NYC Taxi: PageRank Pickup Zone Recommendation

Laurent Aeschbach, Karn Agarwal, William Coningsby, David Kaczmarkiewicz, John Torossian
{laeschbach3@gatech.edu, kagarwal73@gatech.edu, wconingsby3@gatech.edu, djk3@gatech.edu, jtorossian3@gatech.edu}

INTRODUCTION AND MOTIVATION

In New York City (NYC) taxis are heavily used modes of transit. Prior to the COVID-19 pandemic, there were over 200,000 taxi trips per day within the city. The New York City Taxi and Limousine Commission (TLC) publishes data for nearly all taxi trips taken in New York since 2009. The commission data lists each trip with a pick-up and drop-off zone. App based ride hailing services (e.g. Uber) use real time data to direct drivers to the most profitable trips. While there are mobile applications available for hailing yellow taxis, they are lightly used due to poor user adoption. Taxi drivers are left using their own intuition and experience to determine the best location to find their next fare.

Problem Definition

Using the data provided by the TLC, we will build a tool to aid taxi drivers to determine where they should go to obtain the most profitable fare. We will create a weighted directed graph of all taxi zones and use a PageRank algorithm to determine the best taxi zone in which a driver can find a street hail based on expected demand, expected fare, and travel time from their current location.

LITERATURE SURVEY

MI: Main Idea:
US: Why It's Useful
PS: Potentials Shortcomings

[1]: **MI:** DeepWalk is a novel approach for learning latent social representations of vertices. **US:** On large graphs, the method significantly outperforms other methods designed to operate for sparsity.
PS: Sparsity enables the design of efficient discrete algorithms but can make it harder to generalize in statistical learning.

[2]: **MI:** Learning with graph structured data, requires effective representation of their graph structure. **US:** GNNs have achieved state-of-the-art performance in many tasks such as node classification, link prediction, and graph classification. **PS:** There is little theoretical understanding of the properties and limitations of GNNs, and formal analysis of GNNs' representational capacity is limited.

[3]: **MI:** Neural Networks that can accept graphs and learn predictive functions. **US:** DGCNN algo for graph classification. **PS:** The lack of ordered tensor representations limits the applicability of neural networks on graphs.

[4]: **MI:** This article discusses how GNNs compute useful representations of nodes. **US:** The key idea is to train GNNs to predict local and/or global graph properties. **PS:** There may exist more powerful techniques for 'pooling' together node representations.

[5]: **MI:** The authors consider the problem of classifying nodes in a graph, where labels are only available for a small subset of nodes. **US:** GCN model is capable of encoding both graph structure and node features in a way useful for semi-supervised classification **PS:** Strong assumption that connected nodes in the graph are likely to share the same label

[6]: **MI:** The authors demonstrate this model using a graph convolutional network (GCN) encoder and a simple inner product decoder. **US:** This model makes use of latent variables and is capable of learning interpretable latent representations for undirected graphs. **PS:** Hard to find best-suited prior distributions, not flexible generative models and lacks scalability.

[7]: **MI:** The graph extensions of autoencoders (AE) and variational autoencoders (VAE) recently appeared as state-of-the-art approaches for link prediction in numerous experimental analyses. **US:** The authors propose to extend the gravity-inspired method to the graph variational autoencoder framework. **PS:** In the gravity inspired decoding scheme, the models have a quadratic time complexity $O(dn^2)$ with regards to the number of nodes in the graph, as standard graph AE and VAE.

[8]: **MI:** Presentation of the family of spectral clustering algorithms. **US:** Spectral clustering is very simple to implement. **PS:** It separates one individual vertex from the rest of the graph.

[9]: **MI:** The authors evaluate the performance of the kernel and compare it with state-of-the-art graph kernels in terms of runtime, scalability, and prediction accuracy. **US:** Approach is not limited to being used in graph kernels but can be applied in graph mining. **PS:** The kernel computational complexity scales exponentially with graph size.

