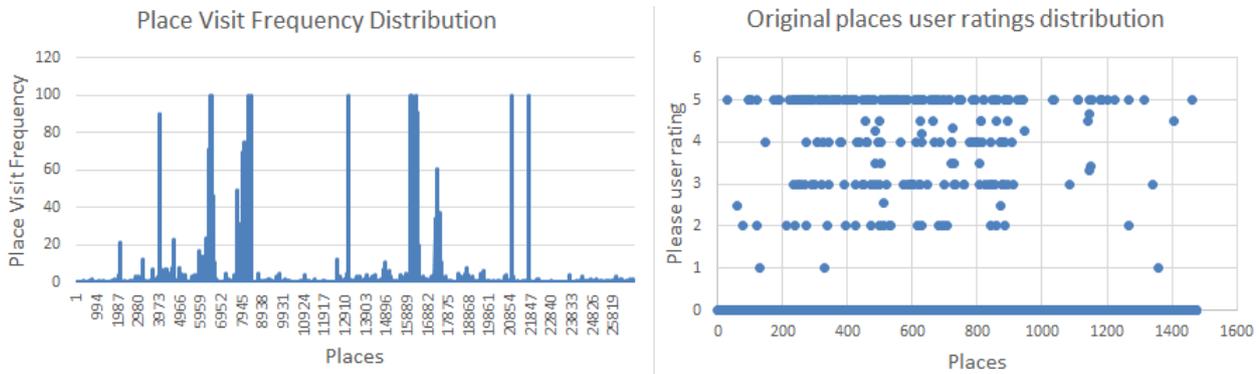


Figure 8: a) Place visit frequency distribution – capped at 100. b) Original user rating distribution provided in the POI database.

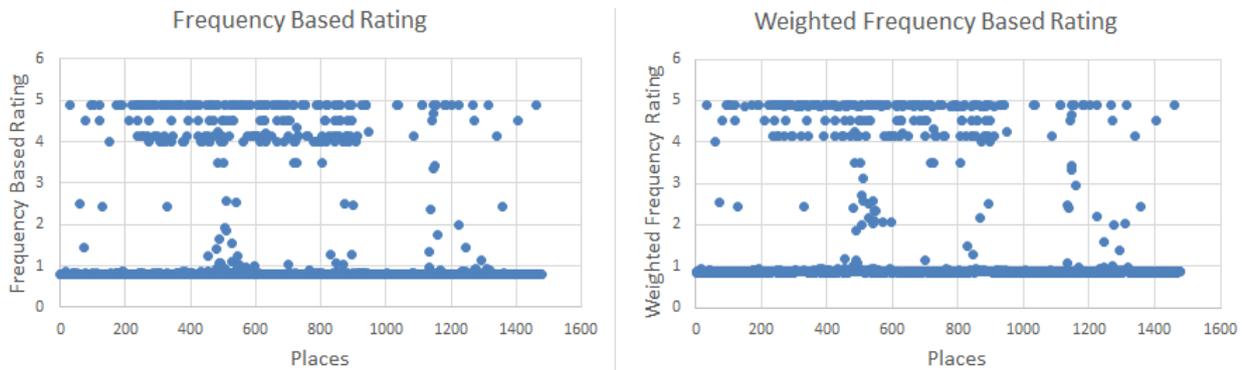


Figure 9: Comparison of ratings distribution – a) frequency based rating, and b) weighted frequency based rating

the ratings from different sources is unfair. The input parameters are entirely different in trajectory based weighted method. Every rating source organization have their own estimation mechanism based on manual feedbacks. In order to validate the correctness of the proposed method, we compared ratings with the actual rating available for POIs in the same region. The results are comparable even though manual ratings are very skewed and sometimes biased and incorrect. The proposed method should be seen as an alternate implicit mechanism of popularity estimation.

