ATLANTIS PRESS
Advances in Computer Science Research (ACRS), volume 54
International Conference on Computer Networks and Communication Technology (CNCT2016)

Identifying Complete Individual Trajectories Using Multi-day Cellular Network Data

Yang ZHAO[1] and Tong-yu ZHU[1,*]

[1] State Key Laboratory of Software Development Environment, Beihang University, Beijing 100191, China

*Corresponding author

Keywords: Spatio-temporal data mining, Cellular network data, Urban computing, trajectory data mining, Kalman filter.

Abstract. Individual Trajectory is the foundation of traveling behavior analysis. Cellular network data contains sufficient spatio-temporal information, which is widely used for trajectory analysis nowadays. However, we found in practice that most of the users' records are incomplete and fragmented. This is because mobile phones leave records of their connected cell tower only when some specific event occurs, such as making calls or sending messages. This paper present a method for identifying complete individual trajectories using fragmented trajectories extracted from multi-day cellular network data. This method firstly extracted a user's all fragmented trajectories when he/she move from one place to another from multi-day data. Secondly, we proposed a greedy algorithm to joint all the fragmented trajectories into a complete trajectory. Eventually, we adopted a modified Kalman filter algorithm to smooth the trajectory. In the end, we experimented with real data to validate our approach. The results show that our approach improves the useable data volume from 17% to 63% and reduce 36% of the trajectories' average bias distance.

Introduction

Urban computing is an emerging field which explores data computing solutions to make cities more efficient, smarter and more livable [1]. An important research aspect of this field is analyzing millions of city residents' mobility pattern. Mobility pattern can be divided into two different scales. Long-time and long-distance mobility pattern describe occasional traces such as business trip or international travel. On the contrary, short-time and short-distance mobility pattern describe the user's daily traces in the region of city, such as commuting and entertainment activity. This kind of trace shows a strong regularity. Study for this scale can compute the regularity of regional population trip and commuting characteristics, in order to develop city transportation systems and improve urban infrastructure. While the mechanism for long time scales has been studied, the underlying mechanism on the daily scale is still unrevealed [2,3]. Therefore, extracting precise trajectories is a key issue for the future research.

Increasing storage capacity and processor clouds make possible to capture petabytes of digital traces from individual activities worldwide. The mobile phone is quite portable nowadays that almost all people have their phones by their sides at all times. Mobile phone has the great potential to provide necessary information for capturing individuals' traces. In the past few years, a crowd of institutions have tried to use cellular network data to extract users' trajectories. Cellular network data automatically collected by telecom operators and archived for billing purposes and network troubleshooting, providing rich spatio-temporal information about all the phones

accessed to the cellular network [4]. Nevertheless, one of the defects of cellular network data is its unstable sampling rate. In most cellular infrastructures today, mobile phones leave a record of their connected cell tower only when some specific event occurs, such as making or receiving calls and sending messages. This characteristic results in that most trajectories we extracted from cellular network data are discontinuous and fragmented, which presents a great challenge to use the incomplete trajectories for following research.

In this paper, we aim to identifying complete individual trajectories using fragmented trajectories extracted from multi-day cellular network data. Firstly, we extract user's all fragmented trajectories when he/she move from one location to another from consecutive days of historical data. We choose the users who have stable travel habits to ensure they choose the same routes every day. Secondly, we propose a greedy algorithm to joint fragmented trajectories into one trajectory. This algorithm generate a rough trajectory with extra-coarse. Finally, we adopt a modified Kalman filter algorithm to make the trajectory smooth. In the last, we experiment with real data to validate our approach. The results show that this method is efficient. The result we obtained by this method has been used for some meaningful researches.

