Comput Syst Sci & Eng (2018) 2: 87–94

International Journal of
Computer Systems
Science & Engineering

Tensor-based User Trajectory Mining

Chen Yu,* Qinmin Hong, Dezhong Yao and Hai Jin

Cluster and Grid Computing Lab
Big Data Technology and System Lab
Services Computing Technology and System Lab
School of Computer Science and Technology
Huazhong University of Science and Technology, Wuhan, 430074, China

The rapid expansion of GPS-embedded devices has showed the emerging new look of location-based services, enabling such offerings as travel guide services and location-based social networks. One consequence is the accumulation of a rich supply of GPS trajectories, indicating individuals' historical position. Based on these data, we aim to mine the hot route by using a collaborative tensor calculation method. We present an efficient trajectory data processing model for mining the hot route. In this paper, we rst model the individual's trajectory log, extract sources and destinations, use map matching to get the corresponding road segments, and nally apply the source-destination-road segments tensor in order to compute the recommended hot route. To prove the validity and efficiency of the method, we conduct a collaborative route recommendation system, and the experimental result indicated that the solution can recommend a route with considerable accuracy

Keywords: Data mining; hot route discovery; GPS logs; route recommendation

1. INTRODUCTION

In the last several years, moving objects researching has become increasingly feasible and commonplace and mobile devices (smartphones and GPS-embedded PDAs for instance) are also becoming ubiquitous in daily life. The proliferation of these devices has given rise to growth in *location-based services* (LBS) that easily provide people with their present locations and also enable the collection of huge quantities of GPS trajectory data. One usefully application is navigation that can search the real-time condition and then calculate the optimal route to their destination. How to efficiently mining hot route is a challenge problem in navigation applications [3].

A GPS trajectory is a time series of coordinates of mobile objects, such as pedestrians and cars. In daily life, for example, people can use Google+ or Foursquare to upload and share their location and GPS trajectory, thus providing data for location-based service [1] applications such as car navigation, travel services and Find my friends [2]. But rare of the apps have mining useful information from user context.

Location is very frequent in user's pattern-mining [3]. But the importance of context depends on not only the location users visited but also the route each user selected. When linked to a real road network, the route can be subdivided into road segments. Once we know where a user is going, we need more detailed information about these road segments in order to recommend the optimal route.

Personal navigation is one of the important services in our daily life and gradually become the research hotspot [4, 25]. As there are multiple choices for the route from one place to the destination, people usually find it difficult to choose a suitable route without road information. So hot route mining and recommendations have become an important and useful topic.

Given the quantity of trajectory history data, we try to combine each user's location trajectory with the trajectories of others to form a more complete route. When people plan a route, they tend to base it on their own experience or that of others, so we can mine the possible route based on a combination of road segment popularity from a GPS database. If a road segment is selected by a user, which indicates that the road segment is superior to others in some ways because it has more points of interest, or better traffic conditions.

In this paper, we intend to mine knowledge based on historical trajectory data, in which we may handle the challenging problem: if we want to go somewhere, which route should we choose? We try to discovery a hot/popular route between the starting location and the destination location. A user's trajectory can be presented

*Corresponding Author. E-mail: yuchen@hust.edu.cn

as a combination of three aspects: source, destination, and road segments. Another associated key characteristic is sparsity that is low-sampling, so we can use tensor object to store and process trajectory data. In a query with a given source and destination, it is possible to find the road segments that are more popular than others. Then we combine the road segments into a complete route as a recommendation for the user's reference. People can refer to it and decide to follow the crowd or avoid the crowded roadways based on other users' real-time situations.

In the previous study, a hot route usually refers to a frequently visited road, which indicates that it is a key point in a city. It can be used to describe the moving pattern of an individual or a moving object. Finding the hot route from trajectory data can assist with city planning, transportation management, and decision analysis in the field of user navigation apps, for example Google Maps, A map, Uber and Waze.

The contributions and novelty of the work is:

- We propose a tensor model to mine the user's trajectory information, which first extracts multiple users' stay points and then uses map matching to determine the corresponding road segments.

- Based on the processed data, we propose a source-destination-road-segment three-dimensional tensor model for mining the hot road segment set. This model is constructed offline, which can ensure the efficiency of our algorithm while calculating the route from a source to a destination on a map.

- Considering trajectory history and road segment popularity, we mine the collaborative hot road segment set and determine the hot route.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 presents an overview of our system and introduces the preparatory work. Section 4 details our three-dimensional tensor model. In Section 5, experimental methods are introduced and the results are analyzed. Finally, we make conclusions about this work and present future work in Section 6.

2. RELATED WORK

In this section, we present previous work on hot route discovery and the tensor calculation method.

