The AAAI-17 Workshop on
AI and Operations Research for Social Good
WS-17-01

Space-Time Graph Modeling of Ride Requests Based on Real-World Data

Abhinav Jauhri,[1] Brian Foo,[2] Jérôme Berclaz,[2] Chih Chi Hu,[1]
Radek Grzeszczuk,[2] Vasu Parameswaran,[2] John Paul Shen[1]
[1]Carnegie Mellon University, USA; [2]Uber Technologies, Inc., USA
{ajauhri, chihhu, jpshen}@cmu.edu; {bfoo, jrb, radek, vasu}@uber.com

Abstract

This paper focuses on modeling ride requests and their variations over location and time, based on analyzing extensive real-world data from a ride-sharing service. We introduce a graph model that captures the spatial and temporal variability of ride requests and the potentials for ride pooling. We discover these ride request graphs exhibit a well known property called "densification power law" often found in real graphs modelling human behaviors. We show the pattern of ride requests and the potential of ride pooling for a city can be characterized by the *densification factor* of the ride request graphs. Previous works have shown that it is possible to automatically generate synthetic versions of these graphs that exhibit a given densification factor. We present an algorithm for automatic generation of synthetic ride request graphs that match quite well the densification factor of ride request graphs from actual ride request data.

1 Introduction

Recent emergence of ride-sharing services is transforming human mobility and transportation in major cities of the world (Buzzfeed 2016). In December 2015, Uber Technologies, Inc. reported completion of a billion rides (Fortune 2015) within five years since it started operations. Didi alone in China reported 1.4 billion ride requests in 2015 (Wired 2016). There is huge potential for such services to transform urban transportation, public policies, and city-scale services.

Prior works have assessed the potential benefits of ride sharing services. More efficient human transportation at the city scale can play a key role in contributing to sustainability. Most previous studies were based on limited amount of data from a handful of cities over a short span of time. Our work is based on extensive ride request data from the Uber ride-sharing service, which has a global footprint covering several hundred cities. We examine ride request data from 40 cities across the world covering a time span of many weeks.

Examining the extensive ride request data from Uber, we quickly observed that the ride request patterns exhibit significant variability from city to city. Furthermore, within each city, ride requests also vary across regions of the city, on different days of the week, and at different times of the day. However, we also observed that, for most cities, there is a strong pattern that tends to repeat on a weekly basis, as shown for San Francisco in Figure 1. Hence, effective modeling of ride requests must capture the variability in both the spatial and temporal dimensions, but can use one week as the representative time period.

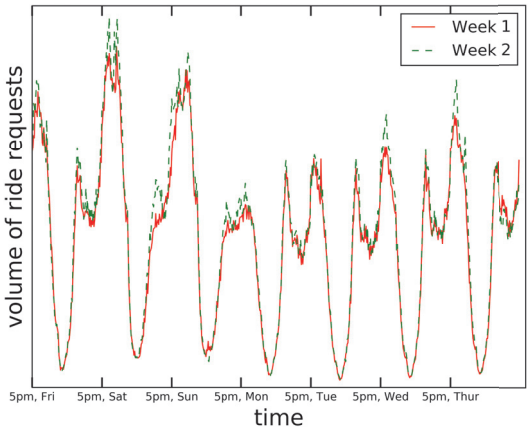


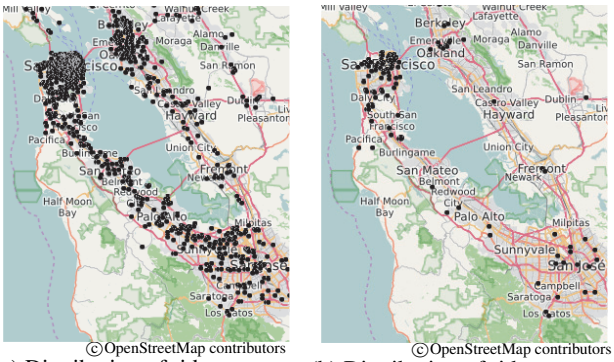
Figure 1: Similarity in the weekly pattern of ride requests for San Francisco for two different weeks.

Several ride-sharing services have recently introduced the notion of ride pooling (combining multiple ride requests into one vehicle) for improving overall service efficiency and at the same time reducing the number of vehicles on the road which can potentially help alleviate traffic congestion. Effective ride pooling requires bundling ride requests that occur in close proximity in both time and location.

