Appendix for To Ride-Hail or Not to Ride-Hail? Complementarity and Competition Between Public Transit and TNCs Through the Lens of App Data

Hengfang Deng, Edgar Castro, Sage Gibbons, Justin de Benedictis-Kessner, Ryan Qi Wang, Daniel T. O'Brien[1]

[1] Corresponding author. d.obrien@neu.edu

BACKGROUND AND DATABASE COMPONENTS

In Massachusetts, there were approximately 91.1 million TNC rides in 2019: a 12.0% increase from 2018 and a 40.6% increase compared to 2017 (Utilities, n.d.). There was a significant drop in the rideshare sector as a result of pandemic-related causes in Massachusetts. However, there were almost 12.5% more rideshare trips in 2021 than there were in 2020. The eastern region of Massachusetts, particularly the greater Boston area witnessed the greatest rebound in ride volume.

There are a few MBTA apps for commuters in the Greater Boston Area. But the Transit App is the only one that has the agency's official endorsement and also has the largest number of users. There were on average over 150,000 users during each month of the study period. As described in the main text, the data used for this paper come from the Transit app, with records of users' interactions with several different aspects of the app stored in separate forms. The main data used in the paper are the trip planning features of the app, which allow us to impute both origins and destinations of users given their input.

Other components of the data include the “nearby view” feature, which is what most users see when first opening the app. This feature allows users to see transit stops and upcoming arrivals in their close vicinity, as well as other alternative transportation options such as bikeshare or TNCs. Our records of users' interactions with this feature of the app include users' taps on each line of transit (or non-transit) transportation option shown to them in this view. Similar to the main data described in the paper, we store the records of users' interactions with the “nearby views" feature in a database consisting of a flattened version of the raw JSON data managed with SQLite3 via Python.

CROSS-VALIDATION WITH 2020 RIDESHARE DATA REPORT

We then utilize the 2020 Rideshare Data Report data (Utilities, n.d.) that provides statistics about the number of rides in Massachusetts provided by TNC companies. Such data reports the trip counts on the municipality level and over 35 million trips had an origin or destination in Boston. For the Transit-App data, the TNC-tapped trips from the Transit App were then geo-tagged using the municipality GIS file and aggregated to the same level. Finally, we compare the overall trip count between the survey data and the aggregated potential trips (based on counts). Fig. 1 below displays the representativeness of the origin-based comparison result (c-d) and the geographical distributions of Transit App Taps (a) and TNC rides (b), we can see that both spatial patterns demonstrate the significant use of rideshare services in the city of Boston compared to the rest of the municipalities. If we exclude the city of Boston, the Pearson Correlation coefficient between two data sources on the municipality level is 0.82 with similar spatial patterns in suburban areas (i.e., higher concentrations in the Worcester, Brockton, and Lowell regions).

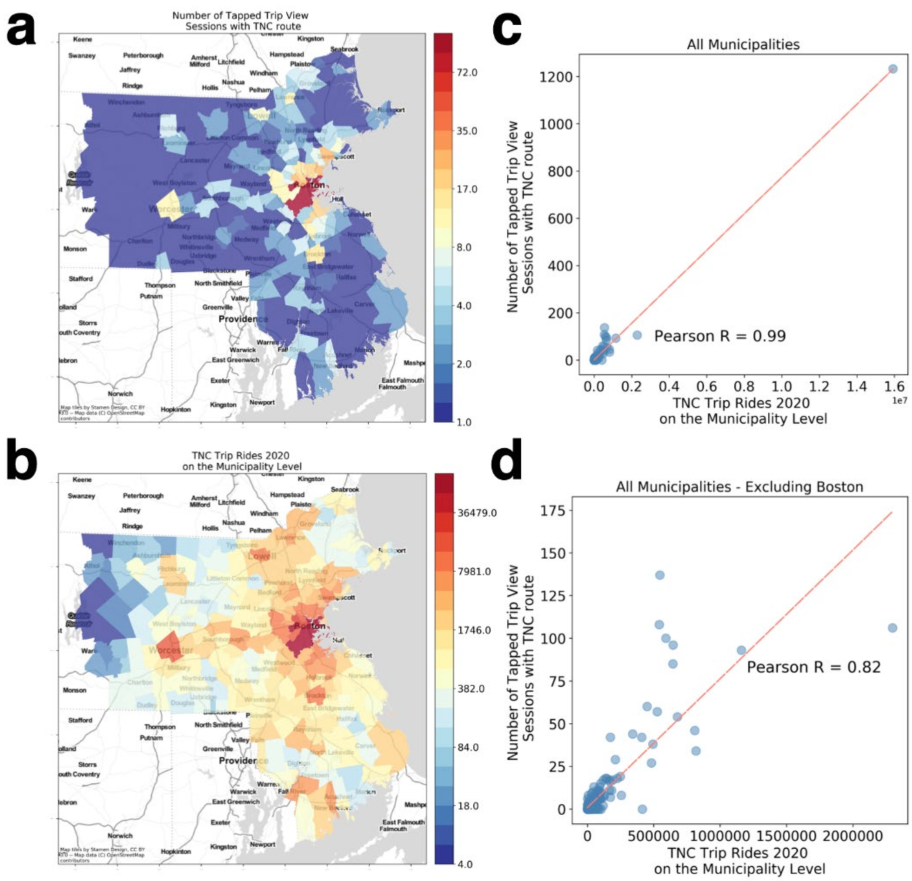

Fig. 1: Cross-validation with the Massachusetts 2020 Rideshare Data Report

TAP NUMBER COMPARISON

Fig. 2 shows that amongst all trip queries, during the vast majority (94.51%) of queries a user only taps on transit-based trips. There is no query during which a user taps on only TNC trips; this might be due to our data source coming from Transit App rather than any TNC App. There are 3.59% of the queries in which the taps on TNC trips are greater than the transit trips. There are only 1.24% where the total tap count for both modes is equal and we remove those trip queries to capture clear inclinations.

