Analyzing the Cascading Structure of Traffic Congestion in New York City Taxi Networks

Jess Peterson, Sanjana Rajan

December 9, 2014

Abstract

In this paper, we use New York City taxi trip data from 2013 and attempt to predict congestion at specific locations in the city at a given time. This paper contains the following three contributions: 1) A method for organizing origin-destination trip data and representing it as a graph; 2) Metrics that that represent congestion at a given location, including whether that location causes (injects) or improves (absorbs) traffic; 3) An attempt to predict congestion at time $t + 1$ at a given node with data about congestion at time t. While we successfully represent the taxi data and measure congestion, our predictive model is generally unsuccessful with an R^2 value close to zero.

1 Introduction

In New York City, the average commuter spends 59 hours a year stuck in traffic, which translates to more than 1000 dollars a year wasted on fuel, according to a recent Business Insider article. Efforts have been made to reduce this wasted time such as initiatives that increase public transportation use and encourage driving during off-peak hours. However, there have been fewer efforts to understand and model the structure of traffic – where congestion originates, how it propagates, and how it then subsides.

In attempting this task, we use extensive data from the New York City taxi system from 2013. Each month in the data set contains more than 13 million taxi trips, meaning taxi trips are a potential source of representative data on New York's overall traffic. More specifically, we ask the following question: Given a data set of taxi data, can we identify how traffic delays propagate and predict where they next spread?

To answer this question, we perform the following organization and analysis of the data. First, we organize taxi trips from any origin and destination in the city into a network with nodes representing square areas in the city and edges representing taxi trips between those nodes. Second, we define metrics for node and edge congestion along with metrics that can become features in a machine learning model. Third, we run a linear regression on these congestion metrics for a given hour and attempt to predict congestion at that same node in the following hour. We believe this kind of work holds promise. If cities can accurately identify and predict congestion based on taxi data alone, they might be able to use an existing data source (city taxis) in real time to preempt problems and minimize the effects of predicted traffic for all drivers.

2 Prior work

There have been many attempts to analyze cascading behavior and delay propagation in transit networks, although few have used the type of car-level data in the New York City Taxi Network. The prior research most similar to our task relates primarily to general cascading behavior and data propagation in airport networks, due to the considerable amount of publicly available airline scheduling data. However, even general analysis of a "contagion" spreading through a network can be applied to the spread of congestion. In general, we found related work to be helpful for defining metrics that our model would use.

2.1 Systemic Delay Propagation in the US Airport Network

Fleurquin et al. model the propagation and magnification of delays through an airline network. To do this, they define metrics to quantify levels of conges-

tion in parts of the network. They utilize data from the Bureau of Transport Statistics which details 9,687,800 flights operated within the United States in 2010 to create networks in which nodes represent airports, and edges between nodes represent direct flights between these airports, weighted with the average delay between those airports during a particular period of time. Cascades of congested airports are observed when a congested airport with many direct flights to another airport causes later congestion in that airport. We adapt the ideas of node-level congestion to taxi trip data.

2.2 How Much Delay does New York Inject into the National Airspace System?

Bolanos et al. attempt a related task to Fleurqin et al., but they add additional complexity to their model which we incorporate into our metrics. Bolanos et al. make a distinction between the amount of air traffic delay that is due to propagation from earlier flights versus delays caused at the current airport. In order to do this, the authors define two types of delays: propagated and original. For example, if a plane leaves airport A 10 minutes late, arrives and departs airport B 10 minutes late, and then arrives at airport C 15 minutes late, the model says that 10 minutes of the delay at airport C is propagated from A, 0 minutes are attributable to airport B, and 5 minutes are attributable to airport C. After calculating these delays for all flights (edges), the authors create an origin-destination matrix where each directed edge's weight represents the total propagated delay between two nodes. They then create in-strength and out-strength totals for each node (row and column totals) and calculated the normalized difference between in-strength and out-strength for each node, which gives a final measure of whether a node propagates, creates, or absorbs (reduces) delays. We take inspiration from the authors' approach and incorporate the notion of absorbing and injecting delays into our congestion metric (defined below).