Our work is connected to hot route discovery, which intend to discover the route people usually traveled. Giannotti et al. [5] presented the problem of trajectory pattern mining, mined regular patterns from people's historical trajectories. [6, 7] proposed some novel methods to deal with mobile data processing. In [8], Li et al. proposed a density-basedalgorithm called FLowScan to extract hot routes based on adefinition of "traffic density-reachable".In [9], Zheng et al. proposed a *history based routeinference system*(HRIS) to mine possible routes from low sampling rate trajectories. Hu et al. [32, 33] have made some prediction based on large scale data set. [10] used multi-camera to build a trajectory mining system. Mobile tourist guide systems in [11] and [12] typically recommended locations and sometimes provided navigation information based on a user's realtime condition. Some energy efficient trajectory tracking systems [13, 18] were also proposed to record density location data. Compared with the methods above, we proposed a novel method of mining different people's trajectory data from searching people's related road segments to understand collective intelligence also model the correlation between routes and historical GPS data.

In recent years, tensor is a popular research topic. The review of tensor decompositions and applications was attributed to Kolda [14], tensor has been applied in such fields as signal processing [15], numerical linear algebra [16], numerical analysis [17], and graph analysis [19]. Recently, tensor based recommendation method have be used in many areas. A collaborative tensor based recommendation method was introduced in [20]. Yao et al. [21] proposed a novel model to predict userâŁ™s mobility based on tensor. A tensor factorization based social tagging system in [26] aimed to solve sparsity and âŁœcold startâŁž problems. Tensor factorization method was also applied in Point-of-Interest recommendation [27].

Our research is based on user's historical trajectory data, we select the popularity of road segment as the critical factor for user context, and then use the tensor calculation method to figure out the recommended route. For example, if a user attempts to travel from a source to a destination, we use tensor to work out the hot road segment set, and regroup it as a complete route according to collective popularity.

3. OVERVIEW

In this section, we first introduce the main structure of our method and then briefly introduce the preparatory work for this paper.

3.1 System Architecture

The architecture of our proposed method is shown in Fig. 1. It is composed of two major components: the preparatory work and user trajectory mining.

The first component operates offline and contains trip partition, stay point ex-traction, and map matching. The second part is the major work that contains the source-destination-road segment tensor construction, hot road segment set extraction, and collaborative route recommendation mining.

In order to better utilize GPS log data, we need to do some preparatory work. We first divide the whole GPS history into trips, which have multiple sources and multiple destinations. Then we extract stay points as either a source or a destination, and use map matching to align the GPS points that lie between the related source and destination on a digital map in order to obtain the corresponding road segments.

After the preparatory work, we construct a tensor with multiple sources, multiple destinations and the corresponding road segments. Then we use the tensor to figure out the hot road segment set for a collaborative route recommendation.

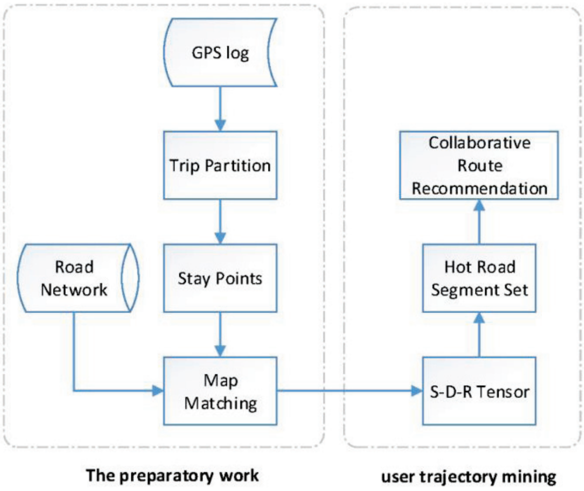

Figure 1 Overview of our system.

3.2 The Preparatory Work

A user's trajectory history is a log that records the user's movement over a period of time. We first divide the log into trajectory segment, where each segment represents a relatively short period of time. Table 1 summarizes the key notations for the ease of reference.

Table 1 Key notations

Symbol	Description
χ	trajectory history data tensor
P	a group of GPS raw data
S	stay point
$T_{\text{threshold}}$	the threshold of time interval
$D_{\text{threshold}}$	the threshold of distance interval
$\mathbf{U}$	identity matrix
$\mathbf{V}$	weight vector of source/destination

Definition 1: **GPS trajectory**. A GPS trajectory T is a time series of coordinates of mobile objects that neighbored points' time delta less than a time threshold ΔT. For example, T : $p_1 \rightarrow p_2 \rightarrow p_3 \ldots\ldots \rightarrow p_n$ where $0 < p_{i+1}.\ t - p_i.t < \Delta T$.