This paper makes the following five key contributions: (1) We introduce a new graph model of ride-sharing services that captures both the temporal and spatial attributes of ride requests; (2) We discover that ride request graphs (RRGs) from this model exhibit the well known "densification power law" (DPL) property often found in real graphs modeling human behaviors (Chakrabarti and Faloutsos 2012); (3) We show it is possible to automatically generate synthetic versions of RRGs that exhibit the same DPL degree as the RRGs extracted from real world data; (4) We introduce a new concept called "ride poolability" that captures the frac-

tion of ride requests that can potentially be pooled; and (5) We show there is a direct correlation between the DPL degree of RRGs and the level of ride poolability of a city.

The paper is organized as follows: In Section 2 we provide a survey of related work. In Section 3 we present space-time evolution of ride requests based on extensive ride request data from cities around the world. We then introduce a concise space-time graph model of ride requests in Section 4. We show that ride request graphs (RRG) exhibit the well known densification power law (DPL) often found in real graphs. In Section 5 we show that it is possible to automatically generate synthetic version of RRGs that exhibit the same DPL degree as RRGs extracted from actual ride requests. In Section 6 we introduce ride poolability and show the direct correlation between the DPL degree and the level of ride poolability. Finally, we summarize our key findings and suggest promising directions for future work in Section 7.



(a) Distribution of ride requests at rush-hour time, 7pm

(b) Distribution of ride requests at non-rush-hour time, 5am

Figure 2: Space-time variability: Each dot represents a source or destination for a ride request in San Francisco[2]

2 Prior Work

Recent studies looked at different formulations to show the potential of ride pooling. (Burns, Jordan, and Scarborough 2013) uniformly distribute ride requests in a geographical area. They model the performance of fleet of vehicles as a queuing system where vehicles are servers and trip requests are customers. Using such an analytical model they derive average capacity utilizations, wait times, and total costs. (Lu 2014) study optimizing the number of miles driven by drivers by pooling riders; ride requests are generated uniformly over square blocks. (Wang 2013) also simulate data for ride requests in the city of Atlanta to study benefits of matching riders. (Knapen et al. 2015) formulate a graph with nodes as users, and edges indicating whether or not a negotiation between two users is possible to carpool. The data used for users here was a synthetic population for a geographical area. (Kamar and Horvitz 2009) use real-world data on trips from a limited part of a city to highlight significant reduction of carbon dioxide per year by having multiple riders share the same vehicle. (Bicocchi and Mamei 2014) developed a recommender system capable of identifying riders which could be pooled by looking at users' location data when they send or receive calls or text messages. (Stiglic et al. 2015) study the benefits of meeting points in a ride sharing system from generating synthetic ride requests within limited distance. (Shmueli et al. 2015) use a graph model to analyze real data set of taxi trips in New York City for assessing the potential of ride pooling.

Previous works have used either synthetic models to generate data or real data which may not accurately represent locations where ride requests originate or terminate. Even if it is representative, temporal changes which affect ride requests have not been considered. For instance, users' locations for making phone calls in the afternoon from offices may not necessarily imply people travel often at the same time from offices. In fact, comparatively low number of ride requests are associated with afternoons on any weekday. For ride pooling to be some percentage of total ride requests such that civic bodies can make decisions, or for ride requests pattern to be studied for its potential of pooling (Huang et al. 2014) and to improve arrival time (Cao et al. 2016) based on anticipated congestion, we need to understand the laws which govern ride requests at any geographical area, and at any given time.

This work models ride requests as a graph. Some important prior works on graphs have helped us to model and discover properties for ride requests. There is a class of graphs that models real networks, e.g. social network graphs and publication citation graphs, that evolve over time. These graphs become more and more dense as they evolve in time, i.e. the edge count grows superlinearly relative to the node count growth. This *densification* of the graph can be modeled concisely by a power law relationship between the edge count and the node count. Graphs for many real networks all seem to exhibit this densification power law (DPL) (Newman 2005). For this class of graphs it is possible to automatically generate synthetic graphs that exhibit the same densification power law without needing the original data (Leskovec, Kleinberg, and Faloutsos 2005). These graphs also exhibit the attribute of having strongly connected subgraphs or communities (Fortunato 2010). In this work, we have discovered that our space-time graph model of ride requests for a city belongs to this class of graphs. This fact allows us to discover interesting attributes and insights about ride requests and the potential of ride pooling for ride-sharing services.

