Biometrics 000, 000–000
000 0000

DOI: 000

Unifying NYC Taxicab Trip Records and Hotel Occupancy Data

Daniel J. Saunders

College of Computer and Information Sciences
University of Massachusetts Amherst
140 Governors Drive
Amherst, Massachusetts 01003
email: djsaunde@cs.umass.edu

and

Christian Rojas* and Debi Mohapatra**
Department of Resource Economics
University of Massachusetts Amherst
80 Campus Center Way
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 rates is presented. We motivate and develop an algorithm used to select an optimal distance threshold for capturing taxi trips relevant to predicting per-hotel occupancy rates. Two outlier removal techniques are discussed and used to improve prediction accuracy by removing hotels with abnormal quantities of taxi trips data. Using ordinary least squares and multi-layer perceptron regression, the accuracy of predicting daily hotel occupany rates with and without conditioning on daily nearby taxi traffic is compared, with and without removing outlier hotels from consideration. 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 science; Distributed computing; Machine learning.

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.

Several interesting studies have recently been carried out on the NYC taxi trips dataset. However, combining daily city-wide taxi trajectories with related NYC datasets should allow researchers to propose and test more powerful economic hypotheses. We study a unification of the taxi records 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 distance threshold d over all hotels, we develop an optimization procedure motivated from a simple distribution matching approach. Hotels with abnormal amounts of nearby taxi traffic may be eliminated from consideration during or after the optimization process.

We hypothesize that hotel occupancy rates can be more accurately predicted from the density of daily nearby taxicab pick-up and drop-off rides combined with information from the hotel occupancy data than from the hotel occupancy dataset alone. This effect should be strengthened as hotels with unusually high or low densities of nearby taxi trips are eliminated from consideration. With this in mind, we argue that many other urban economic data may be better predicted conditioned on observed per-day nearby taxi trips. Estimated values may be used as evidence for policy-making in the hotel and taxi industries, and others besides, depending on the availability of relevant and timely data.

Code for this work can be found at `https://github.com/djsaunde/nyctaxi`.

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). The dynamics of Shenzhen taxis are assessed using graph analysis techniques, and are studied at multiple scales using a graph partitioning algorithm to produce regional 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 the predict traffic dynamics; namely, it it 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 recordings from January 2013 until present day, although there are many hotels for which data is not available until a later starting date. 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 2014 until the end of June 2016, approximately 2.5 years of joint recordings. This allows for the retention of 148 of the 178 hotels in the dataset, an approximate 83% coverage for downstream analysis of the combined hotel and taxi trips data. Our experiments are concerned only with the NYC yellow and green taxi services, 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 independently extract trips from its designated month 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 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 requires approximately 2.5Gb and 300Mb of disk space, respectively, totalling ≈110Gb 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 ≈5Gb each, one of trips beginning near hotels of interest (*nearby pick-ups*), and another ending nearby (*nearby drop-offs*). This represents an approximate reduction of 91% of the original data.

Pre-processing a month of yellow taxi data using the aforementioned Python program and distance criterion $d = 300$ requires approximately 40 minutes and no more than 50Gb of random-access memory. Using the super-computing cluster with a 40-node per-user allotment, processing all 42 months of taxi trip date requires only ≈80 minutes. If equipped with 42 or more nodes with the same computing resources, processing time may be cut in half.

3.2 *Distance threshold 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 per-hotel densities of nearby taxi trip pick-ups or drop-offs to make predictions about future hotel room demand or other quantities of interest. We additionally assume that some unknown distance threshold d maximizes the likelihood, over all hotels, that the number of nearby taxi trips is approximately proportional to the number of the hotel's guests. If d is too small, too few hotel guests are captured in the taxi trips data; if d is too large, too many taxi users are mis-classified as traveling guests.