Next, we extract every possible stay point for every trajectory.

Definition 2: **Stay point**. A stay point is a central point that people stayed for a period of time. The calculation of the stay points relies on these key thresholds: the threshold of time ($T_{\text{threshold}}$) and the threshold of distance ($D_{\text{threshold}}$). A stay point S is a virtual location defined by a trajectory $T = \{p_1, p_2, p_3 \ldots\ldots p_n\}$ where $\forall 1 < i < n$, Distance $(p_1, p_i) < D_{\text{threshold}}$ and $p_n.t - p_1.t \geq T_{\text{threshold}}$. A stay point S can be expressed as $(lat, lon, arrT, levT)$: latitude, longitude, arrival time, and departure time.

$$s.lat = \sum_{i=1}^{n} p_i.lat/n \tag{1}$$

$$s.lon = \sum_{i=1}^{n} p_i.lon/n \tag{2}$$

$$s.arrT = p_1.t \tag{3}$$

$$s.levT = p_n.t \tag{4}$$

As shown in Figure 2, the stay point S is extracted from $\{p_3, p_4, p_5, p_6\}$. The position of S is the center of four GPS points, the arrival time of S is $p_3.t$ and the leave time of S is $p_6.t$.

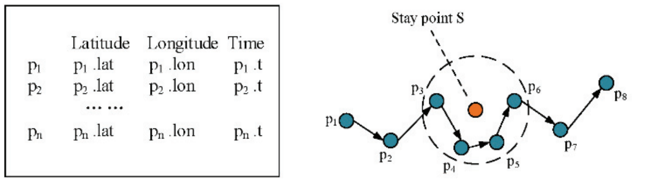

Figure 2 How to extract stay point.

Once we figure out all the stay points, we rearrange the stay points in this form: $(Sour_i, Dest_i)$. A GPS log may cover a long period of time and include travel from multiple sources and multiple destinations. $Sour_i$, $Dest_i$ are stay points and are time and spatially related. The time interval $Dest.arrT$-$Sour.levT$ is not over a single trip time threshold and we can trace the trajectory from the GPS points between source and destination. The whole GPS record is partitioned into pieces by the stay points. The destination point of one segment is the next segment's source point.

Because of the measurement error in GPS points, the observation of a GPS point can't indicate people's real location precisely. So we use map-matching to align these sequence of points on a digital map.

This alignment process yields intermediate results for source-destination pairs($Sour_i$, $Dest_i$) and the corresponding road segments set $\{R\} = \{r_1, r_2 \ldots r_m\}$. The road segment is the real road identity which is fetched from the map dataset. A road segment reflects the junction information and length information in the map. A source-destination pair($Sour_i$, $Dest_i$) is described by a group of road segments.

3.3 Tensor

Tensor is an object that describes linear relationship between vectors, scalars and other tensors. A tensor can be expressed as a N-way array, The order (or modes) of a tensor is the number of dimensions of the array. In this paper, the notation of tensor used is as from [14]. A user's trajectory history can be denoted as a three-order tensor.

Fibers are the sub-arrays of a high-order tensor. A fiber is similar to a single dimensional vector. For a user's trajectory tensor, it has column, row and tube fibers, see Figure 3, expressed as $\chi_{:jk}$, $\chi_{i:k}$, and $\chi_{ij:}$.

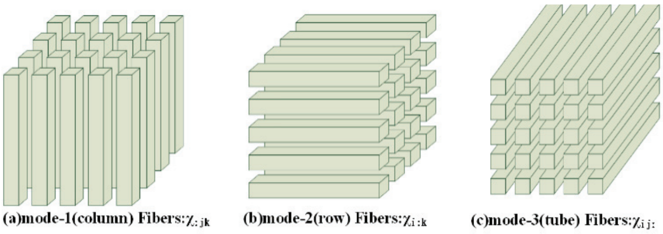

Figure 3 Tensor decomposition: Fibers.

Slices are the sub-matrixs of a high-order tensor. A slice is similar to a digital matrix. For a user's trajectory tensor, it has horizontal, lateral and frontal slices, see Figure 4, expressed as $\chi_{i::}$, $\chi_{:j:}$, and $\chi_{::k}$.

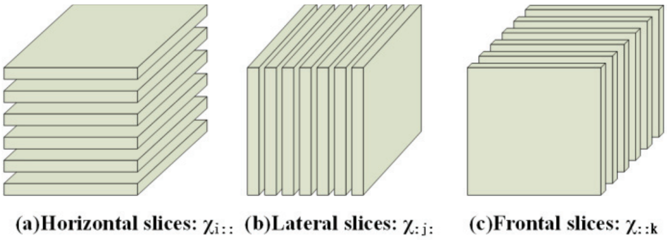

Figure 4 Tensor decomposition: Slices.

