

Unifying NYC Taxicab Records and Hotel Occupancy Data

Daniel J. Saunders

College of Computer and Information Sciences, University of Massachusetts, Amherst, Massachusetts
email: djsaunde@cs.umass.edu

and

Christian Rojas* and Debi Mohapatra**

Department of Resource Economics, University of Massachusetts, Amherst, Massachusetts

**email:* rojas@resecon.umass.edu

***email:* dmohapatra@umass.edu

SUMMARY: A method for fast distributed processing of New York City taxicab trip records with respect to a secondary dataset of daily hotel occupancy and pricing information is presented. We motivate and develop an algorithm used to select an optimal distance threshold for capturing taxi trips relevant to predicting hotel occupancy and pricing rates. Two hotel outlier removal techniques based on distribution matching and simple outlier detection are discussed, and used to reduce prediction error by removing noisy examples from the taxi trips data. We show supporting visualizations of taxicab trips in NYC in relation to hotel locations. The effect of predicting daily hotel occupancy rates and pricing with and without conditioning on daily nearby taxi traffic is demonstrated. Further applications of the taxi trip records dataset and other data sources are discussed as sources of potential future work.

KEY WORDS: Applied economics; Data analysis; Distributed computing.

1. Introduction

The availability of various forms of empirical commercial and industrial data is crucial for the solution of real-world economics problems. At the scale of cities and nations, however, certain data present major data processing challenges. With the advent of massively parallel and relatively cheap distributed computing, however, methods for the processing and analysis of such datasets are not entirely out of reach.

The New York City (NYC) Taxi & Limousine Commission (TLC) Trip Record Data consists in part of yellow and green taxicab trip records spanning the years 2009–2017. Recorded attributes include pick-up and drop-off date and time of day, pick-up and drop-off geospatial or zone-coded location, trip distances, fares, and others. Careful exploration and analysis of such a massive observational dataset may result in powerful insights in transportation issues in NYC. Methods developed on this particular dataset may be generalized to other urban settings in which such data is or becomes available.

Although interesting studies have been done on the NYC taxi records data, combining it with additional relevant datasets should allow researchers to propose and test more powerful economic hypotheses. We study a unification of the taxi data with the occupancy rates of 178 NYC hotels from the years 2013 to 2016. We are interested in trips which *begin or end within distance d* of at least one hotel in the dataset. To find the best choice of d over all hotels, we demonstrate an optimization procedure motivated from a simple distribution matching approach. Hotels with abnormal amounts of taxicab

traffic may be eliminated from consideration during or after the optimization process.

We hypothesize that both hotel occupancy rates and pricing can be predicted from the density of daily nearby taxicab pick-up and drop-off rides, combined with other relevant attributes. We hypothesize that hotel occupancy and, indirectly, pricing may be more accurately predicted as a function of daily nearby taxi ride density along with more generic attributes than without the nearby taxi rides information.

2. Related Work

An analytics model is proposed in Ferreira et al. (2013) which allows users to visually query taxi trips. Standard queries about the data are supported, as well as spatio-temporal “origin-destination” queries, allowing users to discover mobility patterns throughout the city. NYC hotspots are identified in Stoyanovich et al. (2017), in which taxi trips are represented as straight-line trajectories from pick-up to drop-off coordinates. Hotspots are used to identify lack of convenient public transportation options, and to suggest the addition of bus routes and ride-sharing options. A custom software approach is developed on top of the distributed processing system Apache Spark, allowing efficient analysis of taxi trajectory graphs as they evolve. In another work, a visual analytics method is proposed to study urban mobility patterns using graph modeling of taxicab trajectories (Huang et al., 2016). Dynamics of Shenzhen taxis is analyzed using graph analysis, and is studied at multiple scales by using a

graph partitioning algorithm to produce region-level visual analytics. Users of the system can interactively explore street-level city traffic patterns.