[10]: **MI:** The authors have considered similarity measures on graphs based upon three fundamental graph matrices: the adjacency matrix, the Laplacian matrix, and the Markov matrix. **US:** Approach can result in interesting new perspective of graph data, help with clustering based on the chosen similarity metric. **PS:** The appropriate similarity measure may be hard to define, different algorithms give different embeddings.
[11]: **MI:** General overview of unsupervised learning, spectral clustering and Kernel PCA, non-linear dimensionality reduction. **US:** Cluster data based on a similarity measure, dimensionality reduction for visualization. **PS:** General methods may not be appropriate to our specific dataset.
[12]: **MI:** Representing graphs through adjacency matrix, similarity measures and kernel functions, relation between gram matrix and adjacency matrix, graph Laplacian, low dimensional embedding of graphs via eigen decomposition of normalized graph Laplacian, ISOMAP. **US:** Graph embedding of data for visualization and downstream tasks for prediction. **PS:** The appropriate similarity measure may be hard to define, different algorithms give different embeddings.
[13]: **MI:** The authors have examined the problem of detecting community structure in networks. **US:** Method gives a spectral algorithm for community detection. **PS:** Finding the modularity matrix may not be trivial for most problems.
[14]: **MI:** The authors provide a thorough review of currently available KG embedding techniques. **US:** This article provides a systematic review of currently available techniques. **PS:** Needs contextual information e.g., entity types, relation paths, textual descriptions, as well as logical rules.
[15]: **MI:** This paper presents a spectral analysis of New York City taxi data via the graph Fourier trans-form. **US:** Taxi rides representation constrained by an underlying road network. **PS:** Underlying road network is large, directed, and sparse.

METHOD

The method consists of three steps:

1. Assign a score to each dropoff zone based on its popularity.
2. Use the scores to find the best pickup zones by calculating a weighted profit for each pickup zone.
3. Create a visualization showing recommended pickup zones.

Step 1 Dropoff Zone Popularity

Let's consider the taxi network as a graph whose nodes are the taxi zones and the edges between them have a weight given by the number of trips between a given pair of nodes in the given time frame (a month for example). For the purposes of PageRank we consider the graph as a directed graph where the edge $e_{i,j} \neq e_{j,i}$. Thus, the adjacency matrix of the graph is not symmetric.

Consider a taxi driver starting out at a random zone i who then randomly decides to drive to the next zone j. Out of all possible zones j the taxi can drive to, we can use the actual number of trips starting from i and ending at j as a proxy for the probability of driving to j. Upon arriving at j, the taxi repeats the process of randomly driving to a new zone k. Let's consider the taxi repeats this process forever. We can model the above random walk process as a first order Markov chain where the probability of driving to the next zone only depends on the current zone the taxi is in. Let X and Y be random variables where X denotes the zone at time t and Y the zone at time $t+1$. Here time is a discrete integer $t \in \{0,1,2,\dots\}$ indexing the location of taxi in a zone. Let $N_{i,j}$ be the number of trips originating at i and ending at j. Then we define the probability of $i \to j$ as follows

$$\Pr[Y = j | X = i] = \frac{N_{i,j}}{\sum_j N_{i,j}} = p_{i,j}$$

Equation 1

The entries $p_{i,j}$ as defined in Equation 1 then form a state transition matrix. Using this matrix, we can calculate the probability of ending up in zone j at time $t+1$ as follows

$$\Pr[Y = j] = \sum_i \Pr[Y = j | X = i] \cdot \Pr[X = i]$$

$$\Rightarrow y_j = \sum_i p_{i,j}\, x_i$$

Equation 2

In Equation 2 above y_j denotes the probability of ending up in zone j at $t+1$ and x_i denotes the probability of starting at zone i at time t. We can also write the above equation as a matrix vector product as follows

$$y_j = \sum_i p_{i,j}\, x_i \Longrightarrow y = P^\top x$$

Equation 3

Equation 3 above represents one transition from time t to $t + 1$. Since we want to analyze the taxi's path over many time steps, we can write the steady state distribution, $x*$, which is the limit of $x(t)$ as t goes to infinity:

$$\lim_{t\to\infty} x\,(t) = \mathrm{x}^* \equiv [x_i^*]$$

Equation 4

PageRank aims to find the steady state probability distribution of ending up in any given zone after many time steps. We can find x^* by finding the leading eigenvector of the transpose of the state transition matrix $P^\top$ corresponding to the eigenvalue of 1. Our hypothesis is that this steady state distribution gives a measure of the popularity of the drop-off zones. In other words which zones are you most likely to end up in after a number of taxi trips throughout the day.

Step 2 Weighted Profit for Pickup Zones

The next step is to calculate a weighted profit for each pickup zone as a measure of profitability of that zone. Let $T_{i,j}$ be the total amount that a taxi makes in going from zone i to zone j over a given time frame (a month for example). We propose to apply a weight factor w_j to the $T_{i,j}$ based on the importance of the destination zone . This weight factor is given directly by the steady state probability of ending up in zone j as calculated using PageRank in the previous section. Thus, from Equation 4 $w \equiv x^*$. The weighted profit for pickup zone i is then calculated as follows

$$WP_i = \frac{\sum_j w_j T_{i,j}}{\sum_j N_{i,j}}$$

Equation 5

The weighted profit as computed in Equation 5 is a measure of the profitability of the
pickup zone.

Step 3 Create Visualization

The final step is to create a visualization/application showing recommended pickup zones to find the most profitable fare. The visualization has a map with the choropleth zones highlighted for the best pickup areas. The visualization has selections to filter by month, day or night, and weekend or weekday, which will then update the choropleth map.

The driver will be able to select their current zone to obtain an updated map showing the best local pickup zones based on the time remaining in their "shift" and the amount of time they're willing to travel to obtain the next fare. An example of the visualization is shown in the images below.

