Pick-Up Point Recommendation Using Users' Historical Ride-Hailing Orders

Lingyu Zhang[1,2], Zhijie He[2], Xiao Wang[2], Ying Zhang[2], Jian Liang[2], Guobin Wu[2], Ziqiang Yu[3], Penghui Zhang[4], Minghao Ji[4], Pengfei Xu[4], and Yunhai Wang[1](✉)

[1] School of Computer Science and Technology, Shandong University, Qingdao, China
cloudseawang@gmail.com
[2] Didi Chuxing, Beijing, China
{hezhijie_i,yingzhangying,liangjian,wuguobin}@didiglobal.com, wangxiao@didichuxing.com
[3] Yantai University, Yantai, Shandong, China
zqyu@ytu.edu.cn
[4] School of Information Science and Technology, Northwest University, Kirkland, USA
pfxu@nwu.edu.cn

Abstract. The ride-hailing app must provide users with appropriate pick-up points when they submit their travel demands and their locations are recognized, efficiently reducing users' operation complexity and optimizing the software performance. Most apps currently try to search for locations near users' current GPS locations as the Points of Interest (POIs), which is an efficient method of locating, but seriously ignores personal preferences. In this paper, we deeply analyze the historical ride-hailing orders of users on Didi Chuxing platform (http://www.didiglobal.com). We explore the given dataset, get the general regularity of users' commuting, and propose a Pick-Up Points Recommendation Model (PPRM) based on the clustering algorithm. We cluster users' historical orders using Density-Based Spatial Clustering of Applications with Noise (DBSCAN) according to orders' spatial information. In this way, the candidate outputs closest to the user's current environment/feature can be found in a specific category. The linear addition of the candidate outputs severs as the final pick-up point provided. Therefore, our model can offer recommendations of the best pick-up points. In addition, experimental results based on real-world datasets indicate that our model can efficiently and accurately provide users with optimal points.

Keywords: Pick-up point recommendation · Travel pattern mining · Cluster analysis · Ride-hailing system · Data analysis

1 Introduction

Pick-up point recommendation is one of the essential functions of ride-hailing apps. With the increasing of users' historical ride-hailing orders, it is vital to dig out users' travel patterns from users' historical travel records and recommend the optimal pick-up points that meet their preferences.

Location prediction has been a craze for a long time. Numerous location data (such as trajectories, social network sign-in data, and location information obtained by

L. Wang et al. (Eds.): WASA 2022, LNCS 13472, pp. 393–405, 2022.
https://doi.org/10.1007/978-3-031-19214-2_33

various smart terminals) covers the users' travel characteristics and can be used to predict the following locations of users [14,20]. For instance, Li et al. [9] and Yoon et al. [23] predicted individual locations by calculating the similarity of moving behaviors and trajectories, respectively. Tseng et al. [16] tried to mine the mobile sequential patterns related to users' movement paths and time intervals and predicted users' following locations. The space areas can be divided by the Voronoi diagram, and a Markov location prediction model [12,15] based on regional features of users' movements is proposed. Furthermore, based on the trajectory data generated by smartphones and wearable smart devices, Kown et al. [4] proposed a location prediction method via pattern matching and similarity measurement. In addition, Zhang et al. [25] deeply mined the users' moving patterns and used the generated moving rules to predict the following locations.

The migration of users' locations has temporal and spatial correlation, and exploring the temporal and spatial patterns is essential for accurate location prediction. Lei et al. [6] proposed a spatiotemporal trajectory model, which could capture the spatiotemporal features of individual trajectories and improve the accuracy of location prediction. Xu et al. [18] converted the location prediction problem into a classification problem by extracting the spatiotemporal features in the historical trajectory data and proposed a learner based on a modified Support Vector Machine. At the same time, temporal and spatial gates that independently process individual movement information are introduced into Long Short-Term Memory (LSTM) to effectively extract individual trajectory features [19]. Moreover, Zhang et al. [26] proposed a multi-task location prediction framework based on LSTM and Convolutional Neural Networks (CNN). In this model, LSTM is responsible for extracting the location sequence and time attributes, and CNN extracts the spatial correlation of each location. In addition, LSTM can not only predict the short-term location of users but also the long-term movement trajectory by mining the periodicity of the users' movement [17]. Location prediction can also be considered as a classification problem based on the users' current feature [21,29]. For example, Lei et al. [5] proposed a spatiotemporal trajectory framework, which extracted the spatial information by the clustering algorithm and then explored individual travel behavior in the form of a probability suffix tree. Li et al. [10] classified users according to certain classification standards and proposed corresponding prediction schemes for each category of users.

