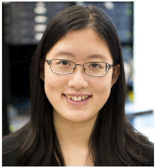
Human Mobility Modeling






Abhinav Jauhri Daniel Chen Carlee Joe-Wong John Paul Shen

Carnegie Mellon University

April 27, 2017

Outline of this talk

- Part 0: Motivation
- Part 1: Modeling (Recap)
- Part 2: Placement
- Part 3: Poolability
- Part 4: Discussion

Part 0: Motivation

Motivation

- Human mobility modeling and understanding.

Motivation

- Human mobility modeling and understanding.
- Ubiquitous mobile devices for sensing at scale.

Motivation

- Human mobility modeling and understanding.
- Ubiquitous mobile devices for sensing at scale.
- Global presence and availability of ride-sharing services.

Motivation

- Human mobility modeling and understanding.
- Ubiquitous mobile devices for sensing at scale.
- Global presence and availability of ride-sharing services.
- Extensive real ride request data from a ride-sharing service.

Motivation

- Human mobility modeling and understanding.
- Ubiquitous mobile devices for sensing at scale.
- Global presence and availability of ride-sharing services.
- Extensive real ride request data from a ride-sharing service.
- Potential of large-scale sensing and analytics for societal good.

Part 1: Modeling

Ride Request Definition

Each ride request is defined by:

1. Time of request: t=<timestamp>
2. Pickup location: s=<latitude, longitude>
3. Dropoff location: d=<latitude, longitude>

Temporal Pattern of Ride Requests

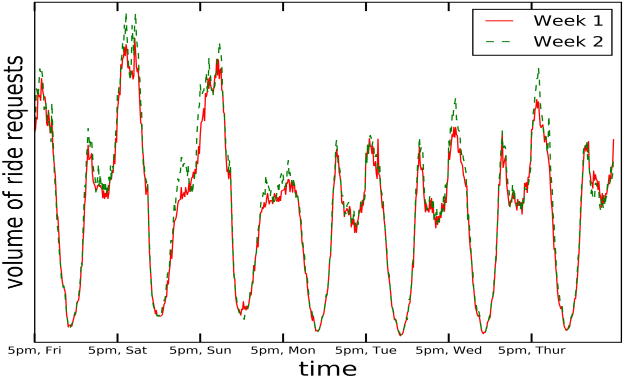


Figure: Similarity in the weekly pattern of ride requests in San Francisco

Carnegie Mellon

Temporal & Spatial Pattern of Ride Requests

Video

<u>Observation:</u> There is significant variability in the **ride request patterns** from city to city, and across space and time.

Question: Is there a rigorous model that can capture the variations of ride request patterns in a city?

Temporal Evolution of Graphs

Densification Power Law:

- networks are becoming denser over time

Temporal Evolution of Graphs

Densification Power Law:

- networks are becoming denser over time
- the number of edges grow faster than the number of nodes – average degree is increasing

Temporal Evolution of Graphs

Densification Power Law:

- networks are becoming denser over time
- the number of edges grow faster than the number of nodes – average degree is increasing

$$E(t) \propto N(t)^{\alpha} \tag{1}$$

Graph Densification

- Densification Power Law:

$$E(t) \propto N(t)^{\alpha}$$

Densification exponent: $1 \leq \alpha \leq 2$:

Graph Densification

- Densification Power Law:

$$E(t) \propto N(t)^{\alpha}$$

Densification exponent: $1 \leq \alpha \leq 2$:

  - $\alpha = 1$: Linear growth - constant out-degree

Graph Densification

- Densification Power Law:

$$E(t) \propto N(t)^{\alpha}$$

Densification exponent: $1 \leq \alpha \leq 2$:

  - $\alpha = 1$: Linear growth - constant out-degree
  - $\alpha = 2$: Quadratic growth - clique

Graph Densification

- Densification Power Law:

$$E(t) \propto N(t)^{\alpha}$$

Densification exponent: $1 \leq \alpha \leq 2$:

  - $\alpha = 1$: Linear growth - constant out-degree
  - $\alpha = 2$: Quadratic growth - clique

Let's look at some real graphs!

