Traffic Prediction in Supervised Learning

Yujun Jiang

Machine Learning

Prof. Aaron Hill

March 4, 2020

In my supervised learning case study, I will focus on the traffic data in the greater New York City area. For data preparation, I will be using the real-time traffic speed data based on geolocation, collected by NYC Open Data with NYCDOT's Traffic Management Center. To simulate a real-life end-user situation, I will be adding external data, including weather conditions, time of the days, and holidays. The final goal is to build a model that can predict the best route in terms of the least travel time between home and workplace for each user.

The original dataset has a continuous (quantitative) variable, SPEED, which is the average speed of a vehicle traveled between end points. This is considered as the input variable (x). The total commute time (TRAVEL_TIME) will be the output variable (y), representing the amount of time that a diver spent on his/her trip. For data transformation, some customized operations are needed because weather conditions and holidays are also important factors contributing to total commute time. For example, the amount of rain or road closure on holidays can increase the commute time. On Google Developers' website, it has introduced to us as to how to represent categorical data as strings or even numbers. At this moment, I will need to transform the categorical variables such us weather conditions into quantitative variables (e.g. cloudy = 0, drizzle = 1, showers = 2, thunderstorms = 3, etc.)

In my research, I have figured that the linear regression algorithm is one of most applicable methods for predicting the traffic time. However, this method cannot train the machine-learning algorithms with multiple features at the same time. Additionally, Naïve Bayes would be an option to determine the conditional probability of the estimated dependent variable values, conditional on the independent variables. The theorem holds strong independence assumptions between features. Unfortunately, the model can only calculate the conditional probabilities of the estimated dependent variable values, but not an actual number. Hence, Decision Trees would be the best model to solve this problem because it is able to address both numerical and categorical data. As the volume of data increases, Decision Trees perform better. One of the most important reasons for this model selection is that it is robust against co-linearity, which hints strong or perfect correlation between two independent variables. This helps the model to avoid redundancy in independent variable selection.

The biggest challenge for testing accuracy of traffic-flow prediction is that there are many sudden changes caused by construction or unpredictable events. Thus, a real-time learning technique could improve the algorithm and help it to adapt to new traffic conditions. For the new data, we can manually train on newer data, and deploy the resulting model once we are satisfied with its performance. On another hand, to schedule training on new data to take place would be an easier and faster way (e.g. once a week and automatically deploy the resulting model). The algorithm must generalize from (real-time) training data to make accurate predictions and provide feasible options to help shorten the total commute time. In this particular case, an acceptable rate of accuracy has to be as high as possible (at least 90% as expectation) to enable people make a real-time decision.