The methods mentioned above only consider the basic temporal and spatial information, while ignore the deeper features such as users' preferences [22], context [2], social correlation [3], and location semantics [27]. By analyzing the location information recorded by smartphones on social networks, Zhou et al. [28] concluded that collective spontaneous mobility would affect users' mobility. And this conclusion is proved to be effective for location prediction. Zhang et al. [24] represented users' preferences with a tensor and used the preference tensor to predict the next location. Moreover, a multi-context-based deep neural network location prediction model was proposed by Liao et al. [11], which captures the deep-level preferences by modeling different contexts. In addition, the sparsity of historical trajectory data often negatively affects the accuracy of position prediction. Therefore, individual-group trajectory prediction models were proposed in [7,8], in which the methods of location extraction, clustering, matching prediction, and probability suffix tree are used to reduce the impact of data sparsity on the prediction results and improve the accuracy.

In this paper, we propose a Pick-Up Point Recommendation Model (PPRM) using users' historical ride-hailing data and can determine whether to provide relevant services according to users' requirements on the recommendations. Our contribution can be summarized in the following points:

1. We deeply dig into the personal travel pattern of users and outline some commonalities through the detailed analysis of the historical order data in many aspects. We further design an unique location recommendation model to provide users with optimal pick-up points which suit their preferences.
2. We introduce DBSCAN, which can be effectively classified without specifying the number of, to cluster user's historical orders, and the idea of order clustering before order matching can effectively reduce the running time and computation complexity of the location recommendation model.
3. The experimental results show that the models designed in this paper can predict users' pick-up points nicely and further optimize the related functions of existing apps to facilitate the operations of users.

2 Dataset and Dataset Analysis

2.1 Dataset

The dataset used in this paper is the historical ride-hailing orders of Didi Chuxing's users from January to June, 2019. The items of each order are as follows:

– **User_ID** and **Order_ID**: Each user or order has a distinctive id.
– **Departure_time**: The user's pick-up time in each order.
– **Starting_lat** and **Starting_lng**: The latitude and longitude of the user's pick-up/starting point, respectively.
– **Starting_name**: The point of interest (POI) of the user's pick-up points.
– **Dest_lat** and **Dest_lng**: The latitude and longitude of the user's destination, respectively.
– **Dest_name**: The POI of the user's destination.

2.2 Dataset Analysis

The distribution of users' orders is shown in Fig. 1. We can notice that the historical order volume of most users within half a year is between 280 to 520 (Noted in the orange box), and only a few have more than 750 orders (Noted in the red box). All order information has been anonymously summarized, and abnormal orders caused by the cancellation of passengers or any other reason have been excluded.

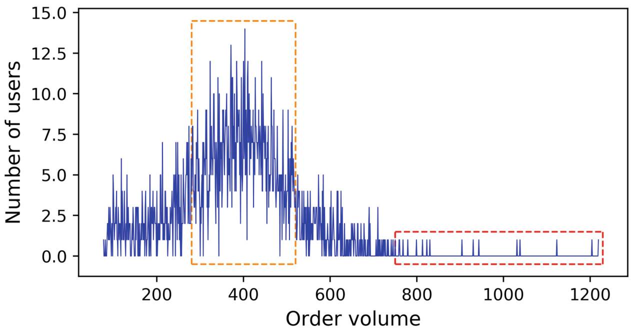
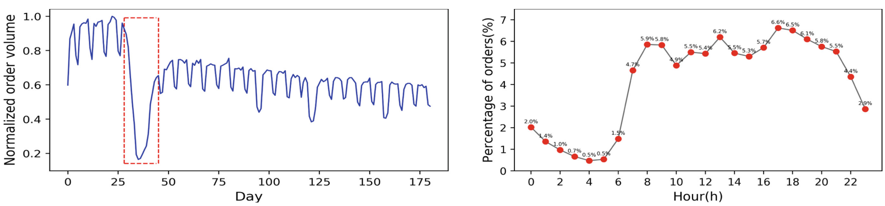


Fig. 1. The numbers of users under different order volumes. (Color figure online)

We separately counted the daily and hourly order volume. The daily order volume has a stage of rapid decay and rise, as shown in the red box of Fig. 2(a). This stage starts at the end of January and ends at the beginning of February. Although the daily order volume later is at a high level, the overall trend is declining. While, the hourly order volume has three peaks, which are corresponding to rush hour at morning/afternoon/evening, respectively, as shown in Fig. 2(b). Moreover, users concentrate on taking taxis during the day, and the daytime (7:00 am–7:00 pm) order volume account for about 68% of the total.