Densification: Physics Citations

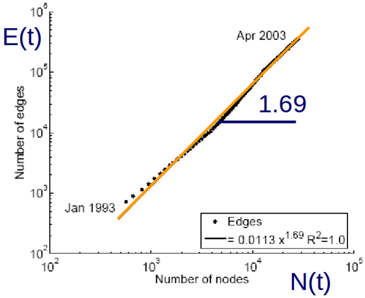


Image source: Leskovec, KDD, 2005.

- 1992: 1,293 papers, 2,717 citations
- 2003: 29,555 papers, 352,807 citations
- For each month *m*, create a graph of all citations up to month *m*.

Densification: Graph of the Internet

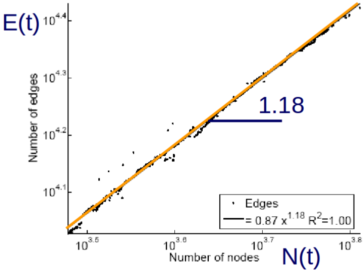


Image source: Leskovec, KDD, 2005.

- 1997: 3,000 nodes, 10,000 edges
- 2000: 6,000 nodes, 26,000 edges
- One graph per year.

Densification: Ride Request Graph

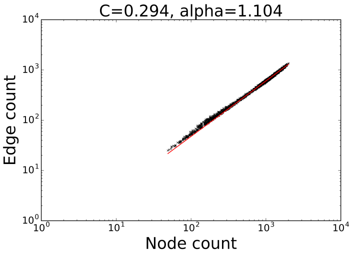
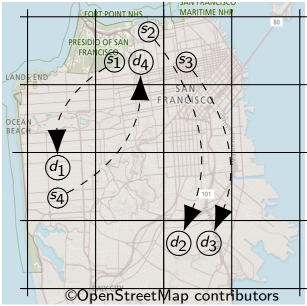


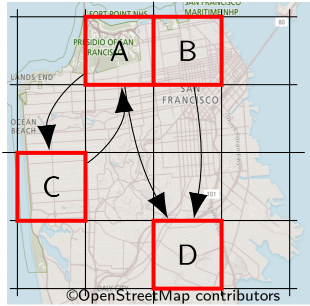
Figure: $n = 5$ minutes

- Non-peak hour: 662 nodes, 383 edges
- Peak hour: 7269 nodes, 7361 edges
- One graph for every n minutes.

Ride Request Graph



(a) Four ride requests distributed spatially over a map



(b) Corresponding Ride Request Graph with four nodes (marked by red boxes) and directed edges.

Figure: Transformation of ride requests, in a particular time interval, into a directed ride-request graph (RRG).

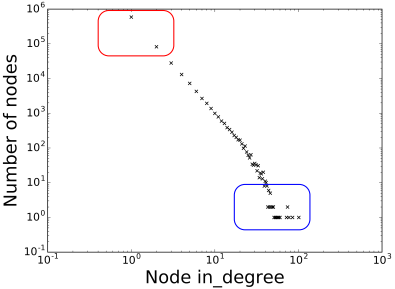
Densification: Ride Request Graph

Densification implies community effect:

Densification: Ride Request Graph

Densification implies community effect:

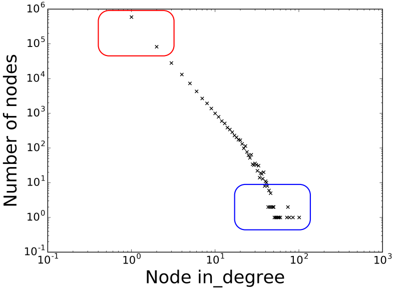
- Few nodes with high degree
- Many nodes with low degree



Densification: Ride Request Graph

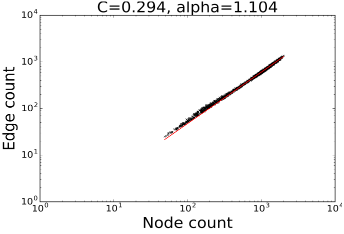
Densification implies community effect:

