Real-World Data Driven Characterization of Urban Human Mobility Patterns

Abhinav Jauhri
ECE Department
Carnegie Mellon University

Thesis Committee:
John Paul Shen (Chair)
Anupam Datta
Jason Hong
Sean Qian (CMU-CEE)

What is known about urban human mobility?

- Human mobility properties shown repetition - *Individuals display significant regularity [Gonzalez 08], as they return to a few highly frequented locations, like home or work.*
- Aggregate individual patterns to summarize human behavior — *[Song 2010] By measuring the entropy of each individual's trajectory, we find a 93% potential predictability in user mobility across the whole user base.*

Individual patterns may not differ across urban areas but city/county specific properties will; what are those city level characterizations of urban human mobility? This is the first work which establishes characterization across more than a dozen cities from large scale datasets from the real world.

Urban Human Mobility Facts

- **Driver efficiency —** In 2020, Los Angeles recorded ~45 hours are wasted on average in congestion per vehicle in a year.
- **Potential Intervention —** In 2020, production of car sales plummeted to 2011 levels but in the 2021 Tesla will manufacture ~1.2%, *capable of self-driving,* of the global production.

Is it possible to improve driver efficiency, and rider experience, and intervene in an automated environment with thorough characterization of urban human mobility?

Image Credits: pixabay.com

Thesis Statement

Based on analysis of real-world datasets from ride-sharing services, we want to understand and characterize human mobility patterns at the city scale, and gain insights on how to improve ridership experience and overall system/service efficiency.

Thesis Structure

Sections	Purpose
• **Urban Human Mobility Characterization (Chapters 2, 3)** Q1. What are the city level patterns? Q2. How to capture the spatial distribution and evolution?	1. For validation of datasets 2. To understand tradeoffs for a policy
• **Synthetic Generation of New Datasets (Chapter 4)** Q3. How to generate synthetic data for any urban area?	1. For city planning simulations 2. For what-if scenarios
• **Applications & Useful Tools for human mobility (Chapters 5, 6)** Q4. What are some applications which can be aided by city level characterizations? Q5. How can the dataset be applied to different what-if scenarios?	1. To validate our characterizations 2. To highlight what-if scenarios

Summary of real-data used in this work

City	Ride Requests
Boston	625k
Chicago	930k
London	1.1M
Los Angeles	1.1M
Mexico City	1.3M
Miami	550k
New Delhi	450k
New Jersey	400k
New York	1.3M
New York (yellow cab)	2.7M
Paris	650k
Rio De Janeiro	400k
San Francisco	1M
Toronto	500k
Washington	800k

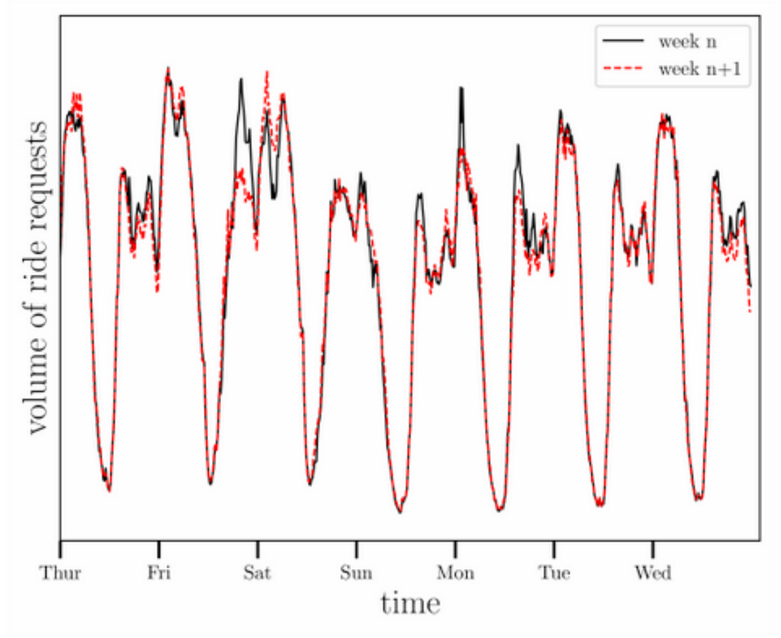
Volume of data for each city represents a typical week.

Each ride request consists -

1. Request timestamp
2. Drop-off timestamp
3. Pickup location
4. Destination location

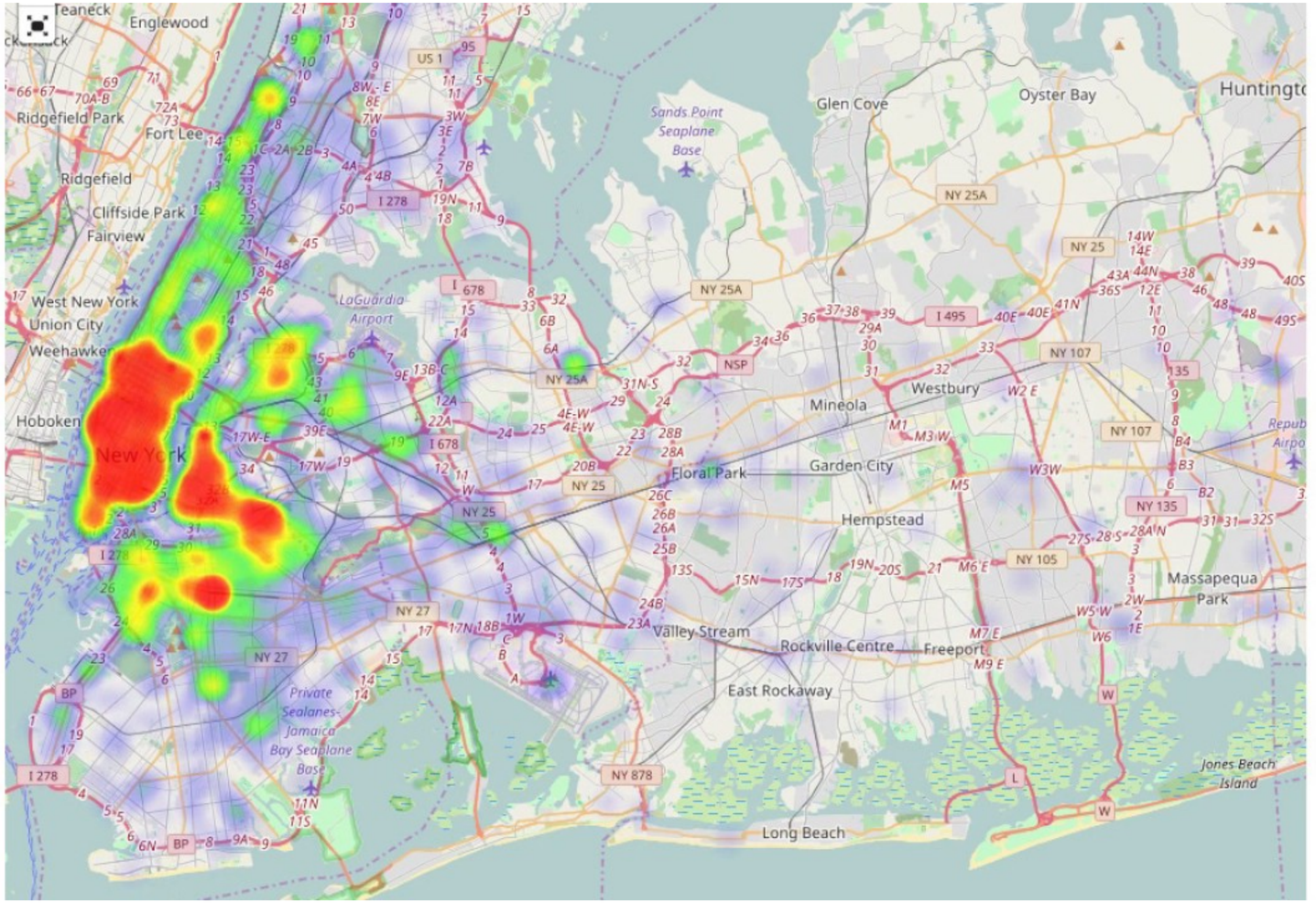
What are the city level temporal patterns?

Ride Request Data

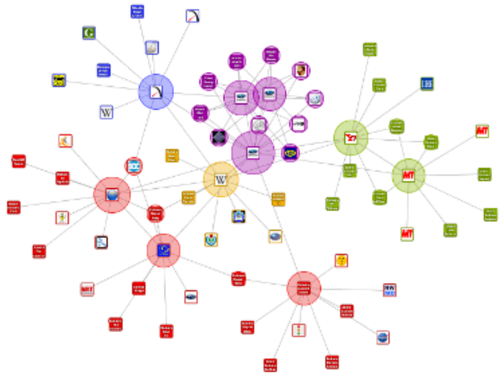


Volume of ride requests received over a week in New York.

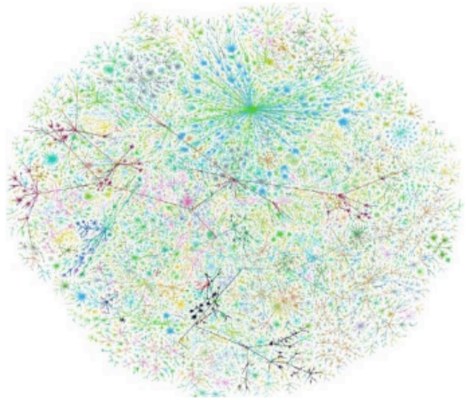
Ride Request Data



Background — Data Graphs

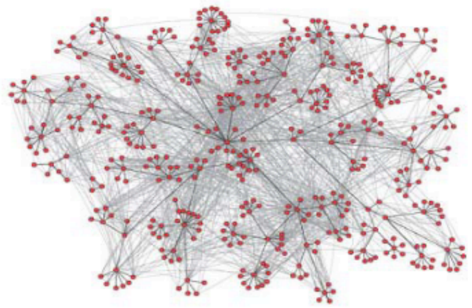


World Wide Web



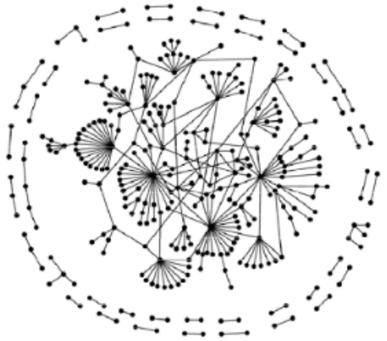
Internet

Social Networks



Communication

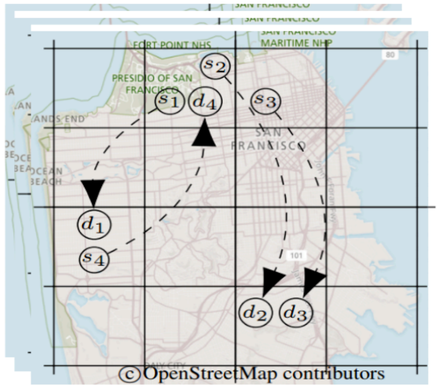
Citation



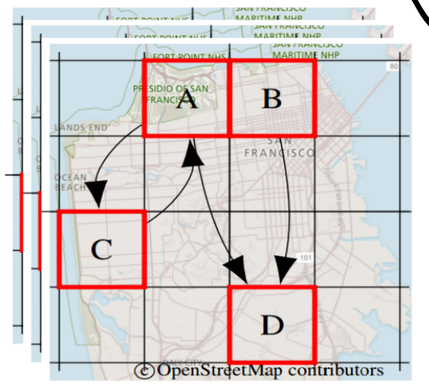
Biological Networks

Image Credits: https://www.cs.cmu.edu/~jure/pubs/thesis/jure-defense.pdf

Temporal Patterns — Ride Request Graph



Stacked sequence of ride requests overlaid on a map. Four ride requests distributed spatially over a map.

