Predicting Next Points of Interests Based on a Markov Model

Jie Xu(✉), Chunxiao Xing, and Yong Zhang

Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
xujl5@mails.tsinghua.edu.cn,
{xingcx, zhangyong05}@tsinghua.edu.cn

Abstract. With the development of Global Position System (GPS) technology, the analysis of history trajectory becomes more and more important. The Location Based Service (LBS) can provide the user's location, the human movement location prediction from the history observations over some period have several potential applications and attract more and more attention. Predicting the user's next position usually includes finding the Points of Interests (POIs) from the historical trajectory and predicting the position with a certain statistical model. In this paper, we present a novel method based on Markov chain for prediction, our method include two contributions: the first one we use GEPETO variant algorithm to cluster for POIs to solve the former algorithm without considering the temporal factor, and the second one we present Mobility Markov Chain (MMC) model which exploits 3 previous states to infer the future location. Our experiments basing on the real Beijing trajectories dataset display that our algorithm can improve the prediction accuracy compared with the baseline algorithm.

Keywords: Next location prediction · Markov chain model · Cluster GPS trajectory analysis

1 Introduction

With the modern geographic position technologies, more and more mobility locations are stored and shared [1]. The researches of the location through Global Positioning System (GPS) devices both from the industry and the research community with sensor, RFID or satellite, have been attracting lots of attention for its application in many different domains [2], such as using the complex sequential location data to observe the individual's activities motive or finding the POIs basing on the past trace, or judging someone whether are at home or at business sites currently [3]. It is important to provide location forecast services accurately, this makes moving object trajectory prediction an active research field [4, 5].

In this paper, we solve the problem of inferring the next location basing on the history mobility traces using the Markov model. This algorithm has two main steps, one is finding the POIs [6], the other is prediction using statistical model. Human historical location are first clustered with their temporal and spatial properties, and then

I. Romdhani et al. (Eds.): CollaborateCom 2017, LNICST 252, pp. 264–274, 2018.
https://doi.org/10.1007/978-3-030-00916-8_25

the previous formed clusters are employed to train the Markov model. More accurately, in our method we present a different cluster method named GEPETO [7] variant taking into account the temporal factors before applying in the Mobility Markov Chain (MMC) [8]. The construct of our approach is present al Fig. 1. Furthermore, we assess the efficiency of our algorithm on a real location mobility dataset named Beijing taxi dataset, and the results demonstrate that our prediction under different configurations can achieve an accuracy for predicting the next location in the range of 60% to 70%.

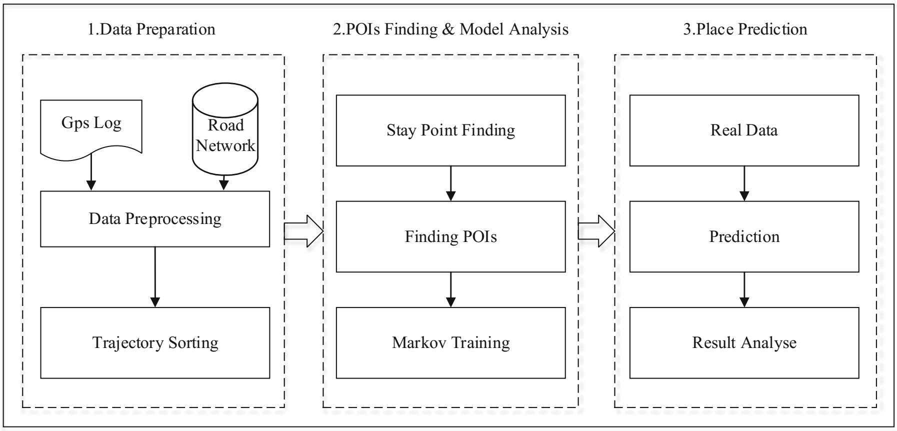

Fig. 1. Overview of system architecture

The rest of this paper is organized as follows: we propose a related important work in Sect. 2, and then discuss the methods of cluster among different scenery in Sect. 3, Afterwards, we describe how to infer the next place prediction with MMC in Sect. 4, experimental results are assessed in Sect. 5, and finally we summarize the paper in Sect. 6.

2 Related Work

Many previous work have been proposed to solve the location prediction. In this section, we introduce the related work and analyze the key technologies for different algorithms.

