Machine Learning Engineer Nanodegree

Capstone Proposal

Maimaitirebike Maimaiti
August 26, 2017

Proposal

Domain Background

Concerns about global climate change, energy security, and unstable fuel prices have caused many decision makers and policy experts worldwide to closely examine the need for more sustainable transportation strategies. Bike-sharing system, the shared use of a bicycle fleet, is one mobility strategy that could help address many of these concerns [1]. Although the 1st generation of bike-sharing programs began in Amsterdam in 1965, bike-sharing has received increasing attention in recent years with initiatives to increase cycle usage, improve the first mile/last mile connection to other modes of transit, and lessen the environmental impacts of our transport activities. Especially, the development of better methods of tracking bikes with improved technology gave birth to the rapid expansion of bike-sharing programs throughout Europe and now most other continents during this decade [2].

In 2013, Bay Area Bike Share (also known as Ford GoBike) was introduced as a pilot program for the Bay Area region, with 700 bikes and 70 stations across San Francisco and San Jose. Similar to car sharing, "bicycle sharing" in Bay Area is a membership-based system for short-term bicycle rental. Members can check out a bicycle from a network of automated stations, ride to the station nearest their destination, and leave the bicycle safely locked for someone else to use. While traditional bike rentals are loaned out for half-day or longer, bike sharing is designed for short, quick trips. Stations connect users to transit, businesses and other destinations, often providing the "last-mile connection." To reflect system design, pricing is set to discourage trips longer than 30 minutes. Users can take an unlimited number of 30-minute trips during their membership period; however, any individual trip over 30 minutes will incur an additional fee [3].

A crutial part of successfully operating such a BBS is the redistribution of bikes. Since the count of bikes in each station, each of which has a finite number of docks, fluctuates, redistribution operation must be performed periodically to make the bike-sharing service more efficient and environmentally friendly. However, staff moving bikes from areas of high supply/low demand to areas of low supply/high demand is time consuming, expensive, and polluting [1]. Thererfore, predicting the number of available bikes in each station over time is one of the key tasks to making this operation more efficient.

The modeling of bike availability using various features like time, weather, built environment, transportation infrastructure, etc., is an area of significant research interest. Froehlich et al. (2009) used four predictive models to predict the number of available bikes at each station: last value, historical mean, historical trend, and a Bayesian network. Kaltenbrunner et al. (2010) adopted an Autoregressive Moving Average (ARMA) model and Yoon et al. (2012) proposed a modified Autoregressive Integrated Moving Average (ARIMA) model considering spatial interaction and temporal factors. In a recent study, Ashqar et al. (2017) used Random Forest (RF) and (Least-Squares Boosting) LSBoost models to predict future bike availability at a given station. In this project we explore a neural network solution for predicting future bike availability.

Problem Statement

The goal of this project is to predict the number of available bikes at a bike share station using various predictors. Specifically, we will treat the number of available bikes at station *i* in the future as the predictor, which is denoted by Yi(*t* + *dt*), where Y is the number of available bikes, *i* (*i* = 1,2,...,70.) is the station number, *dt* is the prediction horizen time. In this project, *dt*=15 minutes will be considered. That means, the prediction would be for the next 15 minutes. The predictors that will be considered are the available bikes at station *i* at time *t* (current time), the month-of-the-year, day-of-the-week, time-of-day, and various weather conditions, like temperature, humidity, visibility, wind speed, precipitation, and events in a day (i.e., rainy, foggy, or sunny). Since the response variable Y takes interger values from 0 to 27, both regression and classification models can be considered. However, the focus of this project will be on using neural network to predict Y by taking a regression approach.

Datasets and Inputs

For this problem we use the dateset provided by Bay Area Bike Share (also known as Ford GoBike) for August 2013 to August 2015. This data is provided according to the Ford GoBike Data License Agreement. They make regular open data releases, plus maintain a real-time API.

The Dataset:

There are four different data files (*status*, *station*, *trip*, *weather*) in this dataset, each of which will be used at different level to solve the problem. The *status* data provides features like current bike availability and time information. The response variable Y will aslo be taken from this dataset by shifting it by *dt*. The various features in *station* and *trip* data will be considered if they are useful in making prediction. Weather related features in *weather* data will be used as predictors after removing redundant features. A breif description of the data is as follows:

***status* data**

station_id: station ID number (use "station.csv" to find corresponding station information)

bikes_available: number of available bikes

docks_available: number of available docks

time: date and time, PST

***station* data**

station_id: station ID number (corresponds to "station_id" in "status.csv")

name: name of station

lat: latitude

long: longitude

dockcount: number of total docks at station

landmark: city (San Francisco, Redwood City, Palo Alto, Mountain View, San Jose)

installation: original date that station was installed.

