Towards Urban Vehicle Autonomy: Estimating Urban Congestion from Taxi Pickups and Deliveries

Federico Rossi Sumeet Singh Rick Zhang

***Abstract*— While there are several advantages that can be attributed to the use of autonomous vehicles for personal mobility, their overall impact on surrounding traffic and levels of congestion is as of yet, poorly understood. A crucial stepping stone towards a better understanding of the relation between large fleets of autonomous vehicles and urban congestion is the creation of a predictive model of urban congestion. In this paper, we present a geospatial, time-variant predictive model for urban traffic congestion within Manhattan using a machine learning approach. The model relies on taxi data collected in New York for the month of February 2012 and encompasses features such as number of taxis on the road, time of day, weather, and day of the week and predicts the average speed in a given region. Using a multi-class support vector machine with a Gaussian kernel resulted in an average RMS error under 2 mph with a classification accuracy between 85%-90% for central Manhattan. The relatively high prediction accuracy suggests a strong correlation between the taxi data and the macroscopic traffic pattern and congestion in most of Manhattan. This model constitutes a first step towards analyzing congestion resulting from a fleet of autonomous vehicles.**

I. Introduction

Autonomous vehicles have been claimed to offer many potential benefits to urban personal mobility, including lower cost to the user, better vehicle utilization, more efficient urban land use, and increased safety. While many of these claims are logically sound, it is far less clear what the effects of vehicle autonomy will be on traffic congestion. One of the most promising uses of autonomous vehicles is one-way vehicle sharing, or equivalently, autonomous taxi service. With this in mind, some have argued that autonomous vehicles will in fact increase urban traffic congestion due to many empty vehicle trips, when the vehicle travels from a passenger destination to the next pickup location. On the other hand, autonomous vehicles, when routed intelligently, may help alleviate traffic congestion and decrease overall travel times. Since the number of autonomous taxis in the city may represent a significant portion of all urban traffic, a predictive model of urban congestion must first be created that takes into account not only how vehicles respond to traffic, but how a fleet of vehicles generate traffic. In this report, we focus on the creation of such a congestion model using existing taxi data from New York City (specifically Manhattan) as a case study.

Congestion data is often directly gathered from probe vehicles, induction loops embedded in asphalt or surveillance cameras. While these approaches offer data with high spatial and temporal density, they present significant drawbacks: deployment of dedicated infrastructure or repurposing of existing facilities typically has a very high cost, while probes offer no information about traffic density and do not scale well because of privacy concerns. Furthermore, these forms of data do not provide information on how large taxi fleets contribute to congestion. On the other hand, large cities routinely collect data from their taxi fleets including pick-up and delivery locations, trip distances, and travel times. The taxi traffic in a large city such as New York arguably presents a strong correlation with the overall traffic flow and reflects macroscopic mobility patterns. However, since the route taken by each taxi is usually not recorded, congestion information can only be indirectly estimated. To build our congestion model, we used a database of pick-up and delivery locations and times of all NYC taxis for the month of February 2012, obtained through the Freedom of Information Act. The challenge intrinsic in estimating congestion from pick-up and delivery pairs is the reconstruction of routes from their endpoints. In an urban center, each taxi may choose to take one of many routes with the same Manhattan distance from start to end points. Since exact routes taken cannot be determined from the data, we employ a simple and computationally efficient model that discretizes the city into a grid and estimates local average velocities in each part of the city. We then correlate the local velocity estimates with features such as local taxicab density, time and day of the week, and weather conditions.

The rest of this report proceeds as follows: section II briefly outlines the existing literature in congestion modeling, section III describes the modelling approaches we have taken, section IV discusses the performance of our model and section V summarizes our results and outlines next steps to be taken.