We suggest that 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 proportions of hotel occupancy as $\hat{p}_{\text{occ}}$ and the observed per-hotel proportions of nearby taxi trips as $\hat{p}_{\text{taxi}}$. In our experiments, these empirical distributions are estimated using per-hotel counts of aggregated hotel occupancy and taxi data from January 1st, 2014 until June 30th, 2016, 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 the empirically observed distributions $\hat{p}_{\text{occ}}$ and $\hat{p}_{\text{taxi}}$ is minimized. We consider summed absolute differences,

$$\sum_x |\hat{p}_{\text{occ}}(x) - \hat{p}_{\text{taxi}}(x)|,$$

and summed relative differences, in which the smaller-valued proportion is used as the denominator in the quotient,

$$\sum_x \frac{\max\{\hat{p}_{\text{taxi}}(x),\ \hat{p}_{\text{occ}}(x)\}}{\min\{\hat{p}_{\text{taxi}}(x),\ \hat{p}_{\text{occ}}(x)\}}.$$

Note that the divergence measures used are not true f-divergences. These measures may be considered useful depending on their ability to remove unpredictable outlier hotels from the regression problem.

After pre-processing the NYC taxi data using a suitably large distance criterion (e.g., $d = 300$ft.), a range of candidate distance criterions are considered (e.g., $d \in [25\text{ft.}, 50\text{ft.}, ..., 275\text{ft.}, 300\text{ft.}]$), 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 $\hat{p}_{\text{taxi}}$ is computed from the data for each choice of d, and we calculate the selected divergence measure with respect to the fixed $\hat{p}_{\text{occ}}$; 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 hotel outliers*

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 the Pennsylvania railroad 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. This approach is assessed in Section 4.1 and 4.2.2.
2. Using a trained machine learning model, hotels may be removed from consideration on the basis of which cause the most prediction error. We may iterate this process until R^2 scores have stabilized, or until the model has reached a satisfactory mean-squared error value. In this case, we are simply throwing away hotels whose occupancy rates are hardest to predict, which depends on the *a posteriori* knowledge gained by trying to fit a predictive model to the data. In addition, the variability of day-to-day rooms sales typically grows with the average room sales rate, so this criterion may need be adjusted to account for this inherent difficulty. This approach is assessed in Section 4.2.3

Of the two methods, the first seems more desirable, since we are still relying on observational data to fit a model. From the nearby taxi trips and hotel occupancy rates alone, we hope that discarding hotels in a principled manner will reliably improve predictive power. On the other hand, using a fitted model to indicate the most variable data samples is typical in the practice of machine learning, and is quite feasible with smaller models and reasonably-sized datasets. Although refusing to make predictions about a small subset of examples sacrifices perfect *coverage*, this is counter-balanced by improving the *precision* of the machine learning models on the remaining data.

3.4 *Predicting hotel occupancy*

We establish a simple estimation baseline using only information from the hotel occupancy dataset to predict daily hotel room demand. An ordinary least squares (OLS) regression model is fit to the hotel occupany data, where it is assumed that 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 demand, as well as to capture temporal trends at multiple timescales. A multi-layer perceptron (MLP) regression model is also fit to this data, with hyper-parameters (hidden layer sizes and regularization constant) selected by grid hyper-parameter search and averaged three-fold cross-validation loss. Importantly, this model is able to learn features codifying non-linear interactions between the observational data.

To evaluate the effect of conditioning hotel occupany 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. Again, hyper-parameters are selected according to a grid search and averaged three-fold cross-validation loss, and the dataset is partitioned into train and test subsets as before. We train and test models using a range of distance thresholds $d \in \{25\text{ft.}, 50\text{ft.}..., 300\text{ft.}\}$, in order to validate the distance thresholds d predicted by the optimization method and chosen divergence measures described above.

All data samples are randomly permuted to avoid leaving any hotels out of any data partitions, and 5 independent realizations of the chosen models are trained and tested. Averaged R^2 and mean-squared error (MSE) of both models are reported for a variety of settings of distance criterions d and with and without outlier hotels removed in Section 4.2.