- Few nodes with high degree
- Many nodes with low degree

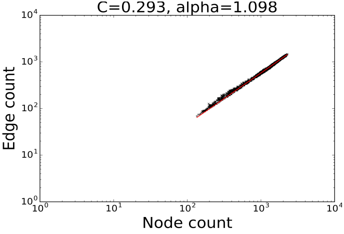


Number of nodes with k in-degree would be $\propto 1/k^c$

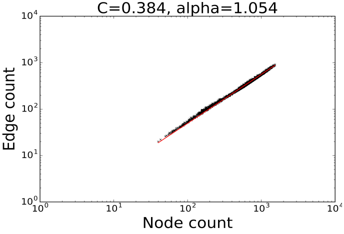
Densification: Ride Request Graphs



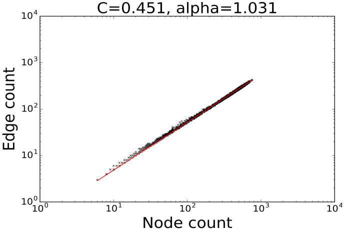
(a) San Francisco



(b) New York



(c) Paris



(d) Hyderabad

Modeling

Summary: RRGs provide a rigorous model to characterize ride requests over time.

Part 2: Placement Problem

Question: Where should drivers go after droping off passengers?

Problem Definition

Let's say at time snapshot t, n vehicles drop-off riders at d_is:

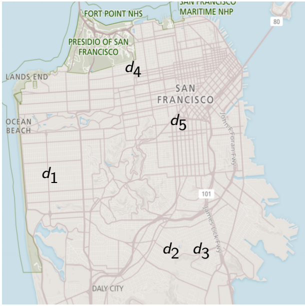


Figure: d_i's denote drop-off points in SF downtown at a time snapshot.

Problem Definition

Let's say at time snapshot t, n vehicles drop-off riders at d_is:

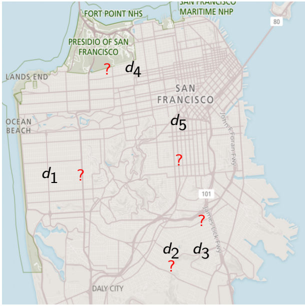


Figure: d_i's denote drop-off points in SF downtown at a time snapshot. Red marks (?) denote possible placements.

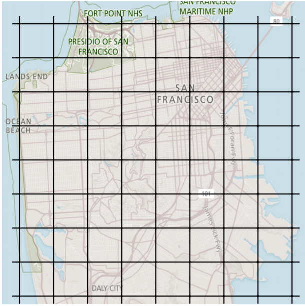
Question: Where should the n vehicles be placed s.t. pickup times for requests at time period $t+1$ are minimized? There are numerous possiblities!

Assumptions

- Drivers don't get tired; willing to pick-up immediately after a drop-off.

Assumptions

- Drivers don't get tired; willing to pick-up immediately after a drop-off.
- Instead of finding exact placement locations, we discretize space into equally sized small nodes/grids:



Problem is simplified to finding a node to place a vehicle.

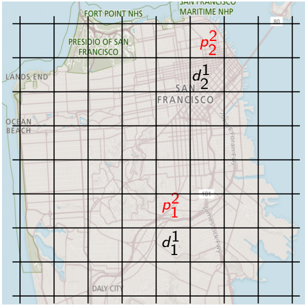
Approach

Online Learning: Data points are arriving over time, and a decision needs to be made on the fly without knowing what will happen in the future.
Decision timescale is as kept as low as three minutes.

Approach

Online Learning: Data points are arriving over time, and a decision needs to be made on the fly without knowing what will happen in the future.
Decision timescale is as kept as low as three minutes.

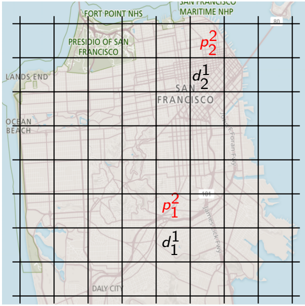
An online learning approach which chooses actions such that the total rewards are close to the best action in hindsight.



Approach

Online Learning: Data points are arriving over time, and a decision needs to be made on the fly without knowing what will happen in the future.
Decision timescale is as kept as low as three minutes.