A technique for event-guided exploration of large spatio-temporal data is introduced in Doraiswamy et al. (2014), on the basis that manual exploration of such data is time-consuming and ineffective. Computational topology is used to discover interesting events in the data, and an algorithm is developed to group and index events which can be interactively explored or queried by users. This approach is validated on NYC taxi trips as well as subway service records. An analysis of the NYC taxi trips database in combination with various relevant urban data is presented in Wu et al. (2016). Correlations between taxi and the external datasets are used to predict traffic dynamics; namely, it is shown that points-of-interest (POIs) can predict regular traffic patterns, geo-tagged tweets can explain traffic caused by atypical events, and weather data may explain abnormal drops in traffic density. Chu et al. (2014) develop a method to discover and analyze information contained in a massive dataset of taxi trajectories. Geospatial coordinates are replaced with traversed street names, which are subsequently modeled with topic modeling tools. The “hidden themes”, or the discovered topics from the employed topic models, are used to analyze mobility patterns with a visual analytics system.

The efficient processing of spatio-temporal datasets (e.g., NYC taxi trip records) often requires specialized software and hardware. A novel indexing scheme over spatio-temporal data is developed for use with general-purpose graphics processing units (GPUs) in Doraiswamy et al. (2016). The index allows sub-second query speeds over large spatio-temporal datasets such as the NYC taxi data, and a massive number of tweets collected from Twitter.

3. Methods

3.1 Data processing

The dataset of hotel occupancy rates contains information from January 2013 until present day. The dataset of NYC hotel taxi records contains data from January 2009 until present day; however, geospatial coordinates of pick-up and drop-off locations are only recorded from January 2009 until June 2016. Given these constraints, our experiments consider only those records which lie in the date range of January 2013 until June 2016, approximately 3.5 years of joint data. Our analyses are concerned only with the yellow and green taxi services in NYC, and we discard all data concerning “for-hire vehicle” services (e.g., Uber, Lyft, and others).

We use a supercomputing cluster in which users are provisioned with a maximum of 40 CPU compute nodes. On each node, a Python program is dispatched to *pre-process* a single month’s worth of taxicab trip data with respect to the set of NYC hotels of interest. In particular, upon specifying a *distance criterion* d , each process will extract trips from its month worth of data which begin or end with d feet of any of the hotels under consideration. Geospatial distance calculations are accomplished using Vincenty’s formula (Bessel et al., 2010). Fast loading and pre-processing of the taxi trip records

is accomplished using the **dask** parallel computation and task scheduling Python library Arabie and Carroll (2016).

Prior to pre-processing, each month of yellow and green taxi trip data amounts to approximately 2.5Gb and 300Mb, respectively, totalling ≈ 115 Gb over the course of the experimental date range. Pre-processing the dataset using a distance criterion $d = 300$ feet reduces this to two files of size ≈ 20 Gb each, one of trips beginning near hotels of interest (*nearby pick-ups*), and another ending nearby (*nearby drop-offs*).

Pre-processing a month of yellow taxi data using the aforementioned Python program and distance criterion $d = 300$ requires *at most* 40 minutes and 50Gb of random-access memory. Using the super-computing cluster with a 40-node per-user allotment, processing all 42 months of taxi trip data requires only ≈ 80 minutes. If equipped with 42 or more nodes with the same computing resources, the processing time would be cut in half.

3.2 Distance criterion optimization

A crucial assumption in the analyses of the taxi dataset is that taxi trips that originate or culminate near a hotel are likely to indicate that guests are arriving or leaving the hotel. With this in mind, we can use the locations to which the guest travels, or where the guest has traveled from, to draw conclusions about typical patrons per hotel. Moreover, we can use the per-hotel density of nearby taxi trip pick-ups or drop-offs to make predictions about the economic viability of each hotel.

The distance criterion d should be selected such that it maximizes the predictability of some quantity of interest. We compare a dataset of hotel occupancy information with counts of nearby taxi trips, per hotel and per day. We denote the observed per-hotel distribution of hotel occupancy as \hat{p}_{occ} and the observed per-hotel distribution of nearby taxi trips as \hat{p}_{taxi} . In our experiments, these distributions are estimated using hotel occupancy and taxi data from January 1st, 2013 until January 1st, 2015, and the distribution of nearby taxi trips depends on the choice of distance criterion d .