II. Related Work

The problem of estimating traffic congestion on highways has seen significant interest in the scientific literature since the 1960s. In particular the Bureau of Public Roads (BPR) model, which correlates traffic flow and road congestion as measured by the time required to cross a given road segment [3], has been used in scientific studies and highway planning with remarkably accurate results. The BPR model estimates

Federico Rossi, Sumeet Singh and Rick Zhang are with the Autonomous Systems Laboratory, Department of Aeronautics and Astronautics, Stanford University, Stanford, CA, 94305, {frossi2, ssingh19, rickz}@stanford.edu

the time S_a required to cross a given road segment as

$$S_a\left(v_a\right) = t_a\left(1 + 0.15\left(\frac{v_a}{c_a}\right)^4\right) \tag{1}$$

where t_a is the free-flow transit time (defined as the transit time in absence of any vehicular traffic), c_a is the route capacity (measured in terms of traffic flow) and v_a is the actual traffic flow on the route considered. Experimental observations show that traffic congestion often manifests itself in three phases:

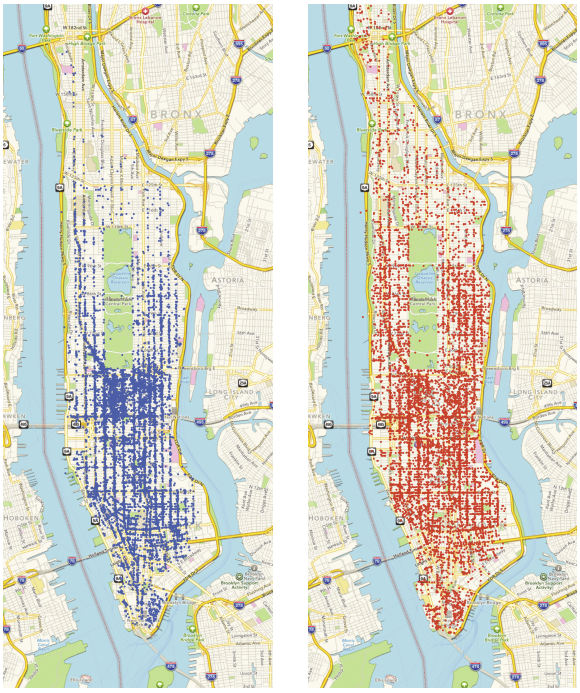
- Free flow, in which vehicles do not interact with one another and move at the free flow velocity (loosely correlated with the speed limit of the road);
- Synchronized flow, a transition region between free flow and traffic jam;
- Traffic jam, characterized by a downstream front moving at low, constant speed;

We refer the reader to [7] for an excellent review of the state of the art in *highway* traffic congestion modeling. *Urban* traffic congestion, on the other hand, is less understood and very much a subject of active research. Sheffi's book [9] offers a comprehensive, if dated, review of modeling approaches to urban traffic: a graph abstraction with variable link weights is typically employed, reducing the problem to congestion estimation and optimal routing on graphs. It is unclear whether highway congestion models are applicable to urban transportation: Sheffi suggests existence of a connection between traffic flow and congestion (as measured by the average speed) but does not propose any analytical shape for this function. Recent works represent urban congestion through Markov models [5], [8]. These models offer very good estimates of overall traffic from limited data points, but they lack *predictive* power, making them unsuitable for estimating the impact of intelligent fleet routing on overall traffic congestion. Most existing works base their estimates on high-fidelity data collected from GPS-equipped probe vehicles, induction loops embedded into the ground and traffic cameras [6]. Data collected through these means has high spatial and temporal resolution; on the other hand, scalability and cost of deployment are major issues barring widespread adoption. To the best of our knowledge, our work is the first to estimate traffic congestion based on low-resolution taxi pickup and delivery points.

III. Methodology

A. Local Velocity Estimation

Prior to investigating possible machine learning algorithms to predict local traffic congestion, our first step is to extract the local average velocity and the local taxi density. Average velocity is the primary congestion indicator, while the number of taxis in each part of the city is an important feature when estimating congestion since it is related to the overall vehicle density. The raw NYC taxi data is sorted chronologically and pruned to isolate trips that are only within Manhattan. Figure 1 shows the pickup and delivery locations within the peninsula between midnight and 1 a.m. on February 1st. Based on the raw data, we estimate that around 84% of the 15 million taxi trips in our database are solely within Manhattan. Having isolated trip data to within