Ashbrook and Starner [3] proposed a method for predicting the next station by using the history POIs, they firstly clustered the location which are frequently visited by multi–user, secondly, they merged these POIs into a Markov model, the transition probability between different POIs represented the different Markov node transition probability. Our work is similar as the ideas [8], the differences lie in the cluster algorithm, the cluster algorithm is very important for prediction accuracy, POIs can be calculated by different methods, such as k-means, DBSCAN. We choose the GEPETO variant which not only considers the minimal threshold number of cluster, but also adds the temporal factors as input parameters.

Qiao [9] proposed a self-adaptive parameter selection algorithm called HMTP, in their approach, to avoid time-intensive distance computation between trajectory points,

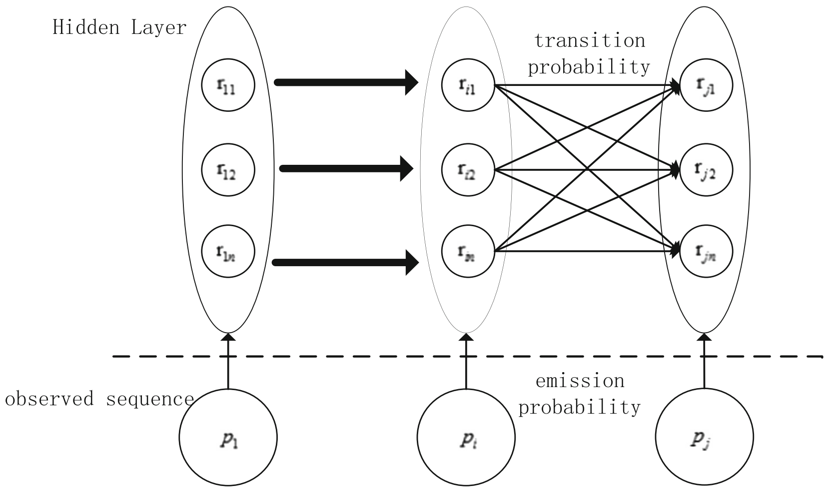

Fig. 2. Hidden Markov model

a density-based trajectory clustering algorithm was introduced. Second, they partitioned trajectory into segments to extract trajectory hidden states [10], third, they captured the parameters necessary for real-world scenarios in terms of objects with dynamically changing speed (Fig. 2).

Asahara [11] proposed an algorithm for predicting pedestrian movement on the basic of Mixed Markov chain mode (MMM), they took a pedestrian's personality and previous states as an unobservable parameter. Observable probability and the transition probability were simultaneously calculated, the user's next position was inferred by using the statistic model and the user's tracking data [12], by using the Expectation Maximization algorithm, the pedestrian's next location with the maximization likelihood was the most probable one.

Some other methods to predict the next position used the trajectories semantics. Krumm and Horvitz [13] presented a measure named predestination which made use of a history of a driver's destinations together with the driving behaviors to predict where the driver was going. They separated behaviors into four different probabilistic cues and then combined them to produce most likely cell destination, afterwards, they availed an open world modeling methodology by using the likelihood of visited unobserved locations based on the background properties of locations and on the trends in the data. The advantage lay in transferring the "out of the box" data into the fully trained data set more smoothly.

Chen [14] put forward an approach to predict both the future destination and the intended route of a person, rather than predicted each other separately. To find the POIs, they made use of a cluster algorithm named FBM (Forward–Backward Matching), and abstracted space partitioning and movement patterns by taking advantage of an extended CRPM (Continuous Route Pattern Mining). From these movement patterns, a pattern tree was built, and the tree was the core method for predicting the future destination and the intended route.

3 Cluster Algorithm

Our method include two steps. Firstly, for the sake of finding the POIs [20], a cluster algorithm is introduced, after that, the transition probability among POIs is calculated. During the second step, a prediction method for next place basing Markov mode is presented. In this section, we first discuss the methods of our cluster algorithm.

K-Means [15, 16]: the well known cluster algorithm is the K-Means which aims to partition n observations into k clusters, the K-Means algorithm takes advantage of the minimum squared distances error for each point to its cluster center, the error term is formal as follows:

$$\mathrm{Error} = \sum_{i=1} \sum_{x \in C_i} d\left(x_{i,} \mathrm{y}_i\right)$$

where the y_i indicates the center of each cluster C, the function d indicates the distance between the point x_i and the center. The algorithm firstly defines the number of k cluster, and then iterates computer distance between each candidate point and the center, decides the x_i point belongs to which cluster until the error is small enough.

