Alexander Tough

Final Project - Intro to Data Science

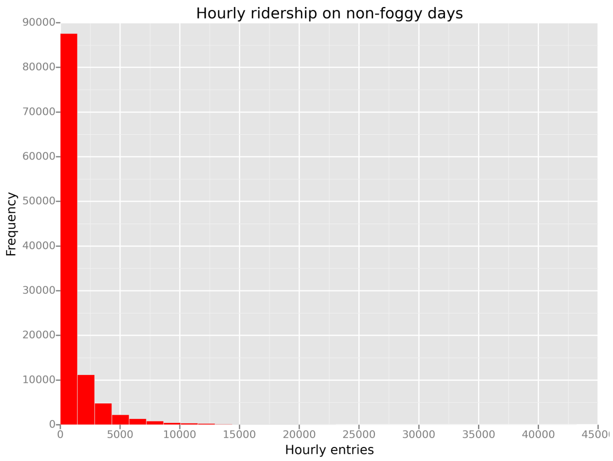
The New York City subway is one of the busiest rapid transit systems on Earth. In 2011 alone, Metrocards were swept 1.6 billion times through the turnstiles of the NYC subway system [1]. With such a large quantity of people using the NYC subway system, one wonders about what drives people to use the subway. In particular, are there non obvious factors that influence subway ridership?

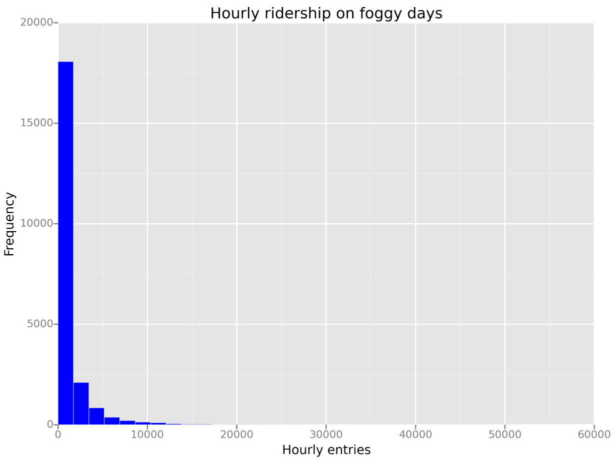
One of these non-obvious factors that comes to mind is fog. We thus want to answer:

“Does fog affect New York City ridership? If so, how?”

To answer this question we analyze New York City subway ridership and New York City weather conditions during the month of May of 2011 [i]. We then use this analysis to predict subway ridership, and discuss how we would handle large amounts of data to answer this question using the MapReduce programming model.

The first thing we must do is to take a look into the data we are hoping to analyze. What specific components of our data are we interested in? Taking our main question into account, it would make sense to analyze hourly subway ridership on both foggy and non foggy days. How does subway ridership behave on both foggy and non-foggy days? In statistical terms, “what is the distribution of subway ridership on both foggy and non-foggy days?”. We can eyeball an answer to this question by plotting two histograms: one with ridership on foggy days and one with ridership on non-foggy days. From this point on, we express subway ridership in terms of the number of entries per hour to a station.





We can see that the both the distribution of hourly ridership on non-foggy and foggy days resemble an exponential distribution. It would be nice if both of these distributions resembled a normal distribution, since we can then derive some very interesting properties. Let’s try to transform both hourly ridership distributions so that they resemble a couple of normal distributions.

Since the hourly ridership is exponentially distributed in either case, we can apply a Box Cox transformation to the hourly ridership distribution [ii]. The Box Cox transformation maps a value of the hourly ridership *y* to the value log(*y*+1). This transformation has the nice property that 0 maps to 0.

To test for normality, we can apply the Anderson-Darling test to both sets of the transformed data. Although the Shapiro-Wilk test is more powerful, it does not work well for large sample sizes [2]. The null hypothesis for the Anderson-Darling test is that the data comes from a population that follows a normal distribution. Applying the Anderson-Darling test for the transformed hourly ridership on non-foggy days, we get A^2 (Anderson-Darling test statistic) value of 2250.51629407. Since this statistic is much larger than the critical value of the normal distribution at a 5% significance level (0.787), we can reject the null hypothesis at a 95% confidence level that the transformed hourly ridership on non-foggy days follows a normal distribution. Applying the same test for transformed hourly ridership on foggy days, we get A^2 = 409.49696682 and arrive at the same conclusion.