(a) Pickups (b) Deliveries

Fig. 1: Pickup and delivery locations in Manhattan for a 1 hour interval

Manhattan only, we subdivide the area into a rectangular grid with 500m by 600m cells and approximate vehicle trajectories using straight line paths. As the given problem is inherently time-varying, the data is further segmented into hour-long blocks so that local congestion information for a given grid-element can be extracted. The task is then to find the number of taxis that passed through a given grid element within each hour and the subsequent average velocity within that area. The average velocity for a grid element is found by calculating a weighted average of the velocities of all taxis that passed through it within a specified time-span. The velocities of all taxis are then approximated as being constant along their routes and found by dividing the total trip distance by the total trip time. The weight assigned to each trip equals the length of the section of the trip within the grid element divided by the net Euclidean distance. Thus:

$$\bar{v}_{ij} = \frac{\sum_k w_{ij}^{(k)} \bar{v}^{(k)}}{\sum_k w_{ij}^{(k)}}$$

where $w_{ij}^{(k)} = l_{ij}^{(k)}/l^{(k)}$ is the assigned weight for the k^{th} trip in the $(i,j)^{th}$ grid element, $\bar{v}_{ij}$ is the average velocity within grid element (i,j), $l_{ij}^{(k)}$ is the length of trip k within the $(i,j)^{th}$ grid element, $l^{(k)}$ is the total Euclidean trip length and $\bar{v}^{(k)}$ is the average speed of the k^{th} trip. The

sum is over all trips through the given grid element. By this method, a higher confidence is assigned to trips that spent a greater proportion of their trip within the given grid element. A modified version of the *Fast Voxel Traversal Algorithm for Line Tracing* [2] is used in performing the calculation. A sample output from one such data-run is shown in Fig. 2. As can be seen, there is a distinct correlation between

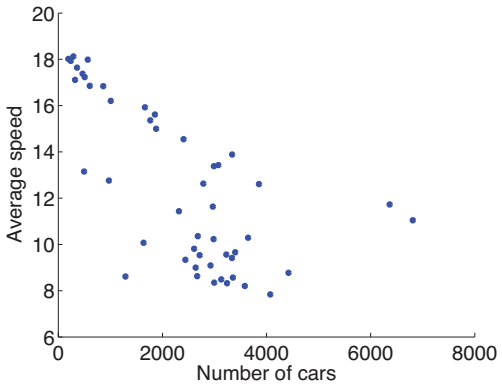


Fig. 2: Sample output for a cell in Midtown Manhattan

the number of taxis within a given grid element and the resulting average velocity within that area. Having obtained our congestion indicator values, we are able to proceed with identifying trends within the data and applying machine learning techniques to predict congestion.

B. Regression

Our baseline approach is to fit the state-of-the-art BPR congestion model to traffic data for a select number of cells. We do not account for features such as time of the day or day of the week, consistently with BPR's hypotheses that congestion is uniquely determined by the number of cars and the road capacity (here, considered constant within a single cell). Results for cell [5,13], located in Midtown Manhattan, are shown in fig. 3: they are rather underwhelming, with a RMS error of 3.13 mph. In particular, BPR grossly overestimates congestion for high traffic volumes.

In highway models, average speed drops to zero in a traffic jam: Manhattan data, on the other hand, shows the jam region moving at a constant, positive speed. We therefore modify the analytical BPR equation, adding an offset term to account for this effect: the fit, also shown in figure 3, is better but still far from optimal, with an RMS error of 2.33 mph after training on the whole dataset. Furthermore, the algorithm has clear difficulties predicting the onset of the synchronous flow phase, placing it around 1500 cars per hour earlier than it should.

The most probable cause of failure is the fact that, while traffic may uniquely depend on the number of *cars* on the road, the fraction of taxis is not constant throughout the day. Informally, taxis make up a significantly higher fraction of traffic at night and outside rush hour. However, controlling for the time of the day is not possible: the number of taxis

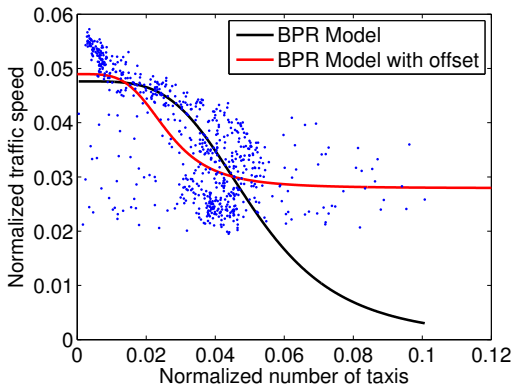


Fig. 3: Fitting speed vs. number of cars to the BPR model

is strongly related to the time of the day, resulting in very poor data on a per-hour basis. This will be discussed in more detail in section III-C. More powerful methods are therefore required to explore the complex dynamics of urban congestion and the correlation between average speed and number of cars, weather, time of the day and date.