2.3 Cascading Behavior in Large Blog Graphs

Although it is not specifically related to delays and congestion, Leskovec et al. discussion of cascading behavior in a blog graph is broadly useful, especially for the idea of a contagion spreading to neighbors, which we adapt in our work. The authors explore the way that information propagates in cascade structures through a blog post network and create a simple model through which to analyze cascades. Leskovec et al. note that most cascades in the post network are tree-like shapes - where the number of edges in a cascade increases linearly with the number of nodes, while the diameter increases logarithmically. Following this analysis, they propose a cascade-generation model in which a randomly chosen cascade initiator *infects* its neighboring nodes with some uniform probability, and infected nodes can then infect their neighbors in the same fashion. We adapt this notion of infection from neighbors for use as a metric in our model.

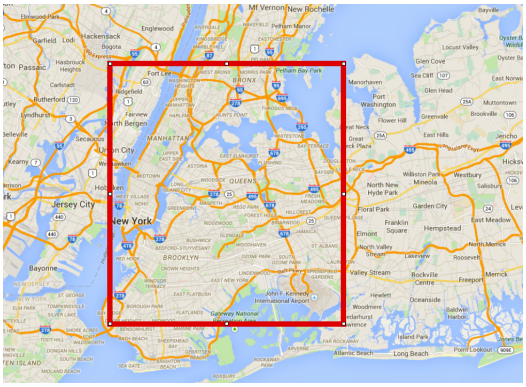

Figure 1: Coverage Area

3 Data

We use a complete data set of 170 million taxi trips taken in the city of New York over 5 months in 2013. For every taxi trip taken, the data set contains a pickup time and location (in latitude and longitude) paired with a drop-off time and location. Our analysis is restricted to taxi trips that both start and end within the boundaries of Manhattan, Brooklyn, or Queens.

4 Network Representation

The traffic congestion network is defined over a particular time interval. In our analysis this is typically a one-hour time interval, although a larger interval (one month) is used initially to gain insight about the general structure of the network.

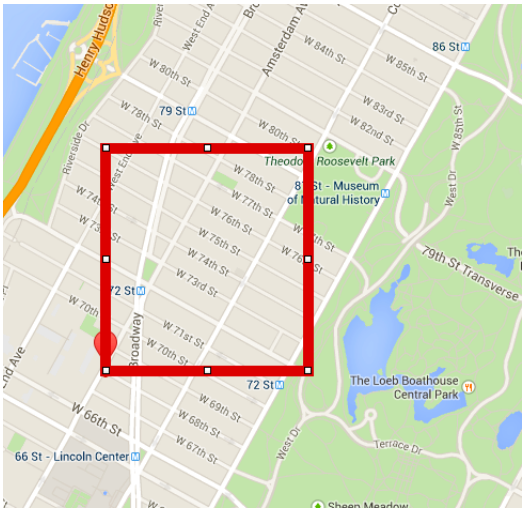

Figure 2: Box Coverage Example. This area represents the area covered by a single node.

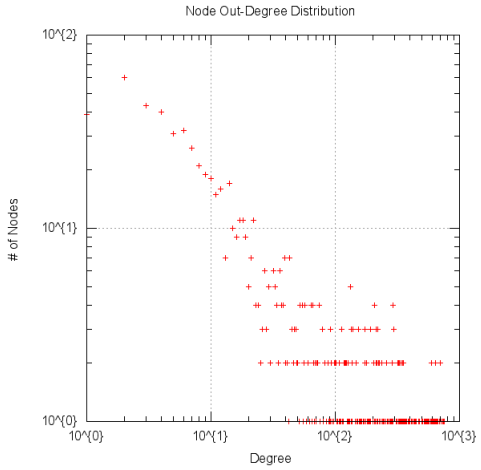

Figure 3: Out-degrees for nodes follow a power-law distribution

4.1 Nodes

In order to create distinct nodes from the continuous latitude, longitude pairs in the data set, we divide New York into a rectangular grid, with each box in the grid of size 0.008 latitude by 0.008 longitude. This means each node represents about 0.36 square miles (an example of a box's coverage area on the upper west side is in Figure 1). The boundaries of this grid are defined by the points (40.626198, -74.015729) and (40.849655, -73.755319), approximately covering Manhattan, Brooklyn, and Queens (see Figure 2). In total, there are 924 nodes in our network with the aforementioned parameters.