An online learning approach which chooses actions such that the total rewards are close to the best action in hindsight.



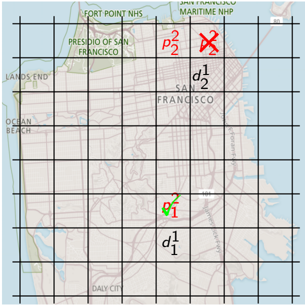
Consider the scenario at time snapshot $t = 1$:

- d_i^j is is the *ith* drop-off at the *jth* time snapshot
- p_i^j is placement of *ith* drop-off in the j time snapshot

Approach

Online Learning: Data points are arriving over time, and a decision needs to be made on the fly without knowing what will happen in the future.
Decision timescale is as kept as low as three minutes.

An online learning approach which chooses actions such that the total rewards are close to the best action in hindsight.



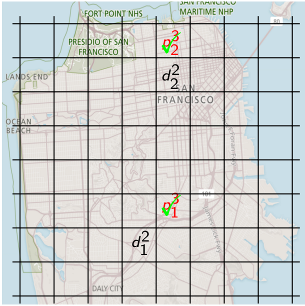
At time snapshot $t = 2$, we realize how good were our placements:

- Only p_1^2 was a good placement
- Reward: $r_2 = 1$

Approach

Online Learning: Data points are arriving over time, and a decision needs to be made on the fly without knowing what will happen in the future.
Decision timescale is as kept as low as three minutes.

An online learning approach which chooses actions such that the total rewards are close to the best action in hindsight.



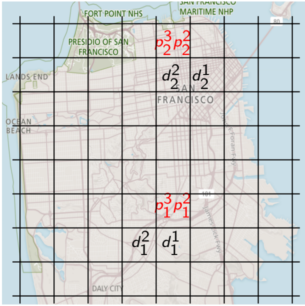
At time snapshot $t = 3$:

- p_1^3 and p_2^3 were good placements
- Reward: $r_3 = 2$

Approach

Online Learning: Data points are arriving over time, and a decision needs to be made on the fly without knowing what will happen in the future.
Decision timescale is as kept as low as three minutes.

An online learning approach which chooses actions such that the total rewards are close to the best action in hindsight.



In hindsight, we could have received rewards:

$r_2 + r_3 = 4$

Reward Percentage Definition

Reward percentage is defined for every time snapshot:

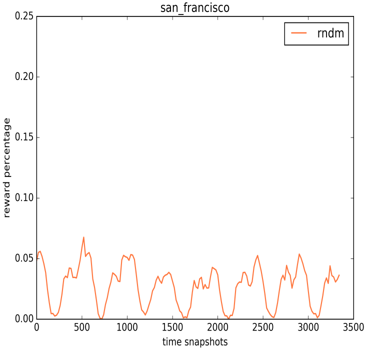
$$R(t) = \frac{\#good\_placements_t}{\#dropoffs_{t-1}} \tag{2}$$

Placement Problem - Random Selection

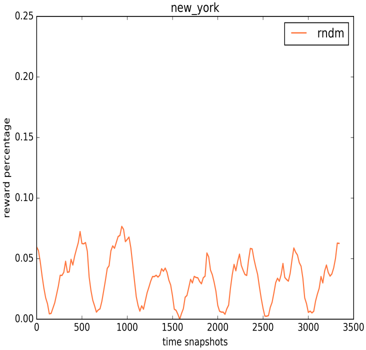
Randomly choose from the allowable placements.

Placement Problem - Random Selection

Randomly choose from the allowable placements.