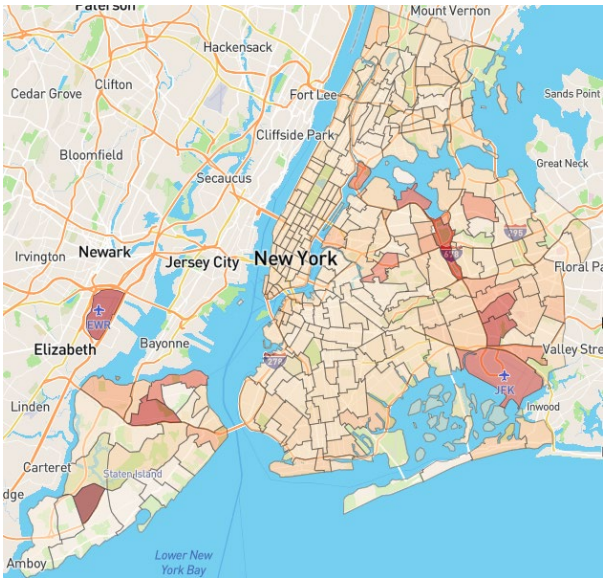


Image 1: Overall Interactive Choropleth Map

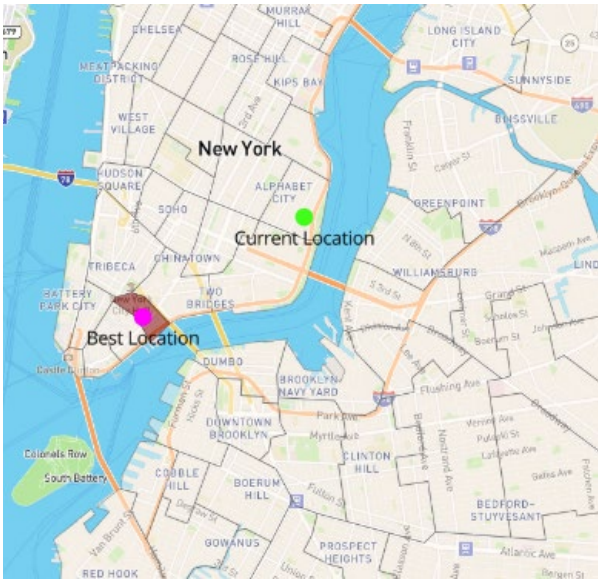


Image 2: Example of Location Recommendation

Summary of Proposed Innovation

Innovation 1: We explored a novel way of determining the best pickup zones using PageRank. To evaluate the algorithm we formulated a random walk simulation to estimate the total amount a driver can make after a specified number of trips. Evaluation shows that it is better than random selection of pickup zones.

Innovation 2: Our visualization and the application itself is tailored to directly benefit the taxi driver themselves. The landing page as well as the zoomed-in location-dependent portion of the application will show where our algorithm predicts the best zone to pick up passengers to maximize the drivers fare. Other work done using this dataset simply looked at aggregating the dataset and letting users explore the data. Our visualization allows users (taxi drivers) to see and understand the output and recommendations of our analysis.

EXPERIMENTS/EVALUATION

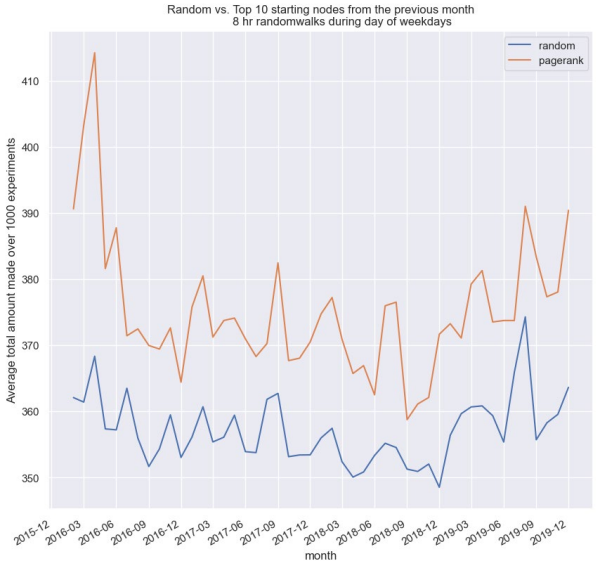
Evaluating PageRank

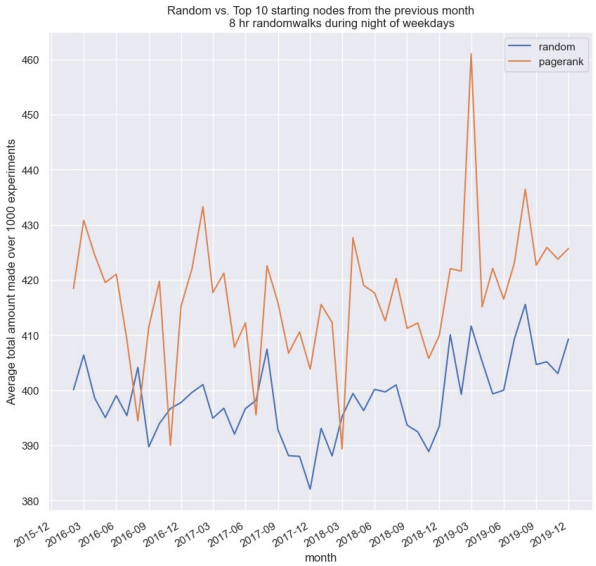
To evaluate the PageRank algorithm we set up a simulation where we compared the average amount a driver can make starting his day in one of the top 10 pickup zones as recommended by our algorithm vs. a random starting zone. We ran 1,000 experiments for each where each experiment consists of an 8 hour random walk. We compared this for the historical dataset from 2016 to 2019.