C. Classification

a) Feature Selection: Underwhelmed by the results of the regression, we approach the problem from a different angle: rather than relying on continuous classifiers, we discretize the number of vehicles and the average velocity, then try to estimate the *discretized* average velocity within each cell as a function of the number of taxis, time of the day, weekday/weekend (boolean), and weather. Figure 4 shows the distribution of the average speeds as well as number of taxis in a cell in midtown Manhattan as it changes throughout the day. We can also clearly see the drastic difference between weekdays and weekends (note that the single "outlier" which seems to exhibit a weekend distribution on a weekday was President's day on February 20th). As discussed in section III-B, the actual number of vehicles on the road (vehicle density) should be a good predictor of congestion. Theoretically, this may be calculated by dividing the number of taxis by the fraction of total vehicles that are taxis. However, this fraction is unknown and does not remain constant throughout the day (and is not the same in different parts of the city). By using the time of the day as a separate feature, the classifier may be able to extract more information about the fraction of total vehicles that are taxis.

Weather conditions, obtained from Weather Underground [1], are used because rain or snow may cause slower speeds on the road. Weather, considered uniform across NYC, is classified as clear, raining or snowing and updated hourly. Average speed is discretized in three bins, roughly corresponding to free flow, synchronized flow and traffic jam; the number of cars is discretized in 50 bins. The features are assembled into a feature matrix and fed into classification algorithms to estimate the local average velocities.

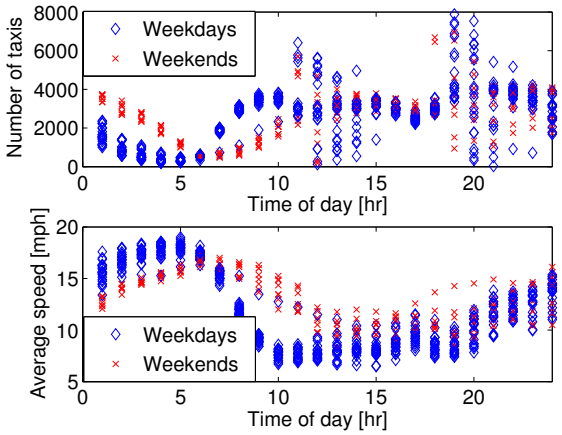


Fig. 4: Number of taxis and average speeds throughout the day, for both weekdays and weekends

b) Algorithm Selection: Several classification algorithms were evaluated on a small number of cells. Sample cells were selected based on the quality of their data: specifically, we sought cells whose features and labels represented a good cross-section of the overall dataset, with high variance both in the number of vehicles and in the range of average speeds. Algorithms are benchmarked and tuned on these cells; their performance is then validated on the whole dataset.

Naïve Bayes was first trained on the number of cars alone to assess baseline performance. Classification accuracy for sample cell [5,13] is between 60% and 65%, far from acceptable. The corresponding RMS error (measured as the difference between the actual average speed and the center of the bin where the sample was classified) averages 3.7 mph. Training Naïve Bayes with a richer feature set, including weather conditions and whether the day in question is a weekday or a weekend, leads to test accuracy consistently between 70% and 75% and RMS error averaging 3.1 mph. In both cases, 70/30 cross-validation is employed. Time of the day is not included in features because of its strong correlation with the number of cars, and since Naïve Bayes assumes all features to be conditionally independent.

Inspection of the training curves, shown in fig. 5, suggests that Naïve Bayes operates in an high bias regime: we therefore switch to an SVM classifier, which explores a wider solution space.

Linear SVM performs exceedingly poorly: its failure is most likely caused by the fact that features are not linearly separable in any meaningful way (even with L1 relaxation) in their original space.