4. USER TRAJECTORY MINING

4.1 Modeling

After the preparatory work, we now have obtained the data for multiple sources and multiple destinations and their corresponding road segments. Now we construct an S-D-R tensor, or source-destination-road segment three-dimensional tensor χ_{sdr}, which provides a natural representation for such trajectory history data.

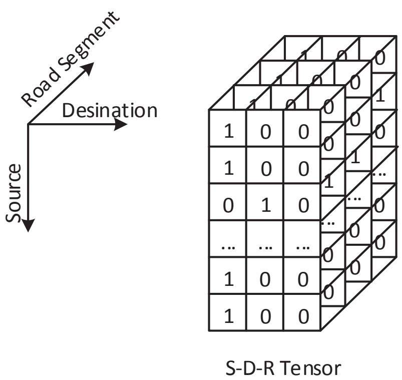

Figure 5 Source-destination-road segment three-dimensional tensor.

First, we summarize some of the key features of the S-D-R tensor. For a tube fiber of χ_{sdr}, the term $\chi_{sd:}$ denotes which road segments set is related for a given source and destination pair. For a horizontal slice of χ_{sdr}, the term $\chi_{s::}$ denotes that for a given source, the multiple destination and corresponding road segments that people traveled. As shown in Figure 6, for a lateral slice of χ_{sdr}, the term $\chi_{:d:}$ denotes that for a given destination, the corresponding sources and road segments. For a frontal slices of χ_{sdr}, the term $\chi_{::r}$ denotes the associated source and destination pair for a given road segment. As we can see from Figure 6, $\chi_{sd:}$ denotes the road segments connecting a given source-destination pair. The expression $\chi_{\mathrm{sdr}} = \{1\}$ means that road segment r belongs to the road segments set connecting source and destination, otherwise it is 0.

In our method, the basic element of "source" is the stay point. If we have a huge quantity of GPS trajectories and stay points, we can split the whole city map into a grid, and use a clustering algorithm to cluster the stay points. In this situation, the basic element of "source" can be turned into a grid region [22]. It also can be used for "destination", which we can adapt to trajectory data changes as time passes.

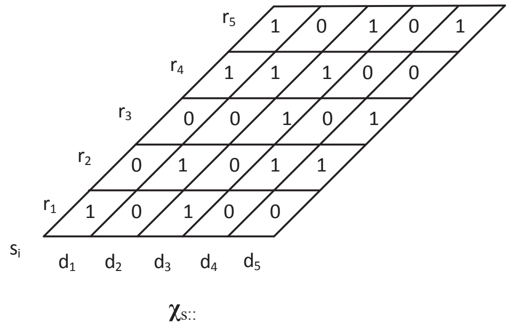

Figure 6 Example of $\chi_{s::}$ denotes for a given source, people go to multiple destination and the corresponding road segments.

Understanding the correlation between road segments may lead us to reason which route people would choose to travel from one source to one destination. For more efficient computing, we transform a tensor into a matrix.

The three mode-$n$unfolding are

$$\chi_{(1)} = \begin{bmatrix} x_{111} & \cdots & x_{1d1}x_{1112} & \cdots & x_{1dr} \\ \cdots & \cdots & \cdots\cdots & \cdots & \cdots \\ x_{s11} & \cdots & x_{sd1}x_{s12} & \cdots & x_{sdr} \end{bmatrix}$$

Figure 7 Unfolding of s-d-r tensor.

Tensor Multiplication: tensors can be multiplied together, the n-mode matrix of a tensor with a matrix. Each mode-n fiber is multiplied by the matrix U.

$$y = \chi \times_n U \Rightarrow Y_{(n)} = U_{\chi(n)} \tag{5}$$

As [9] has observed, the routes of different people overlap and interweave. When people plan the route, they usually base it on their own experience or the experience of others, in case we calculate the historical route collectively,there is a high probability that these dispersed data reform a more integrated route.

In this case, we mine the location-location correlations to aggregate the common road segments together. We use y_s to describe all the possible routes to a destination. For different sources, we present a source correlation matrix U_{is}, which (5) can expressed as

$$[y_s]_{idr} = \chi \times_{(1)} U = \sum_{z=1}^{s} U_{iz}\chi_{sdr} \tag{6}$$

U_{is} is initialized as an identity matrix. The term u_{ij} denotes the association between the two sources. If source i is adjacent to source j, it means the road segments of source i may be chosen to complement the road segments of source j. For every source i, $u_{ii} = 1$, and the more geographically close i and j are, the closer u_{ij} is to 1.