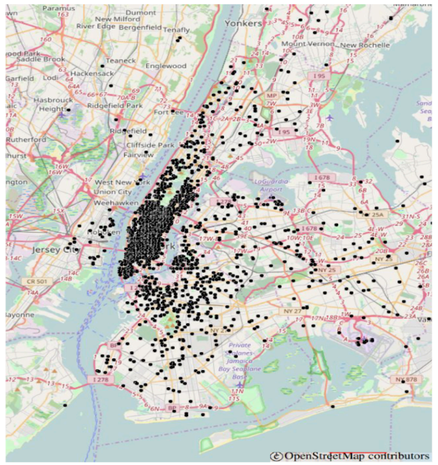


Corresponding Ride Request Graph with four nodes and directed edges.

Sequence of RRGs stacked; each RRG spans for a small window of time.

- Transformation of ride requests into a directed Ride Request Graph (RRG).
- RRGs form a quantized sequence of a succinct representation.

Temporal Patterns - Ride Request Graphs



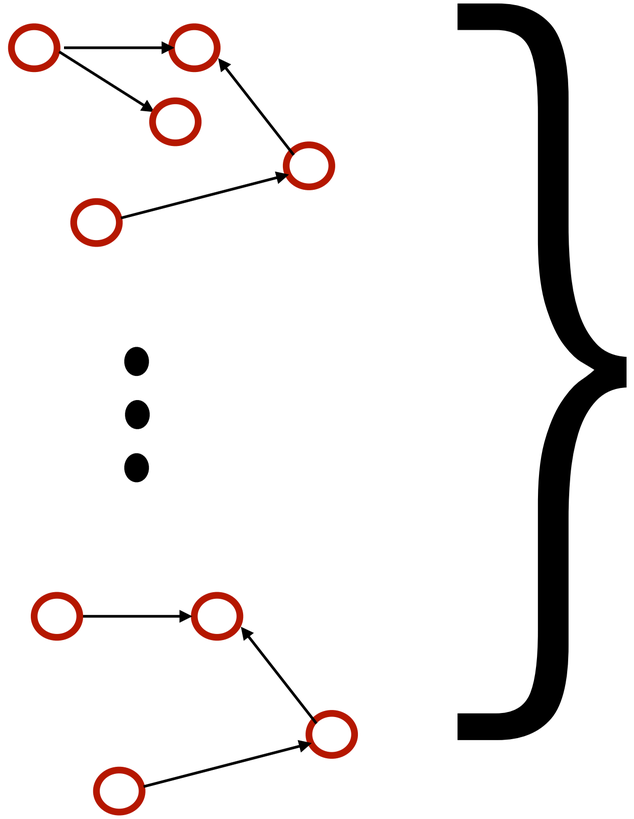
Between 8 & 8:05 pm

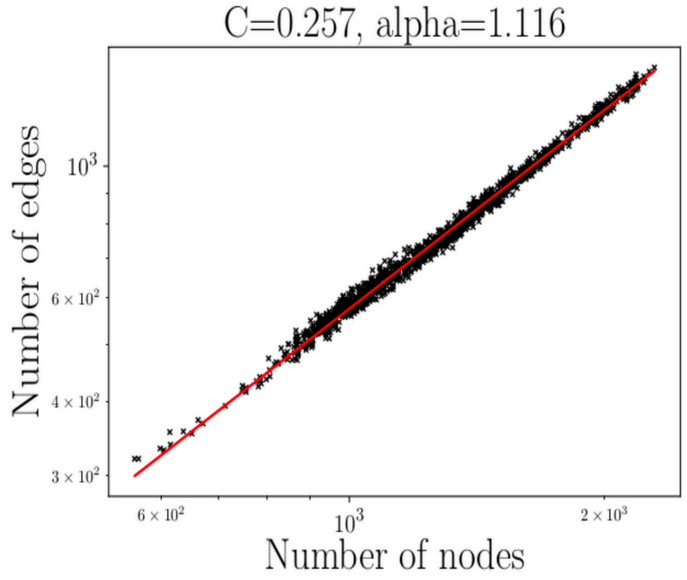
Between 3 & 3:05 am

Black dots denote either pickup or drop-off location; connections between them are not shown.

Separate graphs from different time snapshots

Ride Request Graphs

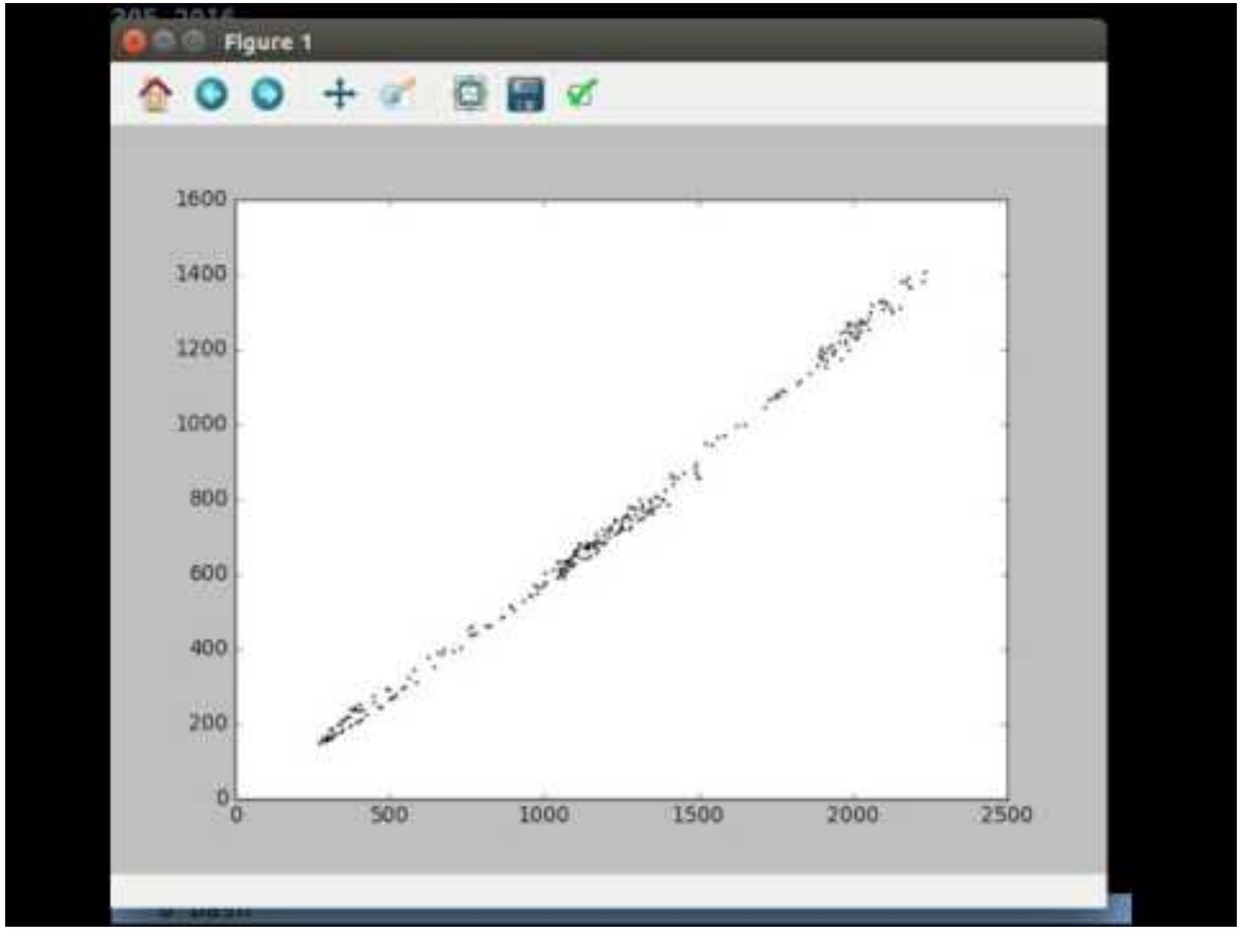




Densification Power Law (DPL): Number of edges E(t) and number of nodes N(t) for any given time t, have the following relation —

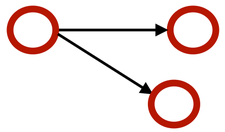
$$E(t) = C \times N(t)^{\alpha}$$

Temporal Patterns - Ride Request Graphs

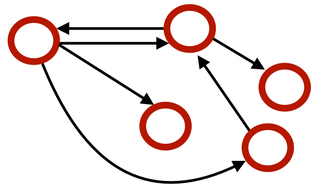


Temporal Patterns - DPL

Interpretation of exponent — Greater value of α signifies higher rate of super-linear growth in number of edges w.r.t. the number of nodes; implying more congested urban areas.



Graph 1: 3 nodes and 2 edges



Graph 2: 5 nodes and 6 edges

Notice the difference in the two graphs. Graph 1's α < Graph 2's α

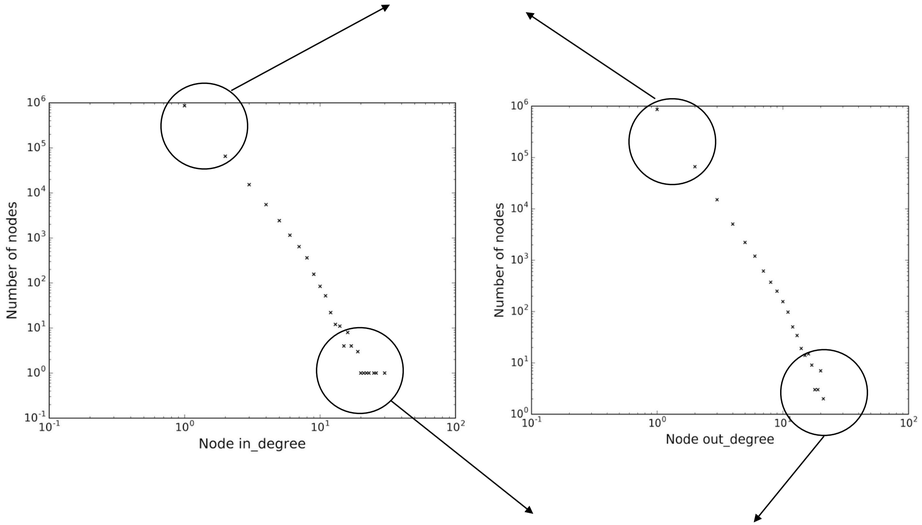
Temporal Patterns — What causes DPL?

- DPL's α is a high level metric of an underlying pattern.
- Probability distribution of degree follows a power law probability distribution —

$$p(x) = cx^{-\gamma}$$

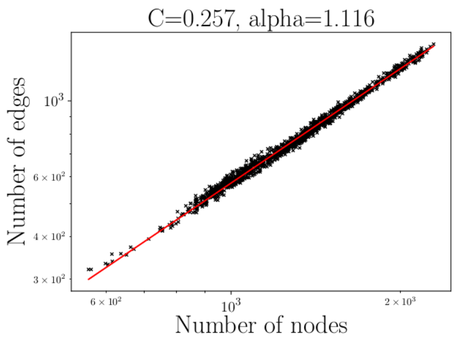
- Closed form solution to derive γ from densification power law slope α.

