Abstract and Status Report

Jiafeng Chen
jiafengchen@college.harvard.edu

Yufeng Ling
yufengling@college.harvard.edu

Francisco Rivera
frivera@college.harvard.edu

December 3, 2017

1 Draft abstract

Using the New York City Taxi and Limousine Commission (TLC) database of taxi rides, we develop a probabilistic model of transportation in New York City. By discretizing the Manhattan geography into a grid structure and by using a mixture model with the latent variable being the route chosen by the driver, we devise a model with a parsimonious number of parameters that is easy to implement even for a large city, fast to estimate, and yields reasonable prediction results relative to benchmarks of linear regression and neural network. Moreover, the parameters of the model can be directly interpreted as measures of congestion on Manhattan's road network.

2 Updates to proposal

2.1 Model

The intuition of our main approach remains the same: break up a trip into its constituent "sub-trips" and learn the latent variables that correspond to these sub-components. However, in the initial proposal, we aimed to make the level of discretization to be the traversal of an actual street. Using the NYC Street Centerline (CSCL) data from NYC Open Data, we (laboriously) constructed the Manhattan street network, which contained 10296 streets (counting two-way streets as two streets since they will have different parameters in the model; See Figure 1). Given the complexity of the data—both in the number of parameters we need to fit, as well running graph algorithms on a large network—we have since refined our approach. We now aim to discretize Manhattan into a grid and have latent variables for the congestion within each of these discrete units. (See Section 4)

2.2 Evaluation

In our initial proposal, we suggested an ℓ^2 norm for penalizing predictions would be natural. Upon further inspection of the data, however, we have realized the distribution of travel times has a very heavy right-tail (some trips apparently taking days). Since these observations are either a product of un-clean data or values which we don't particularly care to predict, we opt for reporting percentiles of the ℓ^2 norm.

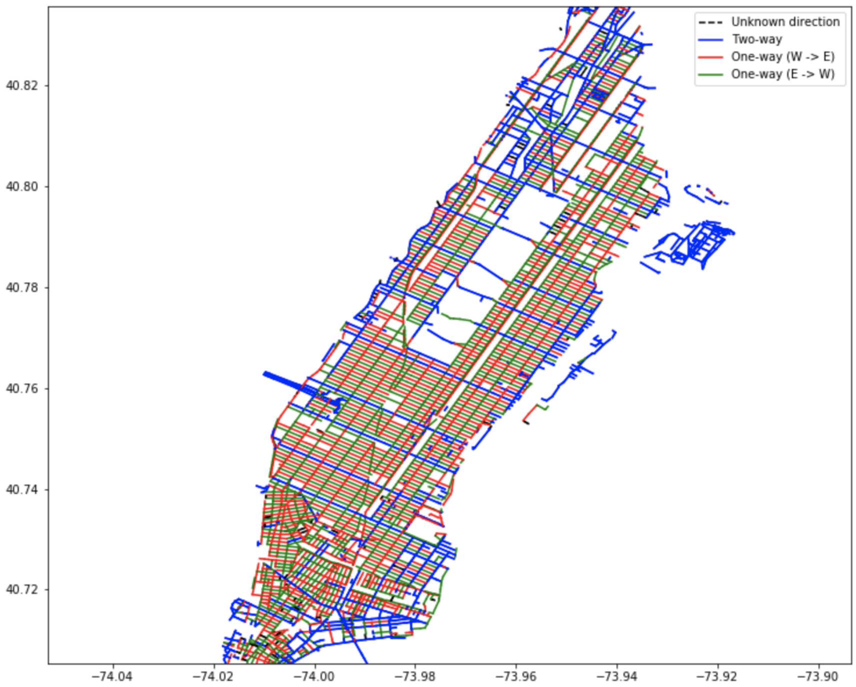

Figure 1: Complexity of the Manhattan traffic structure as a graph

3 Baseline results

We have two models as baseline. We use a ridge regression and a neural network on the following set of features: trip distance, day of the week, and hour of the day. For the ridge regression, we observed the estimated coefficients of Wednesday to Friday to be the highest among days of the week, and coefficients of 2am-6am to be significantly lower than other hours of the day.