Complementing this, we have a similar definition of the destination correlation matrix U_{jd}, which works in the same way as the source correlation matrix described above, and (5) can be expressed as

$$[y_d]_{sjr} = \chi \times_{(2)} U = \sum_{z=1}^{d} U_{jz}\chi_{szr} \tag{7}$$

U_{jd} is initialized as an identity matrix. The term u_{ij} denotes the association between the two destinations. For every destination i, $u_{ii} = 1$, so the geographically closer i and j are, the closer u_{ij} is to 1.

V_s is a weight vector of multiple sources, and V_d is a weight vector of multiple destinations, from which we can cluster the road segment data, and then discover frequent road segment sets.

The n-mode product of a tensor with a vector can be represented as $\bar{\times} V$, which can reduce its dimensionality.

$$(\chi \bar{\times} V)_{i_1 \ldots i_{n-1} i_{n+1} \ldots i_N} = \sum_{i_n=1}^{I_n} x_{i_1} x_{i_2} \ldots x_{i_N} v_{i_N} \tag{8}$$

The proposed method is described in Algorithm 1. There are two main processes in our methods: Road weight calculation and Similarity searching. Given source and destination data as tensor χ_{sdr}, the method first generates source and destination location matrix from given route pairs.

Based on source and destination location matrix, we give the weight for each road. Then we begin to search for optimal route from source point. For each alternate route, we calculate the total weight of each route. Finally, the weight of each route describes the popularity that people usually choose. For the route query and comparison process, the route similarity measurement use (7) as the similarity function. There are three important subroutines in Algorithm 1: (*getRelatedRoad*, *comparison*, *routeSearching*).which are important to the effectiveness and efficiency of this algorithm. To put it simply, first we need to calculate the related roads of corresponding source and destination, then we search route base on related roads, which a little similar to depth-first search, finally we comprise multiple weights of different routes.

GetRelatedRoad: For a given source/destination pair, we try to find some similar history data, and then extract the related road segments. If two points (the given source/destination and historical stay point) are less than 500 meters apart, the weight of related road segments is 1, if the distance is more than 500 meters and less than 1000 meters, the weight is 0.5. If there are more than two points whose distance apart is less than 500m, we will choose the nearest one. If they are the same distance apart, we will randomly choose one point. Thus if a trajectory's history data is closer to the given source/destination, the weight of the related road segment is greater. The weight of a single road segment is cumulative. We sum up each weight for a single road to get the total weight. Sometimes, there are few or no related road segments for a given source/destination pair. In those cases, we try to figure out a route to keep the total length short with rare weights.

Algorithm1: Recommendation route computing

```
Input:dest,sour,χsdr
Output:route
  Mdr ← getRelatedRoad(sour, χsdr)
  Msr ← getRelatedRoad(dest, χsdr)
  roadWeight ← getWeightRoad(Mdr, Msr)
r ← the neighbor of sour
route ← r.sour
If r = r.dest then
  bulid route with road segment set
   TotalWeight ← sum of each weight of r ∈ route
   Comparison()
Else
   routeSearching()
nextr ← the neighbor of r
   if distance(route+next r)+δ<distance(route)
   route.append(next r) // for next recursion
δ ← a new distance threshold
totalLength.add(next r)
Return route
```

In daily life, it may seem impossible for the next road segment always more approach to our destination. Hence we take δ to be a tolerance detour distance that allows the following road segment to move a bit away from the destination which we expected. If the following road segment makes the distance from destination longer, we then reduce δ to make sure it is not too far away from the destination and make a stricter requirement for the next recursion. The threshold δ is to make sure that we would finally get to the destination instead of heading on the wrong track.

Comparison: From all routes we have obtained, we must choose one for the recommendation. So we give each route a score that expresses a combination of the distance, number of road segments and weight (route popularity).

4.2 Theoretical Analysis

The tensor decomposition model is more complicated than other former approaches. The time consumption depends on the number of road segments in a dataset. We denote the size of road segments as n. The route computing process needs to decompose the χ_{sdr} tensor.

Using other similar analysis from previous work [23][25], we can calculate the time complexity and space complexity of the decomposition procedure as $O(n^2)$ and $O(n^3)$.The time complexity and spatial complexity of the unfolding are $O(n)$ and $O(n^2)$ respectively. As a result, the time complexity and spatial complexity of our route computing procedure in Algorithm 1 are

O(n^2) and O(n^3).