(a) Daily order volume. (b) Hourly order volume.

Fig. 2. The volume of orders over time. (Color figure online)

Besides, there are differences between orders on workdays and holidays. Figure 3 respectively depicts the proportion of order volume in each period of workdays and holidays. There are three peaks of order volume on workdays, which is similar to Fig. 2(b). However, there is no morning peaks on holiday. Traffic jams in the morning on workdays often occur due to daily commuting. On the contrary, people often choose to go out relatively late on holidays.

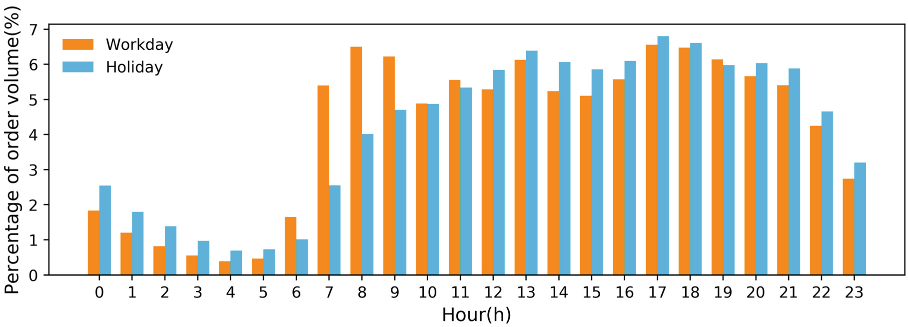
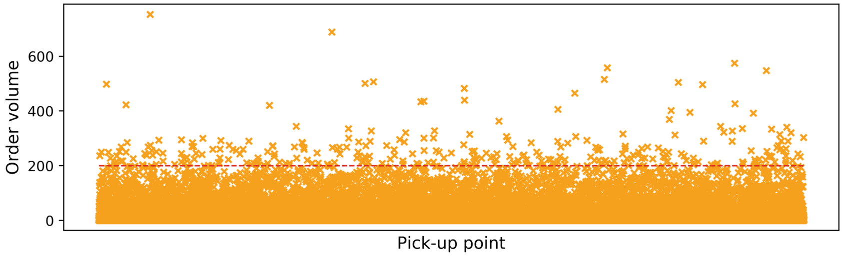
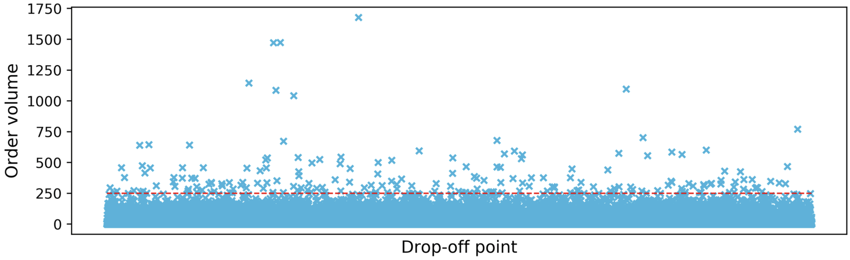


Fig. 3. The order volume on workdays and holidays.

In addition, the dataset of users' orders contains 112581 POIs of pick-up points and 88717 POIs of drop-off points. The order volume of each location is listed in Fig. 4. Since the number of drop-off points only accounts for 78.8% of the number of pick-up points, the orders in the dataset have a certain degree of aggregation in the spatial dimension, which can be seen in Fig. 4 marked in the red lines. It is worth noting that several drop-off points in Fig. 4(b) share extremely high order volume. We check these locations separately and find that they are all located at commercial spots or train stations. These areas have always been the places with high demand for urban commuting.



(a) The order volume for each pick-up point. The value of the red line is 200.