This grid clustering technique ensures that each node in the network represents the same geographical area, making results about congestion comparable and meaningful. An additional advantage came in run time, since every new point with a latitude/longitude pair can be assigned a node in constant time, as opposed to dynamic clustering techniques, which require iterating through existing clusters.

4.2 Edges

The weighted, directed edges between nodes are dependent on the discrete time interval during which traffic is being examined. For a particular time interval, every directed edge between node n_1 and node n_2 holds three values that we consider in parallel – the average trip duration between those two nodes, the number of trips that took place between those two nodes, and the average geographic distance between those nodes.

5 Network Structure

To better understand the structure of the New York City taxi network as well as to validate our dataset, we created the taxi network as described in Section 4 for all 14 million rides that took place in January 2013. In the resulting network, the node with the highest in-degree was centered at (40.754198, -73.991729), which is the area around Penn Station. The node with the highest out-degree was centered at (40.642198, -73.783729), which is a terminal at JFK airport. This matches with intuition about where taxi demand would be the greatest.

In addition to these summary statistics about popular nodes, we also found anecdotal patterns in traffic propagation. We explain these patterns in the discussion since they rely on metrics defined in the subsequent section.

Overall, in-degree and out-degree of nodes generally follow a power-law distribution, as seen in the representative plot in Figure 3. This follows expectations, given that some frequently trafficked areas in the city act as hubs. We also include a plot of the average speeds on each edge (Figure 4). These average speeds appear to be normally distributed around a speed of 16 mph. Here, speed is calculated using the precise distance and time of each trip as reported by the taxi and then averaged over all trips on a given edge, giving an accurate sense of average vehicle speed. Because this speed

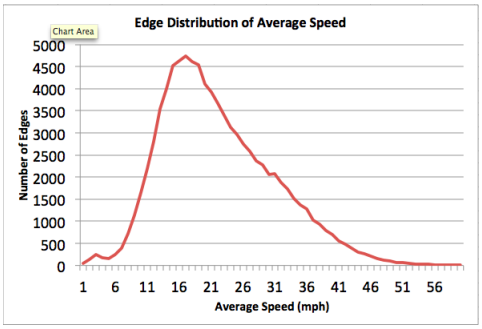

Figure 4: Average Speed on all edges represents

calculation includes time spent at stop signs, etc., this also matches with intuition.

The characteristics of our network validate the quality of our dataset and suggest that it will be feasible to perform our analyses regarding delays (described in the next section).

6 Algorithms and Models

6.1 Obtaining Baseline Values

In order to determine whether an edge is congested during a period of time, we modify the technique from Fleurquin et al., which uses the entire data set to create an average baseline of traffic with which to compare. Rather than averaging over the entire data set to create a single baseline, we create different baselines for each edge for each period of time during which we believe traffic is similar. In designing these baselines, we make several assumptions about the nature of traffic.
The first assumption is that traffic on each day of the workweek is similar, and is divided into three periods – morning rush hour, middle of the day, and evening rush hour. The second is that traffic on each day of the weekend is similar, and is fairly continuous throughout the day. The third is that work nights (Sunday-Thursday) experience similar traffic flow and that non-work nights (Friday, Saturday) experience similar traffic flow. Finally, we assume that traffic from 3am to 7am is similar on all days. Thus, different baseline values are calculated for each edge for the following periods of time:

1. 7 – 10 am Monday-Friday
2. 10 – 4 pm Monday-Friday
3. 4 – 8 pm Monday-Friday
4. 7am – 8pm Saturday-Sunday
5. 8pm – 3am Friday-Saturday
6. 8pm – 3am Sunday-Thursday
7. 3am – 7am every day

We calculate the baseline for each period for each edge as the average trip time along that edge. To calculate these 7 baseline values for each edge, we use all 170 million taxi trips in the data set.