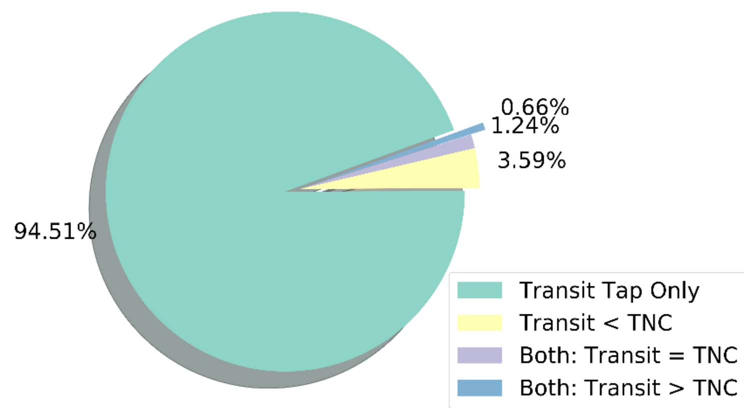

Fig. 2: Comparison of the Number of Taps on both Transit and TNC across all Queries

MULTI-MODE TRIPS

Another initial finding based on the trip data screening is that a large portion of individuals who have tapped on TNC trips tend to tap on the trips that have a combination of both Transit and TNC services. Based on the historical taps, 50.77% of the tapped trips containing TNC are multi-mode trips and across all users who have previously expressed their interests toward the TNC services, 50.41% of them are solely interested in the multi-mode trips. This further suggests that a significant fraction of potential riders who shift from Transit to TNC are using the TNC service as the first or last mile to the transit stop. Fig. 3 reveals the spatial and temporal features of trips with both modes. Where it can be observed that the overall hot spots are similar to the TNC-only hot spots including South Station and the Logan Airport and the majority of the trips are below 20km. There are some additional new destination spots such as the Southeastern areas of the city. The temporal distribution shows a slight peak around 3 pm across the study period. Fig. 4 shows the ratios between the duration and distance of the TNC portion to the Transit portion, and the ratios are close to 0.25 for both metrics highlighting that the Transit service acts as the major portion of such trips.

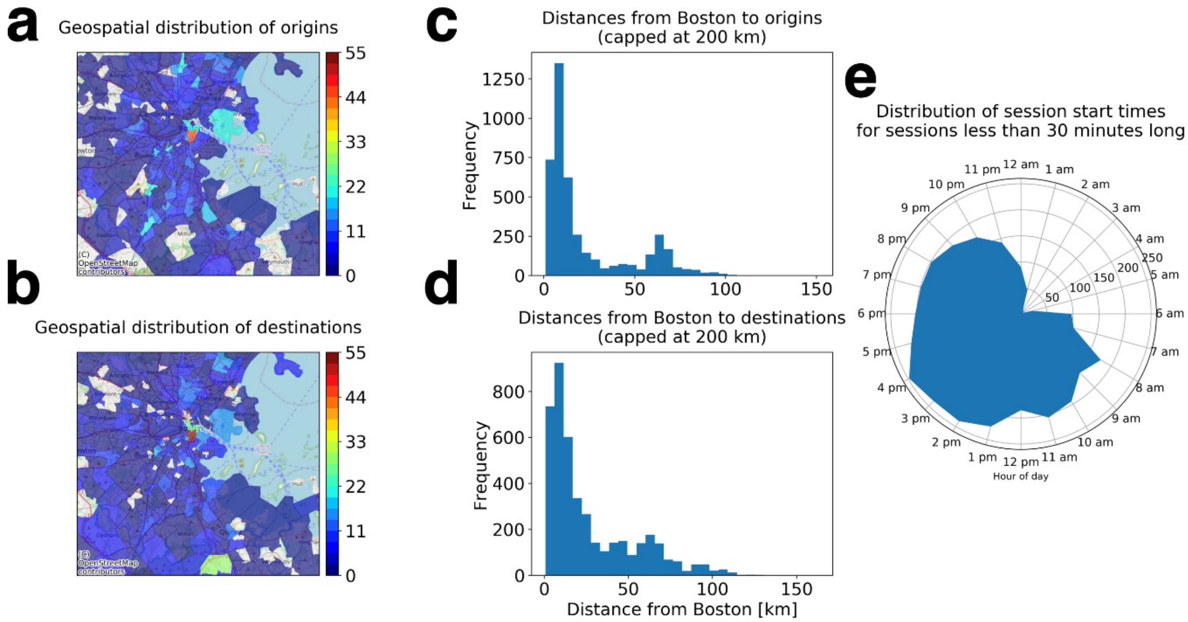

Fig. 3: Spatial Distribution and Spatial-temporal Features of the Multi-mode Trips with Taps