(b) The order volume for each drop-off point. The value of the red line is 200.

Fig. 4. The order volume for each pick-up and drop-off point. (Color figure online)

3 User Travel Behavior Analysis

3.1 The Analysis of Temporal Information in Users' Historical Orders

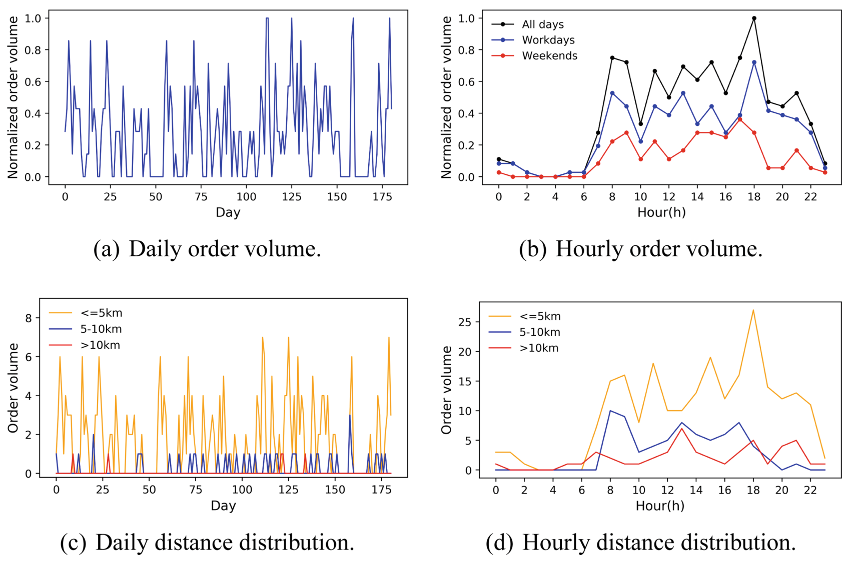
Figure 5(a–b) shows the temporal distribution of users' historical orders. It can be seen that the users took a taxi at least once in 68% of days since the number of workdays is the majority, the trend of orders on workdays is alike as to the overall trend. There are three peaks in the order curve of the workday, which is similar to that in Fig. 2(b). However, the curve of order volume for the weekends is quite different. The trend of orders on weekends is more gradual compared to workdays, and there is only one peak at 17:00.

3.2 The Analysis of Spatial Information in Users' Historical Orders

According to the straight-line distance between the pick-up and drop-off points in each order, we set three levels to divide the distance: short-distance (<5 km), mid-distance (5–10 km), and long-distance (>10 km). Figure 5(c–d) describes the distance distribution. Short-distance and mid-distance trips account for the vast majority of the travel records. In Fig. 5(d), the hourly short-distance distribution curve has a similar trend to the hourly order volume curve in Fig. 5(b). The number of mid-distance trips is smaller than that of short-distance trips but is larger than that of long-distance trips.

The pick-up and drop-off points of most users are highly clustered. We show the relative positions of pick-up points and drop-off points in all historical orders in Fig. 6(a), and we can see that most locations are clustered in a certain space range, as shown in Fig. 6(b).

Through the same analysis of other users, we discover two commonalities of users' travel behavior:



(a) Daily order volume.

(b) Hourly order volume.

(c) Daily distance distribution.

(d) Hourly distance distribution.

Fig. 5. The temporal and spatial distribution of historical orders.

1. The travel time patterns of users can be roughly divided into two categories: regular and irregular. Most users have obvious different travel patterns in different periods, such as workdays and holidays. These users have regular travel patterns of relatively stable travel time and relatively similar pick-up and drop-off points. But there are also users with no regular patterns. In addition, most users' pick-up and drop-off points are clustered, and also, a few users own discrete locations.
2. The average distance of each user's historical orders is distinct, but most users take short-distance and mid-distance trips as their primary travel modes.