3 Space-Time Evolution of Ride Requests

This work is based on extensive real world data from Uber, a ride-sharing service with global presence in several hundred cities. We examine data from 40 cities with average daily city-wide ride requests ranging from 2K to 200K per day. [1]

[1]This paper presents results for only four cities. We have similar results for all 40 cities based on a total of about 50M ride requests.

[2]Map data of Figures 2, 3, and 5 are available from OpenStreetMap under the Open Database License and the cartography is licensed under the Creative Commons Attribution-ShareAlike 2.0. http://www.openstreetmap.org/copyright

Each ride request involves a source location s, destination location d, and the time of the request t. Both s and d are represented by their latitude and longitude. Each ride request is considered independent of any other ride request. In this study we do not consider the actual navigation path taken from s to d for a ride request.

Based on examining the ride request data from the 40 cities gathered over several months in the Spring of 2016, we can make two key observations. For each city, the spatial and temporal ride request patterns tend to repeat from week to week. On the other hand, there is significant variability of ride request patterns from city to city. Furthermore, in each city there is variability across different days of the week, at different times of the day, and across different regions of the city.

Variability in ride request patterns in San Francisco for two snapshots taken at two different times of the day is shown in Figure 2. The figure shows the spatial distribution of ride request density over the Bay Area. San Francisco downtown (top left cluster of points) is clearly denser in ride requests. The two figures illustrate two snapshots of ride request density for two different 5-minute intervals one at 7:00pm and the other at 5:00am. The temporal and spatial variations of ride requests can be clearly seen.

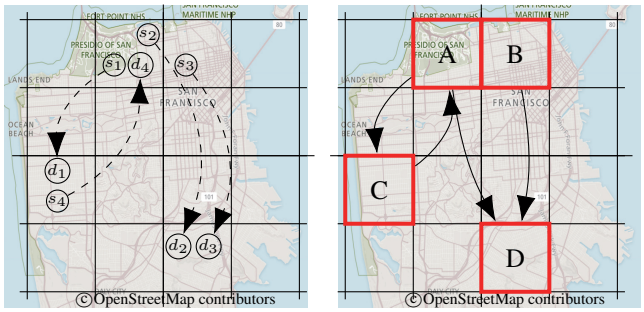
4 Space-Time Graph Model

We now introduce a graph model for ride requests. The graph representing ride requests for a specific time period is called a *Ride Request Graph* (RRG). Each RRG captures the spatial distribution of ride requests across a city within that time period. The RRG evolves from one time period to the next, to account for new ride requests initiated in the next time period. This evolution of the RRG and the resultant sequence of RRGs capture the temporal aspect of ride requests over many time intervals.

Ride Request Graph Generation

For a given time interval we can generate a RRG representing all the ride requests in that interval. We divide the map of a city into equal sized cells of $100m \times 100m$ each. Each cell is considered as a node in the graph only if the source or destination of a ride request falls within that cell. A directed edge connects the source and destination cells of a ride. A directed graph can then be generated to model all the ride requests in that time interval for a given city.

For illustration, consider the ride requests in Figure 3a for a given time interval. The four ride requests are shown on a gridded map (not drawn to scale). The corresponding graph in Figure 3b is formed by four nodes with node A subsuming s_1, s_2, d_4; node B subsuming s_3; node C subsuming s_4, d_1; and node D subsuming d_2, d_3. All edges in Figure 3a have unit weights representing single ride requests. Edge weights represent the number of ride requests from the same source and destination nodes. In this paper, the term *node* is always used in the context of the RRG graph. We use the term *point* to refer to a specific location defined by its latitude and longitude, which could be the source or destination of a ride request, and is associated with a node of RRG.



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

Ride Request Graph Densification

A Ride Request Graph is different from conventional graphs: (1) each node has a geographical area associated with it; (2) RRG is not fixed in time but evolves in time. Each RRG involves a spatial quantization (into $100m \times 100m$ cells) of the geographical space of a city, and a temporal quantization into sequence of time intervals. In this work we use 5-minute intervals for temporal quantization. As an RRG evolves in time, it produces a sequence of RRGs that capture the temporal behavior of ride requests.