The complexity of our proposed method makes it as efficient as FlowScan [8] and HRIS [9]. Those methods use two-cycle loops to find the hot route. As the tensor decomposition and unfolding are matrix operations, multiple source-destination pairs ($Sour_i$, $Dest_i$) can be calculated at the same time. Thus our method is more efficient and useful for multiple user recommendation applications than previous methods.

5. EXPERIMENT RESULTS

In this section, we start with the introduction of detail settings of experiment. And then explain our method for inference evaluation. Finally, we represent experiment results and compare with other route recommendation methods.The computational processes are written in Java, Visual Basic, and MATLAB. The configuration of our computer is a Pentium Dual-Core processor (2.6GHz) with 2GB memory.

5.1 Experimental Settings

Road Network: Our experiment uses a mapset of Peking, the capital of China (see Figure 7), which has 133,547 road segments in VB+MAPX 5.0.

Figure 8 Road network of Beijing.

Dataset: The users' historical GPS trajectories dataset was collected in the Geolife project by 182 people during a period of over five years [23].This dataset contains 17,621 trajectories with a total distance of 1,292,951 kilometers and a total duration of 50,176 hours. Only 8.5 percent of the data is low-frequency sampled.

5.2 Evaluation Approaches

Query: All queries have two points and each query is re-sampled to a source and a destination, which are stay points, and there are also consecutive GPS points between them.

Evaluation criteria: We make evaluation of the method by the quality of inference which we called hit rate.

Hit rate is a model that evaluated the similarity of the recommendation result R_R and query related road segments set.

$$HR = \frac{LCRG(R_R, R_E)}{R_E} \tag{9}$$

R_R represents the road segments of the recommended route, and R_E represents the corresponding road segments of the query before re-sampling. LCR is the longest common road segments between two data sets.

We use hit rate to measure users' satisfaction, if the recommended route is very similar to the original GPS history data, the hit rate will get a high value, and we can say that this is a satisfactory recommended route.

First, as shown in Figure 8, we list all the corresponding road segments (heavy red line segments) of the given source and destination (green dots). From this we can see that there are multiple routes between the source and destination on the digital map, and the road segments that compose the route have different weights based on the accumulation of multiple users' travel experiences.

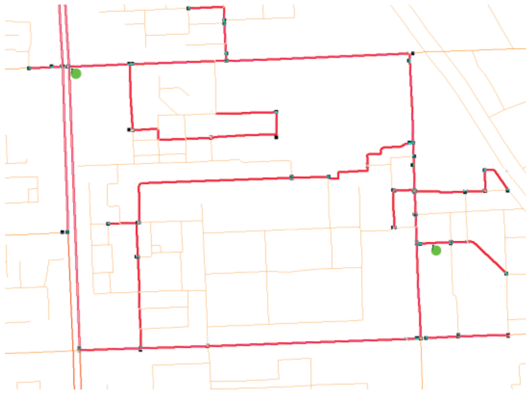

Figure 9 Corresponding road segments.

Second, Figure 9 presents the recommended route for the given source and destination. Considering the spatial information, we may have as many as three routes for recommendation. To avoid a situation where a route with lots of low weight road segments has a greater score than a route with fewer, high-weight road segments, we take the average weight of all road segments as a major factor when comparing the scores of a route.

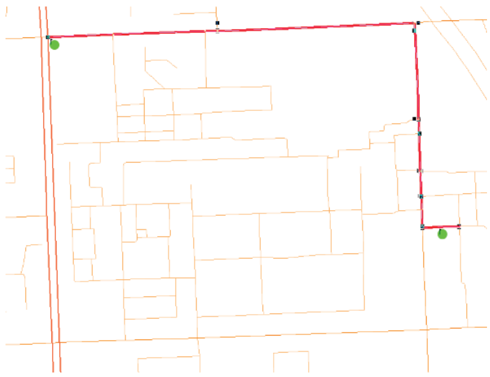

Figure 10 Recommendation route.

The weight of a single road segment is cumulative, based on GPS historical data. We tend to mine out the road segments of other experienced users rather than a single user's history data. For instance, if a user wants to go to somewhere, we would recommend a route that is frequently be selected by the majority of people rather than a route that the user had taken historically.

In the experiment, we have 5608 sets of data with more than 5 road segments, and we compute the hit rate based on the GPS trajectory dataset.

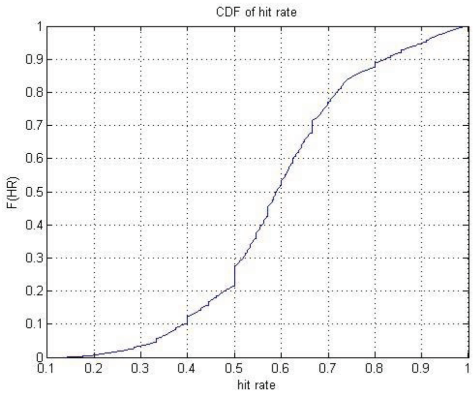

Figure 11 CDF of hit rate.