Density-Based Clustering [16]: the K-Means has some disadvantage for assorting POIs, the biggest one lies on the algorithm in advance needs to know the number of cluster, this is much difficult for the user. To overcome the difficulty for forming arbitrary shape and reducing the influence of the point noise in the computation process, density-based clustering approach is introduced, this algorithm makes use of two parameters: Eps indicates the radius of a circle, MinPts indicates the minimum number of points in the circle. The algorithm first searches each point neighborhood through the database, if the neighborhood points contain sufficient MinPts, then creates a new cluster. If a point is a dense part of a cluster, and its neighborhoods are also part of that cluster, so iteratively adds all points which are found within the threshold ε until there are no new points can be added to any cluster.

Density-Time Cluster: (DT Cluster) [17] is an iterative clustering algorithm which is dependent on two scale parameters: for time threshold t and for spatial distance threshold d from a trail of mobility traces M. First, the algorithm constructs a cluster C, which includes all successive points within threshold distance d from each other. Second, the algorithm checks whether the accumulation time in the moving range is greater than the threshold t. If the condition is true, then creates a cluster to add to the POIs list. For a specific geographic delta scale, if two clusters centroids are within the delta scale, they will be merged.

In our paper, we use the GEPETO variant [7] to cluster the POIs from the trajectory. The GEPETO's inspiration stems from the Density-time cluster (DT Cluster), they are similar to each other. The methods takes in the radius r, the tolerance rate ι, the time window t, the distance threshold d and a trajectory point M as parameters. The algorithm first constructs an iterative cluster from trajectory M which is located in the time window t, then obtains the total number of points and if the distance between the point and the cluster centroids are less than the tolerance ι, adds the cluster to the list of L, otherwise simply discards. At last, the algorithm merges the clusters whose centroids are less than the threshold d value. In some cases, when within a specified radius, we may obscure prediction opportunities on a big radius area, for example, if we predict next location from small town, the accuracy definitely decrease from the city, so we narrow the radius down until the value is 0.1 km. The GEPETO [7] variant algorithm pseudocode is presented as follows:

Algorithm 1. GEPETO variant clustering algorithm

```
Parameters: trajectory T, time window t, radius r, tolerance rate ι , distance threshold d
discard the fast point whose speed is greater than 30Km/h
L put iu POIs; a empty cluster C
for i = 0 to S    do
totaltime = totaltime + (T[i + 1].time -T[i].time)
if totaltime <= t then
   put the trajectory T[i] to cluster C
else
   Compute the centroid of C
while(radius>0.1)            //find the sublocation from the fatherlocation
          cumulTime = cumulTime + (T[i + 1].time -T[i].time)
    if cumulTime <= t then
      Add the mobility trace T[i] to cluster C
    if C is not empty,
             C_sublocaion=C
    else    do nothing
    radius=radius/2;
C=C_sublocaion ; nOutlierradius = 0
for j = 0 to C.number do
if distance(C[j], C.centroid) > r then
    nOutlierradius = nOutlierradius + 1
end if
end for
if nOutlierradius /totalPointsNumber <ι then
    put the cluster C to L
end if
    totaltime= 0; Empty(C)
end if
end for
Merge clusters L if two clusters' centroids distance is less than distance threshold d
return
```

4 Next Place Prediction

In this section, we briefly discuss the notion of Mobility Markov Chain of several users' behavior which is treated as a discrete stochastic process. Once we get the POIs computed from GEPETO variant, we can define a set of states P = {…pi − 1, pi, pi + 1…}, each state pi means a frequent POIs and corresponds to a semantic geographical location, such as home, work, entertainment place, others place which sites are the people daily visit. We also can get the density of the clusters, the radius, and the stay time of mobility trace. A set of transitions, for instance, the phome -> pwork represents the transition probability from home to work. If the user never moves between two places, the transition probability is set to zero.

