Trajectory is the Description of Arrival

Yu Zheng
Shanghai Jiao Tong University

JiaLin Li
Shanghai Jiao Tong University

Hao Liu
Shanghai Jiao Tong University

***Abstract*—In this article, we explore how to model geographical information and taxi drivers's behavior for a challenging task: the next destination prediction in a taxi journey. Next destination prediction is a real-world problem which has rich application scenarios in reality. Effect model can play a great role in reducing traffic accidents and traffic jams. We brought a model that treats this task as a regression problem, instead of a multi-class classification. Finally, we have demonstrated how using multiple embedding features such as trajectory, taxi_ID and periodic time in detail, strongly improves the final result. The proposed approach was tested on the ECML/PKDD Discovery Challenge 2015 dataset based on the city of Porto.We only use less than 10% data while the final result can obtain silver medal in this challenge. Our source code is available on https://github.com/COMoER/Trajectory-is-the-Description-of-Arrival.**

I. Introduction

Nowadays,the widespread application of networks and mobile devices, as well as the remarkable progress of remote sensing satellites and geographic information systems, have enabled people to obtain unprecedented amounts of geographic scientific data. These data are often associated with time series and suggest many patterns that are Highly hidden but potentially useful. Therefore, spatial data mining technology is of great significance for the automatic extraction and analysis of valuable patterns in spatial data, and trajectory analysis is one of the most important fields of spatial data mining.

Due to the large number of passengers and the uniformity of management, it is easier to collect sufficient data that can represent the inherent patterns of taxis. Trajectory analysis for taxis is of great significance to the automatic scheduling and intelligent services of taxis.Effective trajectory prediction can reduce the waiting time of passengers and improve the quality of life of urban residents.This article will focus on the task of taxi trajectory analysis, trying to predict the destination of taxis from their trajectories.

In the competition ECML/PKDD 15: Taxi Trajectory Prediction (I) of kaggle, we are required to build a predictive framework that is able to infer the final destination of taxi rides in Porto, Portugal based on their (initial) partial trajectories.To finish this task, We first looked at the methods used by the winner of the competition[1].They use MLP method with embedding of the meta data to solve this task.

Our method is also using MLP as the skeleton of deep learning model but the process of trajectory is different from the winner method. Inspired by [2], we use the id of Geohash instead of the raw position of the trajectory as the main input, which can be embedded as a dense feature vector and make the method widely used in the aspect of NLP available to the trajectory data, that is, we have treated the trajectory as a sentence or description, in which the points in the trajectory is just the token and the dictionary or tokenizer is Geohash. The information in the description we want to digest is the arrival of the trajectory, which is partial of the whole trip. The arrival could also be presented as a Geohash token but due to the data sparsity problem, it's difficult to predict the token id of the arrival with a great number of candidates, so our method directly outputs the position of arrival, which is a regression method. We use a novel method, that is, not directly predict the pair of latitude and longitude but use an approximate method which is to transform the coordination of latitude and longitude to the 2D Euclidean coordination in which the unit of the two dimension is directly kilometer which is the unit of the measure. Such a simple but novel method greatly improve the performance of the regression model.

In this paper, we will firstly briefly introduce the related work of the taxi arrival prediction problem and the Geohash method in in II. In III, the detail of the dataset will be described. The main idea and more detail of our method will be given in IV. And in V, the result of our experiment with the discussion towards them will be available. In VI, we describe the relationship between our work and data mining course.

II. background

A. Related Work

1) winner method: To reduce the complexity of trajectory, they first used a mean-shift clustering algorithm on the destination points to obtain around 3000 clusters.The key idea of the method is MLP taking as input the first 5 and last 5 GPS points of a prefix and all the raw meta data available (no extra feature engineering). To add more information, they tried a specific embedding for each meta data. After a few RELU layers, they put a softmax over the 3000 clusters and average the corresponding clusters weighted by the softmax probabilities. The final task is to minimize the the mean-square error with SGD and momentum and adadelta.

2) RNN classification method: We also found an article using the same dataset that treats the task as a classification problem[3]. In this paper,author selected the RNN approach that models the taxi drivers' behavior and encoded the semantics of visited locations by using geographical information from Location-Based Social Networks.First the author used K-means method to get a set of location clusters.Just like the winner method, he also used embedding method to enter meta data to the model as the input. At the same time, the author introduces some prior knowledge for the model: points of interest. For some urban areas with a high concentration of people, they will get additional weights when used as the prediction result of the trajectory.

B. Geohash

Geohash is an efficient method to cluster geometric points in latitude-longitude coordination. Its main idea is simple, just to use a binary tree by half dividing the range of latitude and longitude to present different levels of scale of the locations. We can set the max depth of the binary tree to control the granularity of scale of Geohash .A sequence of binary number is used to indicate whether the location is in the corresponding branch.The binary sequence is exactly a binary integer,or to say, a hash code. And the closer to the head of the sequence, the dividing range is larger.So the same prefix indicates that the points are closer and the pair of point who are close in the geometric coordination can also be with fewer distance of the Geohash code in most cases.

The method Geohash used to make the 2D data into 1D hash code is using a specific order. The order is that the binary number in the odd location indicates the longitude branch(range) and that of the even location indicates the latitude branch(range), aim to make the neighborhood,the points whose Geohash binary code is different only in last few bit(or to say, in the suffix), has an order like Z character on the map, but there is a mutation at the corner points just like in 1.

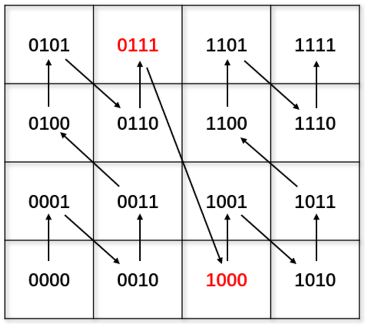

Fig. 1. geohash example

In this work, we use Geohash as our clustering method of the locations in the trajectory due to that it does not need training and can compute the hash code with O(1) complexity while it is a great representation of the locality of the locations.

III. Dataset Description

The dataset for this task describes the trajectories for 442 taxis from 01/07/2013 to 30/06/2014 running in the city of Porto, in Portugal. For each trip trajectory of the taxi, the dataset gives an array of points represented by latitude and longitude coordinates.In order to understand the distribution of trajectories more clearly, we selected the starting point and end point of the trajectory, and drew them on the map (red for start and blue for end). In Figure 2, we can initially observed the distribution of trajectory points in the city.

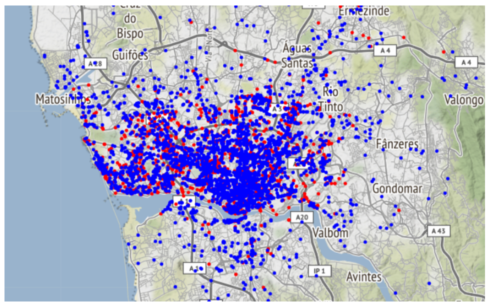

Fig. 2. Distribution of start and end points in all trajectories

We can see that the distribution of the start and end points of taxis is relatively concentrated, but there are some points far from the city center, These points are to avoid because the high offset may cause the model to underfit this task. In addition to the point set of the trajectory, the dataset also contains some metadata. Here are some important kinds of metadata:

- TAXI_ID: It contains an unique identifier for the taxi driver that performed each trip. The id of the taxi may be related to the driver's driving habits, so it is very important information for trajectory prediction.
- TIMESTAMP: It identifies the trip's start.Time may be cyclical on a weekly or monthly scale, with different end points tending to differ from time to time.
- ORIGIN_STAND: It identifies the starting point of the trip. The location type of the starting point is obviously important information for trajectory analysis.

At next chapter we will introduce our methods to solve this task.

IV. Method

A. How To Tokenlize

First, we use the Geohash coding method to cluster the trajectory points in each trajectory. For all trajectory points, there are too many categories formed due to the presence of a small number of outliers that deviate greatly from the center. In order to compress the number of categories, we need to remove these data points that deviate too much. Specifically, we first sample n points in all trajectory points and set lower thresholds α and upper thresholds β. Arrange the latitude and longitude of the n points in descending order to obtain two sequential arrays, and for each point p_i, it can be kept if and only if its longitude and latitude in the corresponding array index i,j satisfy:

$$n\alpha < i, j < n\beta \quad (1)$$

Through this step, we can control the scale of points to an appropriate range. After that, we take the remaining points after Geohash encoding, and take the smallest code as the zero value due to the fact that the points with the same prefix are in the same scale of geometry so we can just present them

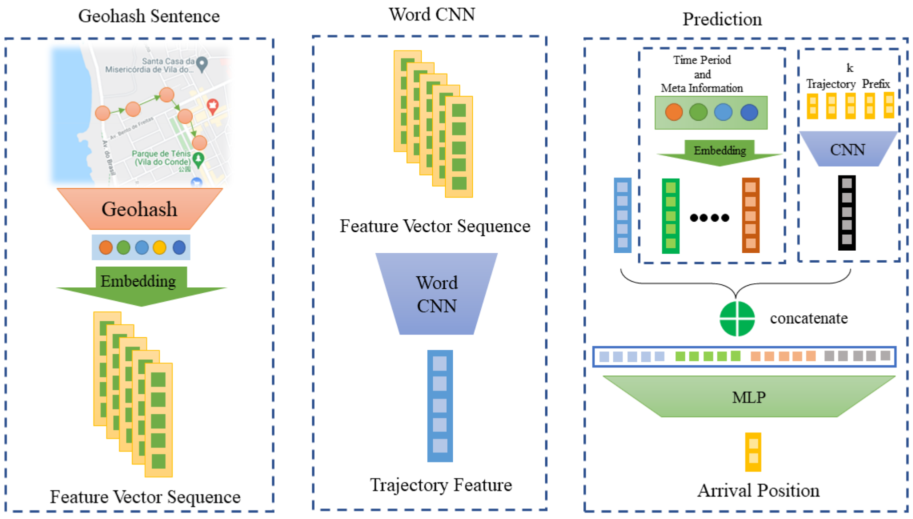

Fig. 3. Architecture of the our model,which is conposed by three conponents. The first named Geohash Sentence, which is to transform the trajectory into a sequence of feature vector using Geohash code and embedding. The second uses the WordCNN network to aggregate the feature vector sequence to a trajectory feature vector.The third is our prediction network, which uses MLP as the output layer and the input of it is the concatenation of the embedded feature vectors from trajectory, time period, meta information and k trajectory prefix location.The output is exactly the arrival position in NED coordination.

using the suffix, that is, for each code k_i for one point, we transform it into:

$$k_i^{'} = k_i - k_{min} \tag{2}$$

The above step is equivalent to take the point in the lower left corner of the map as the origin, and remove the prefix codes that are same for all codes, so as to control all points in one area. After that, we will also include the remaining points into these existing codes, for points that are not in the selected area above, we set their code value to zero.With all steps above, the trajectory points have been transformed to tokens.

Considering the mutation problem in geohash mentioned earlier, we propose a solution called prefix. As shown in Figure 1, geohash code 0111 and 1000 are adjacent in geohash code space while stay away in real world space. In our model, we select n points at the begin of a trajectory. Then we convert these points to two-dimensional rectangular coordinate and regard them as a part of the input feature. Because of the continuity in two-dimensional rectangular coordinate, this method alleviates the mutation problem to a certain degree. In addition, we remove points which are beyond the range we set in order to reduce the geohash code length. Prefix which is in two-dimensional rectangular coordinate can be used at all points. So, by using prefix method, the model will get a better result on the outer points.

B. *Why Regression*

In related work chapter, we mentioned two solutions, and they both treat this problem as a classification task. But with our clustering method there are still more than 50,000 classes remaining in the test. If a classification algorithm is used, due to the large number of categories, it may get a prediction result with low accuracy. But if directly predicting latitude and longitude and using MSE as a loss function, despite a low MSE loss value, which means little latitude and longitude error, it will still cause a great error if using the kilometer unit. So we are trying to convert the latitude and longitude information into continuous position information in kilometer unit thus getting a great performance with a regression model.

We chose to convert geographic coordinates of latitude and longitude, which is in the coordination of WGS-84 coordinate system, to NED coordinate system. WGS-84 coordinate system abstract the earth as a standard ellipsoid:

$$\frac{x^2 + y^2}{a^2} + \frac{z^2}{b^2} = 1 \tag{3}$$

From this we can get the eccentricity of the earth $e = \frac{(a^2-b^2)^{\frac{1}{2}}}{a}$, For any point on the earth's surface, we use the three dimensions of longitude, latitude and height to represent its geographic location as (μ, λ, h)(WGS-84 coordinate system), then we can get the meridian radius of curvature and the prime vertical radius of curvature:

$$R_N = \frac{a(1-e^2)}{(1-e^2\sin^2\mu)^{3/2}} \tag{4}$$

$$R_E = \frac{a}{(1-e^2\sin^2\mu)^{1/2}} \tag{5}$$

In this task, since the height is basically the same, we set the height to zero. Then we select the minimum point (μ_0, λ_0) obtained by the previous clustering as the origin of the NED coordinate system. With these parameters, we can approximately transform the point as (x, y) in NED coordinate system:

$$x = \cos\mu_0 R_E(\lambda - \lambda_0) \tag{6}$$

$$y = R_N(\mu - \mu_0) \tag{7}$$

We have estimated the mean error between Euclidean distance in the NED coordination and Haversine distance in WGS-84 coordination of the sample data, and the error is 0.01068883 km($\leq 0.5\%$ of our model's error), but this method can improve error of regression from more than 3.4 km in [1] to less than 2.8 km as shown in V, which indicates the approximate method is reasonable.Therefore, we map the latitude and longitude information of the trajectory points to a two-dimensional continuous coordinate plane, so that we can start to explore the regression task.

C. Period Analysis

Since this task does not require predicting the time information of the arrival, we just want to explore whether it has an impact on the prediction of the trajectory end point by analyzing the periodicity of the time information. To this end, we analyze time information from the three scales of year, week and day to figure out the impact on the distribution of trajectories[4]. For a specific time scale, it's easy to obtain a series of time with k categories like $s_1 = [t_1^1, t_2^1, \dots, t_{n_1}^1], s_2 = [t_1^2, t_2^2, \dots, t_{n_2}^2], \dots, s_k = [t_1^k, t_2^k, \dots, t_{n_k}^k]$ and the mean value $t_{mean}^j = \frac{1}{n}\sum_{i=1}^n t_i^j$. Then here comes the standard deviation of the series which represents differences in distribution on this category:

$$\sigma^j = \sqrt{\frac{1}{n_j}\sum_{i=1}^{n_j}(t_i^j - t_{mean}^j)^2} \tag{8}$$

Then we calculate the standard deviation of all categories to get the distribution difference across the time scale. The mean value of all standard deviation is $\sigma_{mean} = \frac{1}{k}\sum_{i=1}^k \sigma_i$.

$$\sigma = \sqrt{\frac{1}{k}\sum_{i=1}^{k}(\sigma_i - \sigma_{mean})^2} \tag{9}$$

By comparing the standard deviations of different scales to find which time scale has the most influence on the data distribution. Through the above steps, we finally choose to divide the year by day, the week by quarter week and the day by quarter hour, since that the three scales share the highest standard deviations among all divisions. The three scale information will be added to the embedding method below.

D. Meta Information Embedding

For some additional meta information may also play a role in improving the accuracy of the model, we use the embedding method to combine with the trajectory codes as input of the network.As previously introduced, we picked some metadata that might be of importance and put them into the input vector by the method:

- TAXI_ID: It is described by a unique ID, so that it is easily transformed as an embedding.
- ORIGIN_CALL: It identifies whether the call is from a taxi station or parking lot.Since it has only two values, we can simply represent it by 0 or 1.
- ORIGIN_STAND: It tells the specific place of the taxi station if the ORIGIN_CALL is valid. The stands are also encoded with a unique ID.
- TIMES: It includes day information mainly. Due to possible period pattern, we also enter the date information as input.

E. Network Architecture

The network architecture is shown in 3. First, each point of the trajectory coding by Geohash will be embedded into a dense and fewer dimension feature vector. In the test dataset, the point out of the geohash coding range will be treated as a padding index and its feature vector is all zero.Then the problem is how to overcome variable length of trajectory and fully use the locality of the neighbors in the trajectory. Inspired by graph embedding work Deepwalk[5],we think trajectory is just as the path in deepwalk and they treat the path as a sentence and use word2vec to train the embed ding so trajectory can also treated as a sentence. Instead of using word2vec, We use wordCNN first introduced by [6]. It just performs a 1D convolution over the sequence and then use a over-time-maxpooling to it and thus getting a fixed dimension feature vector. We think wordcnn which sliding window over the sequence just like the word2vec method and the window-wise MLP and maxpooling over time make the end2end training available to make the aggregated feature vector more adapting to our task.

Then we also perform the embedding method to each of the meta information and time period and the sequence CNN architecture to prefix points to get a feature vector of the prefix points. Each of these feature vectors will be normalized to a unit vector to transform them into a unified scale and then concatenate them to acquire the final feature vector as the input of the MLP regressive predictor. The output of the predictor is the arrival position in NED coordination. In the evaluating case, just use an inverse transformation to transform it into WGS-84 coordinate system as the final result.

V. Experiments

The following experiments aim at verifying the effectiveness of our model for predicting the next destination in a taxi ride.We experimented our approach on a taxi dataset of Porto(a Portugal city) by adding our methods step by step.We also compare and analyze the differences of results caused by different methods

A. Experimental Setup

In this experiment,150,000 taxi rides were randomly sampled as our dataset and were randomly splitting into 85%-5%-10% for the train set,validation set and test set.And as it is a competition on kaggle, test set of competition is provided.We also test our model on it and finally obtain the public score and private score as a part of the experimental results.

The evaluation metric for this experiment is the Mean Haversine Distance(mHD) which is commonly used in navigation.It measures distances between two points on a sphere based on their latitude and longitude.The Harvesine Distance between the two locations can be computed as follows

$$a = sin^2(\frac{\phi_2 - \phi_1}{2}) + cos(\phi_1)cos(\phi_2)sin^2(\frac{\lambda_2 - \lambda_1}{2}) \quad (10)$$

$$d = 2 \cdot r \cdot atan(\sqrt{\frac{a}{1-a}}) \quad (11)$$

where ϕ is the latitude, λ is the longitude, r is the Earth's radius, and$d$ is the distance between two points.

B. Training and Hyperparameter Setting

In order to find the impact of each method on the results, we performed the experiment by adding methods step by step , listed in the following:

- **base**: the model was trained only use trajectory information as features.And for each taxi ride, the whole trajectory was input to neural network. This is the base for other models.The following models were built by adding other methods on it.
- **random**: Different from base model, random model only use partial trajectory information for each taxi ride as the input of neural network. We use two methods to get the trajectory fragment, one is sampling from the front part of trajectory, named head, the other is sampling from the middle part of trajectory , named partial.
- **prefix**: On the basis of random, the points at the begin of trajectory translated under x-y coordinate were used as an additional feature. We combine it with partial trajectory as the input of our net.
- **meta**: On the basis of random, more meta data such as time, taxi driver ID were used besides trajectory information. We combine meta features with partial trajectory as the input of our net.
- **all**: All previous methods were used in this model. The model integrates information from trajectory, prefix and meta data.

In order to get better result, we tuned the parameters of our models and finally determined our model's parameters. The embedding size of trajectory, prefix and meta data is 32, 16, 10 respectively.For trajectory geohash encoder, we chose 7 as geohash binary tree depth. For prefix, we finally chose 5 point as the a representative of a trajectory. And for meta data, we chose TAXI_ID, ORIGIN_CALL, ORIGIN_STAND, day, quarter hour, week as meta data. The input dimension of the regression net is equal to the sum of features which were used. The hidden layer size was set to 256 and the output layer size is 2 which is equal to the dimension of one point in x-y coordinate.The dropout rate is set to p = 0.5.

To train the net, we use Adam optimizer. The network is trained using the Mean Squared Error (MSE) as the loss function. We compute the mHD score on the validation set after each epoch and save the network parameters which named best model if a new best mHD. And the network parameters will be saved as the name last model after all training finishing.

TABLE I
RESULT

method		model	score		
			test	kaggle public	kaggle private
base		best	2.651	2.772	2.666
		last	2.653	2.796	2.681
random	head	best	2.193	2.766	2.520
		last	2.194	2.735	2.517
	partial	best	2.467	2.696	2.369
		last	2.470	2.704	2.382
prefix	head	best	2.181	2.851	2.522
		last	**2.178**	2.799	2.529
	partial	best	2.448	2.720	**2.321**
		last	2.453	2.710	2.348
meta	head	best	2.204	2.812	2.509
		last	2.198	2.918	2.454
	partial	best	2.460	2.693	2.348
		last	2.471	2.728	2.352
all	head	best	2.181	2.780	2.416
		last	2.190	2.745	2.489
	partial	best	2.447	**2.647**	2.385
		last	2.470	2.686	2.382

C. Result

All the experiment results can be found in TABLE I. All the models reported in Table I have been trained on the same set of trajectories. We note that:

- By using our methods, the score(take the best results) of mHD on test, kaggle public, kaggle private declined 17.8%, 4.5%, 12.9% respectively than base.
- We can find it that the random model using front part could effectively reduce the mHD on test while the random model using middle part could effectively reduce the mHD on kaggle private. Perhaps it is due to the uneven distribution of test set.
- Prefix and meta data make decrease on test and kaggle private respectively.The model using those two methods finally brought the best results on kaggle public which is 0.049 decline compared to its basics random model.

VI. Association With Data Mining Course

Our work is a typical data mining work. Cluster is a significant conception in data mining and we use Geohash, a cluster method, to encode the trajectory positions. And we use embedding, a popular data mining preprocessing method, to transform each token into a dense feature vector. Specially, to process sequence of trajectory feature vectors, we get motivation from the deepwalk work to treat it as a sequence and thus use NLP technique to get a feature vector.

VII. Conclusion

In this article, we have introduced a new way of looking at the problem of the next destination prediction in a taxi ride. Instead of using the complete taxi trajectory, we brought a new method which use sampled fragments of a taxi ride and get a better prediction result. We creatively use Geohash as a trajectory encode method, which bring benefits of good representativeness and efficient computing speed. In addition, our proposed model captures the taxi driver's habits and the time periodicity of trajectory. Features from multiple dimensions make our model more efficient.

We propose a model that treats this task as a regression problem, instead of a multi-class classification. Finally, we have demonstrated how using multiple embedding features in detail, strongly improves the final result.In this perspective, future works will be devoted to integrate different data sources. Benefit by our flexible model input, features of other dimensions can be easily added to our model.

Finally, our approach can be extended to the more general task of trajectory prediction, not just taxi but human or other means of transport mobility prediction. Our model has powerful multi-dimensional feature integration ability to effectively mining the association between different features, which will be a good reference for traffic policy formulation.

Appendix

A. *Roles and Responsibilties*

The main contributions of our team members are summarized in VII-A

TABLE II
CONTRIBUTION OF GROUP MEMBERS

Member	Contribution
Yu Zheng	Implement of the Geohash Sequence, WordCNN
Hao Liu	Implement of Prediction component
Jialin Li	Paper writing

References

[1] A. de Brébisson, É. Simon, A. Auvolat, P. Vincent, and Y. Bengio, "Artificial neural networks applied to taxi destination prediction," *CoRR*, vol. abs/1508.00021, 2015. [Online]. Available: http://arxiv.org/abs/1508.00021

[2] juzstu, "Tianchi_HaiYang," https://github.com/juzstu/TianChi_HaiYang, 2020.

[3] A. Rossi, G. Barlacchi, M. Bianchini, and B. Lepri, "Modeling taxi drivers' behaviour for the next destination prediction," *CoRR*, vol. abs/1807.08173, 2018. [Online]. Available: http://arxiv.org/abs/1807.08173

[4] R. Holbrook, "Seasonality-create indicators and fourier features to capture periodic change." https://www.kaggle.com/ryanholbrook/seasonality#What-is-Seasonality?, 2012.

[5] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," *CoRR*, vol. abs/1403.6652, 2014. [Online]. Available: http://arxiv.org/abs/1403.6652

[6] Y. Kim, "Convolutional neural networks for sentence classification," *CoRR*, vol. abs/1408.5882, 2014. [Online]. Available: http://arxiv.org/abs/1408.5882