Test-set error descriptive statistics (minutes)	Ridge regression	Neural network
Mean	3.684	3.544
Standard deviation	22.177	21.786
99% percentile	15.268	15.059

Although the neural network model has marginally better performance than the ridge regression and can arguably improve with further parameter tuning, the coefficients of the ridge regression is much more interpretable and generates useful insights. Our approach in Section 4 shares this feature with the linear regression in generating interpretable coefficients of interest.

4 Preliminary results

We implemented a model based on discretization. Since the discretization of Manhattan into literal streets is too cumbersome, we opt to discretize Manhattan into a $P \times Q$ grid (We believe that 40×140 performs well). To put in perspective, each grid point corresponds to roughly a 200ft × 200ft area. We convert the coordinates of start and end locations of each trip to the nearest grid point. Let each grid point be associated with a weight w_i whose collection is $\boldsymbol{W}$. These weights represent expected trip times "passing through" a grid point.[1]

Let $G = \{1, \dots, N\}$ be the set of grid points. Let $\mathcal{Z}_{ij} \subset 2^G$ be the set of paths from i to j. For each trip k starting from i and ending at j, let T_k be the time the trip takes. Let $z_k \in \mathcal{Z}_{ij}$ be a latent variable of the path taken by the trip. We have a mixture model

$$\begin{aligned} z_k | \boldsymbol{W} &\sim \mathcal{L}_{\mathcal{Z}_{ij}} \\ T_k | z_k &\sim \mathcal{N}\left(\sum_{i \in z_k} w_i, \sigma^2\right), \end{aligned}$$

where $\mathcal{L}$ is some distribution over $\mathcal{Z}_{ij}$ and $\boldsymbol{W} = (w_i)_i$ are the weights associated with node i—which should be an estimate of time it takes to traverse the grid node.

We assume $\mathcal{L}$ is the following. The driver would only traverse in the direction of her destination. Given hyperparameter $q \in [0, 1]$, the driver traverses the shortest-time route ($\arg\min_z \sum_z w_i$) with probability q and randomly picks a route in the direction of her destination with probability $1 - q$.

We took an approach that is similar to expectation-maximization and to stochastic variational inference. For each observation, we sample from $\mathcal{L}_{\mathcal{Z}_{ij}}$ conditional on the *current* estimates of $\boldsymbol{W}$. We then update the parameters $\boldsymbol{W}$ via the gradients of the squared error loss between the predicted trip time and the actual trip time (thus corresponding to a Normal model with known variance). This approach is similar to **[?]**, who also use a mixture model, while using a discrete choice model for modeling $\mathcal{L}_{\mathcal{Z}_{ij}}$. However, the model in **[?]** runs on a much smaller scale facing larger computational constraints, whereas we are able to estimate on a larger scale by using a sampling approach for $\mathcal{L}_{\mathcal{Z}_{ij}}$.

There are a few interesting results to note:

1. The model performs well even with a small dimension of points. In Figure 2, we use a model with only 350 parameters (about 1/5 of the neutral network), we are able to achieve test-set error rates that are on par with the neural net. Moreover, we have not stratified the data by time and week, which are important predictors of trip time in the model.

2. The visualization of the model is telling. In all three dimensions of the model that we tested (Figure 3 and Figure 4), the visualization of the model parameters are able to identify similar areas of high and low congestion. This is robust to dimension, initialization, and values of q (Figure 5).

3. The model is a bit sensitive to extreme values (which we suspect to be data errors). The extreme values tend to be unusually long trips, and they tend to update the parameters in extreme directions (the red lines in Figures 4 and 5). Though these errors fade out over time (see attached 100-frame video of visualizations with one frame for 2000 trips[2]), they fade out especially slowly since trips that take long are unlikely to be selected if q is high, and thus it's hard for high parameter values to be corrected on later conditioning.

4. We also observe that q acts as a smoothing parameter. With $q = 1$ in Figure 5, the parameters rigid, forming clear lines that probably represents the disproportionate effects of only a few trips.