Figure 10shows the cumulative distribution function of the hit rate. The curve shows that about 50% of hit rates are greater than 60% and only 10% of hit rates are less than 40%.The mean of the hit rates is 59.73% and standard deviation is 0.1615.

The route recommendation methods can be categorized as three types [28]: text-based route recommendation [29], region-based route recommendation [30], and landmark-based route recommendation [31]. The accuracy of recommendation area is limited by the area descriptions. The proposed method uses road segments as area description, so it has the highest accuracy range as shown in Table 2.

Table 2 Accuracy range for different recommendation methods.

Method	Accuracy range
Text-based	500m
Region-based	100m
Landmark-based	500m
Proposed	50 m

The result shows that for most recommended routes, we obtain considerable accuracy for each query and, using crowd-sourcing techniques, recommend some popular road segments for users to choose.

5.3 Discussion

The proposed method can be applied in many route recommendation applications. It can be used for pedestrians as well as for vehicles. The popularity of a route may result in congestion for the recommended vehicle route, because many cars may choose the same route at the same time. This problem can be solved by limiting the recommendation frequency. The application will select multiple routes as alternative recommendation collocations. Those alternative routes will be recommended to the end user randomly.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we investigate the problem of discovering a hot or optimal route for a given source and destination, by extracting tensor structure from trajectory history data and taking road segment popularity as a major factor when computing the route recommendation. To accomplish this goal, we have proposed a method which includes several novel algorithms to increase the effectiveness of computational performance. In our view, road segments are the basic trajectory block for spatial user trajectory mining.

In the near future, our team intend to improve the efficiency of route searching, search for more attractive candidates such as the optimum efficiency route (the route with minimal popularity that can avoid traffic congestion). We also intend to take more factors in our method, including real-time traffic conditions, weather and points of interest, to strengthening the effects of proposed work. In addition, we try to handle the challenging problem of targeted advertising base on our method.

Acknowledgement

The work is partly supported by NSFC (No.61472149), the Fundamental Research Funds for the Central Universities (2015QN67) and the National 863 Hi-Tech Research and Development Program under grant (2015AA01A203).

REFERENCES

1. Artem Katasonov. "User-centric data querying for location-based services". International Journal of Computer Systems Science and Engineering, 20(2). March 2005.
2. App Store - Find My Friends. Application Store. Apple Inc https://itunes.apple.com/us/app/find-my-friends/id466122094
3. X. Zhou, K. Zheng, H. Jueng, J. Xu, and S. Sadiq, "Making Sense of Spatial Trajectories", In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 2015, pp. 671-672.
4. C. Yu, Y. Liu, D. Yao, L.T. Yang, H. Jin, H. Chen, and Q. Ding,"Modeling User Activity Patterns for Next-Place Prediction", *IEEE Systems Journal*, doi: 10.1109/JSYST.2015.2445919, 2015.
5. F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi. "Trajectory pattern mining", In *Proceedings of SIGKDD*, 2007, pp. 330–339.
6. Waluyo A B, Srinivasan B, Taniar D, "Research on location-dependent queries in mobile databases". International Journal of Computer System Science & Engineering, 2005, 20(2): pp.79-95.
7. Kumar V, Dunham M H, Prabhu N. "Mobilaction: A mobile transaction framework supporting spatial replication and spatial consistency". International Journal of Computer Systems Science & Engineering, 2005, 20(2): pp. 117-131.
8. X. Li, J. Han, J. Lee, and H. Gonzalez, "Traffic density-based discovery of hot routes in road networks", In *Proceedings of the 10^{th} International Symposium,SSTD* 2007, *LNCS*, vol.4605, pp. 441–459.