We want to choose d such that some divergence measure \mathcal{D} between empirically observed distributions \hat{p}_{occ} and \hat{p}_{taxi} is minimized. We consider relative entropy $\left(\sum_x \hat{p}_{\text{occ}}(x) \log \frac{\hat{p}_{\text{occ}}(x)}{\hat{p}_{\text{taxi}}(x)}\right)$, summed absolute differences $\left(\sum_x |\hat{p}_{\text{occ}}(x) - \hat{p}_{\text{taxi}}(x)|\right)$, and summed relative differences, in which the smaller-valued proportion is used as the denominator in the quotient $\left(\sum_x \frac{\min\{\hat{p}_{\text{taxi}}(x), \hat{p}_{\text{occ}}(x)\}}{\max\{\hat{p}_{\text{taxi}}(x), \hat{p}_{\text{occ}}(x)\}}\right)$. Note that the divergence measures used are not necessarily true f -divergences, but were selected on the basis of their usefulness as quantified by their predictive power.

After pre-processing the NYC taxi data using a suitably large distance criterion (e.g., $d = 300$ ft.), a range of candidate distance criteria are considered (e.g., $d \in [25\text{ft.}, 50\text{ft.}, \dots, 275\text{ft.}, 300\text{ft.}]$), whose corresponding datasets are subsets of the pre-processed 300ft. dataset. The empirical per-hotel distribution of nearby pick-up and / or drop-off taxi trips \hat{p}_{taxi} is computed from the data for each choice of d , and we calculate the selected divergence measure with respect to the fixed \hat{p}_{occ} ; that is, the observed per-hotel distribution of rented rooms from the same time period. The criterion d giving the minimal divergence value is said to be the “best”,

and the dataset obtained from this choice of d can be used for downstream analyses.

3.3 Removing outlier hotels

Certain hotels under consideration may have an abnormally large or small number of nearby taxicab pick-ups or drop-offs depending on their location in the city. For example, Hotel Pennsylvania, located adjacent to Pennsylvania station, has a disproportionately large number of both pick-up and drop-off taxi trips due in part to the traffic that the train station causes. This observation makes it difficult to justify using such hotels in our analyses, as they may confound conclusions about hotels with atypical shares of nearby taxi traffic.

Using the same divergence measures as listed above, we propose two methods for the removal of outlier hotels:

1. Iteratively remove hotels from the occupancy dataset during the distance optimization process until the divergence measure between distributions, $\mathcal{D}(\hat{p}_{\text{occ}}||\hat{p}_{\text{taxi}})$, is below some threshold. Hotels are removed on the basis of how poorly their nearby taxi trip proportion matches their proportion of rented rooms. Picking the threshold is a hyper-parameter choice, and may be difficult to select *a priori*. We may also consider removing a fixed number of hotels based on the same distribution matching procedure.
2. After selecting a distance criterion d , apply a standard outlier-detection algorithm based on the per-hotel divergence measure values. **TODO**

3.4 Predicting hotel occupancy and pricing

We establish a simple estimation baseline using only information from the hotel occupancy dataset to predict daily hotel room demand. In particular, we fit an ordinary least squares (OLS) regression model to the hotel occupancy data, in which we assume the room demand is a linear function of the hotel identity, the day of the week, the month, and the year. This information allows the model to discriminate typical per-hotel room demand rates, as well as to capture multiscale time trends in occupancy rates. A multi-layer perceptron (MLP) regression model is also fit to this data, with hyperparameters (hidden layer sizes and regularization constant) selected by random hyper-parameter search. This model is able to learn non-linear interactions between the observations.

To evaluate the effect of conditioning hotel occupancy predictions on available nearby taxi trip densities, we include both nearby pick-up and drop-off trips as separate features in both the OLS and MLP models, as described above. Hyperparameters are re-selected according to a random hyper-parameter search, and the dataset is partitioned into train, validation, and test subsets as before. We train and test models using a range of nearby taxi distance thresholds $d \in \{25\text{ft.}, 50\text{ft.}, \dots, 300\text{ft.}\}$, in order to validate the distance thresholds d predicted by the optimization method and chosen divergence measures described above.

The data is partitioned into 80% training, 10% validation, and 10% test subsets. Both models are trained and test on the training and test subsets, and MLP hyperparameters are selected based on the best model accuracy on the validation subset. All data samples are randomly permuted to avoid leaving any hotels out of any of the subsets, and 5 independent

Table 1: Order of hotel removals and corresponding data using the relative difference divergence measure.

