Demand and Trip Prediction in Bike Share Systems

Team members: Zhaonan Qu SUNet ID: zhaonanq

December 16, 2017

1 Abstract

2 Introduction

Bike Share systems are becoming increasingly popular in urban areas. With growing membership and expansion of service comes many operational challenges. A major challenge in their operations is the unbalanced demand and supply at bike stations as a function of time. Figure **1** shows number of bike trips in Jan 2017, aggregated into time intervals of 30 minutes according to start time, and summed across all days. We see that there is a clear temporal dependence of bike demand. Similarly, work districts have a higher demand during evening rush hours whereas residential areas have a higher demand during mroning rush hours.

Most bike share systems employ active rebalancing to ease the pressure at peak times. This means transporting a certain number of bikes from inactive stations to more active stations, or between stations and storage, in order to maximize the usage of each bike and ease supply and demand inbalance problems across bike stations at different times.

A quantitative, predictive model for the demand and supply would help operators plan bike transports more efficiently. This project aims to build such a model for bike arrivals at stations within one-hour time intervals, as a function of the following parameters:

- time of day, divided into 24 one-hour intervals
- whether a day is a weekend/federal holiday
- maximum and minimum temperature of a day
- precipitation

The output of our model will be the number of bike demand at a station within the one-hour time interval. Moreover, we also employ customer-level data, including:

- gender

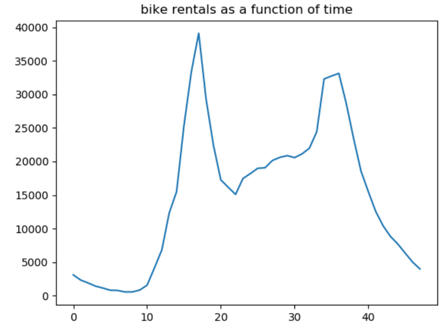
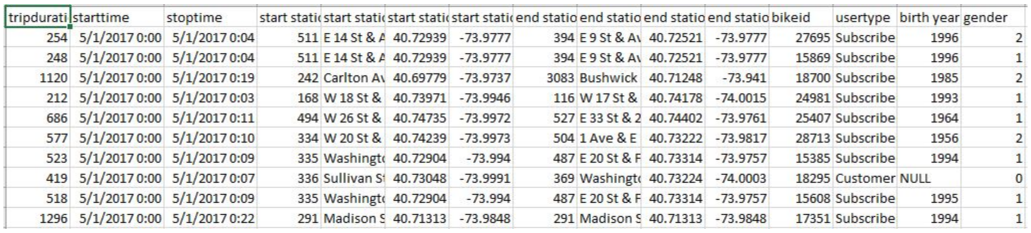


Figure 1: Bike usage in January 2017 as a function of time, discretized to 30-minute intervals.



tripdurati	starttime	stoptime	start stati	start stati	start stati	start stati	end statio	end statio	end statio	end statio	bikeid	usertype	birth year	gender
254	5/1/2017 0:00	5/1/2017 0:04	511	E 14 St & A	40.72939	-73.9777	394	E 9 St & Av	40.72521	-73.9777	27695	Subscribe	1996	2
248	5/1/2017 0:00	5/1/2017 0:04	511	E 14 St & A	40.72939	-73.9777	394	E 9 St & Av	40.72521	-73.9777	15869	Subscribe	1996	1
1120	5/1/2017 0:00	5/1/2017 0:19	242	Carlton Av	40.69779	-73.9737	3083	Bushwick	40.71248	-73.941	18700	Subscribe	1985	2
212	5/1/2017 0:00	5/1/2017 0:03	168	W 18 St &	40.73971	-73.9946	116	W 17 St &	40.74178	-74.0015	24981	Subscribe	1993	1
686	5/1/2017 0:00	5/1/2017 0:11	494	W 26 St &	40.74735	-73.9972	527	E 33 St & 2	40.74402	-73.9761	25407	Subscribe	1964	1
577	5/1/2017 0:00	5/1/2017 0:10	334	W 20 St &	40.74239	-73.9973	504	1 Ave & E	40.73222	-73.9817	28713	Subscribe	1956	2
523	5/1/2017 0:00	5/1/2017 0:09	335	Washingt	40.72904	-73.994	487	E 20 St & F	40.73314	-73.9757	15385	Subscribe	1994	1
419	5/1/2017 0:00	5/1/2017 0:07	336	Sullivan St	40.73048	-73.9991	369	Washingt	40.73224	-74.0003	18295	Customer	NULL	0
518	5/1/2017 0:00	5/1/2017 0:09	335	Washingt	40.72904	-73.994	487	E 20 St & F	40.73314	-73.9757	15608	Subscribe	1995	1
1296	5/1/2017 0:00	5/1/2017 0:22	291	Madison S	40.71313	-73.9848	291	Madison S	40.71313	-73.9848	17351	Subscribe	1994	1