(a) San Francisco



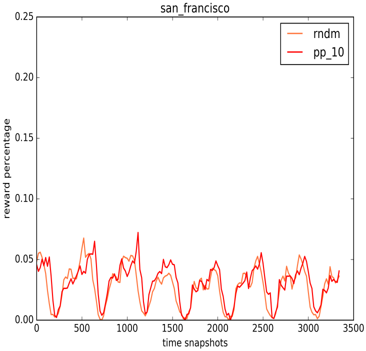
(b) New York

Placement Problem - Poisson Process

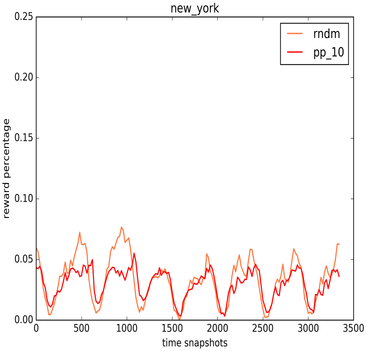
Choose the allowable placement which maximizes the probability of pickup.

Placement Problem - Poisson Process

Choose the allowable placement which maximizes the probability of pickup.



(a) San Francisco



(b) New York

Follow The Leader

On each time snapshot $t = 1, 2, \ldots,$

- For each drop-off $d \in \{1, 2, \cdots, n\}$, pick a set of constrained actions A_d such that $|A_d| = m$.

Follow The Leader

On each time snapshot $t = 1, 2, \ldots,$

- For each drop-off $d \in \{1, 2, \cdots, n\}$, pick a set of constrained actions A_d such that $|A_d| = m$.
- Choose action $a \in A_d$ with maximum reward $r_t[a]$, where $r_t[a] =$ total reward for action a so far.

Follow The Leader

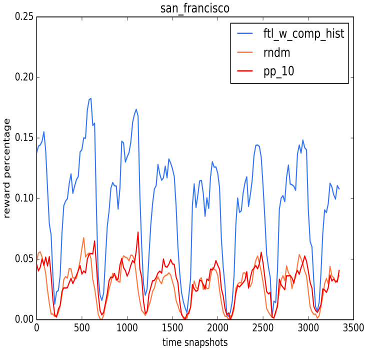
On each time snapshot $t = 1, 2, \ldots,$

- For each drop-off $d \in \{1, 2, \cdots, n\}$, pick a set of constrained actions A_d such that $|A_d| = m$.
- Choose action $a \in A_d$ with maximum reward $r_t[a]$, where $r_t[a] =$ total reward for action a so far.

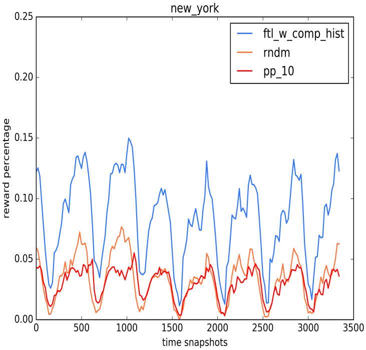
A slightly modified version of the algorithm is instead of using $r_t[a]$, we could use total rewards for k previous time snapshots only.

Placement Problem - Follow the leader

Placement Problem - Follow the leader



(a) San Francisco

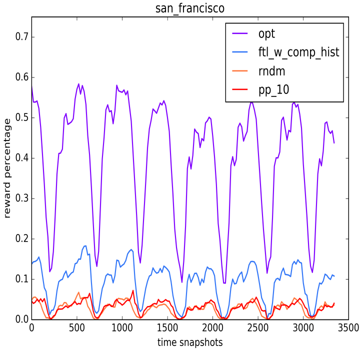


(b) New York

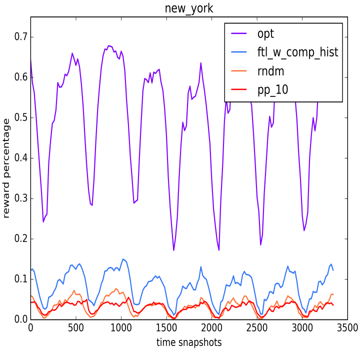
☺

Placement Problem - Optimal

Placement Problem - Optimal



(a) San Francisco



(b) New York

☹

Placement

Observation: Placement of vehicles at granular geo-locations is a hard problem.