Related Work

At present, methods of extracting trajectory from GPS are widely developed in both computer science and transportation field. On the contrary, research on cellular network data is quite a challenge due to its low positioning accuracy and unstable sampling rate. For this reason, a crowd of institutions have made beneficial explorations. Leontiadis developed Cell*, an algorithm that is able to parse the continuous sector observations and identify the segments corresponding to a stationary position, and the segments corresponding to actual mobility [5]. Isaacman proposed new techniques based on clustering and regression for analyzing anonymized cellular network data to identify generally important locations, and to discern semantically meaningful locations such as home and work [6,7,8]. This approach is effective for seeking home and work places for commuters, but cannot identify occasional stay points (i.e. entertainment activity). Although many researchers are using a variety of methods to extract trajectory, however, they only try to solve the problem of the low positioning accuracy of cellular network data, but few researcher face the problem of its unstable sampling rate.

Furthermore, research on origin-destination trajectories and routing behavior are in a fledging period. Alexander presented a method to produce OD trips by purpose and time of day [9]. They validated the distribution of trips using two local surveys and found that the size of the areas used to aggregate trips is a very important factor in how well the CDR and survey data compare. Lima used GPS traces generated by 526 private cars to explore their routing behavior [10]. They investigated how many routes they use and how often they use each of them. However, their experiments had too small a sample size, and GPS data collected from private cars is lack of universality and representativeness. Therefore, extracting more precise trajectories is an important task for future research.

Dataset Description and Preprocessing

In this paper, we use a dataset consisting of anonymous cellular network data collected by telecom operator during the period of November, 2014 in the area of Beijing.

General information of the dataset is summarized in Table 1. Each record contains an anonymous user ID, timestamp, longitude, latitude, cell tower type, event ID, etc.

Table 1. General information of the cellular network dataset

General dataset information	Value
Number of calls	750.5 million per day
Number of users	14.9 million
Average update cycle	21.7 min
Population of Beijing	21.15 million
Area of data coverage	16410.54 km^2

Definition

The raw data is consisted of a series of records which contain the information of time and location. It couldn't represent the user's status, such as stay or move. So we firstly figure out stay point and bass-by point and use them to build the trajectory. Therefore, we propose some definitions.

Raw record: raw cellular network data, is represented as:

$$R(id, t, lon, lat) \tag{1}$$

Stay point: indicates a user stays in an area for a period of time. A stay point *SP* is calculated by a raw record cluster *RC*:

$$SP(RC_{(lon,lat)}, RC_{(ts,te)}) \tag{2}$$

Where *RC(lon, lat)* and *RC(ts, te)* are calculated as follows:

$$\begin{aligned} RC_{(lon,lat)} &= (\frac{1}{n}\sum_{i=1}^{n} R_{i.lon}, \frac{1}{n}\sum_{i=1}^{n} R_{i.lat}), R \in RC, \\ RC_{(ts,te)} &= (min(R_{i.t}), max(R_{i.t})), R \in RC \end{aligned} \tag{3}$$

Pass-by point: indicates a passed place when a user is moving. A pass-by point is calculated by a raw record who couldn't be gathered into a cluster:

$$PP(R_{(lon,lat)}, R_t) \tag{4}$$

Trajectory: is consisted of a series of stay points and pass-by points:

$$TR(SP_1, PP_2, SP_3, ..., SP_n) \tag{5}$$

Preprocessing

We counted the number of records for each user using raw record data, the result is shown in Fig. 1. There are about 10% users whose daily records number is less than 10, which is not enough to present the whole day behavior. We filtered out this kind of users. After that, our method is divided into four steps:

(1) Sort each user's raw records by timestamp;
(2) Adopt a density-based spatial clustering algorithm to identify stay points;
(3) Mark the unclustered points as pass-by points;
(4) Sort the stay points and pass-by points by timestamp.

After above steps, we recognize the daily trajectories for each user, which is the basis for identifying complete trajectories.