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

Removal	Hotel	Best d	$\hat{p}_{\text{occ}}$	$\hat{p}_{\text{taxi}}$	Rel. diff.
1	Marriott New York Marquis	300	0.034678	0.008646	5.504272
2	Hilton New York Midtown	300	0.035888	0.010083	4.703882
3	Sheraton ... Times Square	300	0.034274	0.010320	4.450418
4	Waldorf Astoria New York	300	0.027579	0.009804	4.138792
5	Row NYC	300	0.027087	0.011239	3.993213
6	Holiday Inn ... 57th St	300	0.012890	0.002459	4.047989
7	The Roosevelt Hotel	300	0.021280	0.007908	3.806310
8	Lotte NY Palace	300	0.018808	0.007360	3.727761
9	Wyndham New Yorker Hotel	300	0.021757	0.009457	3.701483
10	Hilton Millenium Hotel	300	0.012483	0.003789	3.681267
11	Yotel NY @ Times Square	300	0.016021	0.006479	3.784685
12	Grand Hyatt New York	300	0.029967	0.018311	3.440488
13	Hudson Hotel	300	0.020890	0.010127	3.271367
14	Westin NY Grand Central	300	0.018530	0.008008	3.231396
15	Crowne Plaza ... Manhattan	300	0.021043	0.009871	3.136236

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 148 hotels in the reduced 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 distance optimization with the relative difference measure, the optimal distances d produced on each iteration largely fally in the range $[250, 300]$. This makes intuitive sense: more observations tend to lead to a better estimate of hotel room demand. If d is increased much more, however, we expect that this may no longer hold, since those trips that are deemed "nearby" with large d may be too far to have any bearing on the nearest hotel's daily sales.

We may check whether the choices of distance thresholds given by both divergence measures produce subsets of taxi trip data which, when used as observations for a machine learning model, give better prediction accuracy. 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 *Predicting hotel occupancy*

We first establish a baseline daily hotel occupancy estimation model 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 number of rooms sold by that hotel on that given day. The data is randomly shuffled to avoid leaving hotels out of any particular subset, and split into 80%, 20% training, test partitions. An OLS regression model is fit to the training data and evaluated on the test data. A multi-layer perceptron regression model is also fit, where network hyperparameters are chosen via a grid search according to the best averaged accuracy on a 3-fold cross-validation.

The results for both models are given at the top of Table 2.

4.2.1 *Full hotel 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. In this first modeling attempt, we utilize the full dataset of hotels, regardless of how atypical their nearby pick-up and / or drop-off taxi trip counts may be.

We train and evaluate all models using nearby taxi trip data using distances $d \in [25\text{ft.}, 50\text{ft.}, ..., 275\text{ft.}, 300\text{ft.}]$, 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.

The results for the OLS regression model and all considered settings of d are given in Table 2. Although the OLS regression model underperforms both baselines with $d = 25$, it eventually significantly outperforms the OLS regression baseline as d is increased, adding more nearby taxi trip counts. With $d = 300$, the OLS model only slightly underperforms the MLP model trained without any taxi data in terms of both MSE and R^2, while exhibiting much less variability in performance.

The results for MLP regression models trained with distance thresholds d are also given in Table 2. Even with the smallest considered subset of taxi data given by choosing the distance threshold $d = 25$ft. produces a significant decrease in training and test mean-squared error, and a significantly higher value of R^2. Though there is some variability in MLP performance as d is increased, MSE values tend to decrease while R^2 values increase. With all hotels included, best candidate distance criterions (according to modeling on the full hotel data) include $d = 225$ft., 275ft., and 300ft.

4.2.2 *Outlier removal: distance optimization.* For each of the three divergence measures detailed in Section 3.2, an ordering of hotel removals based on the magnitude of the measure evaluation can be found. While the procedure relied on the taxi dataset (pre-processed with a distance threshold of 300 feet relative to hotel locations), hotels can be removed from both machine learning models with and without access to taxi data as observations.