[1] If we discretize Manhattan into actual streets, then each street segment would have a weight that indicates how long it takes for a driver to traverse this street segment. Here, the weight of each grid point gives a general sense of how congested a small patch of land is.

[2] The images are snapshots of parameter values in the updating process

4.1 Future directions

Here are a few areas we are looking to explore:

- We are looking to explore the theoretical properties of the model and estimation approach by formalizing them in terms of EM and stochastic variational inference.
- We will split the training sample by some observable characteristics such as day of the week and time of day, and train on each subsample. This approach should improve prediction accuracy, and provide more insight in the evolution of congestion in the temporal dimension.
- We will consider remedies of extreme values. Currently, since we assume drivers behave intelligently, weights that are overly large (usually due to unusually long trips because of unclean data) tend to very slowly be corrected, since intelligent drivers avoid such routes and prevents the routes from being observed and updated. Graphically, this results in abrupt red lines across our Manhattan map and only updated gradually.
- We will attempt to address the intractability of parameter optimization on the exact street network by dividing Manhattan into neighborhoods and optimize the street weights by training on a subsample of the dataset restricted to each neighborhood. With the weight of each streets calculated, we then make prediction on the complete test set.

A Attached figures

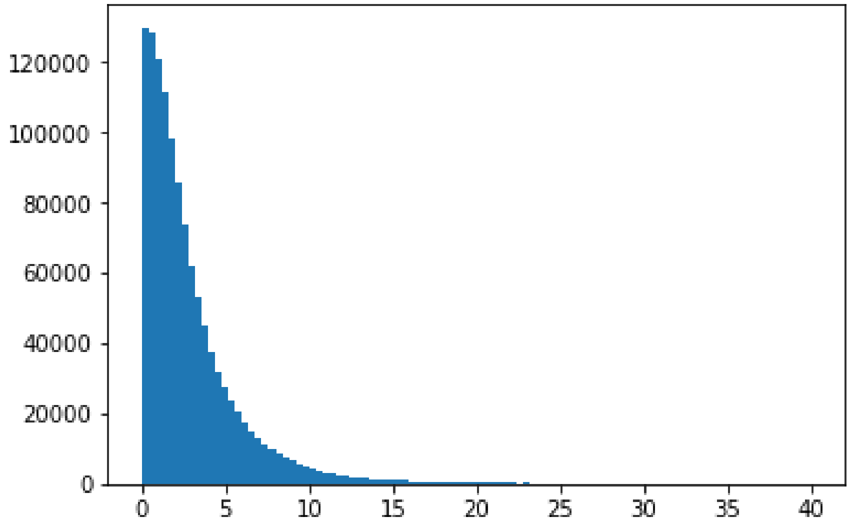

Figure 2: The distribution of error obtained on the test set by using a low-dimensional model with $P = 10, Q = 35$ and $q = 1$. The mean loss is 3.28 minutes and the 99% is 15.49 minutes in error.