Removal	Hotel	\hat{p}_{occ}	\hat{p}_{taxi}	Best d	Rel. diff.
1	Res. Inn ... Trade Center	6×10^{-7}	2.6×10^{-3}	300	4,549
2	Hotel Gansevoort	3×10^{-3}	2×10^{-2}	190	6.33
3	Courtyard ... Herald Sqr.	2.2×10^{-3}	1.3×10^{-2}	180	5.93
4	Hilton ... Park Avenue	5.7×10^{-4}	3.3×10^{-3}	180	5.74
5	Hotel On Rivington	1.7×10^{-3}	9.6×10^{-3}	195	5.62
6	Holiday Inn Express ...	4.3×10^{-3}	8×10^{-4}	195	5.28
7	Hilton NY Midtown	3.6×10^{-2}	7.4×10^{-3}	195	4.91
8	Doubletree ... Fin. Distr.	7×10^{-3}	1.6×10^{-3}	190	4.27
9	Sohotel	9×10^{-4}	3.8×10^{-3}	190	4.2
10	Res. Inn ... Central Park	2×10^{-3}	8.1×10^{-3}	180	4.02
11	Hilton ... Square Central	9.2×10^{-4}	3.6×10^{-3}	180	3.87
12	Marriott NY Marquis	3.8×10^{-2}	9.9×10^{-3}	180	3.8
13	Sheraton ... Times Square	3.5×10^{-2}	9.5×10^{-3}	190	3.72
14	Fairfield ... Penn Station	4.2×10^{-3}	1.1×10^{-3}	195	3.65
15	Holiday Inn ... 57th St.	1.2×10^{-2}	3.5×10^{-3}	195	3.5

realizations of each model are trained. Averaged R^2 and mean-squared error (MSE) of both models are reported in 4.

All machine learning models are fit and evaluated using the `scikit-learn` machine learning library (Pedregosa et al., 2011).

4. Results

4.1 Outlier removal: choosing optimal distance threshold

Using the first outlier removal method, as part of the distance criterion optimization procedure, we may create tables of the order of hotel removals from the dataset for each divergence measure. We include one such table, in which the absolute differences measure is used to decide the hotel removal order. Included also are the removed hotels' absolute difference values, as well as the optimal distance criterion d for that particular iteration of the removal algorithm. Though there are 178 hotels in the dataset, we truncate the table after just 15 iterations in order to give an idea of the removal behavior using the relative difference measure.

Throughout the rest of the optimization, the optimal distances d produced on each iteration typically fell in the range [175, 250]. We may check whether these choice of distances produces subsets of taxi trip data which, when used as features in a machine learning algorithm, give better hotel occupancy and pricing prediction accuracy. The same may be investigated for the range of distances given by the other considered divergence measures. In this context, the most useful divergence measure will yield subsets of taxi data which, when used as observations in a machine learning model, reliably produce the most accurate predictions.

4.2 Outlier removal: outlier detection algorithm

TODO.

4.3 Predicting hotel occupancy and pricing

We first establish a baseline hotel occupancy and pricing learning setup using only the day of the week, the date, and the identity of the hotel. Given these observations, a model is trained to output the daily hotel occupancy and room pricing, both integer targets. The data is randomly shuffled to avoid leaving hotels out of the training or test subsets, and split

Table 2: MSE and R^2 values for OLS and MLP regression models trained without relevant taxi data.

Model	MSE (train)	R^2 (train)	MSE (test)	R^2 (test)
OLS	106,552 \pm 481	0.0166	105,463 \pm 1,923	0.0165
MLP				

Table 3: MSE and R^2 values for OLS regression model trained with relevant taxi data with a range of distance thresholds d .