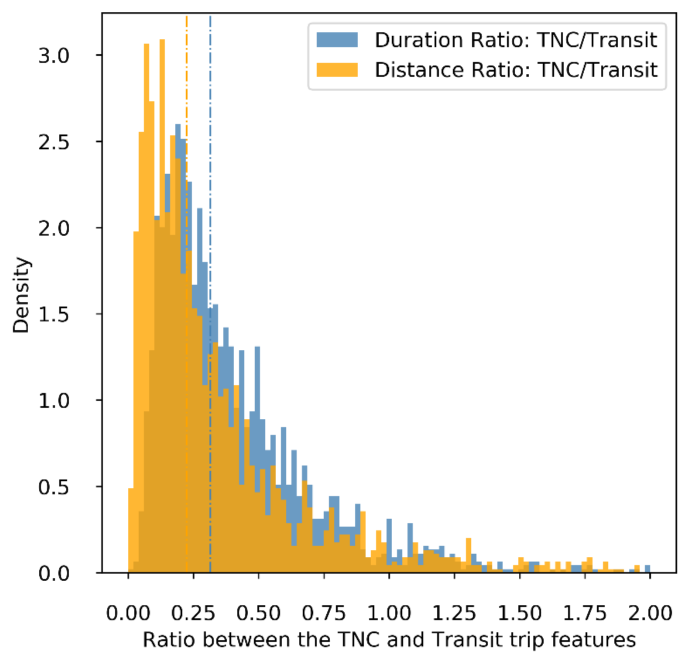

Fig. 4: Duration and Distance Ratios of the Multi-mode Trips with Taps

DATA TRANSFORMATION

For all trip features, we visualize the histogram and compute the skewness following (Mardia, 1974) as shown in Fig. 5. We use a threshold of 1 as suggested by (Mardia, 1974) and perform a log-transformation for any variable that has a skewness value greater than 1 which is shown in Fig. 6. We can see that most heavy-tailed features appear to be more bell-shaped after the transformation with a significant decrease in the skewness value. However, the walk distance feature appears to be an exception hence we use the original scale for modeling and analysis.