Figure 2: Example of original data from Citi bike

- age

combined with the above parameters, to predict the duration of a trip departing from a station. For the demand prediction problem, we use linear regression as a baseline, and in addition use deep neural networks to perform prediction of the number of bikes and the duration of trips. The models are built using Tensorflow's linear regression and deep neural network API.

3 Related Work

There has been several previous academic and non-academic work studying the bike share demand prediction problem. The paper approach differs substantially from the present study in that their main covariate is the number of taxi trips during morning rush hours. They find that there is a substantial positive correlation between the number of bike trips and number of taxi trips. However, such data is not readily available when we want to make real-time predictions. So in our study we only employ data available from bike share system itself, combined with readily-available weather information. A point worth noting from Cornell paper is that they aggregate bike stations based on neighborhoods characterized by economic and demographic variables, and find that analyzing pairwise trips at the neighborhood level instead of looking at individual stations in bike sharing systems can improve the predictions improves the predictions. In our study, we focus on station-level predictions, but plan to pursue their suggestion in the future.

4 Dataset and Features

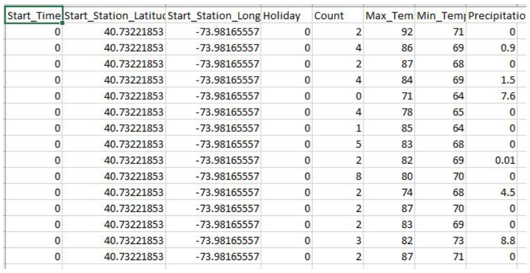
We use system data avaialble from Citi bike's website[1]. The Citi bike data set contains information about each bike trip taken, including starting date and time, starting station, end station, trip duration, and customer age and gender. See Figure 2 for a snippet of data file.

We use trip data from June to August 2017, with 20000 number of training examples and 6000 number of test examples. In order to perform regression on the demand of bikes, we discrete each day into 24 one-hour intervals, count the number of bikes departing from each station in each time interval, determine whether each date is a work day, i.e. not weekend or federal holiday, and finally associate weather data to each date. Weather data is obtained from National Centers' for Environmental Information website[2]. A snippet of the processed data is shown in Figure 3.

As an illustration of how weather affects bike usage, we selected a pair of stations with relatively high usage during morning rush hours. Station "Pershing Square North" is near Grand Central, and Station "East 24th Street and Park Avenue South" is in the Flatiron District, with shops, office space, as well as a university. Figure shows the number of trips from the former to the latter in March 2017 between 6 to 6:30 am, as a function of the temperature during that time. We see a clear dependence on temperature. Moreover, bike usage patterns have a strong dependence on whether a given day is work day or weekend/holiday. So we also select this categorical variable as one of our features. For the problem of predicting trip duration given starting station and additional features such as customer age and gender, we used similar processing methods.

[1] https://www.citibikenyc.com/system-data

[2] https://www.ncdc.noaa.gov/cdo-web/datasets/GHCND/stations/GHCND:USW00094728/detail