So as to predict the next place using the Markov mode, the transitions between different POIs can be represented as a transition matrix base on chronological statistical model, mention from [8], standard MMC forecasting future position depends only on the current location without considering the past states, however there is a semantic correlation between former state transitions, using a single current state means information losses and less accuracy. To solve this problem, we use 3 previous states in which the next state transition probability is dependent on the current state and the previous 2 states, which approach yields more precise predictions. More specially, the home, the work, the Entertainment place, others, if the probability from the 3 previous states to the predicting next location exists, the probability value is correspondingly assigned. The matrix is shown as follows:

From the Table 1, the second column stands for the current and two previous states, the reminder four columns for the prediction location represent the home, work,

Table 1. 3 previous states transition matrix

Order	Source/destination	H	W	E	O
1	WHE	1.00	0.00	0.00	0.00
2	WHO	0.90	0.00	0.10	0.00
3	WEH	0.00	0.86	0.10	0.04
4	WOH	0.00	0.76	0.24	0.00
5	HWE	0.39	0.00	0.00	0.61
6	HWO	0.38	0.10	0.52	0.00
7	HEO	0.91	0.00	0.09	0.00
8	HOW	1.00	0.00	0.00	0.00
9	HOE	0.80	0.00	0.00	0.20
10	EHO	0.00	0.00	0.00	0.00
11	EHW	0.90	0.00	0.00	0.10
12	EWH	0.00	0.90	0.00	0.10
13	EHW	0.92	0.00	0.00	0.08
14	EOH	0.00	1.00	0.00	0.00
15	OWH	0.07	0.93	0.00	0.00
16	OEH	0.00	1.00	0.00	0.00

entertainment, other location. The corresponding value is the transition probability. Take the 1st row for example, the transition probability from the WHE to the prediction H is 1.00, while to the remaining place is zero, meaning the probability is very small and can be nearly ignore. For some kinds of visited sequence, such as the WEO is rarely appear, so this kind of situation is discarded.

5 Experiment Result

In this section, we first introduce our experiments on real trajectory dataset, and second evaluate the accuracy of our prediction algorithm for different algorithm. We implement our experiment on Beijing trajectory dataset, this trajectory dataset contains 118 users' different trajectories over a period of 2 years, every sequence item includes longitude, latitude, direction, velocity, timestamp information, sampling frequency ranged from 10 s to about 5 min.

In order to evaluate the performance of algorithm, we make use of the 88 users' trajectory as labeled data for training the MMC model while the remainder as the test case. we select the HMM [18], MMM [11], MMC(2) [8] as the baseline algorithm, the MMC(2) means the MMC algorithm use two previous states.

Our experiments are implemented in python27 and performed with 16 GB RAM. We compare the precision, recall and F1 measure which are defined as follows [19]:

$$Precision = \frac{True\_positives}{True\_positives + False\_positives} \tag{1}$$

$$Recall = \frac{True\_positives}{True\_positives + False\_negatives} \tag{2}$$

$$F_1{-}measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3}$$

The precision is the ratio between the number of correct predictions over the total number of predictions, the True_positives means the number of predicted as a positive sample, the False_positives is predicted as a positive class which is false, the False_negatives is forecasted as a false class which is true. In our experiment, POIs are deem to be true if predicted with the real location. According the definitions, the algorithm is "good" meaning the precision and the recall are both high.

The prediction precision of the different algorithms are displayed in Fig. 3, the HMM achieved 0.21, MMM achieved 0.37, MMC(2) achieved 0.57, our algorithm achieved 0.635.

As can be seen from Figs. 3 and 4, the HMM is not high, owing to the reason that the algorithm only deel with the transitions of unobserved states, and deems the user can go to anywhere if he want to, this situation is not realistic.

The MMM reach the 0.37, lower than the MMC, because the MMC don't take the temporal factor into consideration.