Part 3: Poolability

Poolability Definition

Each ride request defined by $< t, s, d >$. Pool ride requests if:

1. $\Delta t < m$ time units

Poolability Definition

Each ride request defined by $< t, s, d >$. Pool ride requests if:

1. $\Delta t < m$ time units
2. $\Delta s < S$ distance units

Poolability Definition

Each ride request defined by $< t, s, d >$. Pool ride requests if:

1. $\Delta t < m$ time units
2. $\Delta s < S$ distance units
3. $\Delta d < D$ distance units

Poolability Definition

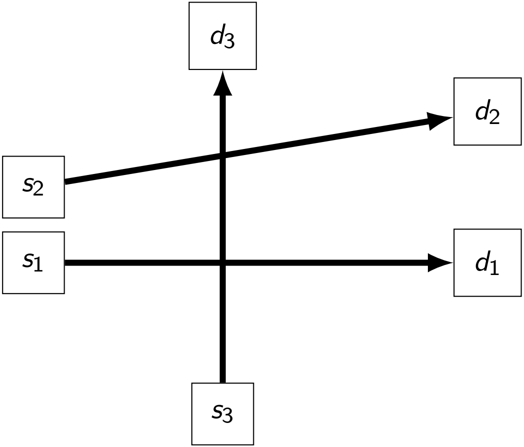
Each ride request defined by $< t, s, d >$. Pool ride requests if:

1. $\Delta t < m$ time units
2. $\Delta s < S$ distance units
3. $\Delta d < D$ distance units

Poolability is the percentage of ride requests poolable. For simplicity, we discretize time into buckets.

Poolability Example

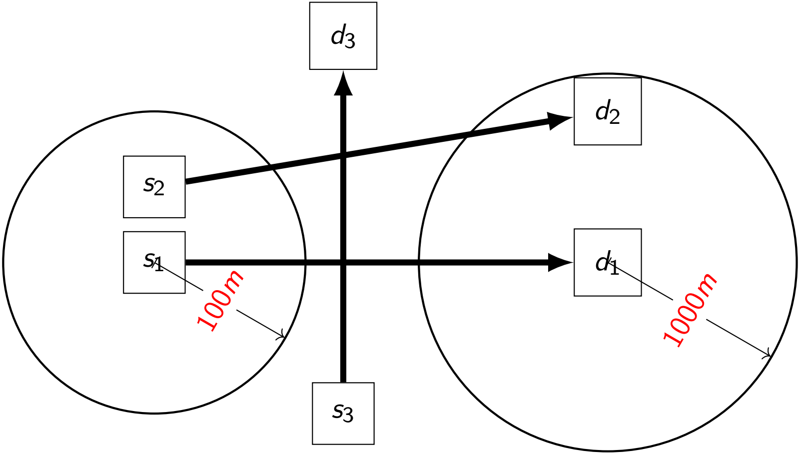
Assume all 3 requests came within 5 minutes.
Case 1: 3 vehicles for 3 requests.



Poolability Example

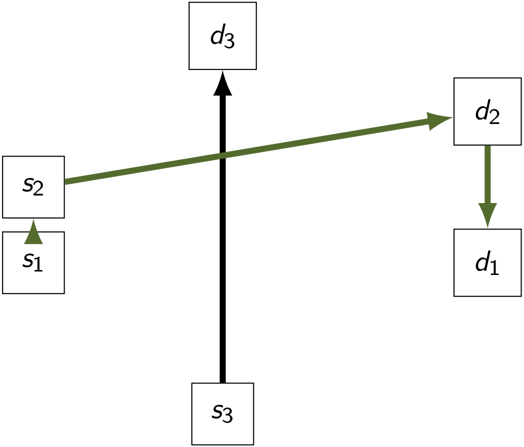
Assume all 3 requests came within 5 minutes.

Case 1: 3 vehicles for 3 requests.