6.2 Metrics for Defining Traffic Congestion

Congestion is defined both for each edge and each node of the network.

6.2.1 Edge Congestion

At a particular one-hour time-slice (e.g. January 25, 2013 from 8:00 am to 9:00 am), we define the congestion on some edge e from node $n1$ to $n2$ as:

$$c(e) = \frac{m - b}{b}$$

where:
m is the average trip time between $n1$ and $n2$ during the time-slice
b is the baseline value for that edge as defined in section 6.1.

6.2.2 Node Congestion

The metric of congestion for a node at a particular time-slice is more complicated, and must take into account the number of incoming and outgoing trips to the node as well as the congestion associated with those trips. Ideally, this value should express the amount of traffic congestion that is concentrated at the node itself - it should be negative if the node is "absorbing" delays, zero if the node is only "propagating" delays, and positive if the node is "creating" delays. Thus, we define the congestion at some node n at time-slice t as:

$$\frac{1}{T_1} \sum_{e_1 \in E_1} t(e_1) * c(e_1) - \frac{1}{T_2} \sum_{e_2 \in E_2} t(e_2) * c(e_2)$$

where:
T_1 is the total number of incoming trips to node n during the time-slice
T_2 is the total number of outgoing trips to node n during the time-slice
E_1 is the set of directed edges into node n during

the time-slice
E_2 is the set of directed edges out of node n during the time-slice
$t(e)$ is the number of taxi trips represented by edge e during the time-slice
$c(e)$ is the congestion on edge e during the time-slice, as defined in Section 6.2.1

6.3 Predictive Model

The aim of the predictive model is to predict the congestion of a node n at time $t+1$ given the entire traffic congestion network at time t.
We obtain our predictive model with a least-squares linear regression, using the following 20 features to predict the congestion of a node n at time $t+1$:

1. The congestion of node n during time t
2. The number of incoming trips to node n during time t
3. The number of outgoing trips from node n during time t
4. The average congestion of the 8 nodes closest to n during time t
5. The average congestion of incoming trips to node n during time t
6. The average congestion of outgoing trips from node n during time t
7. The average baseline value of incoming edges to node n during time t (as defined in Section 6.1)
8. The average baseline value of outgoing edges from node n during time t (as defined in Section 6.1)
9. The average speed (distance/time) of incoming trips to node n during time t (where the distance of each trip is defined as the distance it covers - not the distance between the two associated nodes)
10. The average speed of outgoing nodes from node n during time t
11. - 20. The squares of each of the above 10 features

This model was trained on all of the data from the month of February, 2013 (approximately 13 million taxi trips) in one-hour time intervals to obtain weights for each of the 20 features.
We then tested this model on the data from the month of January. We trained and tested the model leaving out different sets of features, to better understand which features provide more accurate predictions.

6.4 Validation of Predictive Model

To validate the predictive model, we calculate and analyze the coefficient of determination, denoted R^2. This value is calculated as follows

$$R^2 = 1 - \frac{\sum_i (y_i - f_i)^2}{\sum_i (y_i - \bar{y})^2}$$

where:
the observed data set has values $y_1, ..., y_n$ associated with predicted values $f_1, ..., f_n$
$\bar{y}$ is the mean of the observed data set

R^2 values fall between 0 and 1, where 0 indicates that predictions from the model are equivalently good as always predicting the mean, and 1 indicates that the model is a perfect fit. A negative R^2 implies that the predictor is performing worse than if it always predicted the mean.