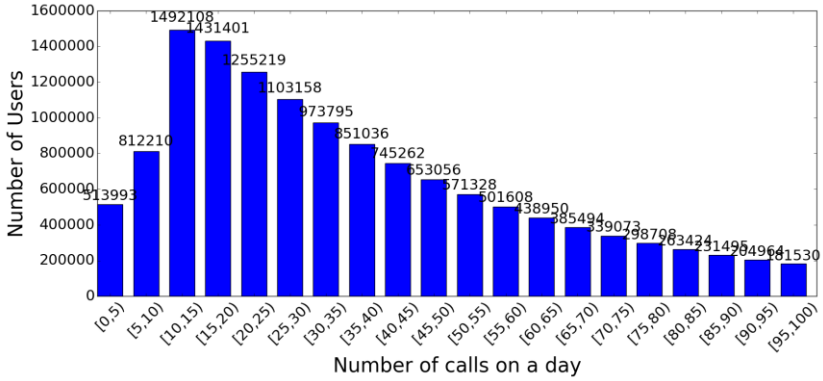


Figure 1. Distribution of number of records for each user

Identifying Complete Trajectory

When a user move from one place to another place for multiple times, he/she tends to choose the same route. According to this rule, we use user's multi-day incomplete trajectories to repair the missed fragments and restore his/her complete trajectory. This method can be divided into three steps, the next three subsections describe each of these.

Origin-Destination Trajectory Extracting

Extracting Origin-Destination trajectories is the first step of this method. Origin-Destination means a pair of locations where one is origin place and another is destination place. An Origin-Destination trajectory is a part of the daily trajectory with two stay points and some pass-by points. The first stay point in the area of origin, the last stay point in the area of destination, meanwhile all the pass-by points in the route between origin and destination. The process of extracting Origin-Destination trajectories is as follows:

(1) Determine the range of origin area and destination area;
(2) Extract the user's all daily trajectories, for each of the trajectory, go to the next step;
(3) For each stay point in the trajectory, if this stay point in the area of origin and next stay point in the area of destination, extract the segment of the trajectory;
(4) Save the segment as an Origin-Destination trajectory.

Trajectories Jointing

Because mobile phone leave a record of their connected cell tower only when some specific events occurs, so most of the Origin-Destination trajectories we extracted are incomplete. In this step, we use the user's multi-day trajectories to repair the missed fragments and build a complete trajectory.

Firstly, define the start *T.Start* and the end *T.End* of the target trajectory as:

$$\begin{aligned} T.Start_{(lon,lat)} &= (\frac{1}{n}\sum_{i=1}^{n} FT_i.Start.lon, \frac{1}{n}\sum_{i=1}^{n} FT_i.Start.lat) \\ T.End_{(lon,lat)} &= (\frac{1}{n}\sum_{i=1}^{n} FT_i.End.lon, \frac{1}{n}\sum_{i=1}^{n} FT_i.End.lat) \end{aligned} \tag{6}$$

n means the number of fragmented trajectories, $FT_i.Start.lon$ and $FT_i.Start.lat$ means the longitude and latitude of the start of the *i*-th fragmented trajectory, meanwhile $FT_i.End.lon$ and $FT_i.End.lat$ means the longitude and latitude of the end of the *i*-th fragmented trajectory.

After that, our algorithm works as follows to visit all pass-by points in the fragmented trajectories:

(1) Marks *T.Start* for visited and add it into the visited queue; marks all pass-by point for unvisited;
(2) Get a point in the tail of the visited queue, find the nearest unvisited point, marks it as a visited point and add it into the visited queue;
(3) Repeat the step 2 until all pass-by points are visited;
(4) Add the *T.End* to the tail of the queue.

After the algorithm is performed, the points sequence in the visited queue is a complete trajectory.

De-noising and Smoothing

Now we have got a rough, approximate complete trajectory. However, due to the low positioning accuracy and measurement error, the current result has noisy. Therefore, we propose a modified Kalman filter to smooth the trajectory.