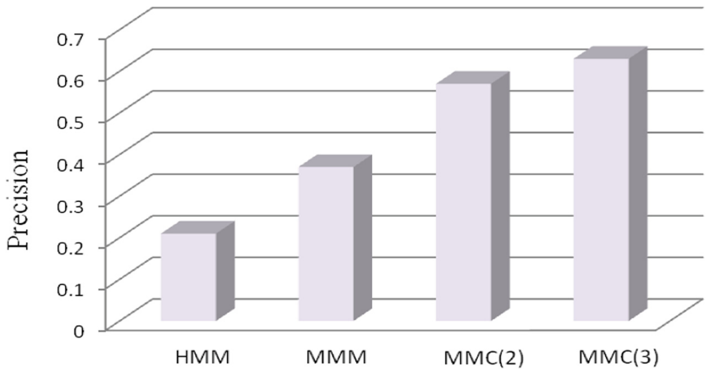

Fig. 3. Compare of precision between algorithms

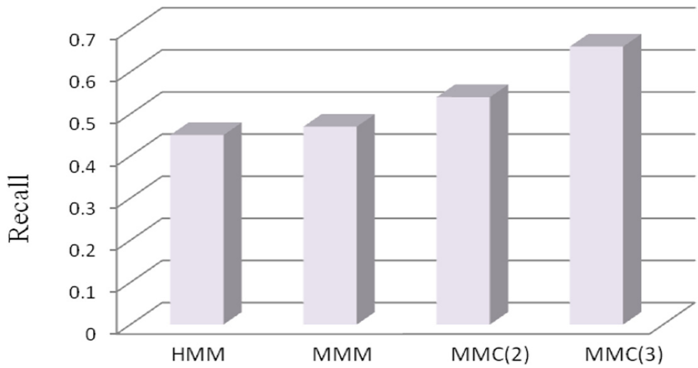

Fig. 4. Compare of recall between algorithms

The MMC(2) is lower than our algorithm, because we choose a GEPETO cluster for considering the temporal and the geographical factor, and for three previous states, we can take advantage of more historical information.

F1-measure: The F1 score is a measure to test the accuracy form both the precision score and the recall score, the F1 value can be interpreted as a weighted average of the precision rates and recall rate, where the F1 value achieves the optimal value at 1 and the worst at 0. From the Fig. 5, the MMC(3) gains the highest value at 0.64, this is in accordance with the precision rate and recall rate.

So as to explain the importance of the cluster algorithm, we also implement different cluster algorithm to find the POIs, we select the DBSCAN, K-Means, DJ, for the baselines, the accuracies are shown as the Fig. 6.

From the Fig. 6, we can see that DBSCAN, are lower than the DT and GEPETO variant, the K-Means is the lowest one, the reason lies in we can't give the suitable default number of cluster, if the k value is too big, the invalid POIs are included. The DT result is a little smaller than GEPETO variant, this can be explained that a tolerance rate ι which is used to control the centroids between clusters decides what

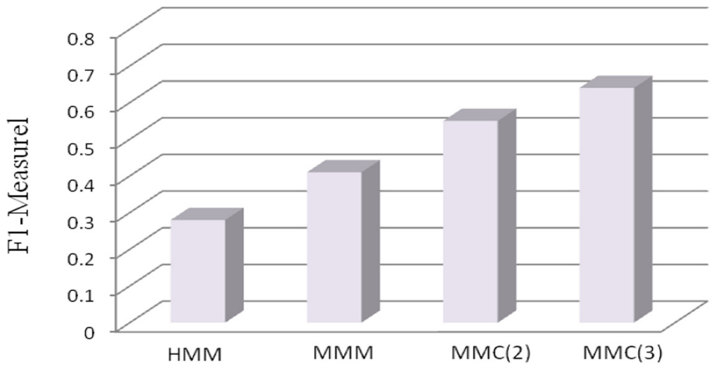

Fig. 5. Compare of F_1 measure between algorithms

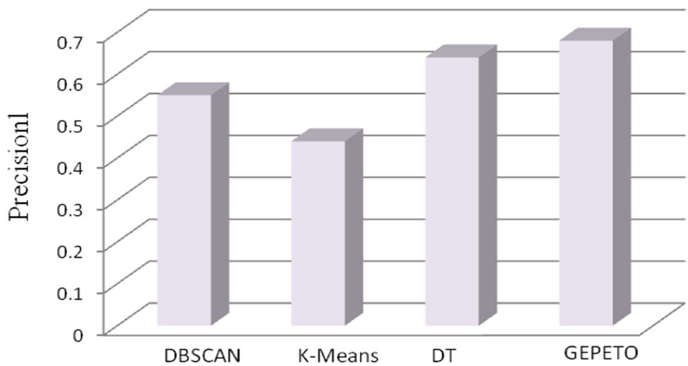

Fig. 6. Compare of precision between different cluster algorithms

kinds of cluster are discarded and merged, and the GEPETO variant cluster algorithm gains the higher accuracy rate in accordance with our expectations.

6 Conclusion

We have presented a novel prediction algorithm basing on Markov model chain which contain two steps, one is finding the POIs, and the other is training the MMC predicting model. First, to solve the previous cluster algorithm which don't consider the temporal factor, we select a GEPETO variant as our cluster algorithm. Second, for making better use of historical data, we exploit 3 previous states to infer the future location using the MMC algorithm.