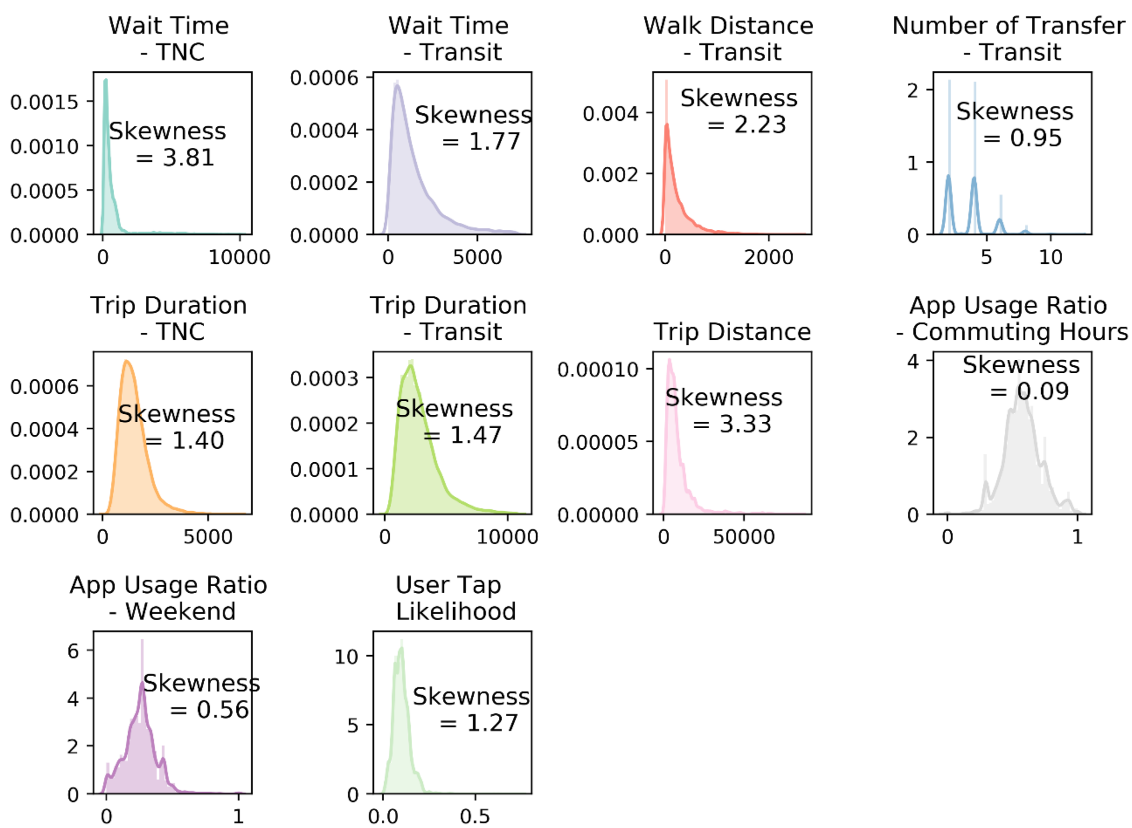

Fig. 5: Histograms of Input Features

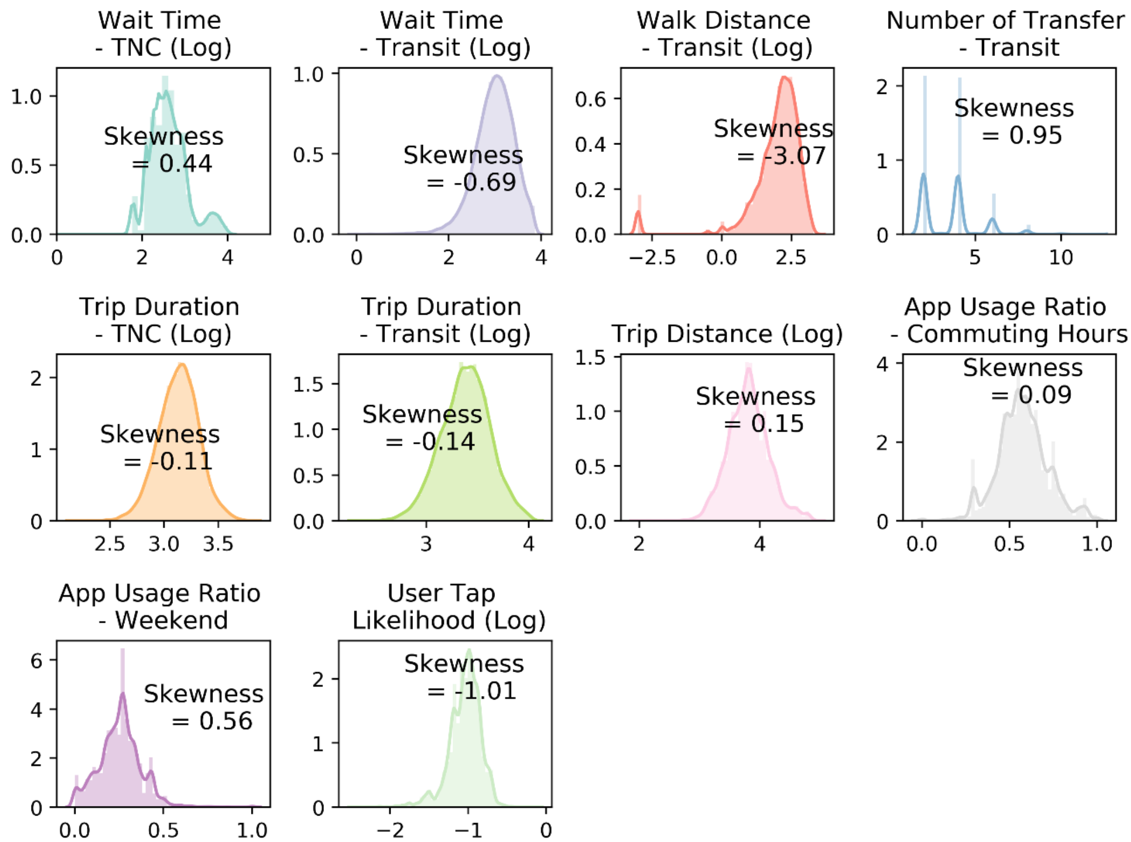

Fig. 6: Histograms of Input Features with Log-transformation on Selected Features

PRECIPITATION DATA

Fig. 7 shows the daily and weekly rainfall based on weather station's historical readings. We can observe some peaks during certain weeks and such events are quite sparse.

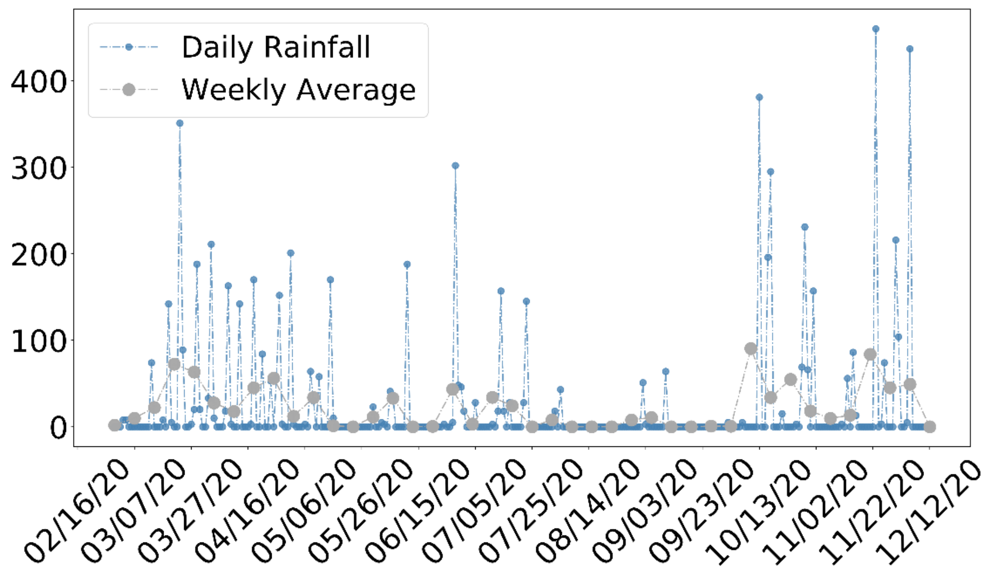

Fig. 7: Time Series of the Historical Daily and Weekly Precipitation (in millimeters)

TRIP TAP TIME SERIES

Fig. 8 shows the temporal progression of daily trip taps from all Transit App users during the study period. The y-axis represents the ratio to the annual median value on each day/week for different types of trip taps. We observe that there is a decrease following the national emergency declaration responding to the COVID-19 pandemic, such decrease can be seen in both daily and weekly patterns and the usage recovered between late July and late October as the reopening takes place. The overall temporal progression trends for both TNC and Transit are similar to each other.