Lot of nodes with less degree

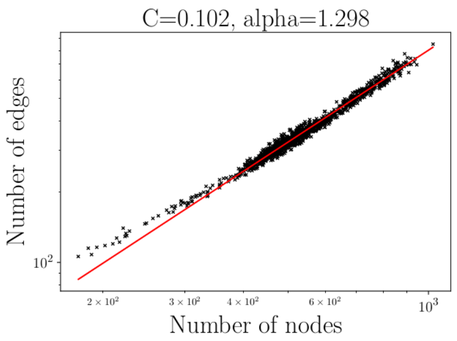


Few nodes with high degree

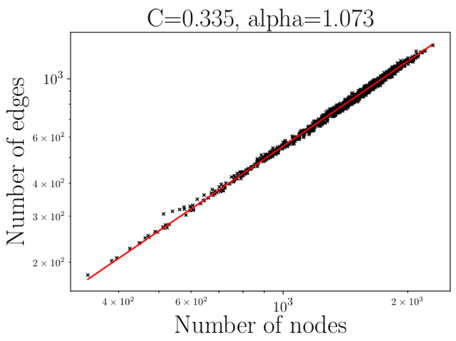
Temporal Patterns — DPL & Degree Exponent



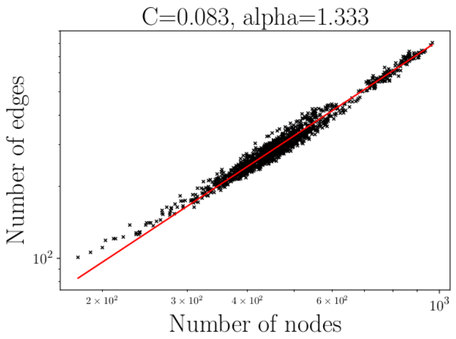
New York



Paris



Mexico City

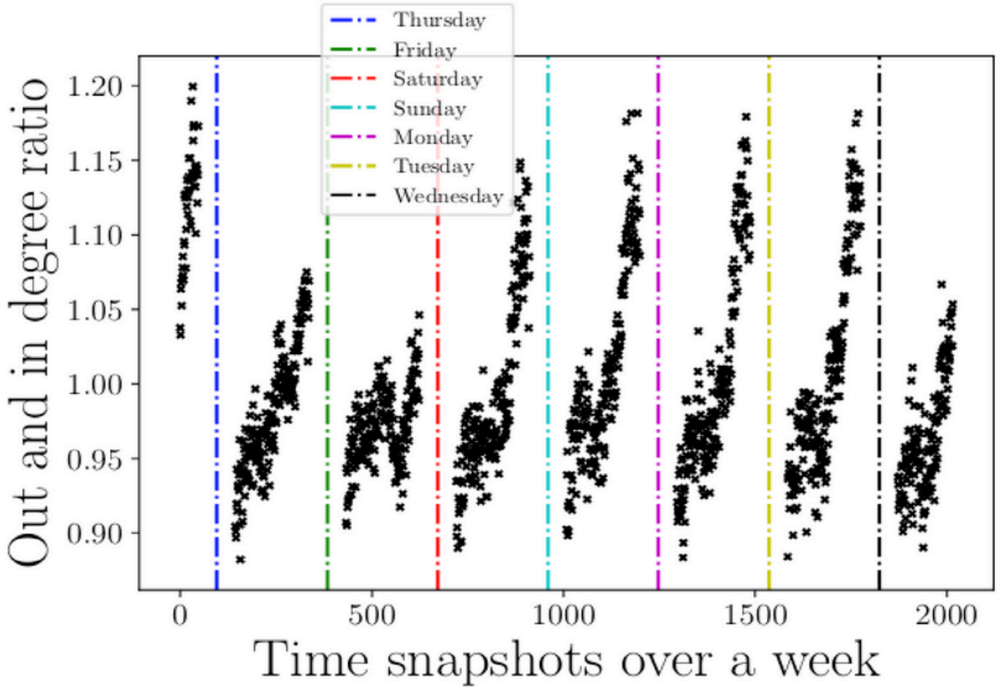


Toronto

City	α	Degree Exponent from real data (average)	Theoretical degree exponent
New York	1.116	1.853	1.792
Paris	1.298	2.037	2.084
Mexico City	1.073	1.849	1.864
Toronto	1.333	2.083	2.069

Temporal Patterns — Community Coefficient

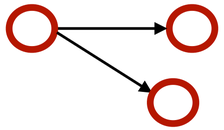
- Community Coefficient (ζ): is the ratio of the average out degree and average in degree.
  - $\zeta >= 1$: more outward movement
  - $\zeta < 1$: more inward movement
- Community Coefficient provides *directionality* of the movement.



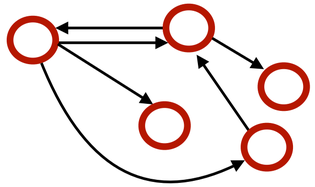
Community Coefficient for New York starting; left-most point is at 20:00 hrs on Thursday evening.

Temporal Patterns - DPL

- Interpretation of exponent — Greater value of α signifies higher rate of super-linear growth in number of edges w.r.t. the number of nodes; implying more congested urban areas.

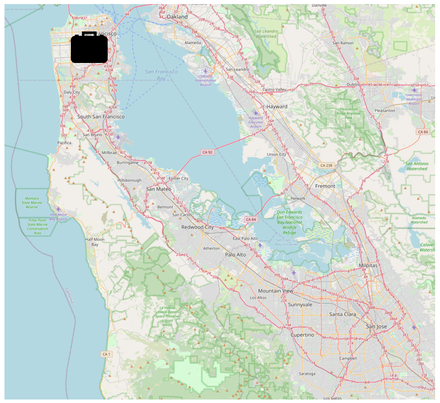


Graph 1: 3 nodes and 2 edges



Graph 2: 5 nodes and 6 edges

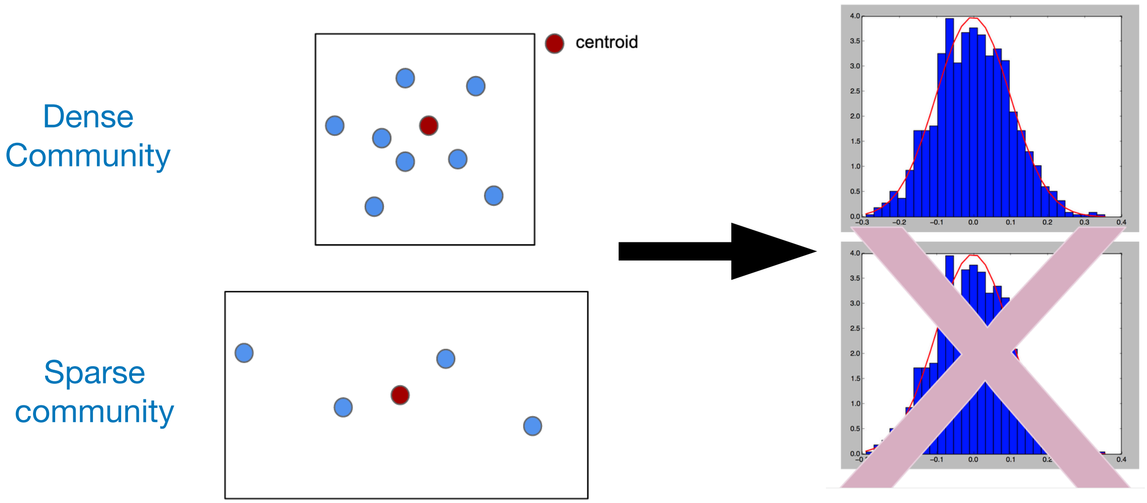
- Planning (oversimplified example)



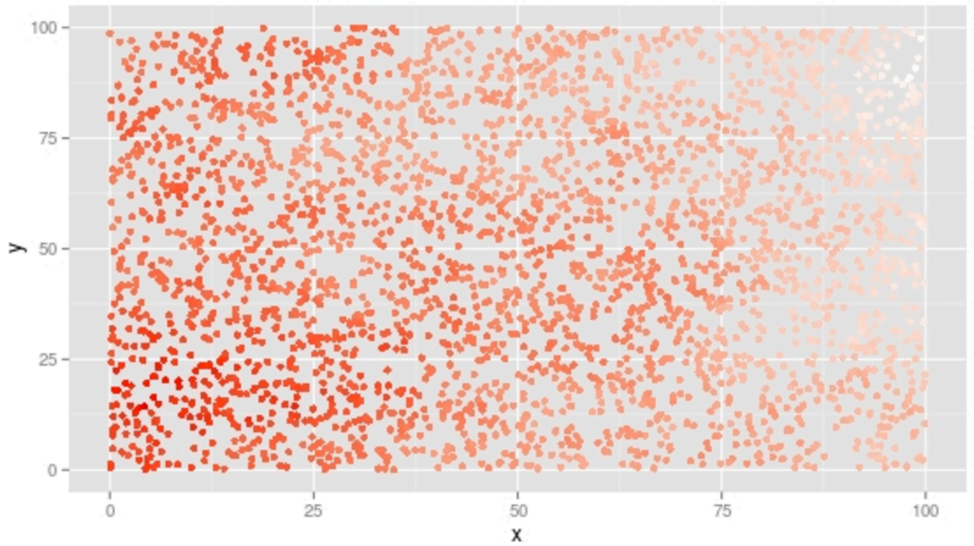
1. Plan to increase office spaces in downtown;
2. Measure the potential impact to α, γ, ζ;
3. Suggest potential ways to reduce congestion. For instance:
    1. bound on the number of people or;
    2. alternative spots
    3. ζ to check balance in the traffic

What are the city level spatial patterns?

Spatial Patterns — why is it difficult to capture?

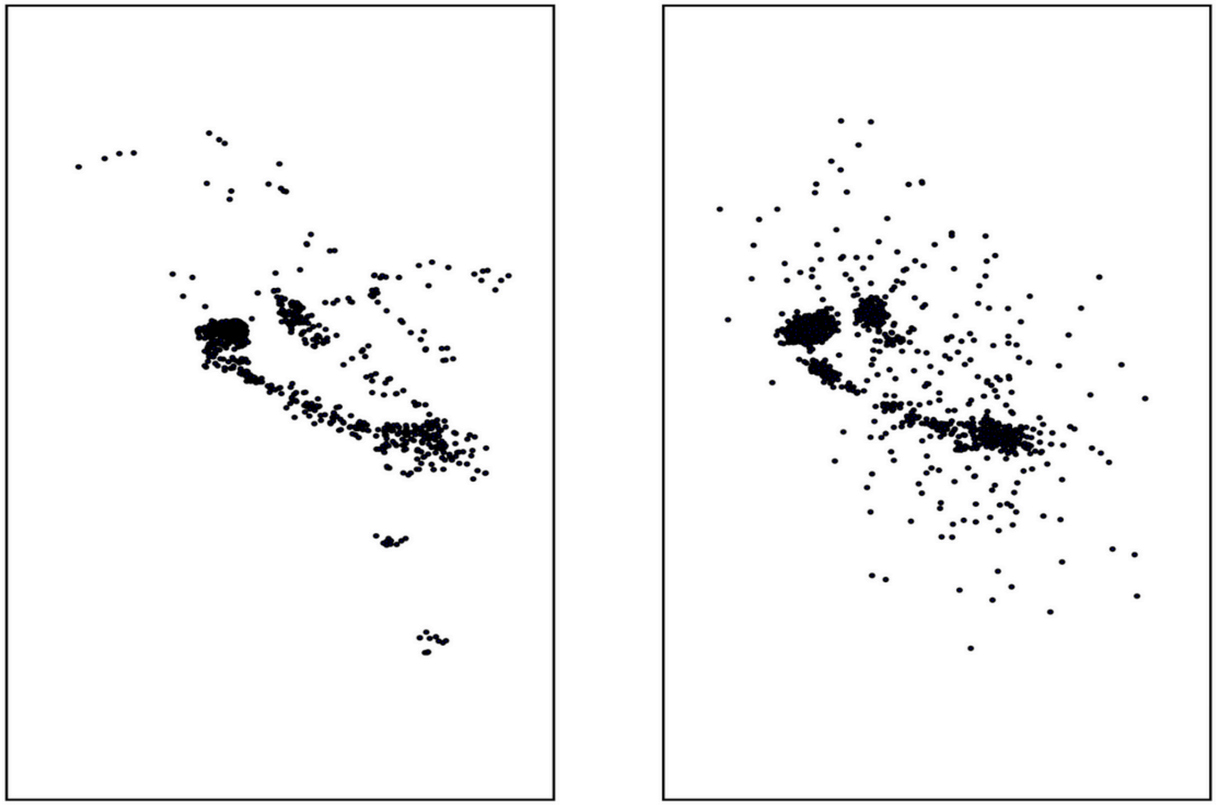


Reason #1: It is difficult to fit a distribution to clusters in sparse areas.