Classical Kalman filter is a recursive estimator. The filter estimate the state at a given time of the process, then obtain the feedback through the measurement variable with noisy. So Kalman filter is conceptualized as two distinct phases: “Time Update” and “Measurement Update”. The time update phase produces an estimate of the state at the current time using the state estimate from the previous time. Because the predicted state estimate does not include observation information at the current time, so it also be called priori state estimate. In the update phase, the state estimate is refined by the current observation information combined with the current a priori prediction.

Time Update:

$$\hat{X}_{k|k-1} = A_k \hat{X}_{k-1|k-1} + B_{k-1} U_{k-1} \tag{7}$$

$$P_{k|k-1} = A_k P_{k-1|k-1} A_k^T + Q_k \tag{8}$$

$\hat{X}_{k|k-1}$ denotes the estimate of the model’s state in the time *k* before the before the *k*-th measurement value has been taken into account. $P_{k|k-1}$ denotes a posteriori error covariance matrix. A_k is the state transition model which is applied to the previous state $\hat{X}_{k-1|k-1}$; B_k is the control-input model which is applied to the control vector U_k. Q_k is noise covariances. In this paper, A_k, B_k, U_k are all set as 1 in the initialization.

Measurement Update:

$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1} \tag{9}$$

$$\hat{X}_{k|k} = \hat{X}_{k|k-1} + K_k (X_k - H_k \hat{X}_{k|k-1}) \tag{10}$$

$$P_k = (I - K_k H_k) P_{k|k-1} \tag{11}$$

K_k is the Kalman gain, H_k is the observation model which maps the true state space into the observed space and R_k is the covariance of the observation noise.

However, in this paper, we use the filter for the geographic data which represented by longitude and latitude. This kind of value is mostly stable and seldom wide-range fluctuating. So we add a penalty coefficient $P_k^{'}$ in the update phase to reduce the sensitivity to the changes of variate.

Modified Measurement Update:

$$\hat{X}_{k|k} = \hat{X}_{k|k-1} + P_k^{'} K_k (X_k - H_k \hat{X}_{k|k-1}) \tag{12}$$

$P_k^{'}$'s value range is $(0,1]$, the more stable the user, the more close to 1 the $P_k^{'}$ is.

Experiment and Result

In this section, we implement a set of experiments to evaluate the performance of our method. Firstly, we show the trajectories jointing result. Then, we adopt the modified Kalman filter and got some results. In our experiments, we analyze real anonymous cellular network data provided by a telecommunication operator in China. The overview of dataset is described in Section 3.

Trajectories Jointing

Because commuters tend to travel the same path when they move between home and office and their travel have a stronger regularity, so we choose the data collected in continuous 20 working days (4 weeks) for our experiments. Firstly, we find some typical commuting Origin-Destination pairs and find out all commuters in these Origin-Destination pairs. These commuters could be divided into three types:

(1) *Not Need Repair*: this kind of users' one trajectory already covers the whole path between origin location and destination location(interval between adjacent points less than 1km).
(2) *Could Be Repair*: this kind of users' one trajectory could not cover the whole path between origin location and destination location, but his/her multiple trajectories could.
(3) *Couldn't Be Repair*: this kind of users' all trajectories could not cover the whole path between origin location and destination location.

We use the algorithm deal with the data and the result shown in Table 2.