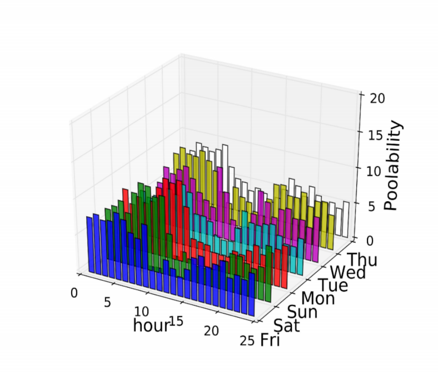


Poolability Example

Assume all 3 requests came within 5 minutes.
Case 2: 2 cars for 3 requests. Poolability $= 66.6\%$



Poolability



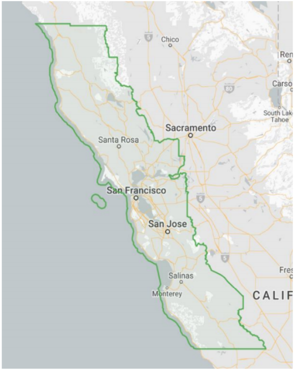


Figure: *Left:* Poolability for a week of data. *Right:* Boundary of the city of San Francisco.

Poolability Experiements

Three metrics to analyze poolability:

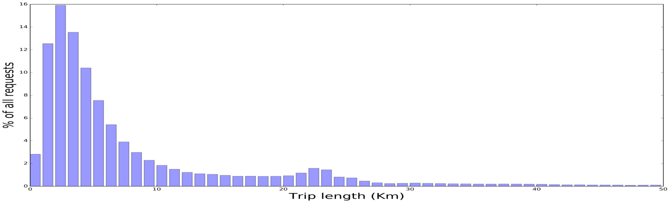
- *Savings:*
  - Total distance covered.
  - Total number of vehicles used.

Poolability Experiements

Three metrics to analyze poolability:

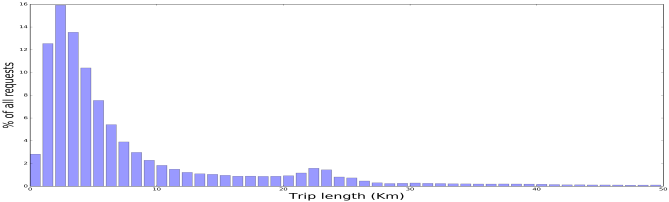
- *Savings:*
  - Total distance covered.
  - Total number of vehicles used.
- *Cost:* Added travel time.