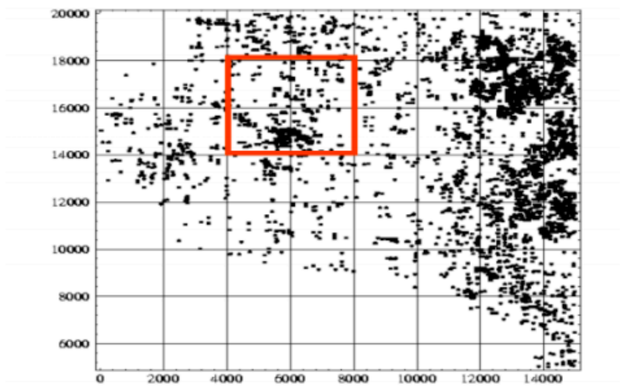
Reason #2: Techniques like spatial auto-correlation assume locations close to each other exhibit more similar values; not true with real-data.

Spatial Pattern — why is it difficult to capture?

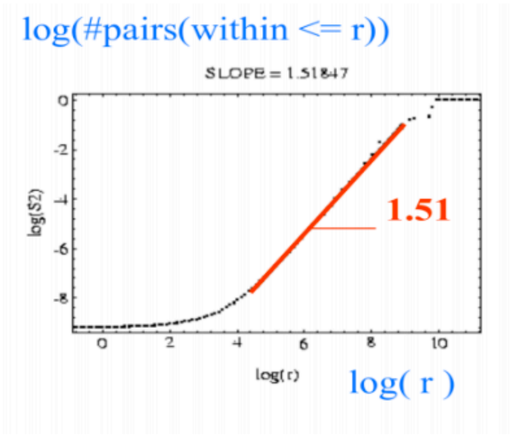


Points of requests in San Francisco from real data (left), and using synthetic data (right)

Spatial Patterns — Fractal Dimension



Every dot is an intersection



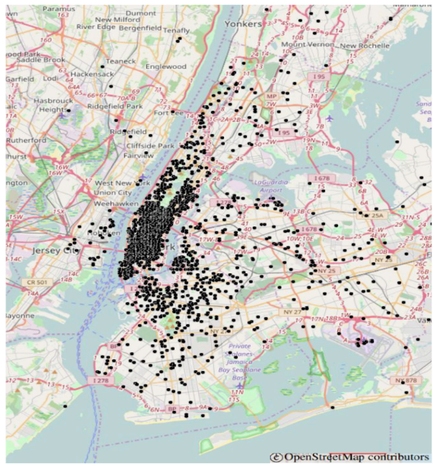
- Fractal dimension for cross roads in Montgomery county [Belussi 98].
- Measure the dispersion affect.

Spatial Patterns — Fractal Dimension

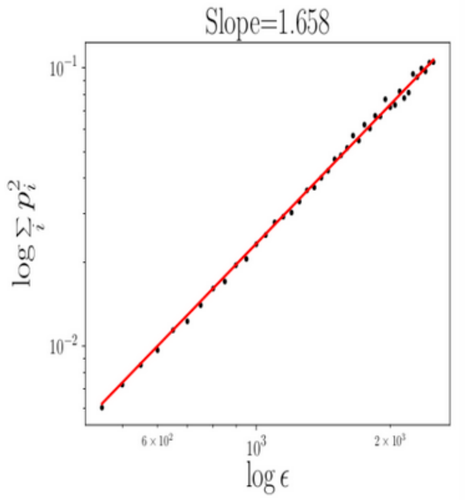
$$D_2 = \frac{\partial \log \sum_i p_i^2}{\partial \log \epsilon} = \text{constant} \qquad \epsilon \in (\epsilon_1, \epsilon_2)$$

where:

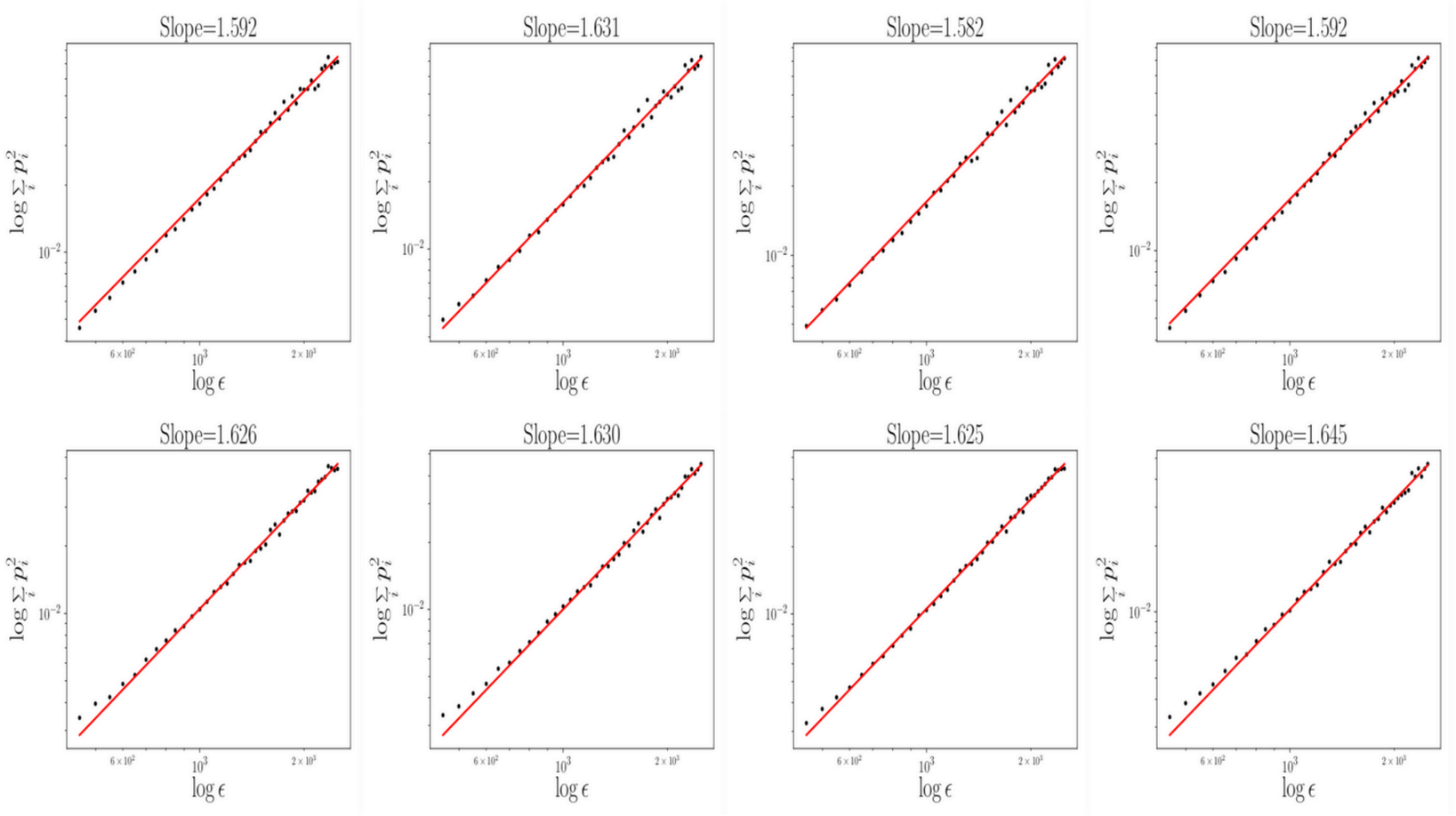
- p_i is the probability of number of points within a cell
- ϵ is the cell length



Between 8 & 8:05 pm



Spatial Variation — Fractal Dimension



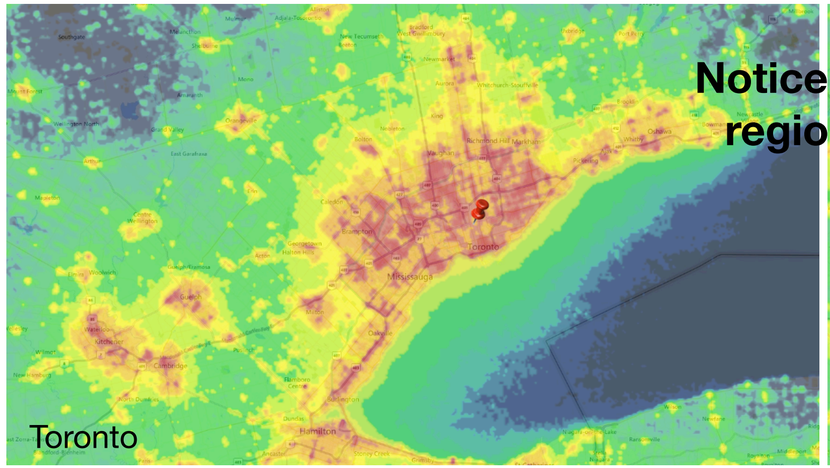
Correlation Fractal Dimension using yellow cab dataset for New York for four consecutive time snapshots each spanning 300 seconds. Top row constructed with pick-up points, and bottom row using drop-off points.