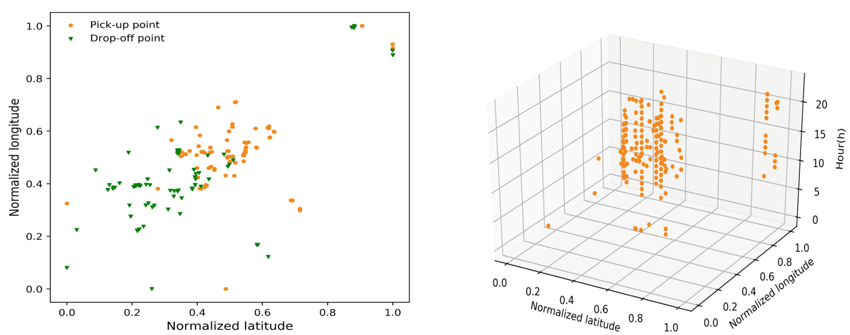
4 Pick-Up Point Recommendation Model

Pick-up Point Recommendation Model (PPRM) can be briefly stated as follows: First, the users' historical orders are clustered by DBSCAN [1] based on the spatial distribution of pick-up points, and the orders that match the users' current environment are searched in a certain category. The overall framework of PPRM is shown in Algorithm 1. Note that all feature vectors in our algorithm have been normalized.

4.1 Order Clustering via DBSCAN

DBSCAN is a density-based spatial clustering method, which treats an area with sufficient density as a cluster (category) and can find arbitrary-shaped clusters in spatial data with noise. Here, a cluster is defined as the largest collection of closely connected points. Compared with other clustering algorithms, like the k-means clustering algorithm [13], DBSCAN has the advantages as follows:

- DBSCAN does not need to specify the number of clusters manually.
- DBSCAN can find clusters of any shape.
- DBSCAN can identify the noise points.



(a) The spatial distribution of pick-up points and drop-off points.
(b) The temporal and spatial joint distribution of pick-up points.

Fig. 6. The spatial distribution of historical orders.

Algorithm 1: PPRM

Input: feature vector v, historical order data X
Output: recommended pick-up point x_{rec}
// Order clustering
The orders are clustered by Algorithm 2;
// Order matching
The set of best matching orders V_{best} is obtained by Algorithm 3;
// Pick-up point recommendation
if *Match failed* **then**
  The pick-up point is not recommended;
else
  $x_{rec} = \frac{1}{n_{best}} \sum_{i=1}^{|n_{best}|} x_{i,best}$,where n_{best} denotes the number of samples in V_{best} and $x_{i,best}$ is the sample in V_{best};
  x_{rec} = Inv-normalized(x_{rec}), where Inv-normalized($\cdot$) denotes the inverse operation of normalization;
end

Figure 6(a) has shown the spatial distribution of pick-up points, and outliers can be regarded as "noise points." First, we randomly select a point from the set of pick-up points and search all the points within the specified radius centered on the chosen point. Then if the number of searched points exceeds the threshold we set, all the searched points are grouped into one cluster, and the point selected is called the core point. Otherwise, use the next point to continue the above operation. Algorithm 2 shows the complete concept of order clustering based on the pick-up points set.

4.2 Order Matching

According to the clustering results in Sect. 4.1, the current best matching order can be found from a certain category. Compared to traversing the entire historical order data, our method has lower time complexity.

First, we calculate the category center of each category C_k, $k = 1, 2, \ldots, m$, which can be defined as

$$c_k = \frac{1}{n_k} \sum_{i=1}^{n_k} x_{ik}, \tag{1}$$

where x_{ik} denotes the i-th sample in C_k and n_k is the number of samples in C_k. Then, the feature vector v (normalized latitude and longitude) is extracted according to the user's current environment. Finally, the model tries to search for the best historical

Algorithm 2: Order clustering via DBSCAN

Input: search radius r, minimum number of samples within the search radius t, sample point collection $X = \{x_1, x_2, ..., x_n\}$, n is the number of sample points
Output: all clusters

```
while unclassified samples are existed do
    // Select the initial point
    Randomly select an unclassified point as the initial point x and define a new cluster
    C_k = {x};
    //Sample search
    while untraversed points are existed in C_k do
        Select an untraversed point x_i;
        if x_i is core point then
            All points within the search radius r of x_i are classified to cluster C_k;
        else
            continue;
        end
    end
end
Output all clusters C_k, k = 1, 2, ..., m;
```

order matching from the closest category. In this paper, the distance d_k between the feature vector v and the category center C_k is defined as Euclidean distance between v and c_k. Similarly, the distance between two feature vectors is also defined as Euclidean distance. Algorithm 3 shows the process of order matching.

$$d_k = \|v - c_k\|. \tag{2}$$

5 Experimental Results and Analysis

In this section, we conduct multiple experiments on our dataset, show the related experimental results, and give the corresponding analysis.