Nonlinear SVM, on the other hand, achieves accuracy significantly higher than Naïve Bayes: direct experimentation shows that a Gaussian kernel with a "One-against-one" multiclass SVM model [4] consistently offers the best testing accuracy. Classification accuracy for sample cell [5,13] is consistently between 85% and 90% (with 70/30 cross-validation) and the RMS error averages 1.3 mph, a nearly twofold improvement over BPR regression. The widening of the gap between the training and test learning curves, shown in fig. 5, suggests that the algorithm may be entering a high variance regime, which warns us against considering new features. In particular, adding spatial information to each cell's features and performing a global estimate of traffic flows results in disastrous performance: while data within each cell is predictable, correlations between neighboring cells (which can have different traffic capacities) are not understood by the SVM algorithm, which is unable to interpret geospatial information and performs only slightly better than random in this case.

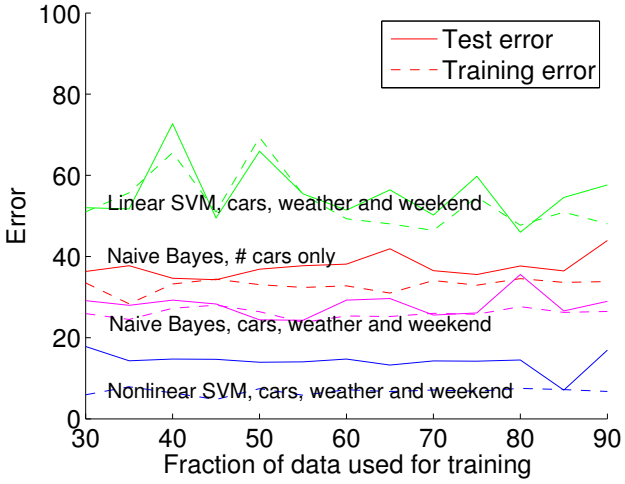


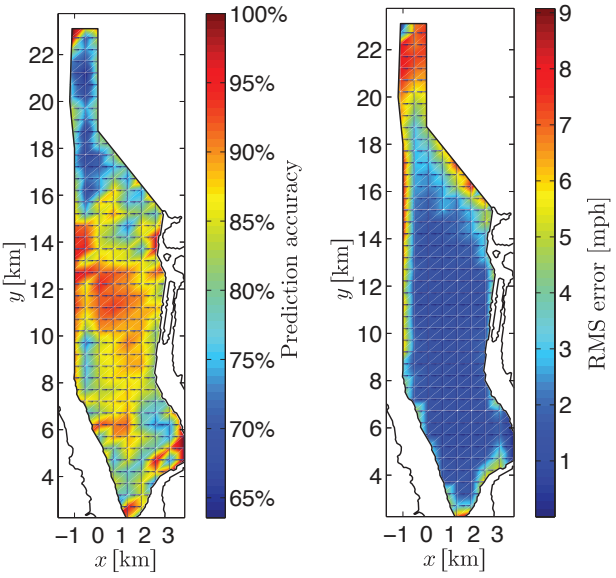
Fig. 5: Learning curves for Naïve Bayes, linear and nonlinear SVM for sample cell [5,13]

IV. Results

The nonlinear SVM algorithm with Gaussian kernel and pairwise comparisons is applied to all the cells in Manhattan to estimate local average speeds. The performance of the classifier is evaluated using 70/30 simple cross validation (70% training, 30% testing), and the testing accuracy is shown in Figure 6a. For most regions of Manhattan, the classifier identifies the correct phase of traffic flow with 85% to 90% accuracy. The accuracy drops down to around 60% in the northern parts of Manhattan (north of Central Park) due to the lack of sufficient data. From Figure 1, it is clear that relatively few taxi trips begin or end north of Central Park. In these regions, the number of taxis become a poor predictor of overall vehicle density.

Figure 6b shows the RMS error between the vehicle speeds and the predicted average speed (average of each bin). For most of the city, the estimated average speed is within 2 mph of the actual speed. Errors are higher in the northern parts of the city due to the lack of data, as discussed. Together, these figures show that our model accurately captures the traffic levels of Manhattan around and south of Central Park.

Figure 7 shows average speed data plotted against the number of taxis as well as predicted average speeds (color



(a) Classification accuracy (b) RMS error

Fig. 6: Classification results for the whole Manhattan grid. Nonlinear SVM classifies the discrete congestion indicator, related to congestion phase, with 85% to 90% accuracy around and south of Central Park. The resulting estimated traffic speed is consistently within 2 mph of the actual one.