7 Results

Figure 5 illustrates the R^2 values obtained by the model trained on all of the data from February and tested both on the data from January and from February (to determine whether overfitting of the model takes place). The model was trained on different subsets of the 20 aforementioned features to determine which features are more predictive of node congestion.
While there is some variation in the R^2 values, all of the values are very close to 0, suggesting that the predictive model performed about as well as it would had it always predicted the mean of the dataset. Analysis of these results follows in the Discussion and Conclusions section.

8 Discussion and Conclusions

In general, these results imply that our model trained on February data is unfortunately not predictive for January data. We can rule out over-fitting as the reason for the failure of the model, as it is nearly equally bad at predicting data from the month of February (the same month that it was trained on).
Although there were 13 million recorded taxi trips

Features	R Squared on Jan test data	R Squared on Feb test data
1-10	-0.01	0.023
1-20	-0.011	0.04
Only 1	0.002	0.003
1 and 11	0.001	0.005
1 and 4	0.002	0.005
1, 4, 10, & 14	0.001	0.005

Figure 5: R^2 values for different features trained on February data and tested on January and February data.

for each month, the data for a single $0.36mi^2$ node at a one-hour time interval was often scarce. We increased both the size of the node and the size of the time-interval from our original plan ($0.09mi^2$ nodes and 10-minute time intervals), but felt that increasing these sizes even further would reduce the explanatory power of our results.

Another issue was that our taxi dataset was limited to only a dropoff and pickup location for each trip, so we had no information about the path of each taxi. Specifically, we could not factor the taxi's path into the congestion of each node it traveled through, information which we believe would significantly improve results. Additionally, rather than assigning features based on intuition, we may have had better results using unsupervised machine learning to determine which features would be helpful to train on. The models that performed the best were the ones that did not use the majority of the 20 features - in fact, they only used the node congestion at time t alone or combined with the congestion of surrounding nodes at time t. While this suggests that these features may be the most predictive of node congestion at time $t+1$, the still very low R^2 value 0.002 makes it difficult to make more concrete conclusions about feature quality.

Although our predictive model did not yield promising results, we believe that both our method of structuring the taxi trips into networks, and our metrics for edge and node congestion, are original and valuable contributions to the study of traffic networks. Anecdotal evidence from the month of January supports this claim. For example, for the entire month of January, the node and hour with the highest node congestion was La Guardia Airport on the night of January 20, 2013 - during Martin Luther King weekend, a time when many people are traveling. Interestingly, the second and third highest congestion values were also at airports during the evening. It seems, then, that our definition of congestion points to airports as the top source of congestion.

9 Contributions

We contributed equally to the algorithms, implementation, and write-up of this project. Jess focused more on organizing the data set and implementing the predictive model, while Sanjana focused more on developing and implementing the metrics and writing the results.

10 References

Bolaños, M. E., and Murphy, D. How Much Delay does New York Inject into the National Airspace System? A Graph Theory Analysis.

Ducruet, C., and Lugo, I. Structure and dynamics of transportation networks: Models, ods and applications. Rodrigue, J.P., Notteboom, T.E., and Shaw, J. The SAGE Handbook of Transport Studies, SAGE, pp.347-364, 2013.

Fleurquin, P., Ramasco, J. J., and Eguiluz, V. M. (2013). Systemic delay propagation in the US airport network. Scientific reports, 3.

Leskovec, J., McGlohon, M., Faloutsos, C., Glance, N. S., and Hurst, M. (2007, April). Patterns of Cascading behavior in large blog graphs. In SDM (Vol. 7, pp. 551-556).

Traffic Modeling - Phantom Traffic Jams and Traveling Jamitons. (n.d.). Retrieved December 10, 2014, from http://math.mit.edu/projects/traffic/

Zhang, B. (2014, March 25). The 10 Most Traffic-Clogged Cities In The US. Retrieved December 10, 2014, from http://www.businessinsider.com/us-cities-with-worst-traffic-2014-3

New York City Taxi Data obtained from http://www.andresmh.com/nyctaxitrips/