5.1 The Analysis of Order Clustering

The clustering algorithm is carried out only on spatial locations, so we made a data preprocessing as follows: First, we extract the latitude and longitude of the pick-up points from each historical order. And then, we set the number of the training set and test set account for 80% and 20%, respectively. In addition, all samples have been normalized.

Since the data in the training set is unlabeled, we introduce Silhouette Coefficient (SC) to measure the effect of data clustering, and the Silhouette Coefficient of sample x_i is defined as

$$SC(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}}, \tag{3}$$

Algorithm 3: Order matching

Input: feature vector v, category $C_k, k = 1, 2, \ldots, m$
Output: the collection of best matching orders V_{best}
Initialize:$V_{best} = \varnothing$
//Compute the category center
for $k = 1, 2, \ldots, m$ **do**
  compute c_k by Eq. 1;
end
// Match the category of v
for $k = 1, 2, \ldots, m$ **do**
  compute d_k by Eq. 2;
end
The category with the short distance is regarded as the category of v, denoted as $C_v = \{x_{1v}, x_{2v}, \ldots, v_{nv}\}$;
// Order matching
for $i = 1, 2, \ldots, n$ **do**
  compute the sample distance d_{iv} between v and x_{iv} by $d_{iv} = \|x_{iv} - v\|$;
end
Select at most 3 samples with the geographic distance less than 100m from C_v as the best matching samples and store them in V_{best};
if $|V_{best}| > 0$ **then**
  Output V_{best};
else
  Match failed;
end

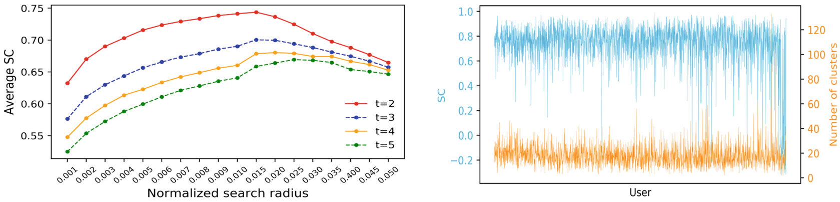
where $a(i)$ denotes the average distance between x_i and other samples in the cluster of x_i, and $b(i)$ denotes the average distance between x_i and all samples in the cluster closest to x_i. $a(i)$ and $b(i)$ can also be called the degree of dissimilarity, and the value range of $SC(i)$ is in $[-1, 1]$. Moreover, the larger the value of the SC is, then the better the clustering effect will be.

Figure 7(a) describes the performance of DBSCAN under various parameter combinations (r and t in Algorithm 2). When $t = 2$ and $r = 0.015$ (the actual geographic distance is about 1 km), the average SC of all users is the smallest. The clustering effect of each user's orders under the best parameter is shown in Fig. 7(b), and the results of most users are satisfactory.

5.2 The Analysis of the Results Obtained by PPRM

In order to ensure the validity of the experiment, we first randomly drift 0–50 m for each sample in the test dataset. Meanwhile, two metrics are introduced to measure the performance of PPRM, i.e., prediction rate (PDR) and distance error (DisErr). PDR is defined as

$$PDR = \frac{n_{rec}}{n_{text}} \times 100\%, \tag{4}$$



(a) Average SC under each parameter, where t is the minimum number of samples within the search radius.

(b) SC and the number of clusters of each user under the best parameters.

Fig. 7. Clustering results via DBSCAN.

where n_{rec} denotes the number of test samples with the pick-up points, and n_{test} is the number of test samples. PPRM outputs the predicted locations in the form of latitude and longitude, so DisErr is defined as the geographic distance by calculating the latitude and longitude of the predicted locations and the actual locations.

According to the parameter combinations of r and t, the results of PPRM are shown in Fig. 8. It shows that the average PDR can reach 82.38%. In other words, PPRM is able to handle 82.38% of the demand scenarios. Moreover, the average DisErr is only 23.14 m, so the recommended and the actual point-up point can be considered as the same POI.