Poolability Savings: San Francisco

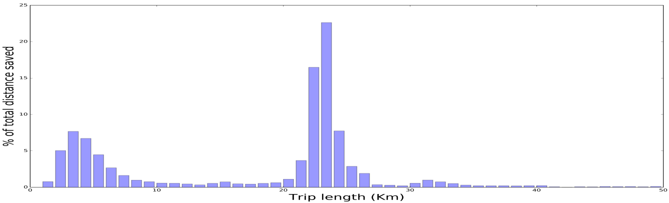


(a) Percentage distribution of trip distances; Mean distance: 8.83km

Poolability Savings: San Francisco

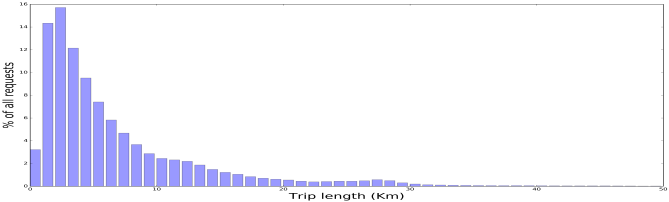


(a) Percentage distribution of trip distances; Mean distance: 8.83km

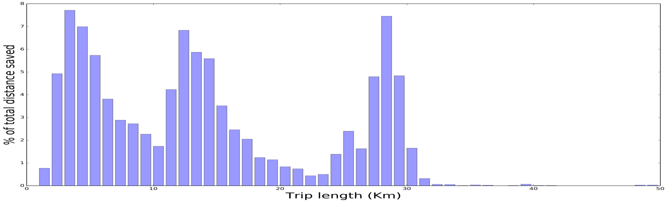


(b) Percentage distribution of reduction of travel distances

Poolability Savings: New York

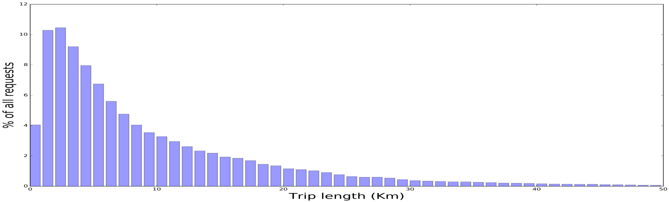


(a) Percentage distribution of trip distances; Mean distance: 6.98km

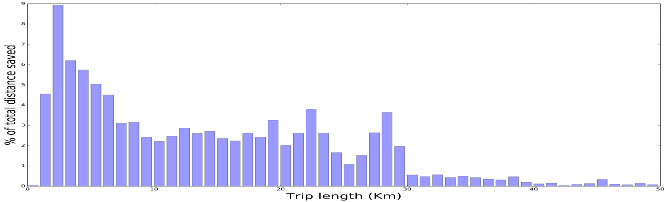


(b) Percentage distribution of reduction of travel distances

Poolability Savings: Los Angeles

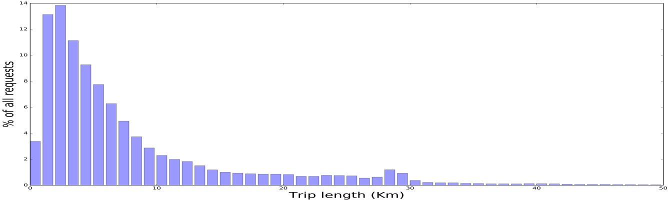


(a) Percentage distribution of trip distances; Mean distance: 9.88km

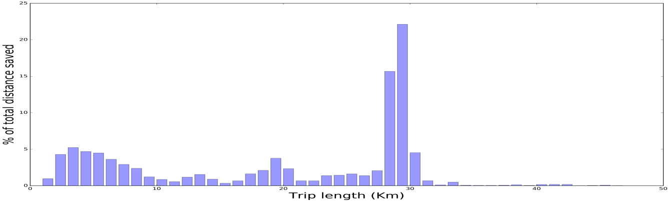


(b) Percentage distribution of reduction of travel distances

Poolability Savings: Chicago

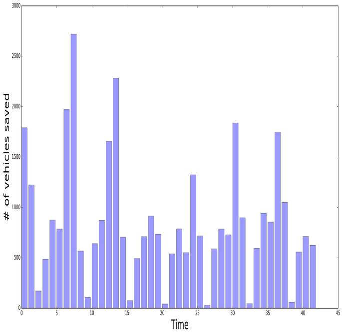


(a) Percentage distribution of trip distances; Mean distance: 8.14km

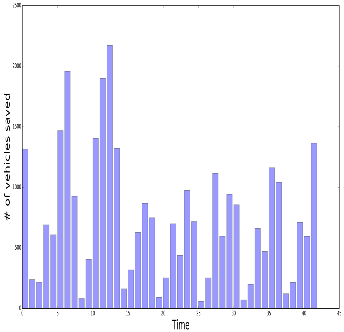


(b) Percentage distribution of reduction of travel distances

Poolability Savings: Vehicle Reductions



(a) San Francisco; Mean 4 hour reduction: 853



(b) New York; Mean 4 hour reduction: 739

Figure: Vehicle reduction plot over time for a week.

Poolability Cost: Travel Time

City	*Mean*	*95th Percentile*
San Francisco	38.49	374
New York	49.34	397
Los Angeles	1.70	274
Chicago	25.70	377

Table: Travel time cost (seconds) due to poolability with $\Delta t = 5min, \Delta s = 100m, \Delta d = 1000m$

Poolability

Observation: Simple pooling algorithms can yield good savings given the observed distribution of travel distances with minimal overhead of travel times.

Part 4a: Our story

The Plot of Our Story

Act 1 Model temporal and *spatial* patterns of mobility.

The Plot of Our Story

Act 1 Model temporal and *spatial* patterns of mobility.

Act 2 Improve placement of vehicles.

The Plot of Our Story

Act 1 Model temporal and *spatial* patterns of mobility.

Act 2 Improve placement of vehicles.

Act 3 Study potential of poolability.

Part 4b: Discussion

<u>Question:</u> Is there a self-similar pattern spatially on how humans move?

Question: How can we rigorously model, and predict about human mobility patterns both temporally and spatially?

Questions?