We know that both distributions (even transformed) are not normal. Unfortunately then, we cannot find out whether hourly ridership in general is greater on foggy days than on non-foggy days. However, we can find out whether the distributions of hourly ridership on both foggy and non-foggy days come from the same population. In simpler terms, we can find out whether there is a difference in ridership on foggy and non-foggy days. To do this, we don't assume anything about the distribution of our data and apply the Mann Whitney U test. Our null hypothesis here would be that the number of entries of foggy days and the number of entries of non-foggy days came from the same population. We obtained a two sided p value of 1.21831138209e-05. Since the two sided p-value is less than 0.05, with a 95% confidence level we can reject the null hypothesis and conclude that the number of hourly entries to a subway station is statistically different between foggy and non-foggy days.

We can thus say that fog must somehow influence the number of hourly entries. Could we somehow use this information to our advantage?

Let's say that we wanted to predict hourly ridership. What could we base our prediction on? We could start off by predicting ridership using a random feature such as temperature. But why not use fog, which we have proven has some significant influence on the number of hourly entries?

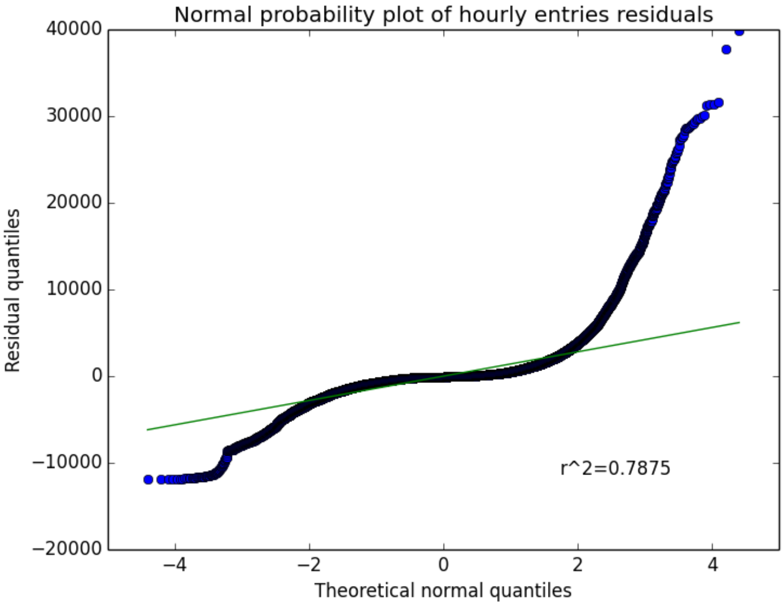
We can apply linear regression with gradient descent to predict hourly ridership. Our model is comprised of:

- Features: fog
- Number of iterations: 75
- Learning rate (α) : 0.1

Running this model **using all of the data in [i] as training data**, predicting hourly ridership of the training set, we get a r^2 value of 0.418171656757.

Although the coefficient of determination (r^2) is a good way of measuring the effectiveness of our model, taking a look at the frequency distribution of the residuals (the difference between the original hourly entries of the training data and the predicted values) will tell us whether our model leaves any structure in the data out. If the frequency distribution of the residuals is approximately normal with a mean of 0, we can conclude that our model does not leave any structure in the data out.

Let’s see if this is the case by creating a normal probability plot of the frequency distribution of the residuals.



We see that the distribution of the residuals is not normal, since the residuals (the blue points) stray far away from the best fit line [iv]. We can thus conclude that the ‘fog’ attribute indeed does not account for some of the structure present in the input data.

We can observe that with ‘fog’ as the only attribute in our linear regression model we get a an average coefficient of determination. Of course, this coefficient could be noticeably improved by including more features in the model, such as temperature, station and time of day, among others. But I wanted to see how far we could go with just one attribute.