In addition, for the situation where the remaining PPRM doesn't work, our solution is to search for the POIs, which are closest to the user from the database as the recommended pick-up points. The recommended locations in this way may not be in line with the user's preferences, but the unnecessary troubles caused by the user entering the wrong location information can be avoided in many cases.

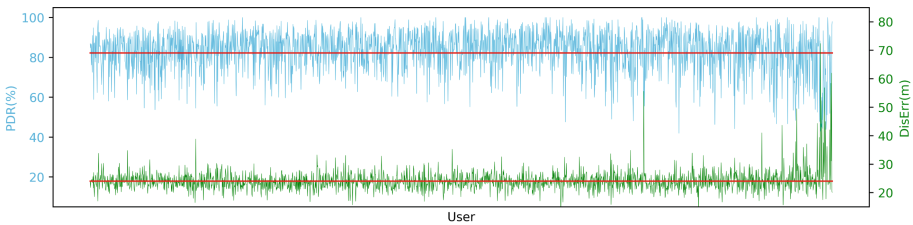


Fig. 8. The results obtained by PPRM, where the two red lines denote the mean values of PDR (82.38%) and DisErr (23.14 m). (Color figure online)

6 Conclusion

In this paper, we deeply mine the users' travel patterns from both temporal and spatial information of users' historical ride-hailing orders and summarize the general regularity characteristics of users' travel behavior. According to the spatial distribution of users' historical pick-up points, we propose a pick-up point recommendation model (PPRM) based on DBSCAN. PPRM consists of two components: order clustering and location recommendation. First, the historical orders of each user are clustered according to the density of pick-up points. Then, the feature vector is extracted from the user's current environment. Finally, the most similar orders are searched and used as the matched orders, and the latitude and longitude of the recommended pick-up point are output by fusing the matched orders. The final experiment results based on real-world datasets show that PPRM can efficiently and accurately provide users with ideal pick-up points.

Acknowledgments. We are grateful to anonymous reviewers for their helpful comments. This work is partially supported by the National Key Research and Development Program of China under Grant No. 2019YFB1600300.

References

1. Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD, vol. 96, pp. 226–231 (1996)
2. Fan, X., Guo, L., Han, N., Wang, Y., Shi, J., Yuan, Y.: A deep learning approach for next location prediction. In: 2018 IEEE 22nd International Conference on Computer Supported Cooperative Work in Design (CSCWD), pp. 69–74. IEEE (2018)
3. Gong, Y., Li, Y., Jin, D., Su, L., Zeng, L.: A location prediction scheme based on social correlation. In: 2011 IEEE 73rd Vehicular Technology Conference (VTC Spring), pp. 1–5. IEEE (2011)
4. Kwon, E., et al.: A novel location prediction scheme based on trajectory data. In: 2019 International Conference on Information and Communication Technology Convergence (ICTC), pp. 1292–1294. IEEE (2019)
5. Lei, P.R., Li, S.C., Peng, W.C.: QS-STT: QuadSection clustering and spatial-temporal trajectory model for location prediction. Distrib. Parallel Databases **31**(2), 231–258 (2013). https://doi.org/10.1007/s10619-012-7115-1
6. Lei, P.R., Shen, T.J., Peng, W.C., Su, J.: Exploring spatial-temporal trajectory model for location prediction. In: 2011 IEEE 12th International Conference on Mobile Data Management, vol. 1, pp. 58–67. IEEE (2011)
7. Li, F., Li, Q., Li, Z., Huang, Z., Chang, X., Xia, J.: A personal location prediction method based on individual trajectory and group trajectory. IEEE Access **7**, 92850–92860 (2019)
8. Li, F., Li, Q., Li, Z., Huang, Z., Chang, X., Xia, J.: A personal location prediction method to solve the problem of sparse trajectory data. In: 2019 20th IEEE International Conference on Mobile Data Management (MDM), pp. 329–336. IEEE (2019)
9. Li, S., Qiao, J., Lin, S.: Location prediction method based on similarity of users moving behavior. Comput. Sci. **45**(12), 288–292+307 (2018)
10. Li, Y., Lei, L., Yan, M.: Mobile user location prediction based on user classification and Markov model. In: 2019 International Joint Conference on Information, Media and Engineering (IJCIME), pp. 440–444. IEEE (2019)
11. Liao, J., Liu, T., Liu, M., Wang, J., Wang, Y., Sun, H.: Multi-context integrated deep neural network model for next location prediction. IEEE Access **6**, 21980–21990 (2018)
12. Lin, S.K., Li, S.Z., Qiao, J.Z., Yang, D.: Markov location prediction based on user mobile behavior similarity clustering. J. Northeast. Univ. (Nat. Sci.) **37**(3), 323 (2016)
13. MacQueen, J.: Classification and analysis of multivariate observations. In: 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297 (1967)