***trip* data**

Each trip is anonymized and includes:

Trip ID: numeric ID of bike trip

Duration: time of trip in seconds (trips <1 min and >24 hours are excluded)

Start Date: start date of trip with date and time, in PST

Start Station: station name of start station

Start Terminal: numeric reference for start station

End Date: end date of trip with date and time, in PST

End Station: station name for end station

End Terminal: numeric reference for end station

Bike #: ID of bike used

Subscription Type: Subscriber = annual or 30-day member; Customer = 24-hour or 3-day member

Zip Code: Home zip code of subscriber (customers can choose to manually enter zip at kiosk however data is unreliable)

***weather* data**

Daily weather information per service area, provided from Weather Underground in PST. Weather is listed from north to south (San Francisco, Redwood City, Palo Alto, Mountain View, San Jose). The futures included in this data are: 'date', 'max_temperature_f', 'mean_temperature_f', 'min_temperature_f', 'max_dew_point_f', 'mean_dew_point_f', 'min_dew_point_f', 'max_humidity', 'mean_humidity', 'min_humidity', 'max_sea_level_pressure_inches', 'mean_sea_level_pressure_inches', 'min_sea_level_pressure_inches', 'max_visibility_miles', 'mean_visibility_miles', 'min_visibility_miles', 'max_wind_Speed_mph', 'mean_wind_speed_mph', 'max_gust_speed_mph', 'precipitation_inches', 'cloud_cover', 'events', 'wind_dir_degrees', 'zip_code'

Note:

Precipitation_In: "numeric, in form x.xx but alpha ""T""= trace when amount less than .01 inch"

Cloud_Cover: "scale of 0-8, 0=clear"

Zip: 94107=San Francisco, 94063=Redwood City, 94301=Palo Alto, 94041=Mountain View, 95113= San Jose"

Solution Statement

This is a supervised learning problem. We first implement Random Forest and Least-Squares Boosting to predict the future bike availability (Y) based on features X which includes current bike availability, datetime information, and various weather parameters, etc.. Random Forest is a good choice for this problem because (1) there are very few assumptions needed in building a model, (2) it is considered to be robust against overfitting, and it runs efficiently and relatively quickly with a large amount of data and many input variables, which is the case in this problem. Least-Squares Boosting will also be used because it is a gradient boosting of regression trees that produces highly robust and interpretable procedures for regression. However, the focus of this project is to solve the problem with a neural network approach. Specifically, we will use Multi-layer Perceptron (MLP) regressor because Y would be treated as a continues variable. An important advantage of MLP is that it can learn complex non-linear models. Also it is suitable for large dataset like the one in this problem. Various number of layers and different number of neurons per layer will be explored in model tuning procedure and the final model will be selected based on mean absolute error. Then the results will be compared with the Benchmark model results.

Benchmark Model

In a recent paper, Ashqar et al. (2017) used Random Forest and Least-Squares Boosting algorithms to predict the bike availability in a given station and used mean squared error (MAE) as an evaluation metric. The average MAE results they obtained for RF and LSBoost are 0.37 bikes/station and 0.58 bikes/station. In this project these two values will be taken as Benchmark.

Evaluation Metrics

Since this is a regression problem, mean absolute error and mean squared error are good options for model evaluation. A recent study [5], whose results are taken as Benchmark in this project, used MAE as evaluation metric. In this project, mean squred error (MSE) will also be used as an evalluation metric in addition to MAE.

Project Design

The workflow for reaching the solution for the stated problem can be divided into the following steps:

Data Preprocessing

The four files in this dataset are in csv format and are well structured. However, there are missing values and incorrect entries, which issues need to be addressed first.

The status.csv file size in this dataset is 1.9 GB and it contains about 72 million data points. It is difficult to work with such a large dataset using an ordinary computer. Therefore, downsampling approach will be taken to reduce it to a managable size. Initial experimentation shows that downsampling the original data (1-min resolution) to 15-min resolution reduces the file size to about 130 MB and the number of data points reduces to about 4.8 million. Together with the other three files, size of the total dataset will be around 210 MB.

Feature selection is also necessary for this problem because some features in the dataset could be redundant. For example, features like 'max_temperature_f', 'mean_temperature_f', and 'min_temperature_f' can be highly correlated such that one of them may be sufficient for prediction.

For the ease of building a model, it is better to pull all the predictors and the reponse variables in a single dataframe. This requies joining the data from the four different files and put them together into a single table. Once data is joined together, categorical features need to be transformed into numerical features before feeding them into a model. One-hot encoding can be used for this purpose. Then, data will be splitted into training and test datasets. The training dataset will be used for training a model and the test data for evaluating the trained model performance.

Model Building and Evaluation

A Multi-layer Perceptron (MLP) regressor will be implemented for predicting the future bike availability (Y). MLP regressor trains a model with no activation function in the output layer, which can also be seen as using the identity function as activation function. Thus the output will be a set of continuous values, which is desired because reponse variable Y is also a continues variable. The first layer will have the same number of neurons as the number of encoded-features, excluding the bias. Then different hidden layers will be tested. The tuning procedure will include tuning the hyperparameters like the number of hidden neurons, layers, and iterations. The final model will be selected based on mean absolute error. Then its resutls will be compared with that reported by Ashqar et al. where RF and LSBoost models have been used.