Spatial Variation — Fractal Dimension

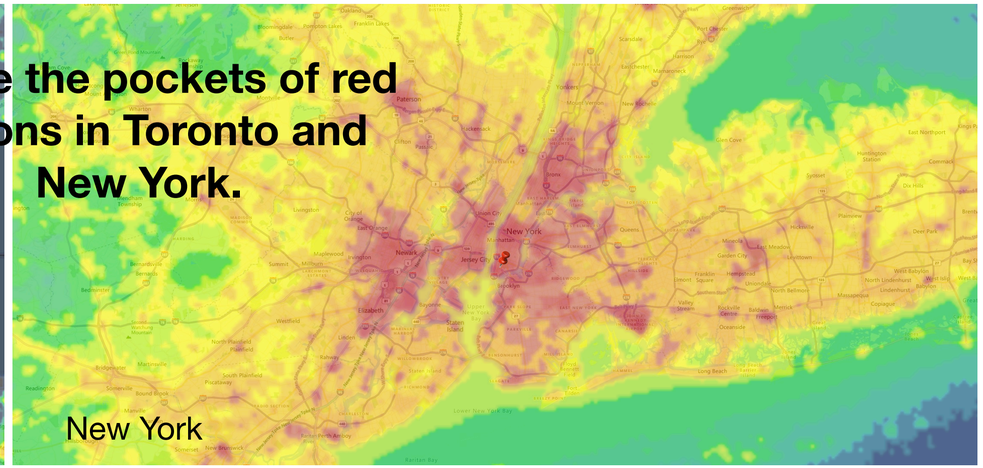
City	D_2 mean	Fractal Range (m.)
New York	1.457	(450, 2500)
Mexico City	1.529	(600, 2500)
Paris	1.586	(900, 4000)
Toronto	1.292	(500, 2500)

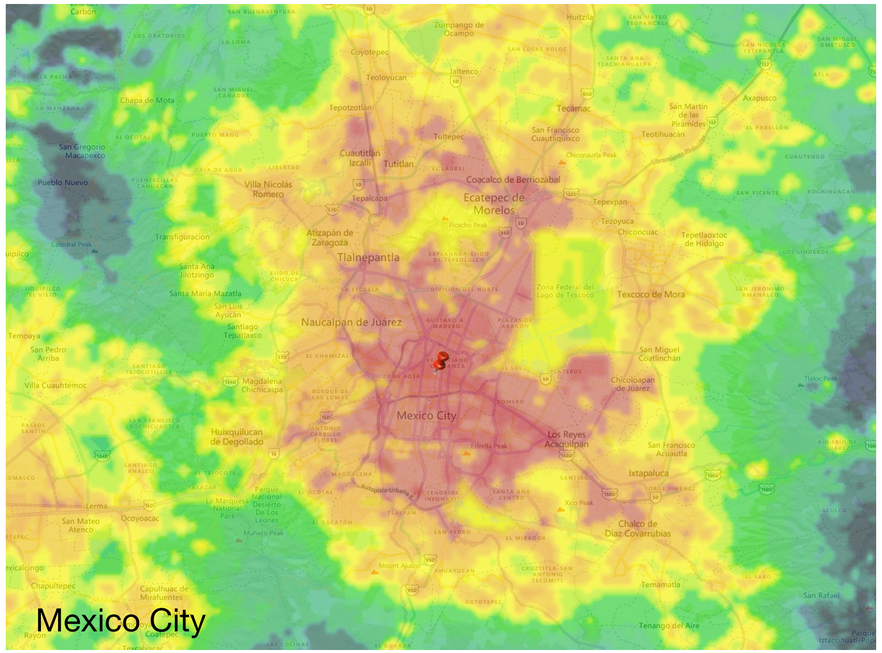
Spatial Variation — Fractal Dimension

City	D_2 mean
Toronto	1.292
New York	1.457
Mexico City	1.529
Paris	1.586



Notice the pockets of red regions in Toronto and New York.





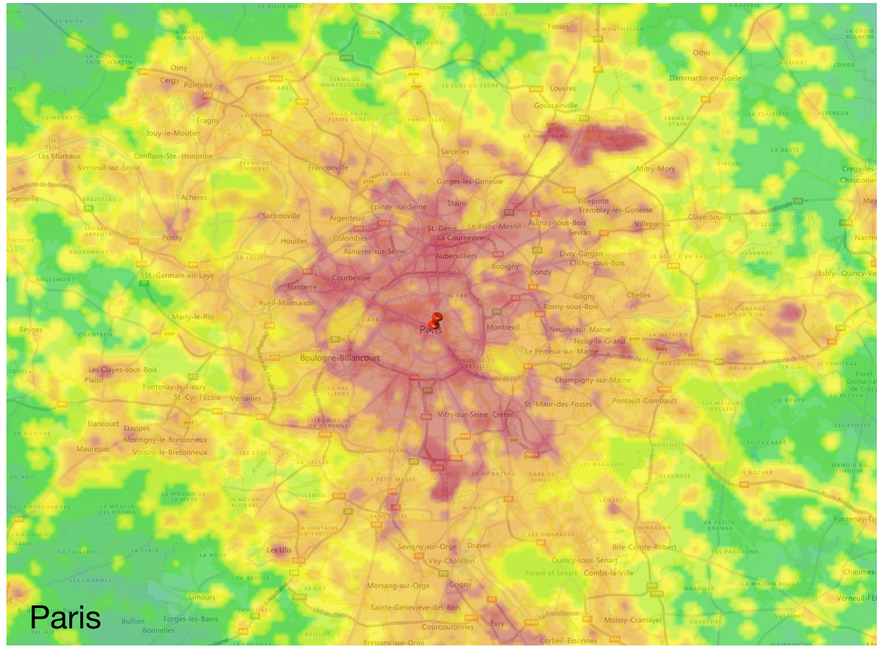
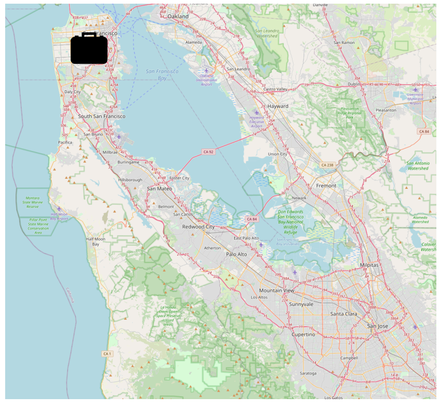


Image credits — https://www.lightpollutionmap.info

Spatial Variation — Fractal Dimension

- Planning (same oversimplified example!)



1. Plan to increase office spaces in downtown;
2. Measure the potential impact to $\alpha, \gamma, \zeta, D_2$;
3. Suggest potential ways to reduce congestion. For instance:
    1. bound on the number of people or;
    2. alternative spots

Summary of Urban Human Mobility Characterization

- Temporal (DPL) characterization depicts the temporal pattern of human movement.
- Fractal dimension provides a statistic for the spatial distribution of requests.
- Both, temporal and spatial characterizations, form qualitative metrics to validate urban level characteristics of ridership.

How to generate synthetic data for any urban area based on its characteristics?

Synthetic Data - Why GANs?

- Allows to model Pr()

- Generator capable of generating realistic looking images.

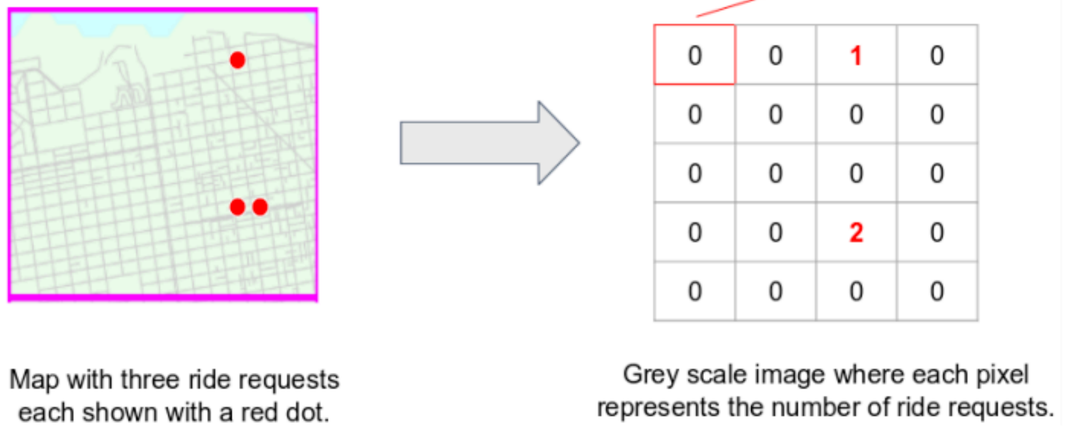
Quality of images generated by GANs over the years

- Minimize the divergence between generated distribution and target distribution.
- Gives us a parameterized model to augment the number of rides.

Image credits: https://twitter.com/goodfellow_ian/status/1084973596236144640?lang=en

Synthetic Data Generation — Spatial Generator

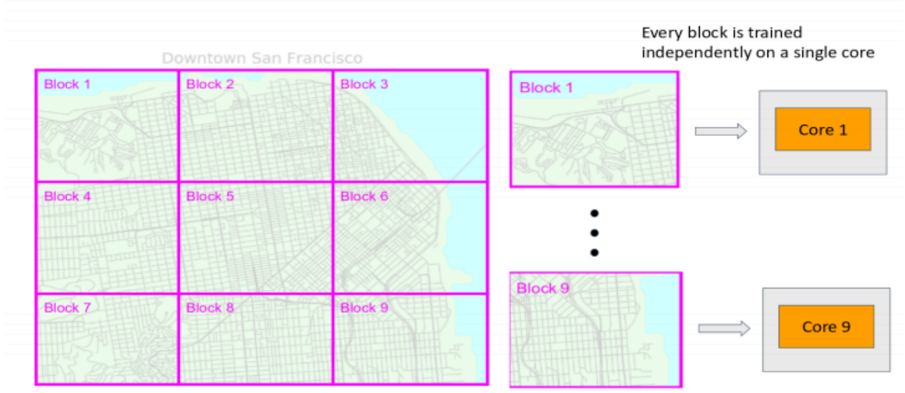
1. Ride Requests to Images



Map with three ride requests each shown with a red dot.

Grey scale image where each pixel represents the number of ride requests.