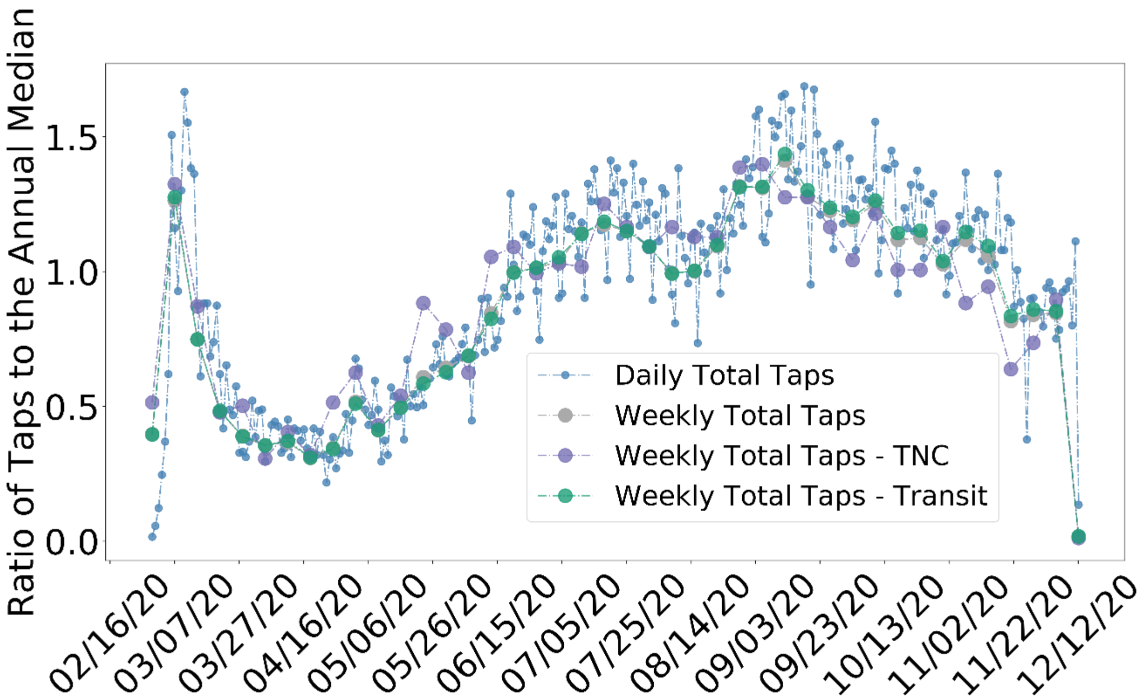

Fig. 8: Time Series of the Trip Taps on Transit and TNC Modes

ADDITIONAL METHODS DETAILS

We examine the questions of mode choice between public transit and TNCs using several machine learning classification models. We deploy these classification models on our data, which contain the features described above, including trip-level, user-level, and contextual features, along with y_i, the target mode choice corresponding to the user's tap for that query i (with 1 = TNC, 0 = Transit). The overall prevalence of TNC sessions by this metric is 4%. We limit the analyses to users who have tapped on both transit trips and TNC trips at least once each during the study period, ensuring we are not erroneously examining individuals who would never tap on one mode or the other (given the numbers, we are most likely excluding individuals who would never consider TNC). Within these users, we only include sessions in which: (1) the user utilized the trip planning feature (i.e., requested a set of trip options from the app); (2) the trip planner generated at least one public transit option and at least one TNC option; and (3) the user tapped on at least one type. We log-transformed numerical predictors with significant skewness to account for outliers and then standardized them for all numerical variables. We then perform One-Hot Encoding for the categorical variables and after the construction of features, we use

Logistic Regression, Support Vector Machines (SVM), Random Forest (RF), and Adaptive Boosting (AdaBoost) to train models. The data were randomly split, with 75% samples used for training and the remaining 25% samples used for testing.

The low prevalence of TNC taps creates an imbalanced data set and thus can impact the performance of prediction from conventional classification algorithms. To address this issue, we assign the class weight to each instance based on the inverse ratio of the corresponding label during the training process to balance the label distribution. Additionally, to ensure the robustness of the results, we rerun all analyses 100 times on bootstrapped samples and report both the means and standard deviations of all evaluation metrics.

Logistic Regression

As one the most widely used classification methods, the basic function of the Logistic Regression model is:

$$P(y_i = 1|x_i, w) = sigm(w_0 + w_i * x_i)$$

where x_i represents the predictive feature vector and $y_i = 1 \; or \; 0$ represents whether this session's user choice is classified as one particular label or not. The *sigm* function refers to the sigmoid function and it is defined as:

$$sigm(x) = \; e_x/(e_x + 1)$$

The loss function is the cross-entropy loss which is defined as:

$$L(f(x), y) = -\frac{1}{n}\sum_{i=1}^{n}(y_i * lnf(x_i) + (1 - y_i) * ln\,(1 - f(x_i)))$$

where x is the set of the inputs and y represents the corresponding labels, and $f(x)$ denotes the output of the Logistic Regression model given input x_i and binary vector y_i with n as the total number of instances. Table 1 shows the coefficients of the logistic regression model with all samples as input, in which five variables were statistically significant predictors. The strongest predictor, based on z-score, was walking distance, wherein longer distances were associated with a greater likelihood of tapping on TNC (with a coefficient of 0.27, $p < .01$). Other significant predictors included contrasting effects of trip distance and duration. For trips with a greater distance, people were less likely to tap on TNC options, but when the public transit option would take a greater amount of time people were more likely to tap on the TNC option. Note that since these effects are controlling for each other, they distinguish between trips that are far and how efficiently they are traveled by public transit. The last two findings were that users were more likely to tap on TNC if the optimal transit trip involved commuter rail; and if the wait time for the public option was longer.

Coefficients	Estimate	Std. Error	z value	P value
Intercept	-3.45	0.08	-32.45	0.00
Transit Vehicle Type – Commuter Rail	0.78	0.17	4.53	0.00
Trip Distance	-0.56	0.09	-6.23	0.00
Trip Duration – Optimal Transit	0.41	0.08	5.18	0.00
Walk Distance – Optimal Transit	0.27	0.03	8.27	0.00
Wait Time – Optimal Transit	0.19	0.05	4.05	0.00
Nearby View Tap	-0.15	0.12	-1.21	0.23
Number of Transfer	0.08	0.08	1.56	0.12
Wait Time – Rideshare	0.07	0.05	1.4	0.15
App Usage Ratio – Commuting Hours	0.05	0.04	1.31	0.19
Trip Start Time in Commuting Hours	-0.04	0.12	-0.29	0.77
Transit Vehicle Type – Subway	0.03	0.1	0.32	0.75
Wait Time – Rideshare	0.02	0.09	0.22	0.83
Trip Start Time in Non-commuting Hours	0.01	0.09	0.05	0.96

Table 1: Coefficients of the Logistic Regression Model

Random Forest

As an extension of the traditional Classification and Regression Tree (CART) method, Random Forest is a tree-ensemble algorithm that uses a significant number of decision trees simultaneously to control the bias and variance of the model at the same time (Liaw & Wiener, 2002). Training a Random Forest model involves two statistical techniques: bootstrapping and bagging. First, N bootstrapped sample sets are used from the entire training set with random sampling with replacement. Each set is then supplied to construct a regression tree without pruning, enabling the tree to grow independently to its maximum. Instead of considering all predictors (i.e., features), only a small and fixed number of total features would be considered as split candidates, as using only a subset of the predictors can eliminate the correlations among all trees. Such a principle also provides a unique opportunity of evaluating the variable importance.