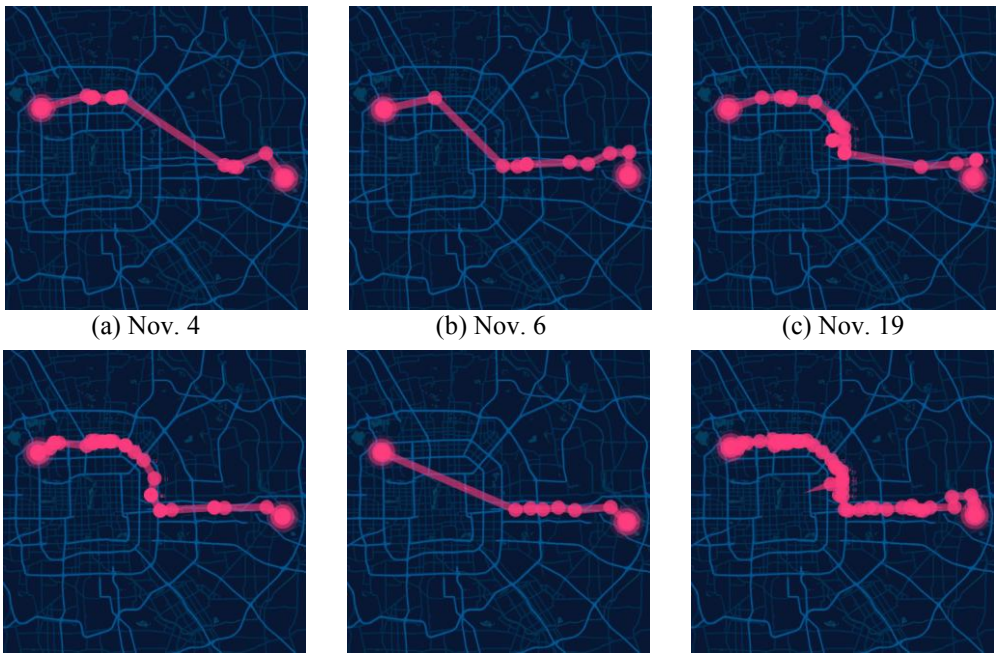
Table 2. The user types accounts

User type	Not Need Repair	Could Be Repair	Couldn't Be Repair
Ratio	17%	46%	37%

The result shows that trajectory jointing method improves the useful sample size from 17% to 63%. Fig. 2 shows a user's trajectory jointing result. Fig. 2(a) to (e) is the fragmented trajectories the user leaved in 5 days. Fig. 2(f) is the jointed result using trajectory jointing algorithm.

De-noising and Smoothing

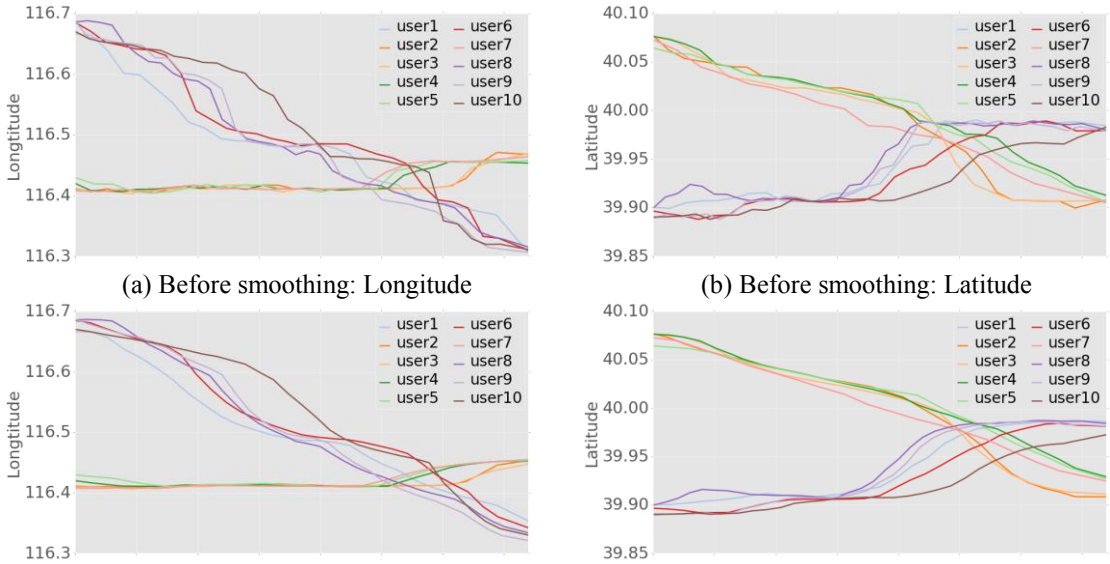
After jointing, we use the modified Kalman filter to smooth the noisy of the trajectory. The modified Kalman filter respectively smooth the longitude and latitude data. We select 10 users' smoothing result, as shown in Fig. 3. The results show that filter process make the trajectories more smooth.