The results for models trained with various distance thresholds d and without taxi data are shown in Figure 1 in which the absolute differences metric is used, and in Figure 2 in which the relative differences metric is used. The "Naive" line plot corresponds to the MSE and R^2 results for models trained without nearby taxi trip counts. It is difficult to spot any global trend in the MSE and R^2 results for either the fitted OLS or MLP models as hotels are iteratively removed, for either divergence metric. This suggests that the chosen metrics are not necessarily good indicators of outlier hotels, or that the relative proportions of per-hotel occupancy and nearby taxi trip counts need not be equal to be informative.

4.2.3 *Outlier removal - highest MSE.* Hotels with the highest mean-squared errors are removed one by one from

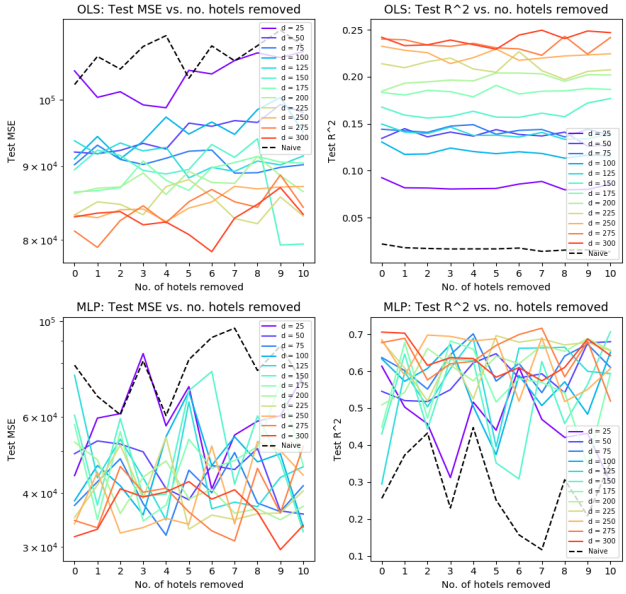

Figure 1: MSE and R^2 values for OLS and MLP models fit with various distance thresholds d. Hotels with the largest absolute value differences between proportions of nearby taxi trip data and hotel occupancy are iteratively removed. This seems to have a near-random effect on the fit of the predictive models.

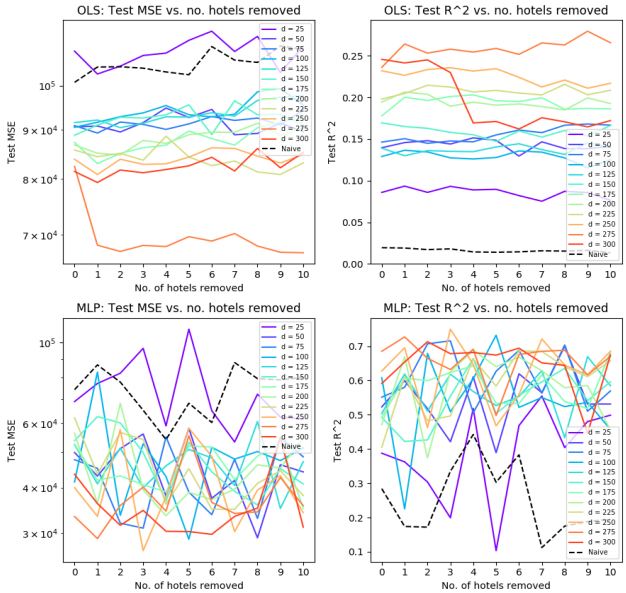

Figure 2: MSE and R^2 values for OLS and MLP models fit with various distance thresholds d. Hotels with the largest relative differences between proportions of nearby taxi trip data and hotel occupancy are iteratively removed. This seems to have a near-random effect on the fit of the predictive models.