As we analyze the RRGs extracted from historical data of ride requests from all the cities, we make an interesting observation about these RRGs. *Densification* refers to graphs that evolve in time, and how the edge count grows relative to the growth of the node count. Many graphs modeling aspects of human behaviors, such as social network graphs and publication citation graphs, among others, exhibit densification over time that follows a power law, i.e. the number of edges grows as a power of the number of nodes (Leskovec, Kleinberg, and Faloutsos 2007). We have discovered that the RRGs for ride requests exhibit the same power law densification behavior and belongs to this class of graphs.

We observe RRGs at different snapshots of time, with each spanning five minutes. For each snapshot, we study the *Densification Power Law* plot (DPL plot) (Leskovec, Kleinberg, and Faloutsos 2005) i.e. log-log plot of the number of edges $e(t)$ versus number of nodes $n(t)$.

Top row in Figure 4 shows the Densification Power Law (DPL) plot for four cities based on real data for a typical week in 2016. It is observed that for every time interval t:

$$\begin{aligned} e(t) &\propto n(t)^{\alpha} \\ &= Cn(t)^{\alpha}, \end{aligned} \tag{1}$$

where $e(t)$ and $n(t)$ are the number of edges and number of nodes respectively, formed by all ride requests occurring in the time interval t. C and α are constants. Number of edges is a good approximation of the number of ride requests. We observe that all the cities follow the densification power law but the parameters of the power law vary from city to city.