9. K. Zheng, Y. Zheng, X. Xie, and X. Zhou, "Reducing uncertainty of low-sampling-rate trajectories", In *Proceedings of the IEEE 28th International Conference on Data Engineering*, 2012, pp. 1144-1155.
10. Y. Hu, S. Liao, D. Yi, Z. Lei, andS.Z.Li, "Multi-camera trajectory mining: database and evaluation", In *Proceedings of the 22nd IEEE International Conference on Pattern Recognition*, 2014, pp. 4684-4689.
11. D. Gavalas,C. Konstantopoulos, K. Mastakas, and G. Pantziou,"Mobile recommender systems in tourism", *Journal of Network and Computer Applications*, vol.39, pp. 319-333, 2014.
12. L. Baltrunas, B. Ludwig, S. Peer, and F. Ricci,"Context relevance assessment and exploitation in mobile recommender systems", *Personal and Ubiquitous Computing*, vol.16, no.5, pp.507-526. 2012
13. D. Yao, C. Yu, A.K. Dey, C. Koehler, G. Min, L.T. Yang, and H. Jin, "Energy efficient indoor tracking on smartphones", *Future Generation Computer Systems*, vol.39, pp. 44-54, 2014.
14. T. G. Kolda and B. W. Bader, "Tensor decompositions and applications", *SIAM review*, 2009, pp. 455-500.
15. A. Cichocki, D. Mandic, L. Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H.A. Phan, "Tensor decompositions for signal processing applications: From two-way to multiway component analysis", *IEEE Signal Processing Magazine*, vol.32,no.2, pp.145-163, 2015
16. A. N. Langville and W. J. Stewart, "Kronecker product approximate precondition for SANs", *Numerical Linear Algebra with Applications*, vol.11, pp. 723-752, 2004.
17. W. Hackbusch andB. N. Khoromskij, "Tensor-product approximation to operators and functions in high dimensions", *Journal of Complexity*, vol.23, pp. 697-714, 2007
18. D. Yao, C. Yu, H. Jin, and J. Zhou,"Energy efficient task scheduling in mobile cloud computing", In *Proceeding of the 10^{th} IFIP International Conference on Network and Parallel Computing*, 2013, pp. 344-355.
19. B.W. Bader, R. A. Harshman and T. G. Kolda, "Temporal analysis of semantic graphs using ASALSAN", In *Proceedings of ICDM*, 2007, pp. 33-42.
20. V.W. Zheng, Y. Zheng, X. Xie, and Q. Yang, "Towards mobile intelligence: Learning from GPS history data for collaborative recommendation", *Artificial Intelligence*, vol.184, pp.17-37, 2012.
21. D. Yao, C. Yu, H. Jin, and Q. Ding, "Human mobility synthesis using matrix and tensor factorizations", *Information Fusion*, vol.23, pp. 25-32, 2015.
22. T.M.T. Do and D. Gatica-Perez, "Contextual conditional models for smartphone-based human mobility prediction", In *Proceedings of Ubicomp*, 2012, pp. 163-172.
23. Y. Zheng, X. Xie, andW. Ma, "GeoLife: A Collaborative Social Networking Service among User, location and trajectory", *IEEE Data Engineering Bulletin*. vol.33, pp. 32-40, 2010.
24. R. T. Ledyard, "Some mathematical notes on three-mode factor analysis", *Psychometrika*, vol.31, pp. 279-311, 1966.
25. B. Giuseppe, "Spatial data mining for highlighting hotspots in personal navigation routes", *Int. J. of Data Warehousing and Mining*, vol.8, pp. 45-61, 2012.
26. C. Yin, Z. Xiong, H. Chen, J. Wang, D. Cooper, B. David, "A literature survey on smart cities", *SCIENCE CHINA Information Sciences*, vol.58, no.10, pp.1-18, 2015
27. L. Yao, Q.Z. Sheng, Y. Qin, X. Wang, A. Shemshadi, and Q. He, "Context-aware Point-of-Interest Recommendation Using Tensor Factorization with Social Regularization", In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2015, pp. 1007-1010
28. H. Yin, C. Wang, N. Yu, and L. Zhang, "Trip mining and recommendation from geo-tagged photos", In *Proceedings of IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, 2012, pp. 540-545.
29. D. Yang, D. Zhang, Z. Yu, and Z. Wang, "A sentiment-enhanced personalized location recommendation system", In *Proceedings of the 24th ACM Conference on Hypertext and Social Media*, 2013, pp.119-128.
30. B. Hu, M. Jamali, and M. Ester, "Spatio-temporal topic modeling in mobile social media for location recommendation". In *Proceedings of IEEE 13th International Conference on Data Mining (ICDM)*, 2013, pp. 1073-1078.
31. T. Kurashima, T. Iwata, G. Irie, and K. Fujimura, "Travel route recommendation using geotags in photo sharing sites", In *Proceedings of the 19th ACM International Conference on Information and knowledge Management*, 2010, pp.579-588.
32. H Zhu, M Dong, S Chang, "ZOOM: Scaling the Mobility for Fast Opportunistic Forwarding in Vehicular Networks". In *Proceedings of IEEE INFOCOM*, 2013, pp.1073-1078.
33. Ota K, Dong M, Zhu H, et al, "Traffic information prediction in Urban Vehicular Networks: A correlation based approach", In *Proceedings of the IEEE Wireless Communications and Networking Conference*, 2011, pp. 1021-1025.