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

Model	d	MSE (train)	R^2 (train)	MSE (test)	R^2 (test)
OLS	-	106,552 ± 481	0.0166	105,463 ± 1,923	0.0165
MLP	-	79,592 ± 3,382	0.2620	80,718 ± 5,113	0.2610
OLS	25	109,353 ± 567	0.0963	107,669 ± 2,276	0.0921
	50	95,367 ± 512	0.1428	94,824 ± 2,046	0.1437
	75	94,594 ± 144	0.1390	94,970 ± 583	0.1462
	100	96,411 ± 227	0.1225	98,077 ± 907	0.1158
	125	95,403 ± 631	0.1308	94,481 ± 2,521	0.1307
	150	92,105 ± 301	0.1554	93,720 ± 1,205	0.1598
	175	90,397 ± 446	0.1787	88,569 ± 1,787	0.1765
	200	88,093 ± 308	0.1973	87,165 ± 1,231	0.1992
	225	86,192 ± 275	0.2141	85,492 ± 1,097	0.2164
	250	84,369 ± 249	0.2292	85,484 ± 999	0.2226
	275	**83,345 ± 371**	**0.2386**	83,810 ± 1,484	0.2376
	300	83,633 ± 263	0.2382	**82,655 ± 1,053**	**0.2394**
MLP	25	60,384 ± 8,877	0.4994	60,428 ± 9,796	0.4967
	50	50,176 ± 5,267	0.5476	50,122 ± 5,589	0.5530
	75	43,835 ± 4,942	0.6014	44,454 ± 4,662	0.5985
	100	45,588 ± 5,509	0.5851	45,779 ± 5,020	0.5872
	125	47,437 ± 4,877	0.5673	47,630 ± 5,204	0.5637
	150	50,678 ± 10,050	0.5379	50,648 ± 9,794	0.5355
	175	46,218 ± 6,021	0.5788	46,501 ± 6,034	0.5730
	200	50,535 ± 12,527	0.5372	51,004 ± 13,138	0.5402
	225	37,277 ± 6,933	0.6603	37,064 ± 6,297	0.6593
	250	44,971 ± 10,325	0.5896	43,966 ± 9,011	0.5895
	275	**35,077 ± 1,456**	**0.6809**	**35,401 ± 1,668**	**0.6727**
	300	38,173 ± 5,142	0.6514	38,251 ± 4,996	0.6513

consideration in the estimation of the multi-layer perceptron models, with and without taxi data. The MSE and R^2 results for MLP and OLS regression and all distance thresholds $d \in \{25, 50, ..., 300\}$ and without taxi data (*"Naive"*) are plotted in Figure 3. For the MLP models, the same grid search and 3-fold cross-validation procedure is used to pick a best setting of the hidden layer sizes and regularization constant hyper-parameters at each iteration of hotel removal.

In most cases, the networks with access to the nearby taxi trip counts obtain lower MSE and higher R^2 values, and larger values of d tend to produce better predictions and explanation of variance. This further validates the claim that the inclusion of nearby taxi trips reduce uncertainty in the prediction of hotel occupancy. As observed above, the multi-layer perceptron training produces more accurate but less stable predictions, demonstrated by the smooth, near-monotonic decrease in OLS MSE with increasing d versus the jagged decrease of MLP MSE.

As hotels are removed from the dataset, both metrics trend towards improvement, but not monotonically so. We argue that this is due to two opposing forces: while removing noisy hotels reduces the degree of unpredictibility, it also has the effect of shrinking the size of the dataset. Without large amounts of data, it is often difficult to train complex machine learning models; i.e., the multi-layer perceptron. The drastic drop in OLS R^2 values may indicate that, for some values of d, too much training data has been removed. The trend towards improved modeling validates the use of this method for identifying hotels with abnormally unpredictable occupancy rates.