For different implements on real trajectory, experiments results show that our approach can reach 0.635 accuracy rate than the baseline HMM, MMM, MMC(2) algorithm. In terms of our future work, we will apply our model to lane changing vehicle prediction, indeed, this is useful for the upcoming self-driving car area.

References

1. Zheng, Y., Li, Q., Chen, Y., et al.: Understanding mobility based on GPS data. In: Proceedings of the 10th International Conference on Ubiquitous Computing, pp. 312–321. ACM (2008)
2. Liu, P., Kurt, A.: Trajectory prediction of a lane changing vehicle based on driver behavior estimation and classification. In: 2014 IEEE 17th International Conference on Intelligent Transportation Systems (ITSC), pp. 942–947. IEEE (2014)
3. Ashbrook, D., Starner, T.: Using GPS to learn significant locations and predict movement across multiple users. Pers. Ubiquit. Comput. **7**(5), 275–286 (2003)
4. Lee, J.G., Han, J., Li, X., et al.: TraClass: trajectory classification using hierarchical region-based and trajectory-based clustering. VLDB Endow. **1**(1), 1081–1094 (2008)
5. Nascimento, J.C., Figueiredo, M.A.T., Marques, J.S.: Trajectory classification using switched dynamical hidden Markov models. IEEE Trans. Image Process. **19**(5), 1338–1348 (2010)
6. Liu, X., Liu, Y., Aberer, K., et al.: Personalized point-of-interest recommendation by mining users' preference transition. In: Proceedings of the 22nd ACM International Conference on Information and Knowledge Management, pp. 733–738. ACM (2013)
7. Gambs, S., Killijian, M.O., del Prado Cortez, M.N.: Show me how you move and I will tell you who you are. In: Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS, pp. 34–41. ACM (2010)
8. Gambs, S., Killijian, M.O., del Prado Cortez, M.N.: Next place prediction using mobility markov chains. In: Proceedings of the First Workshop on Measurement, Privacy, and Mobility, p. 3. ACM (2012)
9. Qiao, S., Shen, D., Wang, X., et al.: A self-adaptive parameter selection trajectory prediction approach via hidden Markov models. IEEE Trans. Intell. Transp. Syst. **16**(1), 284–296 (2015)
10. Ling, Z.H., Richmond, K., Yamagishi, J.: An analysis of HMM-based prediction of articulatory movements. Speech Commun. **52**(10), 834–846 (2010)
11. Asahara, A., Maruyama, K., Sato, A., et al.: Pedestrian-movement prediction based on mixed Markov-chain model. In: Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 25–33. ACM (2011)
12. Govea, A.D.V.: Incremental Learning for Motion Prediction of Pedestrians and Vehicles. Springer, Berlin (2010). https://doi.org/10.1007/978-3-642-13642-9
13. Krumm, J., Horvitz, E.: Predestination: inferring destinations from partial trajectories. In: UbiComp 2006: Ubiquitous Computing, pp. 243–260 (2006)
14. Chen, L., Lv, M., Chen, G.: A system for destination and future route prediction based on trajectory mining. Pervasive Mob. Comput. **6**(6), 657–676 (2010)
15. Zhou, C., Frankowski, D., Ludford, P., et al.: Discovering personal gazetteers: an interactive clustering approach. In: Proceedings of the 12th Annual ACM International Workshop on Geographic Information Systems, pp. 266–273. ACM (2004)
16. Yuan, G., Sun, P., Zhao, J., et al.: A review of moving object trajectory clustering algorithms. Artif. Intell. Rev. **47**(1), 123–144 (2017)
17. Hariharan, R., Toyama, K.: Project Lachesis: parsing and modeling location histories. In: Egenhofer, M.J., Freksa, C., Miller, H.J. (eds.) GIScience 2004. LNCS, vol. 3234, pp. 106–124. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30231-5_8
18. Mathew, W., Raposo, R., Martins, B.: Predicting future locations with hidden Markov models. In: Proceedings of the 2012 ACM Conference on Ubiquitous Computing, pp. 911–918. ACM (2012)

19. Yang, J., Xu, J., Xu, M., et al.: Predicting next location using a variable order Markov model. In: Proceedings of the 5th ACM SIGSPATIAL International Workshop on GeoStreaming, pp. 37–42. ACM (2014)
20. Zheng, Y., Zhang, L., Xie, X., et al.: Mining interesting locations and travel sequences from GPS trajectories. In: Proceedings of the 18th International Conference on World Wide Web, pp. 791–800. ACM (2009)