2. Parallel Training using GANs

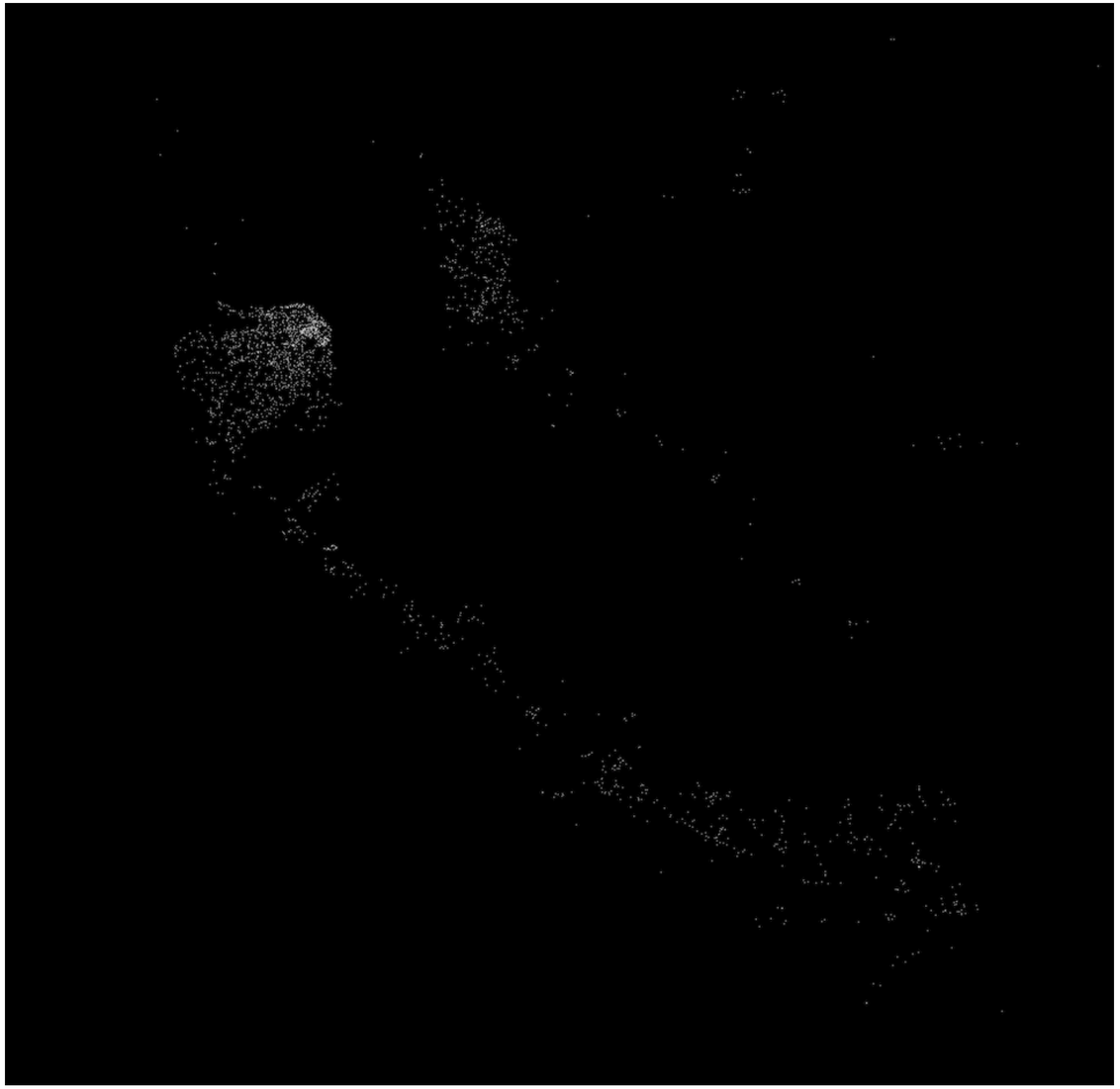


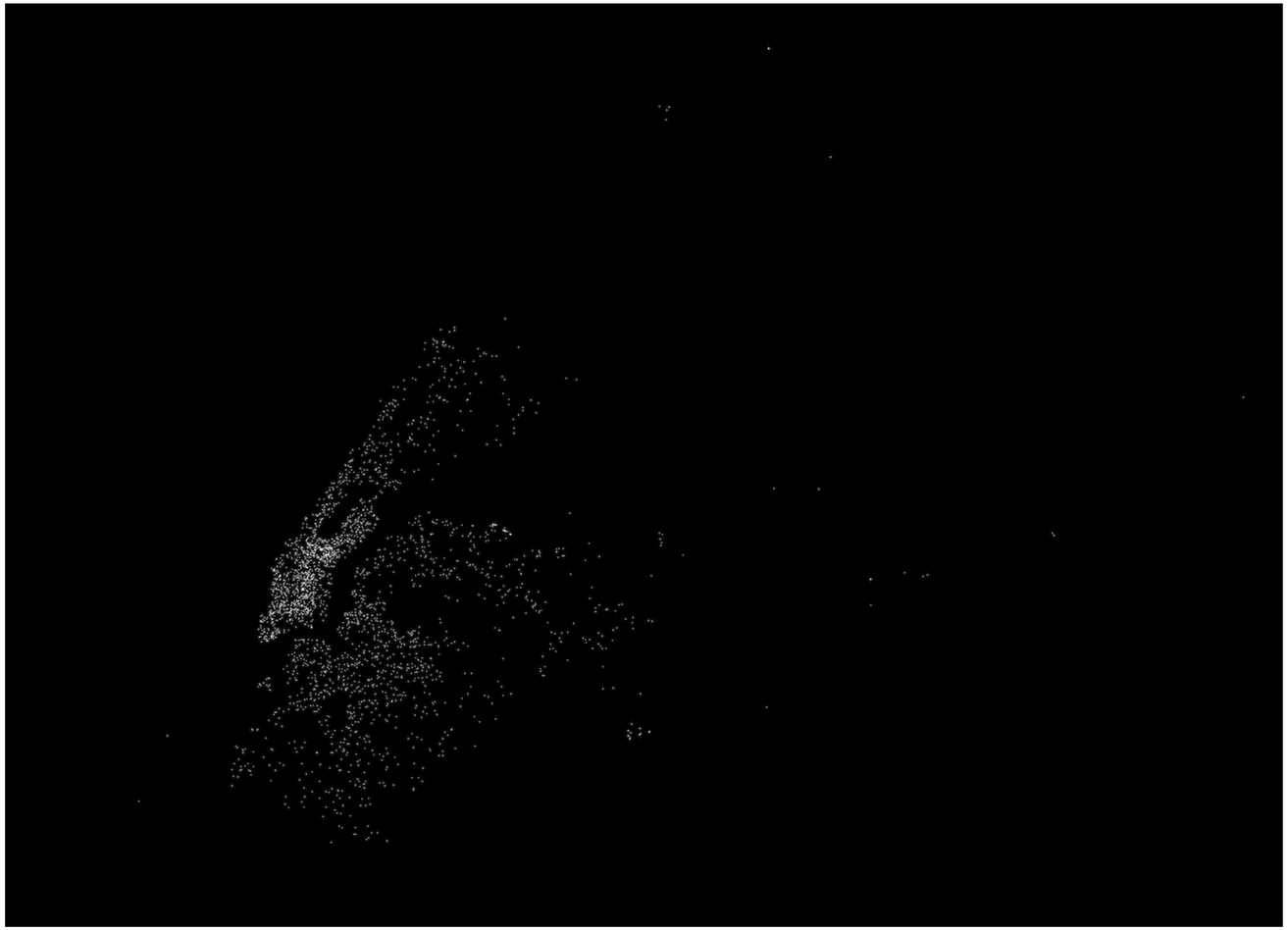
Each block represents a grey scale image of 24×24 pixels.

We use conditional GANs where time snapshot is the label for each image.

Synthetic Data Generation — Spatial Generator

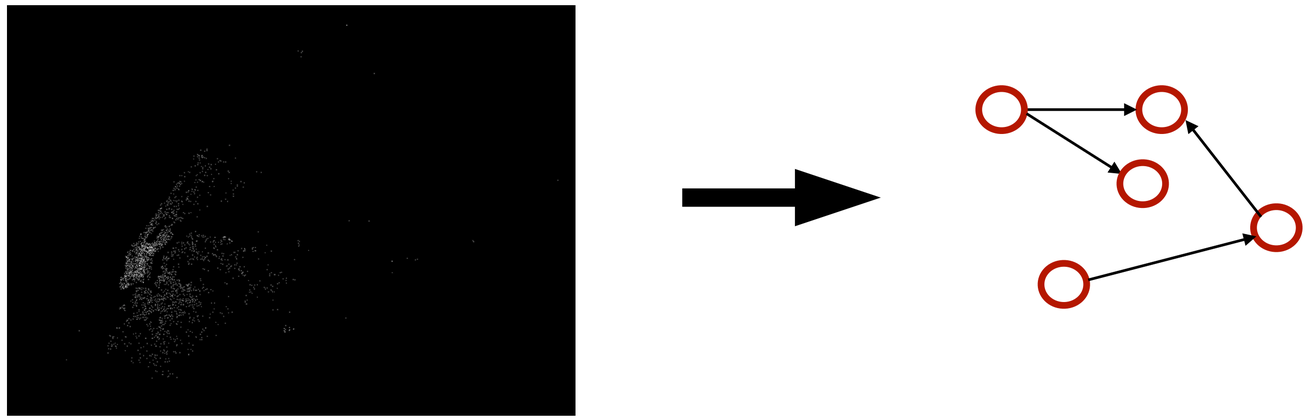
Synthetic overlap of ride requests for San Francisco (Bay Area) and New York after stitching blocks trained in parallel —





Synthetic Data Generation — Graph Generator

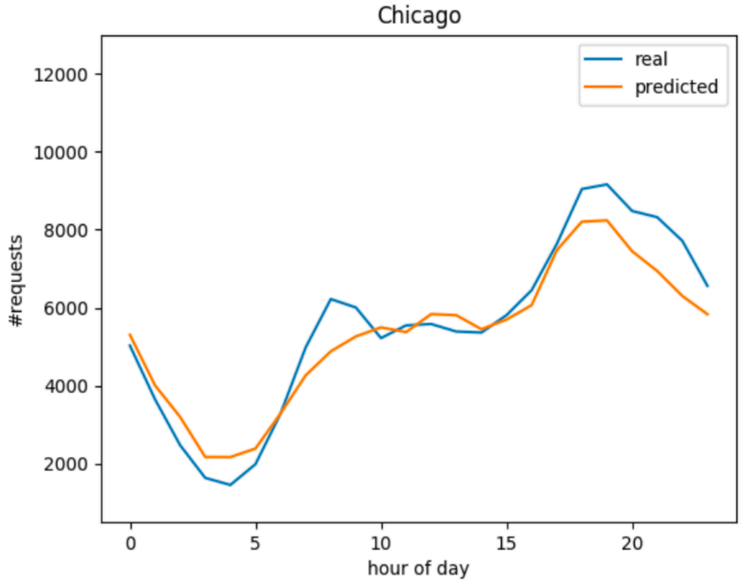
How to convert points into valid ride requests?

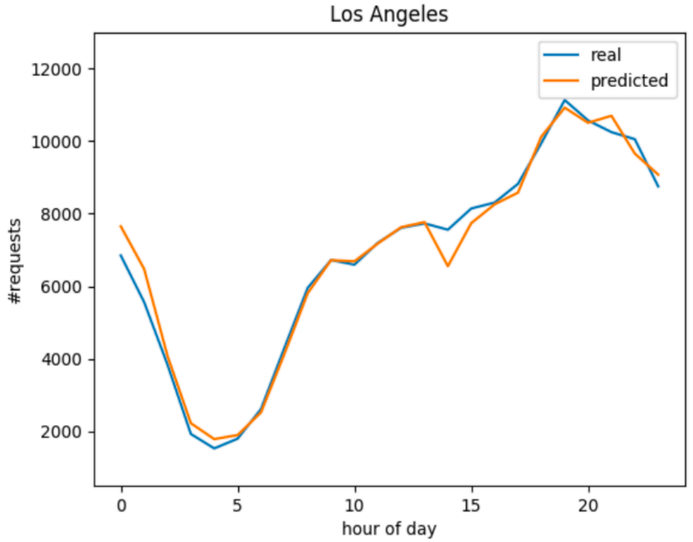


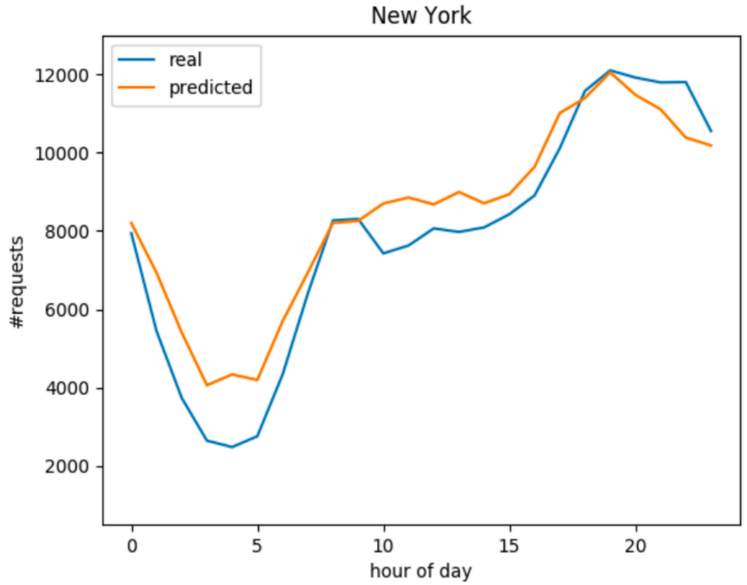
Given a set of nodes M, empty rich set R:

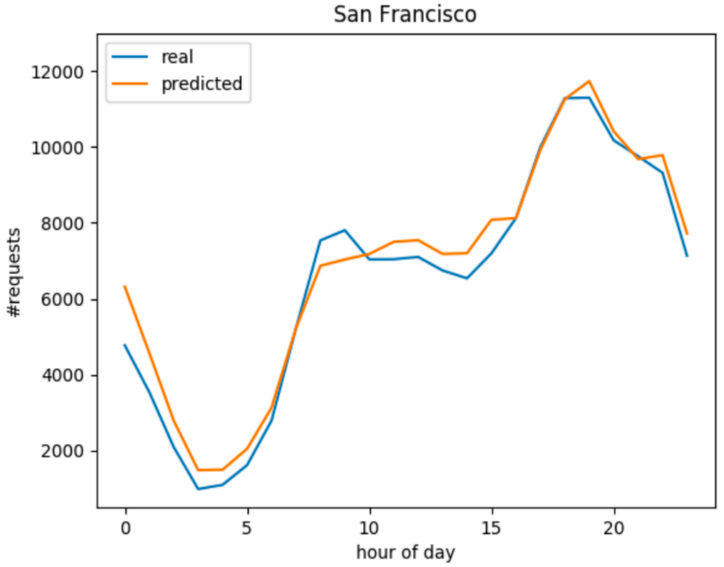
1. Uniformly randomly choose a source node from M, and some number of edges.
2. With some low probability choose destination from a rich set of nodes R, or else just choose from M.
3. Add source node to R.

Validation of Real & Synthetic Datasets









Temporal Validation of Real & Synthetic Datasets

City	Real Data Sets (α)	Synthetic Data Sets (α)
Chicago	1.415	1.492
New York	1.299	1.361
Los Angeles	1.053	1.614
San Francisco	1.250	1.341

Increase of 0.1 in exponent translates to ~10% decrease in number of nodes.

Spatial Validation of Real & Synthetic Datasets

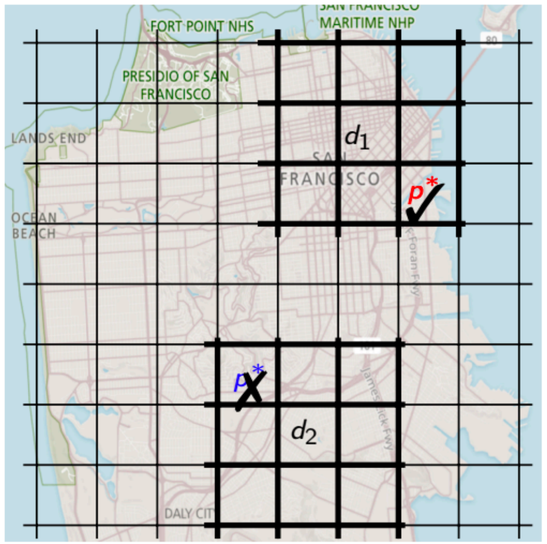
City	Real Data Sets (D_2 mean)	Synthetic Data Sets (D_2 mean)
Chicago	1.384	1.435
New York	1.648	1.540
Los Angeles	1.352	1.314
San Francisco	1.548	1.442

What are some applications which can be aided by the city level characterization?

Real-time Vehicle Placement Problem

- Advantageous for ride-sharing services to reduce average waiting time for rider.
- Also, beneficial for driver, and vehicle efficiency; more savings if cars can be directed to the right place beforehand.

Real-time Vehicle Placement Problem



- d_i - dropoffs at time snapshot t
- p^* - placement for d_1 by time snapshot $t+1$
- p^* - placement for d_2 by time snapshot $t+1$

Two placements are made using some algorithm.

Reward R is computed for every time snapshot:

$$R(t+1) = \frac{\#\text{good placements}}{\#\text{total placements}}$$

Objective: Maximize reward over time

Real-time Vehicle Placement Problem

- We explore a bunch of online algorithms -
    1. Follow the leader — go to the node with maximum number of requests based on historical data.
    2. Uniformly at random choose a node for placement.
    3. Assume every node follows a poisson process for incoming ride requests.

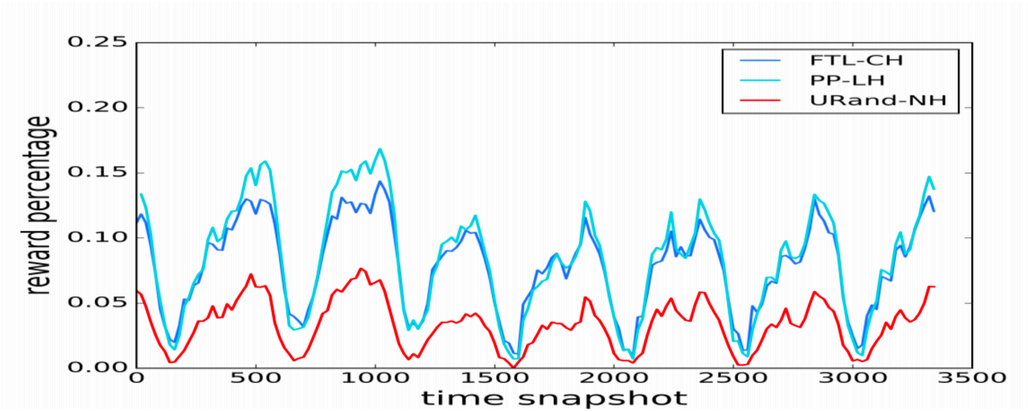
Real-time Vehicle Placement Analysis

Theorem 1: Using fractal dimension, the expected reward with follow the lead with complete history would be strictly better than algorithm which chooses a node uniformly at random.

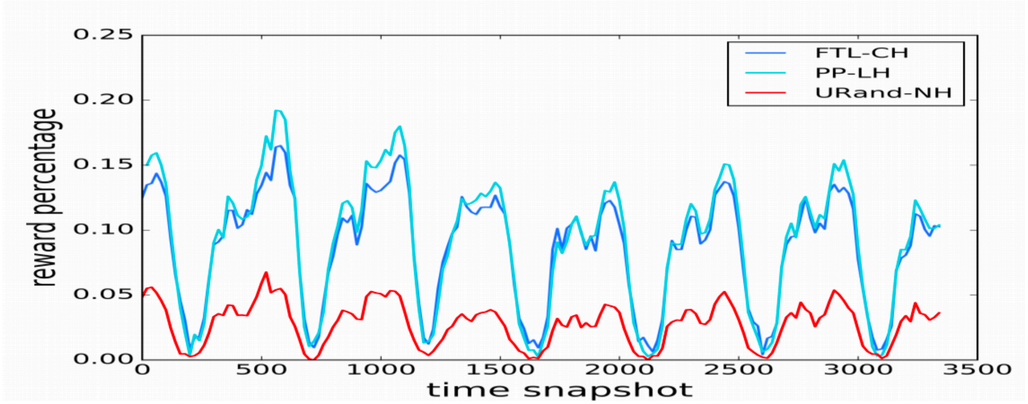
Real-time Vehicle Placement Analysis

Theorem 2: Follow the leader with complete history would have an expected performance equivalent to an algorithm which assumes that every node observes a poisson process for ride requests.

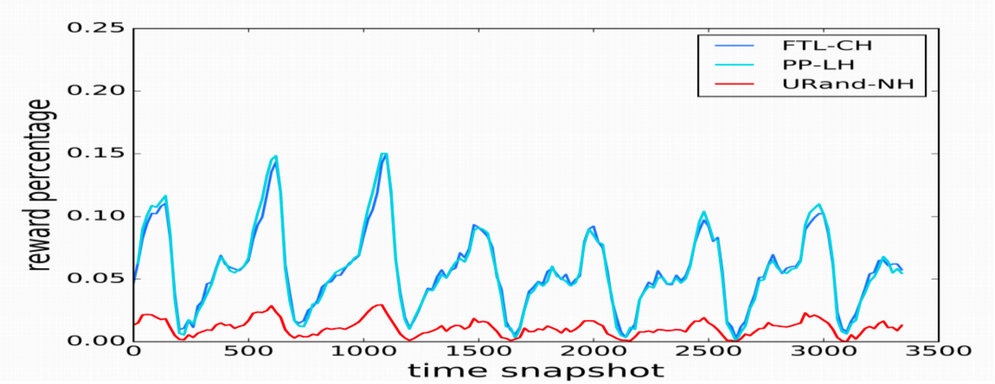
Vehicle Placement: Real-data results



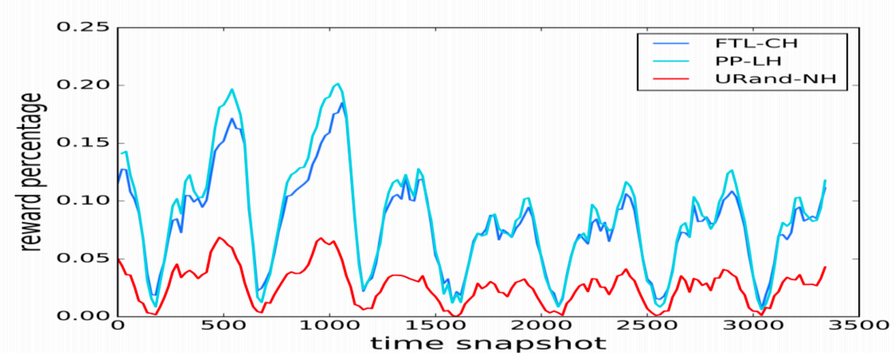
New York



San Francisco

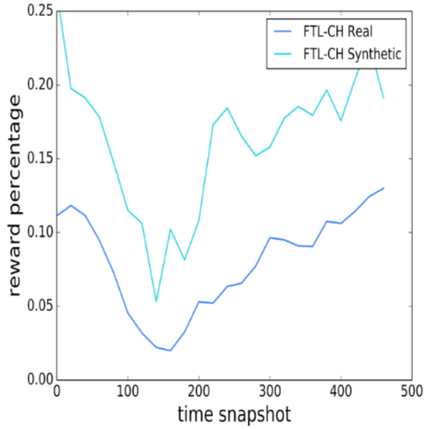


Los Angeles

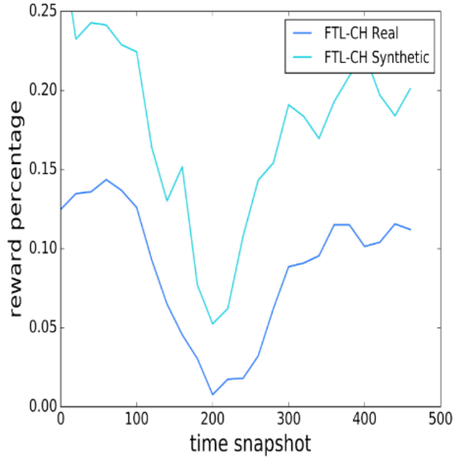


Chicago

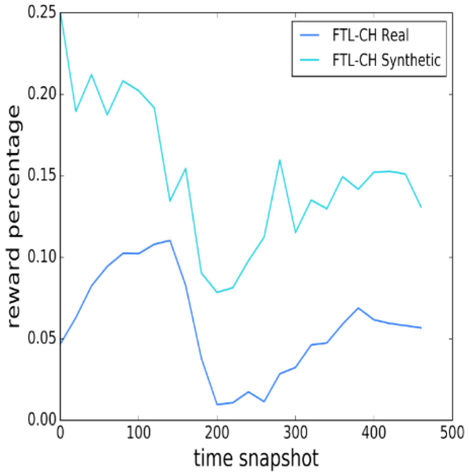
Vehicle Placement: Real vs. Synthetic Results