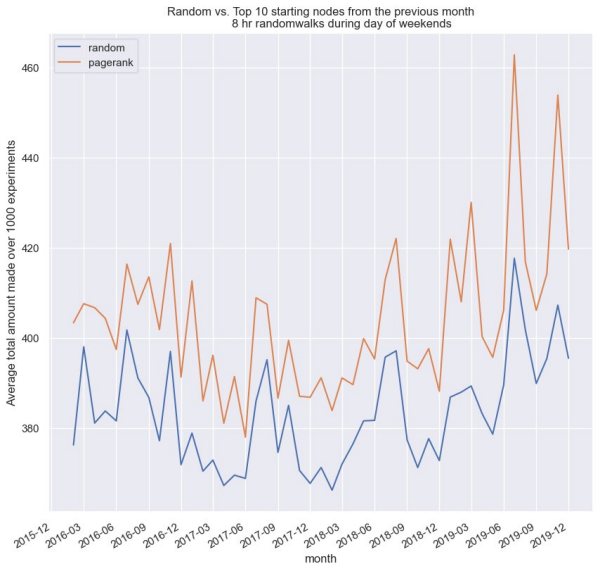
As shown in the figure below, on average the expected amount that can be made from a pickup zone recommended by our algorithm is more than a random selection. The functionality of finding the best neighboring pickup zones given a current zone is best demonstrated through our visualization. The following table shows the average earnings per 8 hour shift based on 2019 data by day of week and time of day.

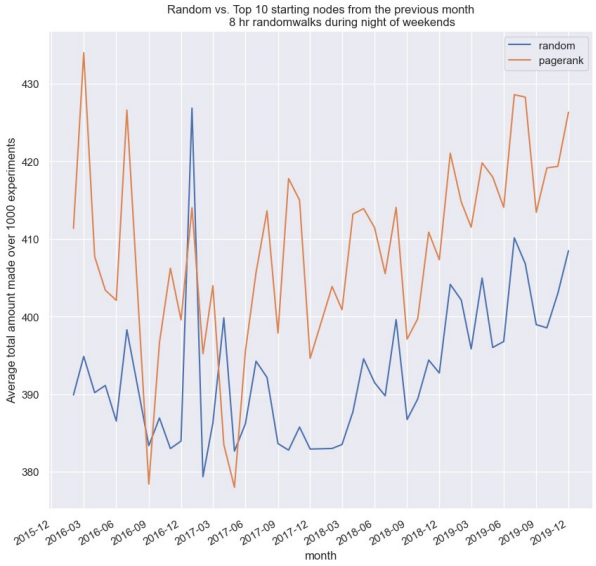
	Baseline	Model	Increase
Weekday Day	$360.80	$378.84	5.00%
Weekday Night	$406.05	$426.33	4.99%
Weekend Day	$393.62	$419.64	6.61%
Weekday Night	$402.14	$419.52	4.32%

The graphs to the right and on the following page show the average amount made over 8 hours across 1000 simulations for each combination of weekday/weekend and day/night.