Support Vector Machine

Support Vector Machine (SVM) is commonly used as a classifier that performs classification by finding the best hyperplane as a decision boundary (Schölkopf & Smola, 2002). One main advantage of the SVM is that it implements instinctive complexity control to reduce overfitting. The main objective of SVM is to maximize the margin around the hyperplane and the classifier is determined by a subset of all samples. A simple linear Support Vector Classifier (SVC) can be built in the following mathematical process:

$$\min_{w,b,\varepsilon} \quad \frac{1}{2}w^T w + C\sum_{i=1}^{n} \varepsilon_i$$

$$y_i(w^T \emptyset(x_i) + b) \geq 1 - \varepsilon_i$$

$$\varepsilon_i \geq 0, i = 1, \dots, n.$$

where ε is the threshold value that controls the number of support vectors for model fitting, C is a penalty parameter that controls the flexibility of the model, and $\emptyset(x)$ denotes the features mapped from the original input with the kernel function. W and b represent the weight vector and bias term, respectively.

AdaBoost

AdaBoost is an adaptive and ensembled boosting algorithm that essentially integrates multiple classifiers into one “great classifier”. The ensemble of many decision stumps and weighting them as emphasizing on the challenging instances and less on those already trained well provides an opportunity to optimize the model performance, which leads to generally higher accuracy compared to linear models. The Pseudo-code for the algorithm originally introduced by (Freund & Schapire, 1997) is shown in Fig. 9.

Algorithm AdaBoost$(\mathbf{Z}, T)$

Input: l examples $\mathbf{Z} = \langle(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)\rangle$

Initialize: $w_1(\mathbf{z}_i) = 1/l$ for all $i = 1 \ldots l$

Do for $t = 1, \ldots, T$,

1. Train classifier with respect to the weighted sample set $\{\mathbf{Z}, \mathbf{w}^t\}$ and obtain hypothesis $h_t : \mathbf{x} \mapsto \{\pm 1\}$
2. Calculate the training error ε_t of h_t:

$$\varepsilon_t = \sum_{i=1}^{l} w^t(\mathbf{z}_i)(h_t(\mathbf{x}_i) \neq y_i) \ , \tag{1}$$

abort if $\varepsilon_t = 0$ or $\varepsilon_t \geq \frac{1}{2} - \Delta$, where Δ is a small constant

3. Set

$$b_t = \log \frac{1 - \varepsilon_t}{\varepsilon_t}. \tag{2}$$

4. Update weights $\mathbf{w}^t$:

$$w^{t+1}(\mathbf{z}_i) = w^t(\mathbf{z}_i) \exp\{-b_t(h_t(\mathbf{x}_i) = y_i)\}/Z_t \ , \tag{3}$$

where Z_t is a normalization constant, such that $\sum_{i=1}^{l} w_{t+1}(\mathbf{z}_i) = 1$.

Output: Final hypothesis

$$f(\mathbf{x}) = \sum_{t=1}^{T} c_t h_t(\mathbf{x}), \quad \text{where} \quad c_t := \frac{b_t}{\sum_{t=1}^{T} |b_t|} \tag{4}$$

Fig. 9: Pseudo-code of the AdaBoost Algorithm

EVALUATION OF TRANSIT APP TAPS

Overview

We acknowledge that taps on routes in Transit do not guarantee that someone then took said route. It is possible that they were looking for more information to evaluate the option and then chose another. To evaluate these alternative interpretations, we conducted several empirical exercises, the most elaborate of which was the analysis of a subset of 181 sessions for which a user tapped on a single route and then had a subsequent session using the app before the estimated arrival time for the tapped route. In these sessions we were able to examine their actual trajectory via GPS location. (Note that most individuals understandably do not open the app again once they have boarded their mode of transportation, offering no data on future locations for these users). Of this subset, 41.9% were within 25m of the route at that time (see the Results subsection below). This suggests that a substantial proportion of people tapping on routes did indeed use these routes of transit. Though this rate of matching user trajectories to tapped routes is not perfect, it is in line with the accuracy rates of techniques using cell phone-generated mobility to assess mode choice (see the Background subsection below), which are largely considered to be the most accurate method for measurement in this area. Especially given the sparsity of the data, this match rate is likely a lower bound on the degree to which users' taps indicate real-world mode choice.

Background

Prior research utilizing real-world mobility data and validating transportation analysis includes two main types of studies: transportation mode detection and map matching with crowd-sourced GPS data sets. The accuracy from these studies varies substantially, ranging from 25% to 98.4%. Firstly, the frequency of data collection can determine the accuracy. For instance, Feng et al. (2018) collected about 3,000 data points per person for 5,000 individuals in one month (i.e., approximately 4 per hour) and achieved a prediction accuracy of 69.4%. Shin et al. (2015) designed their smartphone app that recorded the location every second of 495 individuals and was able to achieve an accuracy of 82%. In contrast, each device in the Transit data set only reports a median of 42 data points per year. The median increases to 285 for users who made at least one tap, which is less than once per day. Bedogni, Di Felice & Bononi (2016) reported that although their algorithms could achieve an accuracy of mode detection of more than 80%, a lowered sampling rate could reduce the accuracy to about 40%. Secondly, mode detection for public transportation systems is more difficult compared to driving. In the study of Bedogni, Di Felice & Bononi (2016), the overall accuracy for mode detection was above 80% but the accuracy for the bus was less than 55%.

