Ride Hailing Supply and Demand Forecasting using Didi-Tech Dataset
Data Collection Report

Kimberly Williamson, Gopal Menon
Data Mining
Spring 2017, University of Utah

I. Background

Didi Chuxing is the leading ride hailing company in China and processes over 11 million trips, plans over 9 billion routes and collects over 50TB of data per day. They organized a worldwide algorithm challenge in the year 2016 [1] for forecasting ride supply and demand. Teams were asked to come up with a ride supply and demand forecasting algorithm based on a train and test dataset that was provided. Algorithms submitted by the teams were evaluated based on a published metric that measured the average error in the forecast.

II. Project Scope

The project team will use the 2016 Didi algorithm competition dataset to forecast taxi trip supply, demand, and expected fare for any given date, time, and location using the regression methods that will be covered in the Data Mining 2017 Spring semester at the School of Computing of the University of Utah. The accuracy of the forecast will be evaluated using the same average forecast error metric that was used in the competition.

III. How the Data was obtained

Training and Testing data was provided by Didi as part of the 2016 competition.

IV. Data Size

TABLE I
Train and Test number of rows

Dataset	Orders	Traffic	Weather	Cluster Map	POI
Train	8,540,614	193,553	4811	66	66
Test	557,985	8381	78	66	66

V. Data Format

Didi divides a city into n non-overlapping square districts $D = \{d_1, d_2, \ldots, d_n\}$ and divides one day uniformly into 144 time slots $t_1, t_2, \ldots, t_{144}$, each 10 minutes long. The training set contains 3 consecutive weeks of data for City M in 2016, and we need to forecast the supply-demand gap for a certain period in the 4^{th} and 5^{th} weeks of City M. Following are the tables in the dataset in tab separated format [1]:

1) **Order Info:** The basic information of an order and contains order id, driver id, passenger id, start district hash, destination district hash, price and time.
2) **District Info:** The district hash mapped to a district id.
3) **POI Info:** The points of interest(POI) for each district. The POI includes the POI class which details the type of POI and the count of that type of POI in the district. The district is represented with a district hash.
4) **Traffic Jam Info:** The overall traffic status on the roads in a district and contains the district hash, number of road sections at different congestion levels and recorded time of congestion.
5) **Weather Info:** The temperature and pollution levels recorded every 10 minutes.

VI. Data Processing

Our plan is to create an OrderItem for each order row by joining all the above tables. The categorical values (POI and Traffic Jam) will be stored as separate elements, where the value will be 0 or the value will reflect a count or indicator of 1, depending on if the element is present or not in the OrderItem. The OrderItem will be structured as a key-value pair, where the key is the Order Id and the value is the OrderItem elements stored in a list.
Example of the structure below. Detailed breakdown in Table II.
$\{70fc7c2bd : (5601832, 238de35f4, 1, 10, 37.5, 2016 - 01 - 1500 : 35 : 11, 0, 5, 0, 350, 0, 32, 7, -9, 66)\}$

VII. Data Simulation

The dataset has missing or incomplete data due to orders that were not fulfilled by drivers, multiple orders generated by a customer for the same ride, orders generated by third party applications, and data that could not be collected due to glitches or technological limitations. This missing and incomplete data will need to be filled in before it is used for prediction of ride volumes and pricing. Unfullfilled orders will be removed and not used as inputs into the regression models. We plan to use matrix completion techniques for this purpose at this point of time. The Didi dataset is already separated into train and test parts and we plan to use k-fold cross validation in order to come up with a good model for prediction.

REFERENCES

[1] "Algorithm Competition." *Algorithm Competition*. N.p., n.d. Web. 28 Jan. 2017.

TABLE II
BREAKDOWN OF ORDERITEM-NOTE:POI AND TRAFFIC JAM WILL BE REPEATED. ONCE FOR THE START DISTRICT AND ANOTHER TIME FOR THE DEST DISTRICT.

Order-Hash values truncated for example	Order Id Driver Id Passenger Id Start District Id Dest District Id Price Time	70fc7c2bd 5601832 238de35f4 1 10 37.5 2016-01-15 00:35:11
POI-one element per POI class, more elements than shown.	1#1 2#1 1#2	22 0 5
Traffic Jam Levels	1 2 3 4	0 350 0 32
Weather	Condition Temperature Pollution	7 -9 66