Machine Learning Nanodegree Capstone Proposal

Sourav Jha

September 13, 2017

1 Domain Background

The taxi industry is evolving rapidly. New competitors and technologies are changing the way traditional taxi services do business. While this evolution has created new efficiencies, it has also created new problems.

One major shift is the widespread adoption of electronic dispatch systems that have replaced the VHF-radio dispatch systems of times past. These mobile data terminals are installed in each vehicle and typically provide information on GPS localization and taximeter state. Electronic dispatch systems make it easy to see where a taxi has been, but not necessarily where it is going. In most cases, taxi drivers operating with an electronic dispatch system do not indicate the final destination of their current ride.

Another recent change is the switch from broadcast-based (one to many) radio messages for service dispatching to unicast-based (one to one) messages. With unicast-messages, the dispatcher needs to correctly identify which taxi they should dispatch to a pick up location. Since taxis using electronic dispatch systems do not usually enter their drop off location, it is extremely difficult for dispatchers to know which taxi to contact.

To improve the efficiency of electronic taxi dispatching systems it is important to be able to predict how long a driver will have his taxi occupied. If a dispatcher knew approximately when a taxi driver would be ending their current ride, they would be better able to identify which driver to assign to each pickup request.This problem is part of two kaggle competitions(ECML/PKDD 15*[1]* and NYC taxi *[5]*).

2 Problem Statement

In this project, I will build a predictive framework using supervised learning that is able to infer the trip time of taxi rides in Porto, Portugal based on their (initial) partial trajectories.It is a regression problem to predict the taxi trip.

3 Datasets and Inputs

The dataset contains a complete year (from 01/07/2013 to 30/06/2014) of the trajectories for the 442 taxis running in the city of Porto, in Portugal . These taxis operate through a taxi dispatch central, using mobile data terminals installed in the vehicles.Each ride is categorized into three categories: A) taxi central based, B) stand-based or C) non-taxi central based. For the first, there is an anonymized id, when such information is available from the telephone call. The last two categories refer to services that were demanded directly to the taxi drivers on a B) taxi stand or on a C) random street.

Each data sample corresponds to one completed trip. It contains a total of 9 (nine) features, described as follows:

- **TRIP_ID**: (String) It contains an unique identifier for each trip.
- **CALL_TYPE**: (char) It identifies the way used to demand this service. It may contain one of three possible values:
  - 'A' if this trip was dispatched from the central.
  - 'B' if this trip was demanded directly to a taxi driver on a specific stand.
  - 'C' otherwise (i.e. a trip demanded on a random street).
- **ORIGIN_CALL**: (integer) It contains an unique identifier for each phone number which was used to demand, at least, one service. It identifies the trip's customer if CALL_TYPE='A'. Otherwise, it assumes a NULL value.
- **ORIGIN_STAND**: (integer): It contains an unique identifier for the taxi stand. It identifies the starting point of the trip if CALL_TYPE='B'. Otherwise, it assumes a NULL value.
- **TAXI_ID**: (integer): It contains an unique identifier for the taxi driver that performed each trip.
- **TIMESTAMP**: (integer) Unix Timestamp (in seconds). It identifies the trip's start.
- **DAYTYPE**: (char) It identifies the daytype of the trip's start. It assumes one of three possible values:
  - 'B' if this trip started on a holiday or any other special day (i.e. extending holidays, floating holidays, etc.).
  - 'C' if the trip started on a day before a type-B day.
  - 'A' otherwise (i.e. a normal day, workday or weekend).
- **MISSING_DATA**: (Boolean) It is FALSE when the GPS data stream is complete and TRUE whenever one (or more) locations are missing.
- **POLYLINE**: (String): It contains a list of GPS coordinates (i.e. WGS84 format) mapped as a string. The beginning and the end of the string are identified with brackets (i.e. [and], respectively). Each pair of coordinates is also identified by the same brackets as [LONGITUDE, LATITUDE]. This list contains one pair of coordinates for each 15 seconds of trip. The last list item corresponds to the trip's destination while the first one represents its start.

4 Solution Statement

I have planned to use following supervised learning algorithms:

- Support Vector regression*[2]*
- XGBoost*[3]*
- Neural Networks*[4]*

Using the dataset ,i will preproccess and select the relevant features to train the regression model. I will use above three algorithms separately for regression model.

5 Benchmark Model

The problem is one of the kaggle competition ECML/PKDD 15*[1]*.The winner's solution had Root Mean Squared Logarithmic Error of 0.5354.For the project, I will compare my results with winner's result.

6 Evaluation Metrics

I will be using the Root Mean Squared Logarithmic Error (RMSLE) to evaluate different models. The RMSLE is calculated as

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}((\log(p_i+1)) - \log(a_i+1)))^2}$$

Where:

- n is the number of hours in the test set
- p_i is predicted count
- a_i is the actual count
- log(x) is the natural logarithm

7 Project Design

7.1 Programming Languages and Libraries

- Python 2
- scikit-learn - Machine learning library for python
- numpy and pandas - Libraries for data manipulation and visualization
- XGBoost - Library for developing fast and high performance gradient boosting tree models.

7.2 Preprocessing and Feature Selection

The first step in my project will be preprocess the data.As already mentioned in the "dataset and input" section,there is a feature "MISSING_DATA" if it is true, we don't have complete GPS data.Also,there are some data points,which has GPS data empty.I will drop all the rows which have missing data(both "MISSING_DATA" = True and POLYLINE= []).

The second step will be removing unnecessary features like MISSING_DATA and TRIP_ID which doesn't contain any relevant information. Then, i will calculate the trip length of each trip which will be regression value of each data point.

7.3 Training and Evaluation

I will take features generated in step 7.2 to train using supervised learning algorithms described in "Solution Statement" section.For training ,i will use K-Fold cross validation technique to avoid overfitting.

References

[1] https://www.kaggle.com/c/pkdd-15-taxi-trip-time-prediction-ii

[2] http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html

[3] http://xgboost.readthedocs.io/en/latest/model.html

[4] http://scikit-learn.org/stable/modules/neural_networks_supervised.html

[5] https://www.kaggle.com/c/nyc-taxi-trip-duration