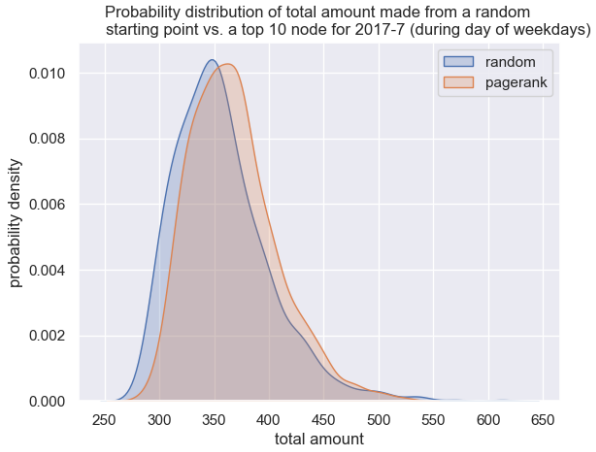






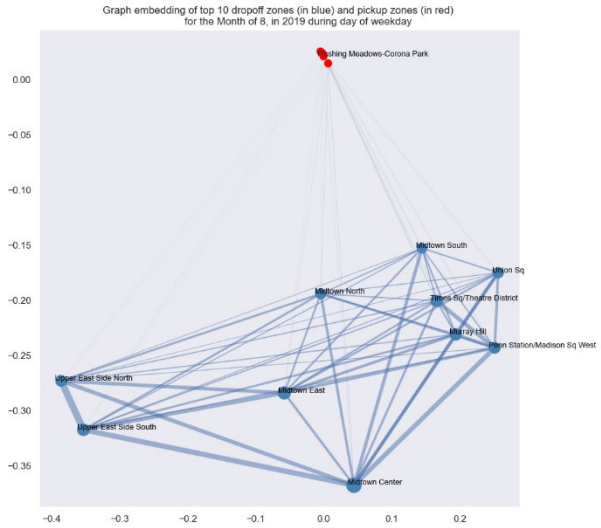


The following graph shows the probability density of earning amounts for a typical month for weekday days. Other months, weekday/weekend, and day/night combinations show a similar distribution.



Comparison with other methods

We explored other models including a two-dimensional embedding approach in order to identify the best zones, but this approach did not result in an improvement in a driver's earnings. An example of a graph embedding is shown in the column to the right.



CONCLUSIONS AND DISCUSSION

Taxi drivers can make more per shift by using our PageRank pickup zone recommendations. This is significant as there is increasing pressure on taxi drivers due to the rise of app hailing services and the impact of the COVID-19 pandemic. Drivers are under economic pressure to improve their earnings per hour. Using a data model based on finding the most profitable zones will help drivers increase their earnings as they manage an overall decreasing demand for taxi trips.

DISTRIBUTION OF EFFORT

All team members contributed a similar amount of effort. Each team member contributed to all task areas.

REFERENCES

[1] Perozzi, B., Al-Rfou, R., & Skiena, S. (2014, August). Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 701-710).

[2] Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2018). How powerful are graph neural networks?. *ICLR (2019)*.

[3] Zhang, M., Cui, Z., Neumann, M., & Chen, Y. (2018, April). An end-to-end deep learning architecture for graph classification. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[4] Daigavane, A., Ravindran, B., & Aggarwal, G. (2021). Understanding Convolutions on Graphs. *Distill*, *6*(9), e32.

[5] Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *ICLR (2019)*.

[6] Kipf, T. N., & Welling, M. (2016). Variational graph auto-encoders. *NIPS Workshop on Bayesian Deep Learning (2016)*

[7] Salha, G., Limnios, S., Hennequin, R., Tran, V. A., & Vazirgiannis, M. (2019, November). Gravity-inspired graph autoencoders for directed link prediction. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (pp. 589-598).

[8] Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and computing*, *17*(4), 395-416.

[9] Shervashidze, N., Vishwanathan, S. V. N., Petri, T., Mehlhorn, K., & Borgwardt, K. (2009, April). Efficient graphlet kernels for large graph comparison. In *Artificial intelligence and statistics* (pp. 488-495). PMLR.

[10] Avrachenkov, K., Chebotarev, P., & Rubanov, D. (2019). Similarities on graphs: Kernels versus proximity measures. *European Journal of Combinatorics*, *80*, 47-56.

[11] Hastie, T., Tibshirani, R., & Friedman, J. (2021). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition (Springer Series in Statistics)* (2nd ed.) Chapter 14. Springer.

[12] Vlachos, M., Domeniconi, C., Gunopulos, D., Kollios, G., & Koudas, N. (2002, July). Non-linear dimensionality reduction techniques for classification and visualization. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 645-651).

[13] Newman, M. E. (2006). Modularity and community structure in networks. *Proceedings of the national academy of sciences*, *103*(23), 8577-8582.

[14] Wang, Q., Mao, Z., Wang, B., & Guo, L. (2017). Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, *29*(12), 2724-2743.

[15] Deri, J. A., & Moura, J. M. (2015, November). Taxi data in New York city: A network perspective. In *2015 49th asilomar conference on signals, systems and computers* (pp. 1829-1833). IEEE.