(a) Nov. 4 (b) Nov. 6 (c) Nov. 19

(d) Nov. 21 (e) Nov. 28 (f) Jointed

Figure 2. A sample of trajectory jointing



(a) Before smoothing: Longitude (b) Before smoothing: Latitude

(c) After smoothing: Longitude (d) After smoothing: Latitude

Figure 3. A sample of trajectory smoothing

To validate the result, we calculate the average bias distance between the trajectory and the road using a map matching program. We respectively calculate the raw trajectories and the repaired trajectories. The result is shown in Table 3. The result shows that the trajectory after smoothing has a smaller average bias distance than the trajectory without smoothing. The average improvement is 36%.

Table 3. Average bias distance between trajectory and road

User ID		1	2	3	4	5	6	7	8	9	10
Average Bias Distance (km)	Before Smooth	1.0	1.2	0.9	0.9	1.0	1.1	0.9	1.1	1.2	0.8
	After Smooth	0.7	0.8	0.6	0.5	0.6	0.7	0.6	0.7	0.6	0.6

Conclusions

Analysis of people's traveling behavior using cellular network data is a challenge task. In this paper, we propose a method to identify complete individual trajectory using his/her historical fragmented trajectories extracted from cellular network data. Compare with other researches' approaches, our research uses massive historical data to repair the missed fragments and improve the precise of the trajectory. In order to do that, we come up with a greedy algorithm to joint fragmented trajectories and a modified Kalman filter algorithm to make the trajectory smooth. We use the real-world data to validate our method. The results show that our method could repair 46% of the users' trajectories, which increases the useful data from 17% to 63%. Our future work will focus on using these data to analyze citizen's traveling rules.

Acknowledgement

This research was financially supported by the National High Technology Research and Development Program of China (863 Program) No.2015AA124103.

References

[1] Batty, Michael, et al. "Smart cities of the future." The European Physical Journal Special Topics 214.1 (2012): 481-518.

[2] Schneider, Christian M., et al. "Unravelling daily human mobility motifs." Journal of The Royal Society Interface 10.84 (2013): 20130246.

[3] Toole, Jameson L., et al. "The path most traveled: Travel demand estimation using big data resources." Transportation Research Part C: Emerging Technologies 58 (2015): 162-177.

[4] Ficek, Michal, and Lukas Kencl. "Inter-call mobility model: A spatio-temporal refinement of call data records using a Gaussian mixture model." INFOCOM, 2012 Proceedings IEEE. IEEE, 2012.

[5] Leontiadis, Ilias, et al. "From cells to streets: Estimating mobile paths with cellular-side data." Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies. ACM, 2014.

[6] Isaacman, Sibren, et al. "Identifying important places in people's lives from cellular network data." International Conference on Pervasive Computing. Springer Berlin Heidelberg, 2011.

[7] Isaacman, Sibren, et al. "Human mobility modeling at metropolitan scales." Proceedings of the 10th international conference on Mobile systems, applications, and services. ACM, 2012.

[8] Becker, Richard, et al. "Human mobility characterization from cellular network data." Communications of the ACM 56.1 (2013): 74-82.

[9] Alexander, Lauren, et al. "Origin–destination trips by purpose and time of day inferred from mobile phone data." Transportation Research Part C: Emerging Technologies 58 (2015): 240-250.

[10] Lima, Antonio, et al. "Understanding individual routing behaviour." Journal of The Royal Society Interface 13.116 (2016): 20160021.