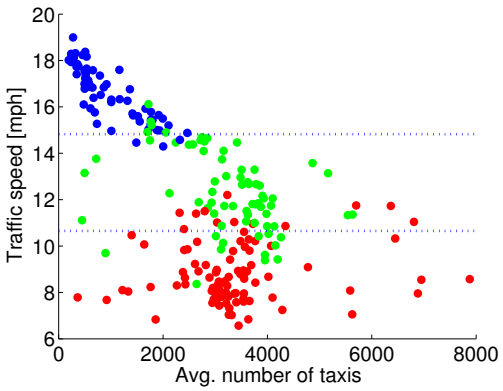


Fig. 7: Performance of the nonlinear SVM classifier for one cell. The horizontal dashed lines mark the boundaries of each speed bin. The colors represent the predictions made by the classifier. A correct prediction is made if the data point lies within its predicted bin.

coded into 3 bins) for cell [5,13]. The three speed bins roughly correspond to the three phases of traffic flow, and the classifier is able to correctly classify most data points (even outliers with very low speeds and low number of taxis).

V. Conclusions and Future Work

In this report, we detail the derivation of a predictive model of urban traffic based on data from pickup and delivery locations of taxi trips. After showing that the standard BPR congestion model is not suitable for estimating urban congestion, we propose a classification-based approach: our model locally estimates a discrete congestion indicator (related to the three phases of traffic flow), then reconstructs the average traffic speed. Our model is tuned and its behavior is evaluated on a small number of cells located in Central Manhattan: in this document, the procedure is outlined on a single cell for simplicity. The model is then trained on the whole grid. Results are encouraging, with very accurate prediction of the discrete congestion estimator and precise reconstruction of the actual traffic speed: we consistently obtain 85%-90% classification accuracy and the estimated traffic speed is generally within 2 mph of the actual speed in most areas south of 110th Street.

One of the key limitations of our model is its inability to capture directionality of congestion: average speed strongly depends on the direction of travel in both urban and non-urban settings. Future work will adopt a graph-based model rather than our current grid-based paradigm, allowing us to better capture the traffic patterns of road networks and direction of travel. It will also allow us to better estimate the number of cars on each segment of the road network by using routing heuristics that mimic driver behavior to reconstruct routes in a realistic yet computationally efficient way.

This predictive model allows us to perform high fidelity simulations of trips within Manhattan that take into account urban traffic congestion. As new personal mobility solutions become practical with the help of vehicle autonomy (for example, autonomous Mobility-on-Demand systems [10]), this model can be used to evaluate the impact of these mobility solutions on urban congestion.

References

[1] Weather Underground API and historical database. http://www.wunderground.com, Retrieved 12 Oct. 2013.

[2] John Amanatides and Andrew Woo. A fast voxel traversal algorithm for ray tracing. In *In Eurographics '87*, pages 3–10, 1987.

[3] Bureau of Public Roads. Traffic assignment manual. Technical report, U.S. Department of Commerce, Urban Planning Division, Washington, D.C (1964), 1964.

[4] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *Neural Networks, IEEE Transactions on*, 13(2):415–425, 2002.

[5] S. Kataoka, M. Yasuda, C. Furtlehner, and K. Tanaka. Traffic data reconstruction based on Markov random field modeling. *ArXiv e-prints*, June 2013.

[6] James H Kell, Iris J Fullerton, and Milton K Mills. Traffic detector handbook. Technical report, 1990.

[7] Boris S. Kerner. Modeling approaches to traffic congestion. In *Encyclopedia of Complexity and Systems Science*, pages 9302–9355. Springer, 2009.

[8] Vahid Moosavi and Ludger Hovestadt. Modeling urban traffic dynamics in coexistence with urban data streams. In *Proceedings of the 2nd ACM SIGKDD International Workshop on Urban Computing*, page 10. ACM, 2013.

[9] Yosef Sheffi. *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods*. Prentice-Hall, Englewood Cliffs, New Jersey, 1985.

[10] Kevin Spieser, Kyle Treleaven, Rick Zhang, Emilio Frazzoli, Daniel Morton, and Marco Pavone. Toward a systematic approach to the design and evaluation of automated mobility-on-demand systems: A case study in singapore. In *Road Vehicle Automation*. Springer, 2014 (in press).