There is huge possibility to improve the proposed method. It would be interesting to explore the person's profile based popularity estimation. For example, if an international pop singer or a prime minister of some country visits a place, it increases the POI's popularity. The fleet management companies need to travel mostly long distances; therefore, it might consider the highway side restaurants as highly popular based on the travel distance of the truck drivers. In this experiment, we have estimated the places across the categories and hence it is good for the same category places as the competition should be among the same category places. However, it is fairly easy to extend the system considering the specific scenarios and by

handling the edge cases in the implementation without loss of generality.

References

- [1] Giannotti F., Nanni M., Pedreschi D., Pinnelli F., Trajectory Pattern Mining, In proceedings of: KDD'07, San Jose, California, USA, August 12–15, 2007
- [2] Guo D., Liu S., Jin H., A graph-based approach to vehicle trajectory analysis, *Journal of Location Based Services*, 4(3-4), 183-199, 2010
- [3] Giannotti F., Nanni M., Pedreschi D., Pinnelli F., Mining of sequences with temporal annotations, In proceedings of: SIAM Conference on Data Mining, pages 346–357, Dijon, France, April 23 -27, 2006
- [4] Tiwari S., Kaushik S., Modelling On-the-Spot Learning: Storage, Landmarks Weighting Heuristic and Annotation Algorithm, In proceedings of: CIAA'13 Saida (Algeria), May 04–06, 2013
- [5] Dalumpines R., Scott D., GIS-based Map-matching: Development and Demonstration of a Postprocessing Mapmatching Algorithm for Transportation Research, *Lecture Notes in Geoinformation and Cartography*, 1, 101-120, 2011
- [6] Lammerena R., Goossen M., Ligtenberg A., Interactive location-based services: problems and perspectives on the example of a cultural site, *Journal of Location Based Services*, 2, 105-119,

- 2010
- [7] Yan Z., Towards Semantic Trajectory Data Analysis: A Conceptual and Computational Approach, VLDB PhD Workshop, Lyon, France, August 24–28, 2009
- [8] Zenger B., Trajectory based Point of Interest Recommendation, M.Sc. Thesis, School of Computing Science, Simon Fraser University, Canada, 2009
- [9] Kang J., Yong H., Mining Spatio-Temporal Patterns in Trajectory Data, *Journal of Information Processing Systems*, 6(4), 521-536, 2010
- [10] Lee J. Han J., Li X., Gonzalez H., TraClass: Trajectory Classification Using Hierarchical Region-Based and Trajectory-Based Clustering, *Journal of VLDB Endowment*, 1(1), 1081-1094, 2008
- [11] Lee J., Han J., Whang K., Trajectory Clustering: A Partition-and-Group Framework, In proceedings of: SIGMOD'07, Beijing, China, June 11–14, 2007
- [12] Patel D., Incorporating duration and region association information in trajectory classification, *Journal of Location Based Services*, 7(4), 246-271, 2013
- [13] Zheng Y., Zhang L., Xie X., Ma W., Mining interesting locations and travel sequences from GPS trajectories, In proceedings of: International conference on World Wild Web (WWW 2009), Madrid, Spain, April 20–24, 2009
- [14] Zheng Y., Xie X., Ma M., GeoLife: A Collaborative Social Networking Service among User, location and trajectory. *IEEE Data Engineering Bulletin*. 33(2), 32-40, 2010
- [15] Zadeh L. A., Fuzzy sets, *Journal of Information and Control*, 8(3), 338-353, 1965
- [16] Greenfeld, J.S., Matching GPS observations to locations on a digital map, In proceedings of: the 81th Annual Meeting of the Transportation Research Board, Washington, DC, January 2002
- [17] Tiwari S. and Kaushik S., Boundary Points Detection Using Adjacent Grid Block Selection (AGBS) kNN-Join Method, In proceedings of: MLDM 2012, Berlin, Germany, July 16 - 20, 2012
- [18] Klir G., Yuan B., *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, 1st Edition, Prentice Hall PTR (ECS Professional), Upper Saddle River, New Jersey, 1997