Start_Time	Start_Station_Latituc	Start_Station_Long	Holiday	Count	Max_Tem	Min_Temp	Precipitatio
0	40.73221853	-73.98165557	0	2	92	71	0
0	40.73221853	-73.98165557	0	4	86	69	0.9
0	40.73221853	-73.98165557	0	2	87	68	0
0	40.73221853	-73.98165557	0	4	84	69	1.5
0	40.73221853	-73.98165557	0	0	71	64	7.6
0	40.73221853	-73.98165557	0	4	78	65	0
0	40.73221853	-73.98165557	0	1	85	64	0
0	40.73221853	-73.98165557	0	5	83	68	0
0	40.73221853	-73.98165557	0	2	82	69	0.01
0	40.73221853	-73.98165557	0	8	80	70	0
0	40.73221853	-73.98165557	0	2	74	68	4.5
0	40.73221853	-73.98165557	0	2	87	70	0
0	40.73221853	-73.98165557	0	2	83	69	0
0	40.73221853	-73.98165557	0	3	82	73	8.8
0	40.73221853	-73.98165557	0	2	87	71	0

Figure 3: Processed data for bike demand prediction. Count is the number of bikes rented from the station within the time interval.

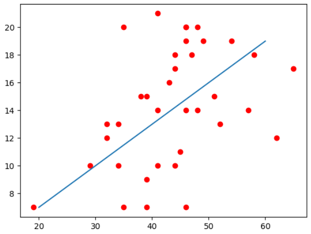


Figure 4: Weather dependence of bike usage between Pershing Square North and East 24th Street and Park Avenue South

5 Methods and Experiments

5.1 Linear Regression

Our baseline model for both demand and trip duration predictions is linear regression. The loss function for linear regression is given by

$$\ell(\theta) = \frac{1}{m}\sum_{i=1}^{m}(\theta x^{(i)} - y^{(i)})^2 + \alpha\|\theta\|^2$$

where α is the strength of regularization to prevent overfitting and we have used the convention that $x^{(i)}$ has entry 1 for the intercept term. The loss is minimized with gradient descent,

$$\theta \leftarrow \theta - \beta\nabla\ell(\theta)$$

with a learning rate β of 0.01.

5.2 Regression with L2 loss and Neural Network

As a comparison for both prediction problems, we used a two-hidden-layer neural network with L2 loss with regularization. The forward propagation for this neural network is given by

$$\begin{aligned}
\hat{y} &= W^{[2]}a^{[1]} + b^{[2]} \\
a^{[1]} &= \text{Sigmoid}(W^{[1]}a^{[0]} + b^{[1]}) \\
a^{[0]} &= \text{Sigmoid}(W^{[0]}x + b^{[0]})
\end{aligned}$$

with regularized loss function

$$\ell(W,b) = \frac{1}{m}\sum_{k=1}^{m}(\hat{y}^{(k)} - y^{(k)})^2 + \alpha\sum_{i=1}^{2}\left(\|W^{[i]}\|^2 + \|b^{[i]}\|^2\right)$$

RMSE (1/1000 seconds)	Training Error	Test Error
Linear Regression	0.552630	0.593969
Neural Network Regression	0.556546	0.545661

Table 1: RMSE of Trip Duration Prediction with linear regression and neural network regression.

The conventional optimization method for back-propagation is gradient descent. For each mini batch, we calculate the gradients of ℓ with respect to $W^{[i]}, b^{[i]}$, and use the update rule

$$\theta \leftarrow \theta - \alpha \nabla_p \ell_{MB}$$

where θ is a parameter and $\alpha = 5$ is the learning rate.

For the implementation of neural network through DNNRegressor on Tensorflow, we use AdamOptimizer proposed in that comes with the Tensorflow package. Briefly, the idea of the Adam (adaptive moment estimation) optimization method is as follows. As in gradient descent, it updates weights based on gradient, but the main difference with gradient descent is that Adam maintains parameter-specific learning rates and also adpatively changes them based on the first and second moments of the gradients of the weights in recent iterations. More specifically, it calculates an exponential moving average of the gradient and its norm squared, and parameters β_1 and β_2 control the decay rates of the two moving averages. After some tuning of parameters, we selected $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 0.1$ in the input to AdamOptimizer in Tensorflow.

6 Results and Discussion

6.1 Trip Duration Prediction

We used data from June, July and August 2017, randomly selected 50000 data points, split them randomly into training data (70%) and test data (30%), and performed 10-fold cross-validation on the training data. We use root mean squared error as the metric to evaluate our model. The RMSE is given by

$$\text{RMSE}=\sqrt{\frac{1}{m}\sum_{i=1}^{m}(y^{(i)} - \hat{y}^{(i)})^2}$$

for each run, we have 10 cross-validations each yielding a root mean squared error. We take the average of the 10 to obtain the training error. Similarly, the test error is simply the RMSE using the parameters trained using the training sample. See Table 1 for the final training and test errors for linear regression and neural network regression. As mentioned before, we used AdamOptimizer on Tensorflow to minimize the loss. After some tuning of parameters, we selected $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 0.1$ in the input to AdamOptimizer in Tensorflow.