One shortcoming of the model is that it could be possibly learning specific patterns about the training data that does not generalize to the population. This is known as overfitting.

One way to combat overfitting is described in [3] :

“Many techniques exist to combat model overfitting. The simplest method is to split your data set into training, testing and validation sets. The training data is used to construct the model. The model constructed with the training data is then evaluated with the testing data. The performance of the model against the testing set is used to further reduce model error. This indirectly includes the testing data within model construction, helping to reduce model over fit. Finally,the model is evaluated on the validation data to assess how well the model generalizes.”

So herein concludes our preliminary analysis of how fog affects New York City ridership for the month of May of 2011. You must be thinking: what if the month of May of 2011 was not a typical month of May for ridership and weather-wise? Any conclusions we may have derived are “overfitted” (pun intended). Indeed, this may be the case. What if we had more data, for instance, NYC ridership with weather for the past 30 months of May? What if we had subway ridership and weather data of the month of May for every city in America? Our analysis would then be more valid. But we would also be facing the fact that perhaps our data exceeds 5 TB, which merits the use of MapReduce.

As noted in class, MapReduce consists of a mapper and a reducer. The mapper takes our data and produces a series of intermediate key-value pairs which are then collapsed by the reducer.

Thus, to apply MapReduce, we must ask ourselves a couple of questions:
- What will the intermediate key-value pairs look like?
- How will the reducer collapse these key-value pairs?

To answer the first question, remember that we are interested in the distribution of hourly subway ridership on both foggy and non-foggy days. Hence, for both foggy and non-foggy days, we need each value of hourly entries to a subway station with the total count of how many times that value of hourly entries appears. Therefore the key-value pair emitted by the mapper would look like this at a high level:

{key: (ENTRIESn_hourly, fog), count : 1}

ENTRIESn_hourly corresponds to a value of hourly entries.
fog has a binary value: 0 means no fog, 1 means fog.
count is the number of times the value of ENTRIESn_hourly appears. It is set to 1 initially.

To answer the second question, with this key-value pair, the reducer would then simply group all of those entries with the same (ENTRIESn_hourly, fog) value and add up their counts. This way, we will accomplish our goal: for both foggy and non-foggy days, we get each value of hourly entries with the total count of how many times that value of hourly entries appears, i.e. the frequency distribution of the hourly entries according to the presence of fog.

We therefore conclude our analysis on subway ridership according to fog during May 2011. We have learned that the frequency distribution of hourly entries on both foggy and non-foggy days does not follow a normal distribution. We employed a Mann Whitney U test to conclude that fog does indeed play a factor in NYC hourly subway ridership. We then used this fact to yield a linear regression with gradient descent model with a coefficient of determination of 0.418171656757. We realized that our model may have overfitted, and described a possible remedy. We finally considered how we would employ MapReduce to handle potentially large datasets.

Footnotes:

[i] The New York City subway ridership data for the month of May of 2011 is publicly available at http://web.mta.info/developers/turnstile.html and the New York City weather conditions data is available for the same month is available at http://www.wunderground.com/weather/api/ . A combined version of both data sets which was the main data set analyzed for this report is available at https://www.dropbox.com/s/meyki2wl9xfa7yk/turnstile_data_master_with_weather.csv

[ii] More information about the Box Cox transformation can be found here: http://robjhyndman.com/hyndsight/transformations/ . For this model $\lambda 1 = 0$ and $\lambda 2 = 1$.

[iv] More information about normal probability plots can be found here: http://en.wikipedia.org/wiki/Normal_probability_plot

Bibliography:

[1] Visualizing The New York Subway System's 'Data Exhaust'. Emily Badger. Available at: http://www.theatlanticcities.com/commute/2012/07/visualizing-new-york-subway-systems-data-exhaust/2720/

[2] Shapiro, S. S. and Wilk, M. B. (1965). "*An analysis of variance test for normality (complete samples)*", Biometrika, 52, 3 and 4, pages 591-611. See section 6.1, "Evaluation of test".

[3] The Field Guide To Data Science. Booz Allen Hamilton. p.88. Available at http://www.boozallen.com/insights/insight-detail/data-science-field-guide