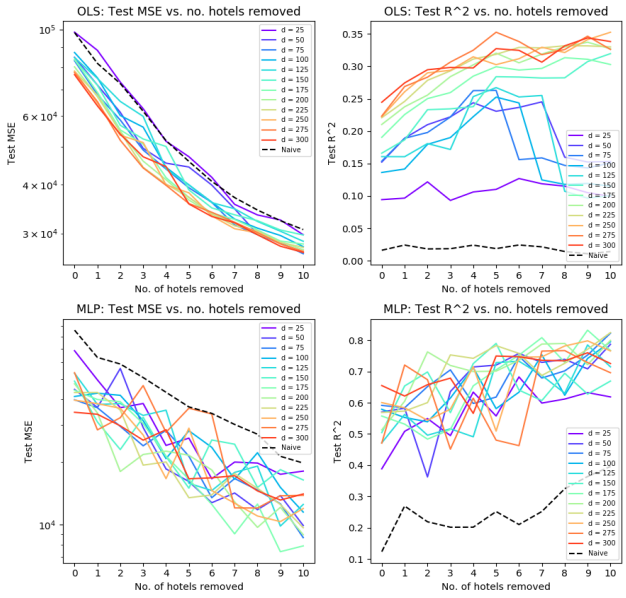

Figure 3: MSE and R^2 values for OLS and MLP models fit with various distance thresholds d. Hotels with the largest MSE are iteratively removed, decreasing available training data but increasing hotel occupancy predictability. For a small number of hotel removals, this causes better MSE and R^2 values.

5. Conclusions and Future Work

We have demonstrated a fast method for processing large quantities of taxi data in relation to auxiliary information on hotel room sales. This method facilitates the reduction of uncertainty in downstream data analysis tasks without discarding too much relevant data. This point is demonstrated with a simple prediction task and two standard machine learning models. Models which learn nonlinear interactions between observation variables are much more accurate predictors.

Methods for finding a distance threshold which enables maximal predictive power are developed and assessed. The optimal distance threshold comprises a point past which additional data is not useful or provides diminishing returns, and trades off practically with computation time and storage requirements. The taxi data, however, is not always a robust indicator of hotel room sales, and several hotels appear to have inherently less predictable occupancy rates, with or without it. In the case that nearby taxi trip density does not match well with the per-hotel room sales distribution, we can expect that hotels with the worst divergences are more difficult to predict. This effect is difficult to determine, however, especially since the metrics employed to estimate it are chosen in an ad-hoc manner based on their desired properties. The design of a better divergence metric based on a more detailed analysis of the relationship between hotel and taxi trips data may be needed to make better outlier removal recommendations and improve model accuracy. Indeed, choosing different per-hotel distance thresholds may be warranted for hotels situated in areas of the city with qualitatively different traffic patterns.

Our methods are applicable also to other scenarios in which a large quantity of spatiotemporal data is available, but where desired analyses concern only a small subset of it. Distributed processing of large data is crucial for rapidly estimating the parameters of statistical models and answering questions with them in a timely fashion. Loading all available data into the memory of a distributed system would enable even faster processing, but at a potentially much higher dollar cost. For this reason, our method is especially suitable for institutions or researchers with restrictive distributed computing resources. A comparison of the relative computation time and cost requirements of different taxi data processing methods is needed to determine the most efficient strategy.

A more complex modeling approach, suitable to the problem of predicting time series data, may increase our system's predictive performance. Although we use the weekday, month, and year as inputs to our regression models, considering the data as a sequence (e.g., with a m-order Markov chain or recurrent neural network) may allow our regression models to capture more of the inherent structure of the data, leading to more accurate predictions. On the other hand, a loss function more effective than mean-squared error on daily room sale counts could be designed. Outputting predictions of daily per-hotel shares (i.e., proportions) of room sales may encourage our regression models to discover useful relationships in inter-hotel room demand. Although an interesting problem on its own, a highly accurate model of daily per-hotel room sales is not the main focus of this work.

With evidence that nearby taxi data informs hotel occupancy rates, it seems likely that it can be used to predict other variables of socioeconomic interest. With the assumption that nearby taxi trips inform hotel guest rates, we may additionally characterize where those supposed guests arrive from or travel to, enabling estimation of the degree to which hotels compete to provide access to nearby attractions. These observations may be combined with data on the relative popularity of attractions to enable more accurate predictions, as captured by public visitation records or nearby geotagged tweets and taxi trips. This is an exciting direction for potential future work.

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.*

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*.