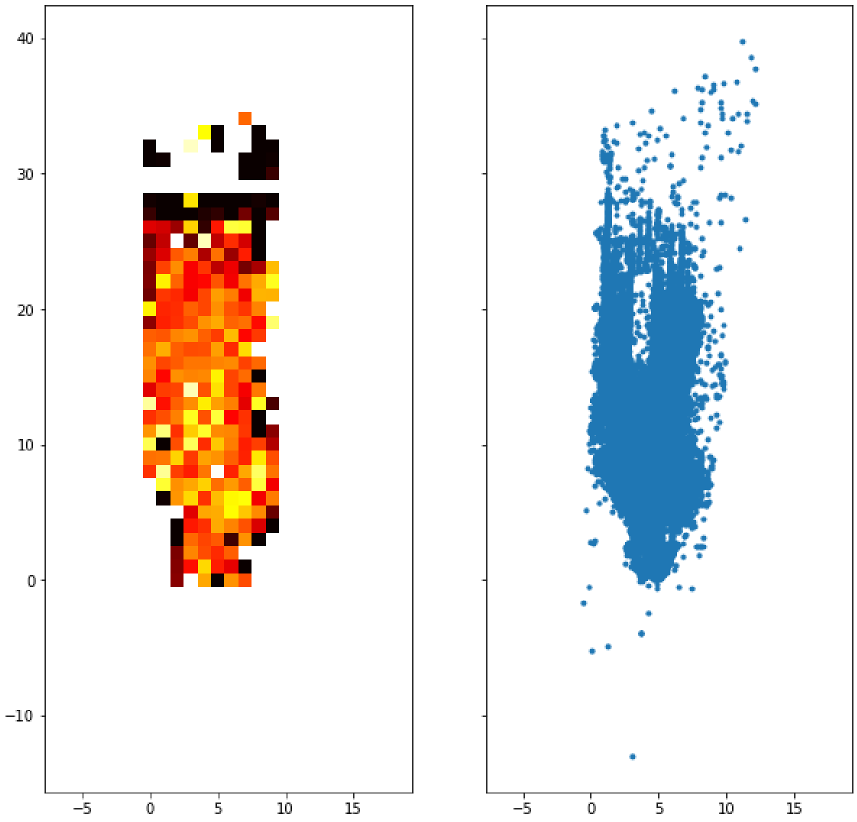

Figure 3: We plot via `plt.imshow(...,cmap='hot')` the distribution of parameter values since they have physical interpretations on a heat map. The right panel is the distribution of trip starting locations which trace out the shape of Manhattan. This is on a low dimensional model with $P = 10, Q = 35$. Here, brighter colors (yellow) means higher parameter value, which points to more congestion. We also initialize grid points that are above non-terrestrial areas (e.g. Hudson river) to have high values and omit them in the plotting. We later realized that the initialization does not matter, and the model is able to pick up the shape of Manhattan automatically.