There is also a large body of studies developing algorithms matching the sequence of GPS points to transportation routes or networks. Initially, features such as trajectory similarity between road networks and location sequence were proposed as input features for the map-matching procedure (White, Bernstein & Kornhauser, 2000). In addition to the similarity or closeness estimation techniques, methods such as Kalman Filter, Fuzzy Logic, and Hidden Markov Models have also been previously demonstrated as candidate methods with accuracies varying between 60% and 90% (Quddus, Noland & Ochieng, 2006; Obradovic, Lenz & Schupfner, 2006; Maher Atia et al., 2017). However, most of these methods do not work well for sparse trajectories. The GPS locations reported from the Transit App also show a high level of sparsity as the users tend to check the route services or progress at irregular intervals. Hence, the ability of the Transit data to confirm the route of choice, then, would be expected to be on the lower end compared to previous studies.

Objective

The main objective of our validation process is to match GPS observations recorded from the Transit App in the backend to the Transit Routes. The hypothesis is that given a user has tapped on specific routes in the current trip planning query session, the users who continue to use the Transit App for GPS-based navigation services, their movement trajectory is likely to align with the routes that they tapped on.

Methods

We implement the matching using a subset of users who allow the Transit App to continuously collect their GPS data after the query. Note that the percentage of such sessions is rather low (<= 1%). Hence, despite a relatively large original sample of sessions in which a user tapped on at least one route (>20,000), the number of sessions for which we were able to detect post-tap movement is small (<200). The trajectories were recorded in the format of [***session_id, user_id, longitude, latitude, timestamp***]. We propose a matching procedure with both simplicity and robustness by computing the point-line distance on trajectories with spatial displacements (>= 500m). In addition to the minimum spatial displacement condition, the trajectories were collected only between the time that the user planned the trip for which a route was tapped and the estimated arrival time with the longest duration. We then determined whether the subsequent GPS point was within 50m of the tapped route. Considering most of the GPS sequences are non-continuous and rather scattered (see Fig. 12), we set the minimum number of matching points as 2 and the maximum point-to-line distance as 25m. The bus and rapid transit routes data are from

MassGIS data and the layers were developed by the Central Transportation Planning Staff (CTPS) of the Boston Region Metropolitan Planning Organization (MPO)[2].

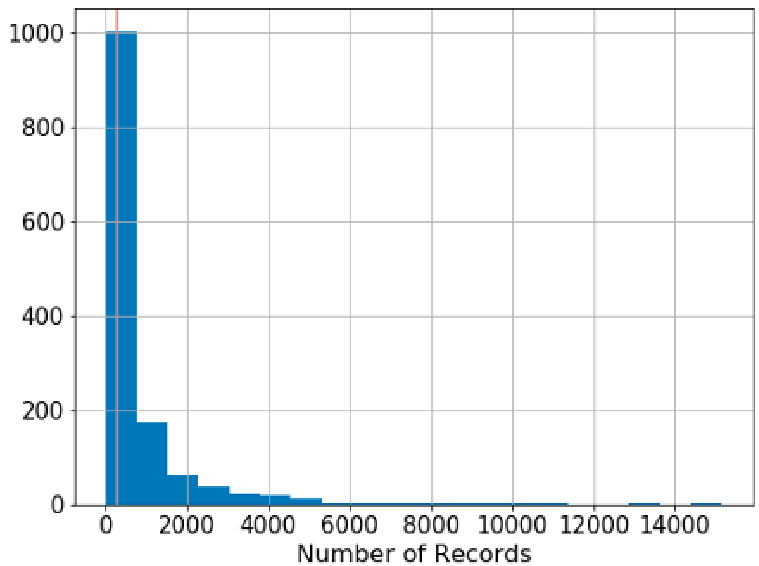

Fig. 10: Mobility Data Statistics for Users with at least One Trip View Taps

Results

Fig. 10 shows the histogram of the number of GPS records following the tap on specific routes within the same Transit App usage session, and it has a long tail distribution. This is suggesting that there are only very few users whose GPS records can be used for map matching purposes. We first filter to users who have at least 1,000 GPS points across the study period and then compute accumulated spatial displacement between every two GPS records at each session. Similarly, the displacement variable also has a heavy tail pattern that is shown in Fig. 11. This indicates that only within a small fraction of sessions, a user not only reports GPS sequence but also makes movements. After setting the 500m as the displacement threshold, only 181 queries were left.

Out of the 181 candidate queries, there are 76 in which we can observe that the user occurs at another location on the public transit route the user has tapped on with displacement validated

[2] https://www.mass.gov/orgs/massgis-bureau-of-geographic-information

(illustrated in Fig. 12). This was determined by computing the minimum point-to-line distance with a threshold of 25m. This gives us a 41.9% matching rate that is lower than previous studies and there are several plausible reasons behind this performance gap. First, we can only match with public transit routes. This means if one uses TNC services, we would not be able to validate unless the user provides continuous location sharing during their TNC ride. Second, the sparsity of the GPS data is very severe due to the nature of transportation navigation App usage and reduces the candidate trajectories significantly as a result. Finally, the perturbation in ridership during COVID-19 could also play an important role given that people can work from home and make more flexible travel plans.