d (ft.)	MSE (train)	R^2 (train)	MSE (test)	R^2 (test)
25	117,339 \pm 499	0.0193	116,856 \pm 1,990	0.0193
50	108,431 \pm 631	0.0160	105,588 \pm 2,524	0.0154
75	106,501 \pm 258	0.0167	106,314 \pm 1,030	0.0167
100	108,146 \pm 454	0.0211	106,199 \pm 1,818	0.0211
125	107,166 \pm 549	0.0215	107,298 \pm 2,196	0.0217
150	107,491 \pm 693	0.0213	105,996 \pm 2,779	0.0223
175	107,277 \pm 684	0.0212	106,929 \pm 2,740	0.0228
200	107,378 \pm 495	0.0214	106,525 \pm 1,979	0.0220
225	107,256 \pm 525	0.0212	106,999 \pm 2,098	0.0226
250	107,251 \pm 411	0.0213	107,006 \pm 1,643	0.0223
275	106,844 \pm 506	0.0218	108,628 \pm 2,023	0.0205
300	107,212 \pm 634	0.0214	107,157 \pm 2,538	0.0220

into 80% / 10% / 10% training, validation, and test subsets. An OLS model is fit to the training data and evaluated on the test data. A multi-layer perceptron regression model is also fit to the training data, where network hyper-parameters are chosen via random search according to the best predictive accuracy on the validation data, and is evaluated on the test data.

To compare, the same regression models are fit to the same observations, albeit with the counts of nearby pick-up and / or drop-off taxi trips per hotel and per day.

We train and evaluate all models using nearby taxi trip data using distances $d \in [25\text{ft.}, 50\text{ft.}, \dots, 275\text{ft.}, 300\text{ft.}]$, in order to test predictions from the distance optimization. In all experiments, we report the values of both (training and test) mean squared error (MSE) and the coefficient of determination, R^2 . Using recommendations from both outlier detection approaches, we investigate whether removing certain hotels from the dataset produces qualitatively better predictions, as measured by R^2 values.

5. Conclusion

ACKNOWLEDGEMENTS

The authors would like to thank the College of Computer and Information Sciences for their gracious allowance of supercomputing resources.

REFERENCES

Arabie, P. and Carroll, J. D. (2016). *Dask: Library for dynamic task scheduling*.

Table 4: MSE and R^2 values for MLP regression model trained with relevant taxi data with a range of distance thresholds d .

d (ft.)	MSE (train)	R^2 (train)	MSE (test)	R^2 (test)
25				
50				
75				
100				
125				
150				
175				
200				
225				
250				
275				
300				

- Bessel, F. W., Karney, C. F. F., and Deakin, R. E. (2010). The calculation of longitude and latitude from geodesic measurements. *Astronomische Nachrichten* **331**, 852–861. arXiv: 0908.1824.
- Chu, D., Sheets, D. A., Zhao, Y., Wu, Y., Yang, J., Zheng, M., and Chen, G. Z. (2014). Visualizing hidden themes of taxi movement with semantic transformation. *2014 IEEE Pacific Visualization Symposium* pages 137–144.
- Doraiswamy, H., Ferreira, N., Damoulas, T., Freire, J., and Silva, C. T. (2014). Using topological analysis to support event-guided exploration in urban data. *IEEE Transactions on Visualization and Computer Graphics* **20**, 2634–2643.
- Doraiswamy, H., Vo, H. T., Silva, C. T., and Freire, J. (2016). A gpu-based index to support interactive spatio-temporal queries over historical data. *2016 IEEE 32nd International Conference on Data Engineering (ICDE)* pages 1086–1097.
- Ferreira, N., Poco, J., Vo, H. T., Freire, J., and Silva, C. T. (2013). Visual exploration of big spatio-temporal urban data: A study of new york city taxi trips. *IEEE Transactions on Visualization and Computer Graphics* **19**, 2149–2158.
- Huang, X., Zhao, Y., Ma, C., Yang, J., Ye, X., and Zhang, C. (2016). Trajgraph: A graph-based visual analytics approach to studying urban network centralities using taxi trajectory data. *IEEE Transactions on Visualization and Computer Graphics* **22**, 160–169.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* **12**, 2825–2830.
- Stoyanovich, J., Gilbride, M., and Moffitt, V. Z. (2017). Zooming in on nyc taxi data with portal. *CoRR* abs/1709.06176.
- Wu, F., Wang, H., and Li, Z. (2016). Interpreting traffic dynamics using ubiquitous urban data. In *SIGSPATIAL/GIS*.