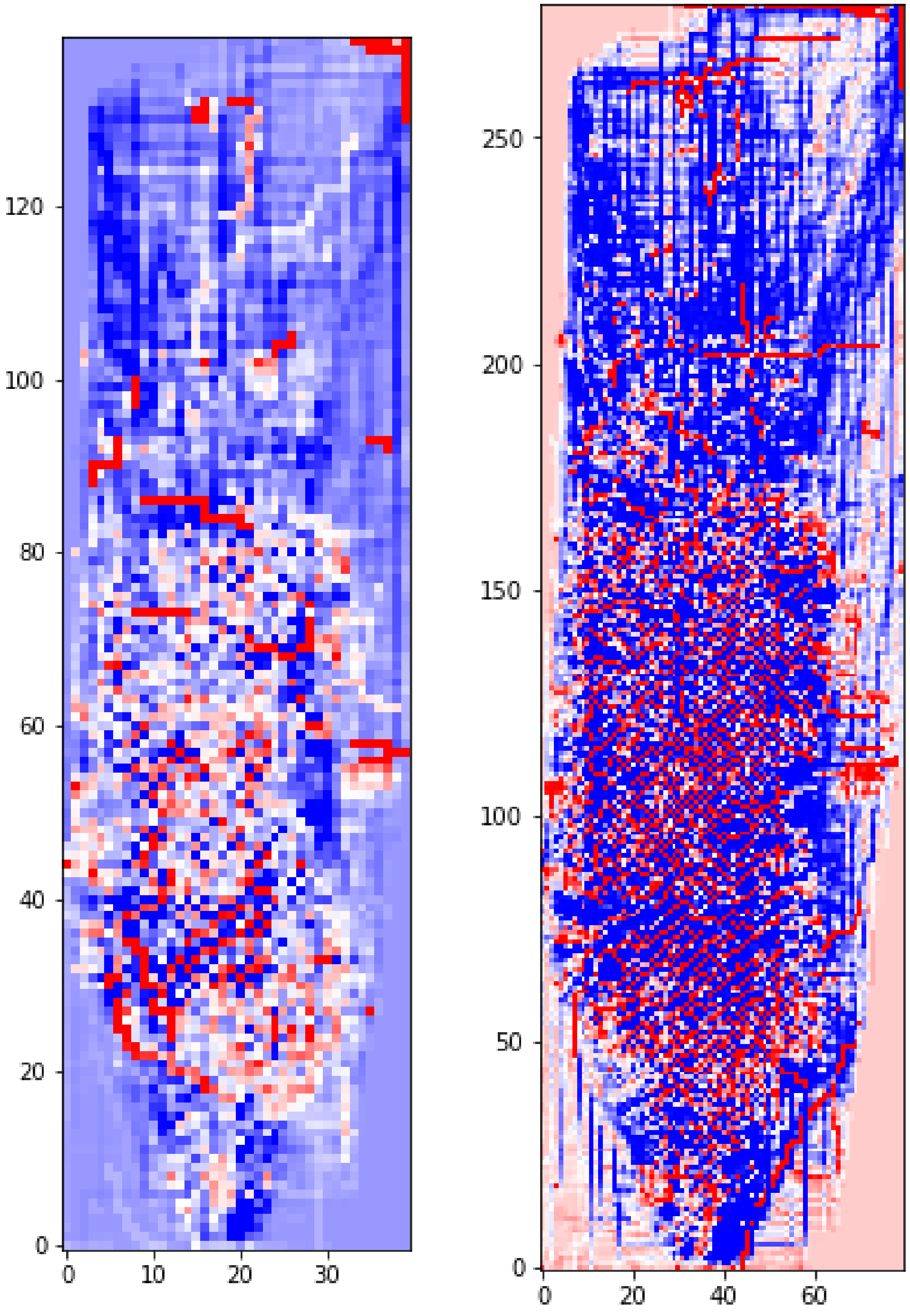

Figure 4: Left: an iteration of the model with 40 × 140 parameters. Right: an iteration of the model with 80 × 280 parameters. In both, $q = 0.5$.

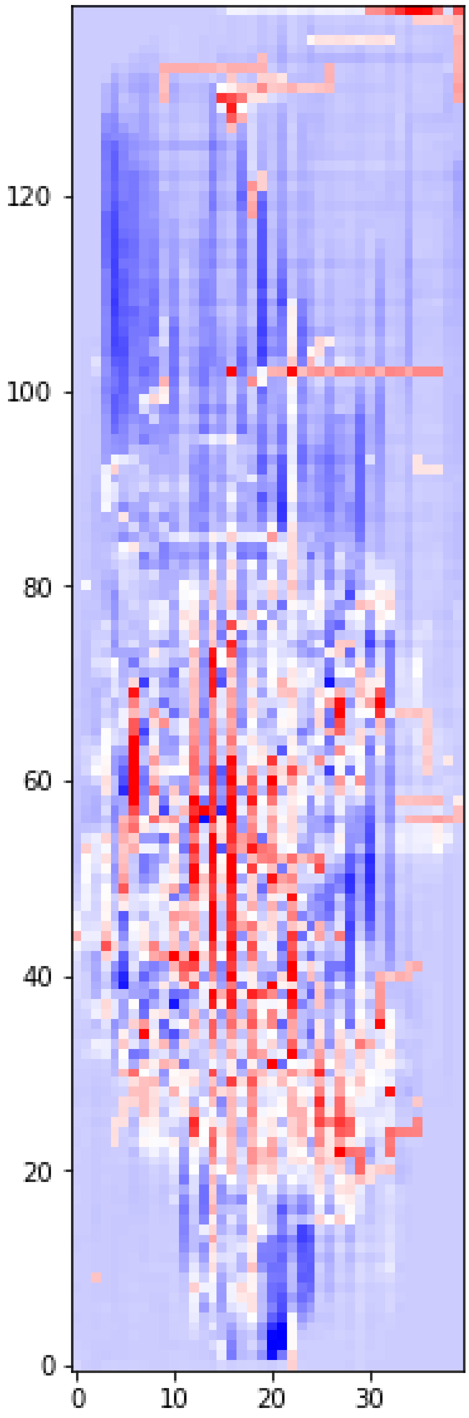

Figure 5: An iteration of the model with $q = 1$. We see that without smoothing, the parameters doesn't really look right.