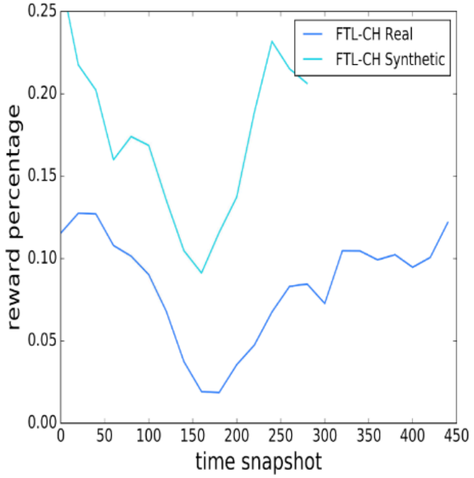
New York



San Francisco



Los Angeles

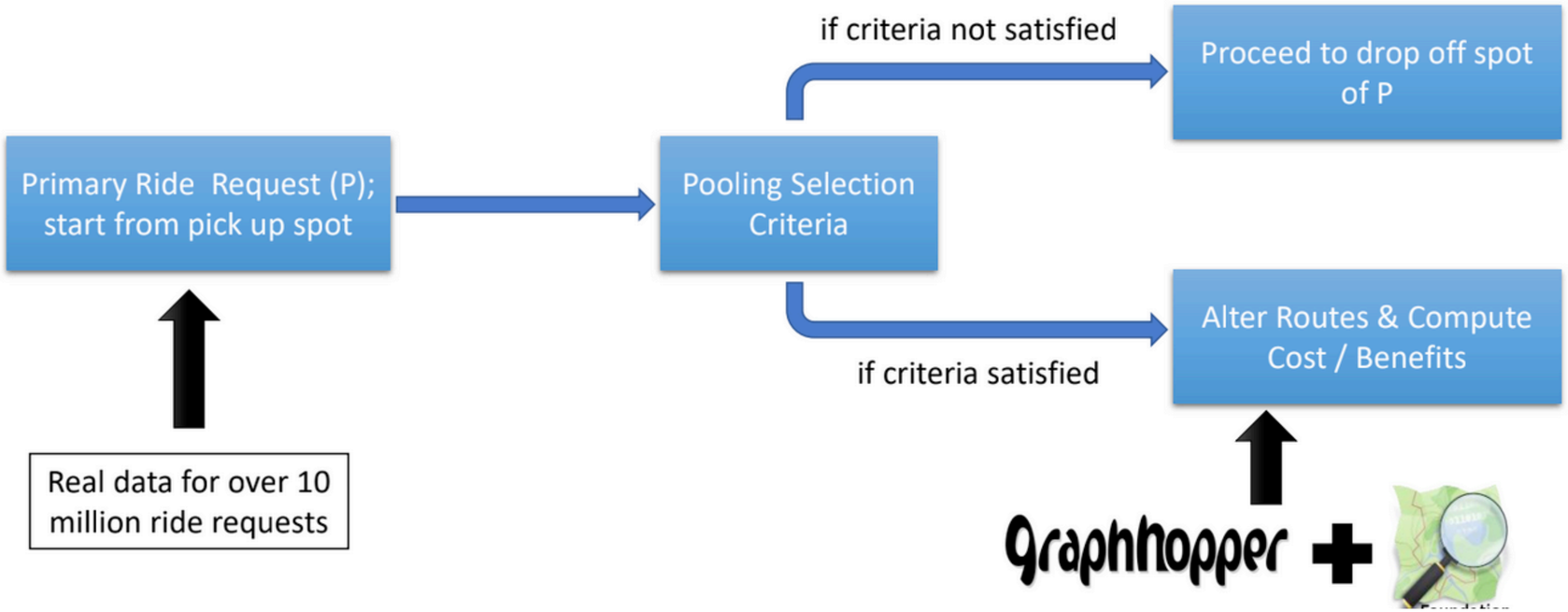


Chicago

Dynamic Ride Pooling

- Propose the design space for real-time pooling of riders.
- The decisions for pooling take into account the temporal and spatial proximity of the ride requests.
- Such a method can be used by ride-sharing services, and also for different what-if scenarios to assess the societal benefit.

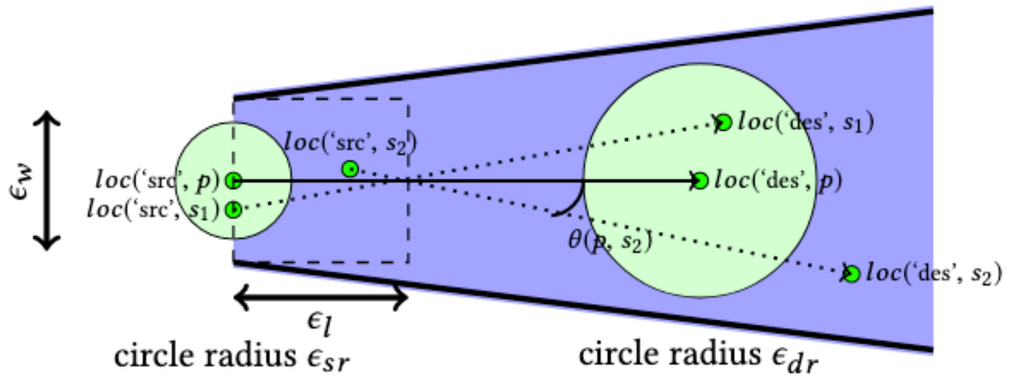
Dynamic Ride Pooling



Dynamic Ride Pooling

Design Space:

1. Time interval
2. Vehicle Occupancy
3. Distance from pickup (ϵ_{sr})
4. Distance from drop-off (ϵ_{dr})
5. Rectangular Width (ϵ_w)
6. Rectangular length (ϵ_l)
7. Angular difference (ϵ_θ)



Dynamic Ride Pooling — Results

Metric	*San Francisco*	*New York*	*Los Angeles*	*Mean across cities*
Total Travel Distance Reduction (%)	17.13	19.06	11.01	15.76
Total Vehicle Count Reduction (%)	33.76	36.93	23.03	31.23
Mean Poolability (%)	48.94	56.39	34.52	46.61
Mean Travel Time Penalty (sec)	162.12	97.55	148.17	135.94

Summary of benefits and costs. Parameters used $\epsilon_t = 5$ **mins.,** $\epsilon_{sr} = 500m$, $\epsilon_{dr} = 1000m$, $\epsilon_w = 2000m$, $\epsilon_\theta = 20°, k = 3$

Contributions

- By access to massive amounts of data, highlighted different ways to characterize dynamics of urban human mobility:
  - Ride Request Graph for temporal patterns
  - Fractal dimension for spatial patterns
- Parallel privacy preserving method for generating synthetic data which can easily be parallelized.
- Use temporal, and spatial characterizations to demonstrate real-world applications for reducing traffic congestion; and make cities more eco-friendly.
- Urban human mobility toolkit for analysis, data generation, and applications:

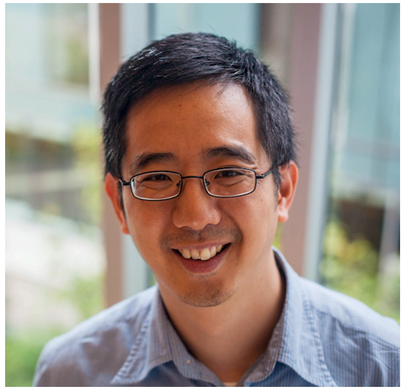
http://github.com/ajauhri/mobility-modeling

Future Works

- Better sampling methods from GANs to eradicate training bias.
- Urban planning from synthetic data; what-if scenarios.
- Covid-19 drastically changed mobility patterns; ride-sharing services survived due to food delivery services. It will be interesting to look pattern of food delivery services and compare it with ride-sharing services.
- What is the appropriate balance between applying differential privacy and preserve characteristics of urban human mobility data?
- How can vehicles act as smart sensing objects; learn real-time human mobility patterns and respond with intervention?

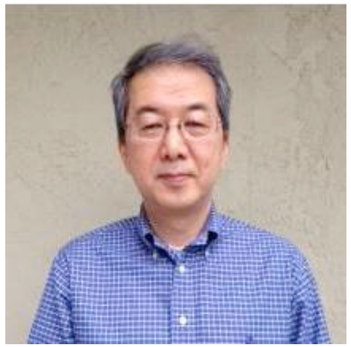
Thanks!














Contributions

- By access to massive amounts of data, highlighted different ways to characterize dynamics of urban human mobility:
  - Ride Request Graph for temporal patterns
  - Fractal dimension for spatial patterns
- Parallel privacy preserving method for generating synthetic data which can easily be parallelized.
- Use temporal, and spatial characterizations to demonstrate real-world applications for reducing traffic congestion; and make cities more eco-friendly.
- Urban human mobility toolkit for analysis, data generation, and applications:

http://github.com/ajauhri/mobility-modeling

List of Publications & Internships

Jauhri, Abhinav, et al. "Generating Realistic Ride-Hailing Datasets Using GANs." *ACM Transactions on Spatial Algorithms and Systems (TSAS)* 6.3 (2020): 1-14.

Jauhri, Abhinav, et al. "Space-Time Graph Modeling of Ride Requests Based on Real-World Data." *AAAI Workshops*. 2017.

Chen, Min Hao, Abhinav Jauhri, and John Paul Shen. "Data driven analysis of the potentials of dynamic ride pooling." *Proceedings of the 10th ACM SIGSPATIAL Workshop on Computational Transportation Science*. 2017.

Jauhri, Abhinav, Carlee Joe-Wong, and John Paul Shen. "On the real-time vehicle placement problem." *arXiv preprint arXiv:1712.01235* (2017).

Jauhri, Abhinav, Martin Griss, and Hakan Erdogmus. "Small Polygon Compression." *2016 Data Compression Conference (DCC)*. IEEE Computer Society, 2016.

Erdogmus, H., M. Griss, B. Iannucci, S. Kumar, J. Falcão, A. Jauhri, and M. Kovalev. "Opportunities, options, and enhancements for the wireless emergency alerting service." *Carnegie Mellon University, Technical Report CMU-SV-15-001* (2015).

Jauhri, Abhinav, Bradley McDanel, and Chris Connor. "Outlier detection for large scale manufacturing processes." *2015 IEEE International Conference on Big Data (Big Data)*. IEEE, 2015.

Jauhri, Abhinav, Jason D. Lohn, and Derek S. Linden. "A comparison of antenna placement algorithms." *Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation*. 2014.

Nvidia, Santa Clara, CA (Summer 2013)

Intel, Hillsboro, OR (Summer 2014 & 2015)

Uber, San Francisco, CA (Spring & Summer 2016)

Facebook, New York, NY (Summer 2017)

Intel, Santa Clara, CA (Summer & Fall 2018)