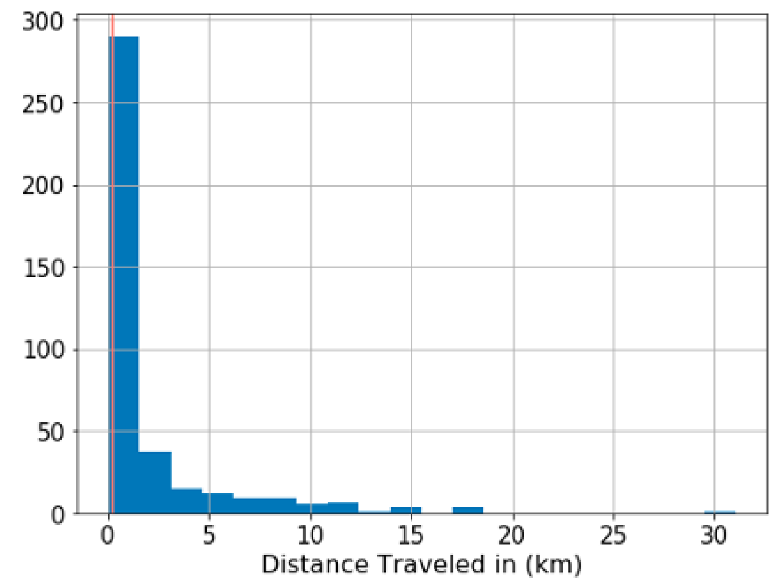

Fig. 11: Distance Traveled by Users after Trip Taps

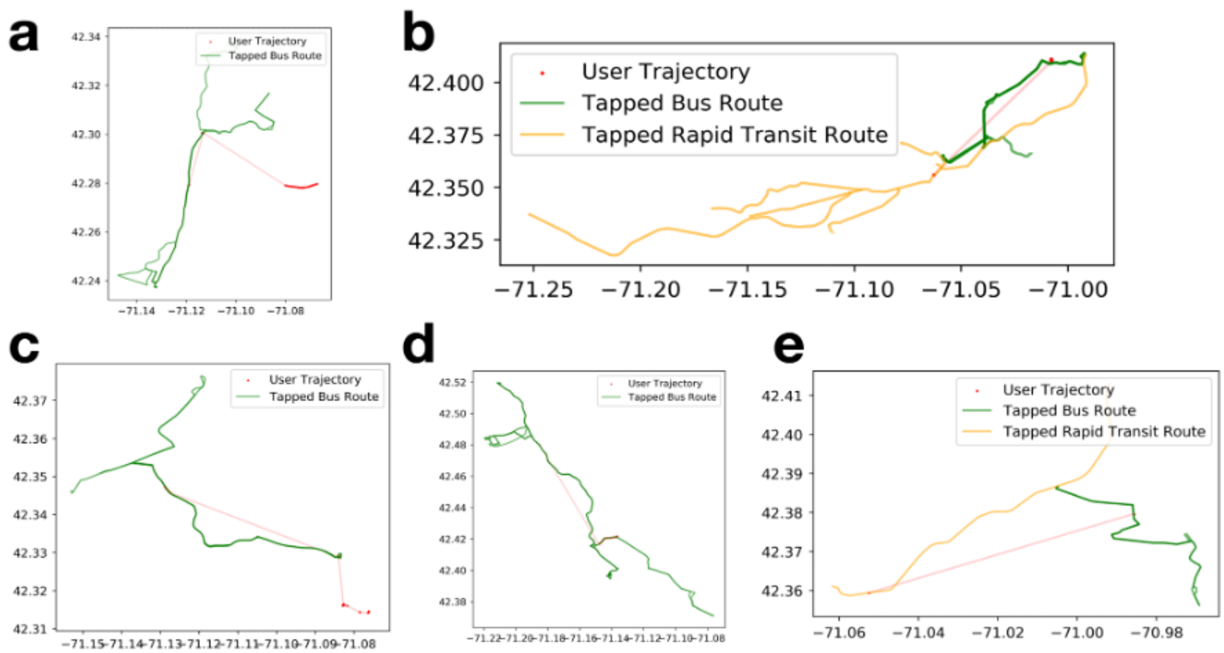

Fig. 12: Example of Map Matching Results

REFRENCES

Bedogni, L., Di Felice, M., & Bononi, L. (2016). Context-aware Android Applications through Transportation Mode Detection Techniques. *Wireless Communications and Mobile Computing, 16*(16), 2523-2541.

Feng, J., Li, Y., Zhang, C., Sun, F., Meng, F., Guo, A., & Jin, D. (2018). Deepmove: Predicting human mobility with attentional recurrent networks. *Proceedings of the 2018 world wide web conference*, (pp. 1459-1468).

Freund, Y., & Schapire, R. E. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences, 55*(1), 119-139.

Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R news, 2*(3), 18-22.

Maher Atia, M., Hilal, A. R., Stellings, C., Hartwell, E., Toonstra, J., Miners, W. B., & Basir, O. A. (2017). A Low-Cost Lane-Determination System Using GNSS/IMU Fusion and HMM-Based Multistage Map Matching. *IEEE Transactions on Intelligent Transportation Systems, 18*(1), 3027-3037.

Mardia, K. V. (1974). Applications of Some Measures of Multivariate Skewness and Kurtosis in Testing Normality and Robustness Studies. *Sankhyā: The Indian Journal of Statistics, Series B*, 115--128.

Massachusetts Department of Public Utilities, T. D. (n.d.). *2020 Rideshare Data Report*. Retrieved from https://www.mass.gov/info-details/2020-rideshare-data-report

Obradovic, D., Lenz, H., & Schupfner , M. (2006). Fusion of Map and Sensor Data in a Modern Car Navigation System. *Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology, 45*(1), 111-122.

Quddus, M. A., Noland, R. B., & Ochieng, W. Y. (2006). A High Accuracy Fuzzy Logic Based Map Matching Algorithm for Road Transport. *Journal of Intelligent Transportation Systems, 10*(3), 103-115.

Schölkopf, B., & Smola, A.J. (2002). Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT press.

Shin, D., Aliaga, D., Tunçer, B., Arisona, S. M., Kim, S., Zünd, D., & Schmitt, G. (2015). Urban sensing: Using Smartphones for Transportation Mode Classification. *Computers, Environment and Urban Systems, 53*, 76-86.

Stenneth, L., Wolfson, O., Yu, P. S., & Xu, B. (2011). Transportation Mode Detection using Mobile Phones and GIS Information. *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, (pp. 54-63).

White, C. E., Bernstein, D., & Kornhauser, A. L. (2000). Some Map Matching Algorithms for Personal Navigation Assistants. *Transportation Research Part C: Emerging Technologies, 8*, 91-108.