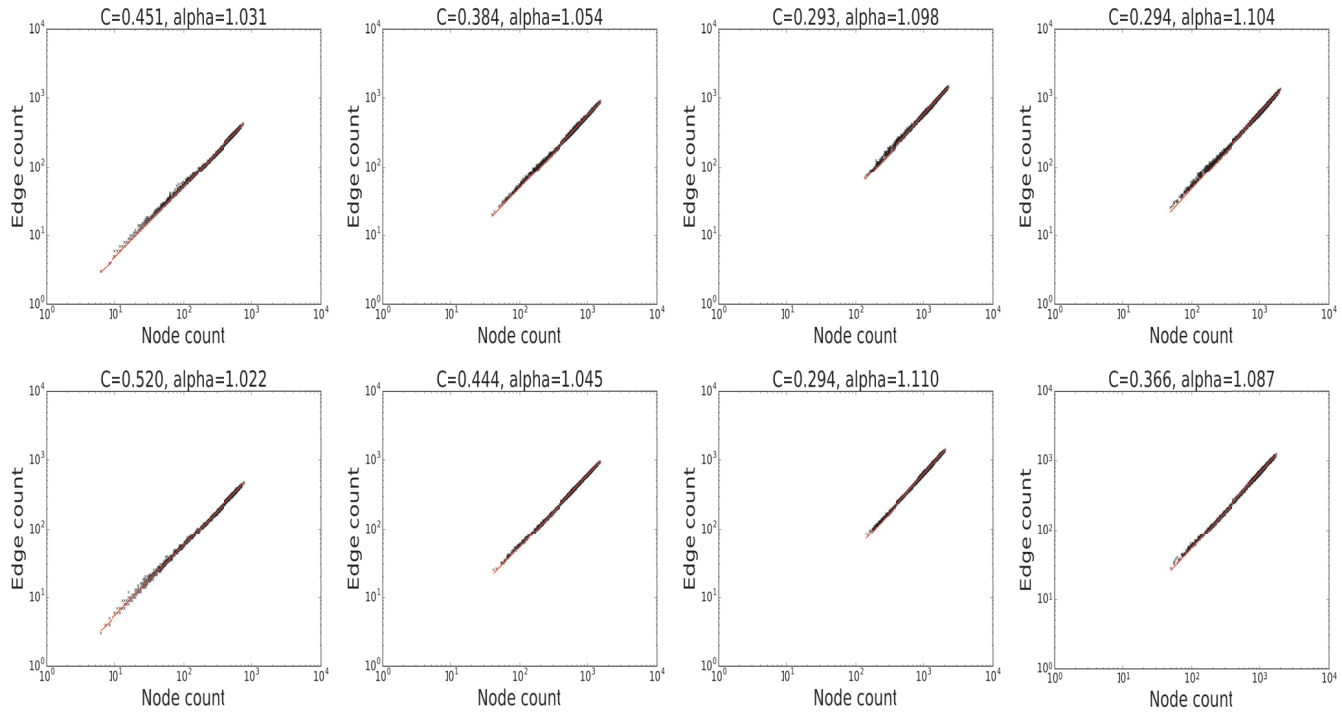


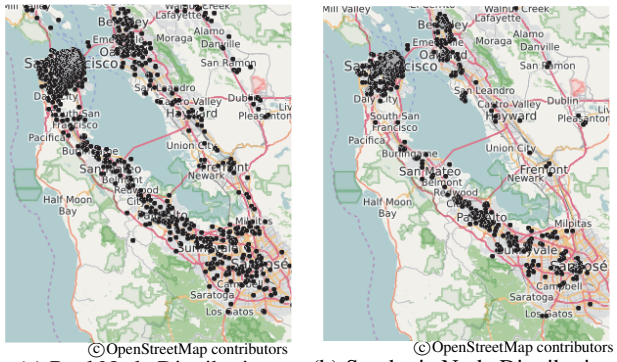
Figure 4: DPL plots from real data (top row) and synthetic data (bottom row) for four cities. The red line is the least square fit of the form $y = Cx^{\alpha}$, where y and x are number of edges and nodes respectively. $R^2 \approx 1.00$ for all of them.

Characteristic Attributes of RRGs

It appears that the pattern of ride requests for a city can be characterized concisely by C, and α derived from the power law of RRGs for that city (see top row of Figure 4). The exponential α depicts the densification of ride requests within a city. This *densification factor* α can range in value from 1.0 to 2.0. If $\alpha = 1.0$, this means the number of edges is growing linearly with respect to the number of nodes; if $\alpha = 2.0$, then the RRGs become fully connected graphs.

It is interesting to note that all four cities exhibit densification factors greater than 1.0. This means the edge count is growing superlinearly to the node count, implying the densification of ride requests. We speculate this demonstrates the human tendency towards the creation of clustered/connected communities, perhaps reflecting the *small world* effect (Watts and Strogatz 1998).

DPL graphs exhibit a fascinating attribute. (Leskovec, Kleinberg, and Faloutsos 2005) and (Chakrabarti and Faloutsos 2012) have shown that for graphs that evolve according to the densification power law, it is possible to automatically generate these graphs that exhibit specific densification factors. This means that we can automatically generate RRGs that exhibit similar densification factor as that of RRGs extracted from real data. This can potentially allow us to generate synthetic RRGs, exhibiting similar densification factor, for a city without needing the real ride request data.



(a) Real Node Distribution (b) Synthetic Node Distribution

Figure 5: Plots of nodes for San Francisco for a single time interval of five minutes. On the left, is the real spatial distribution, and on the right is the synthetically generated.

5 Synthesized Space-Time Graph Model

In this section, we explore the automatic generation of synthesized space-time graph models that mimic the attributes of RRGs generated using real ride request data. The two attributes of interest are: 1) the spatial distribution of nodes; and 2) the temporal evolution or the densification factor of the RRG.

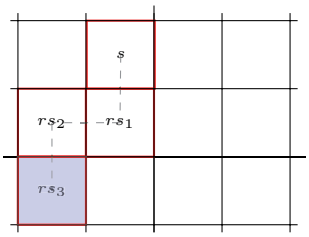


Figure 6: Random walk starting at node s with three random steps to reach node rs_3. Final point is selected uniformly random within the blue colored grid.

Spatial Properties

Spatial properties provide information on the source and destination locations of ride requests as shown in Figure 3a. The synthesized graph model should capture the spatial distribution of these ride request locations.

To capture spatial properties, we compute the likelihood of a node to either possess a source or destination location by using geospatial information from OSM's public data (OpenStreetMap 2016) on node density[3]. To avoid confusion with a node of the Ride Request Graph, we shall refer to an OSM node[4] as *Point of Interest* (PoI). A PoI is defined by a tuple of latitude and longitude. PoIs are used to define standalone features such as traffic signals, businesses, schools, hospitals, and many others. Alternatively, any other dataset providing a quantitative measure over geographical space which is correlated with the likelihood of ride requests can also be used.

Algorithm 1 performs node selection using vector of probabilities $pr \in \mathbb{R}^n, n = |S|$, where S is a subset of nodes of interest. pr is computed by aggregating all PoIs present at a RRG node, and then normalizing to get the probability mass function across nodes in S. