Taxi Destination Prediction Based on Partial Trajectories

Li-Yi Lin, Qun Gao

JOHNS HOPKINS UNIVERSITY

Description

Motivation

The major shift of taxi dispatch system is the adoption of electronic dispatch system, in which it switches from broadcast-based radio messages for service dispatching to unicast-based messages. To improve the efficiency of electronic taxi dispatch systems it is important to be able to predict the final destination of a taxi while it is in service.

Goal

Predict the destination of taxi trips based on initial partial trajectories and other metadata (pick up time, date...)

Dataset

The dataset describes a complete year (from 01/07/2013 to 30/06/2014) of the trajectories for all the 442 taxis running in the city of Porto in Portugal and some other metadata(e.g. call type, time, etc.) described as follows.

Main features

- **Partial trajectory:** A list of GPS coordinates measured every 15 seconds, each with value of longitude and latitude.
- **Call type:** the way to request a taxi, including
  A: call to the taxi center
  B: call at a specific taxi stand
  C: call on a random street
- **Origin call:** ID for each phone number which is used for the demand.
- **Taxi stand:** ID of the taxi stand.
- **Time:** the pick up time of a ride.
- **Day type:** the day type of a trip's start time,including
  A: If this trip started on a holiday or any other special day (i.e. extending holidays, floating holidays, etc.);
  B: If the trip started on a day before a type-A day;
  C: Otherwise (i.e. a normal day, workday or weekend).

Preprocess: Input

Partial trajectory

The dataset contains a complete trajectory for each ride and we randomly generate a partial one as input with length between 0 and 1/3 of the whole. We take the last GPS coordinate of the full trajectory as the destination of this ride.

Call type

We embed this variable to a two dimensional vector: the first dimension stores the binary value of whether requesting through a phone call and the second one stores the taxi stand ID if requesting at a taxi stand. The vector with zero value means requesting on a random street.

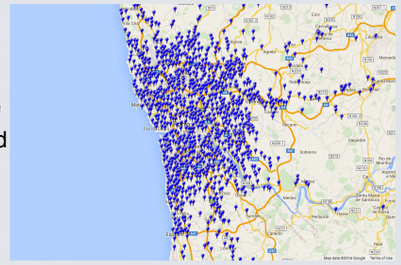
Pick up Time

The pick up time in the dataset is stored as the Unix Timestamp (in seconds). We first convert it to a normal format and then embed into a two dimensional vector: the first dimension represents whether the taxi ride starts between Monday and Thursday, or on Friday, or in weekend(Sat. and Sun.) and the second one represents the ride starts in one of the time period of a day (we divide a day into 48 intervals, 0.5 hour each).

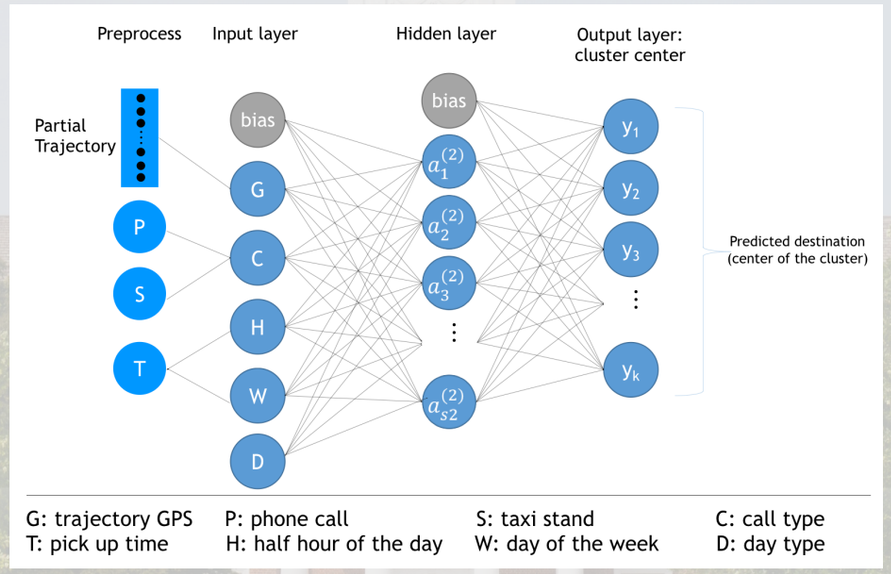
Preprocess: Output

We do the clustering to the destination points of the training set and use the cluster centers as the candidate destinations for the model.

The figure in the right is the result of clustering using mean-shift method. There are 1665 clusters in total. And we will predict the cluster labels in the model and take the cluster centers as the predicted destination.



Model: Multilayer Perceptron (MLP)



Training algorithm

- Randomly initialize weights Θ in the model.
- In each iteration:
  - Implement **forward propagation** to get the activations of each layer.
  - Compute the cost function $J(\Theta)$.
  - Compute the partial derivatives of $\frac{\partial}{\partial\Theta}J(\Theta)$.
  - Use stochastic gradient descent with **back propagation** algorithm to minimize $J(\Theta)$, and update weight Θ .

- **Input layer**

We take the preprocessed partial trajectories and metadata as input and normalize them to get better results.

Note: Since the MLP model need fixed size input, we try to convert the variable length of partial trajectories to a fixed one by taking the first 5 and last 5 GPS points of the trajectories. If the length are less than 10, we just repeat the first or the last point to reach the full size.

- **Hidden layer**

We use one hidden layer in the model as it's sufficient for most problems. And we try different units(neurons) in it to find the best.

- **Output layer**

The units in the output layer are the clusters generated before (1665 in total).We will take the cluster with highest probability in the output layer as the prediction result and use the cluster center as the predicted destination.

Results and analysis

We randomly select 90 thousand rides from the dataset as the training set for learning and 10 thousand as test set for prediction. We evaluate the model for prediction by computing the geographic distance between the predicted location and actual destination based on the Haversine distance, in which we regard the cluster center as the predicted destination.

$$a = sin^2\left(\frac{\phi_2-\phi_1}{2}\right) + cos(\phi_1)\,cos(\phi_2)\,sin^2\left(\frac{\lambda_2-\lambda_1}{2}\right)$$

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

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

The results are in the following table:

Model	# of hidden units	# of iterations	mean Haversine distance(km)
MLP	100	2000	3.2779
	200	3000	3.2729
	500	2000	5.757
	800	2000	5.97
SVM-RBF			5.4354

Results: By comparing different models, we find the MLP with 200 hidden units and 3000 iterations is the winner.
Analysis: Too many hidden units could cause overfitting.

Discussion

We used the multilayer perceptron model to predict the destination of a taxi ride based on its partial trajectory and other metadata and evaluate the result by computing the mean Haversine distance. Out best result is 3.2729 for MLP with 200 units and 3000 iterations. We can improve it by trying different hyper parameters (by time limit we didn't do well). In the preprocessing step we fixed the size of partial trajectories in a way and lost some information. We can improve our model by using recurrent neural networks (RNN), which can collect time series data and read the GPS points one each time.

References

- ECML/PKDD 15: Taxi Trajectory Prediction (I) https://www.kaggle.com/c/pkdd-15-predict-taxi-service-trajectory-i/data
- "Artificial Neural Networks Applied to Taxi Destination Prediction" http://arxiv.org/abs/1508.00021
- Comaniciu et al, *Mean Shift: A Robust Approach Toward Feature Space Analysis*, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 24, NO. 5, MAY 2002