Note that before we ran the algorithms, we performed data normalization. More specifically, we rescaled each feature column with

$$\frac{f - f_{mean}}{f_{max} - f_{min}}$$

and we rescaled the label vector, which is the vector of trip durations in seconds, by dividing it by 1000. This will give us an easy interpretation of the RMSE for the experiments. Figure 5 shows the training loss of over 10000 epochs, and test error using averaged cross-validated for linear regression and neural network.

As we can see, the cross-validated training error is close to test error for neural network, meaning that the model is not overfitted. A 0.55 RMSE corresponds to 550 seconds of error, which is about 9 minutes. Linear regression and neural network have comparable training error, but linear regression appears to be overfitted compared to neural network.

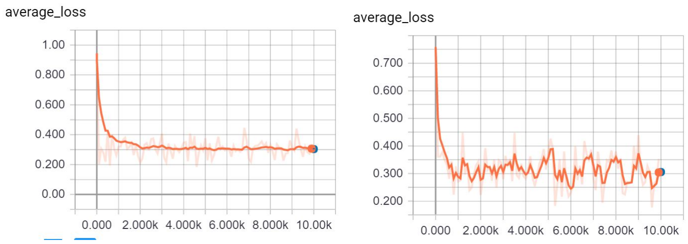


Figure 5: Left, training loss of DNN and right, training loss of linear regression. Blue dots are test errors using corss-validated average training parameters.

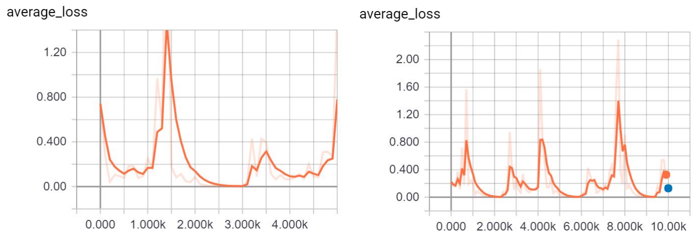


Figure 6: Left, training loss of DNN and right, training loss of linear regression. Blue dots are test errors using corss-validated average training parameters.

6.2 Demand Prediction

We used data from June and July 2017 as training data, and August 2017 as test data. We performed 10-fold cross validation to produce 10 different sets of trained parameters and averaged them as the parameter for test set. We also calculated the average loss of the 10 experiments as the training loss. Figure 6 shows the training loss as a function of epochs.

As before we rescaled each column of the feature and label data.

The training loss for both linear and neural network have periodically increasing loss. We changed several hyperparameters, but this pattern persists. We suspect that this is due to improper normalization of data, and will investigate this further in future work.

7 Conclusion and Future Work

We performed linear and neural network regression to predict the demand for bikes at bike share stations, as a function of time, day of week, temperature, and precipitation, and to predict the duration of bike trips as a function of age and gender of customer. We see that the duration prediction has a good fit of training data and small generalization error, but prediction of number of trips seems to yield periodically increasing loss, which is something that should be addressed in future work.

As the weather data we could import on a large scale was only for each 24-hour period, the dependence on weather was not very strong. However, we are working on scraping finer historical weather data, and that should improve the prediction results considerably. It would also be interesting to model trips between pairs of stations, and build models to predict the destination given customer information and starting station. We plan to do this in future work. As rebalancing efforts by Citi bike accounts for a significant amount of bike transfers, it is also important to incorporate those transfers, if we are able to access more detailed data.

References

[1] Adam Optimizer: https://arxiv.org/abs/1412.6980

[2] Tensor Flow: http://download.tensorflow.org/paper/whitepaper2015.pdf

[3] Citi Bike: https://www.aaai.org/ocs/index.php/WS/AAAIW15/paper/viewFile/10115/10185

[4] https://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/viewFile/9698/9314