14. Mantoro, T., Olowolayemo, A., Olatunji, S.O., Osman, A., et al.: Extreme learning machine for user location prediction in mobile environment. Int. J. Perv. Comput. Commun. **7**(2), 162–180 (2011)
15. Qiao, J., Li, S., Lin, S.: Location prediction based on user mobile behavior similarity. In: 2017 IEEE 23rd International Conference on Parallel and Distributed Systems (ICPADS), pp. 783–786. IEEE (2017)
16. Tseng, V.S., Lu, E.H.C., Huang, C.H.: Mining temporal mobile sequential patterns in location-based service environments. In: 2007 International Conference on Parallel and Distributed Systems, pp. 1–8. IEEE (2007)
17. Wong, M.H., Tseng, V.S., Tseng, J.C.C., Liu, S.-W., Tsai, C.-H.: Long-term user location prediction using deep learning and periodic pattern mining. In: Cong, G., Peng, W.-C., Zhang, W.E., Li, C., Sun, A. (eds.) ADMA 2017. LNCS (LNAI), vol. 10604, pp. 582–594. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-69179-4_41
18. Xu, C., Xu, C.: Predicting personal transitional location based on modified-SVM. In: 2017 International Conference on Computational Science and Computational Intelligence (CSCI), pp. 340–344. IEEE (2017)
19. Xu, F., Yang, J., Liu, H.: Location prediction model based on ST-LSTM network. Comput. Eng. **9**, 136–138 (2019)
20. Yamada, N., Katsumaru, N., Nishijima, H., Kimoto, M.: Location prediction based on smartphone multimodal personal data for proactive support services. In: 2018 Eleventh International Conference on Mobile Computing and Ubiquitous Network (ICMU), pp. 1–2. IEEE (2018)
21. Yasser, K., Hemayed, E.: Novelty detection for location prediction problems using boosting trees. In: Gervasi, O., et al. (eds.) ICCSA 2017. LNCS, vol. 10405, pp. 173–182. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-62395-5_13
22. Ying, J.J.C., Lee, W.C., Tseng, V.S.: Mining geographic-temporal-semantic patterns in trajectories for location prediction. ACM Trans. Intell. Syst. Technol. (TIST) **5**(1), 1–33 (2014)
23. Yoon, T.B., Park, K.H., Lee, J.H.: A spatiotemporal location prediction method of moving objects based on path data. J. Korean Inst. Intell. Syst. **16**(5), 568–574 (2006)
24. Zhang, D., Yang, N., Ma, Y.: Explicable location prediction based on preference tensor model. In: Cui, B., Zhang, N., Xu, J., Lian, X., Liu, D. (eds.) WAIM 2016. LNCS, vol. 9658, pp. 205–216. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39937-9_16
25. Zhang, H., Jiang, J., Zhou, H.: Method of mining user mobile rule based on pattern matching degree and location prediction. Comput. Sci. **36**(11), 3258–3261+3296 (2019)
26. Zhang, R., Guo, J., Jiang, H., Xie, P., Wang, C.: Multi-task learning for location prediction with deep multi-model ensembles. In: 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), pp. 1093–1100. IEEE (2019)
27. Zhang, W., Sun, L., Wang, X., Huang, Z., Li, B.: SEABIG: a deep learning-based method for location prediction in pedestrian semantic trajectories. IEEE Access **7**, 109054–109062 (2019)
28. Zhou, C., Huang, B., Tu, L.: Exploiting collective spontaneous mobility to improve location prediction of mobile phone users. In: 2015 IEEE International Conference on Data Science and Data Intensive Systems, pp. 117–122. IEEE (2015)
29. Zolotukhin, M., Ivannikova, E., Hämäläinen, T.: Novel method for the prediction of mobile location based on temporal-spatial behavioral patterns. In: 2013 IEEE Third International Conference on Information Science and Technology (ICIST), pp. 761–766. IEEE (2013)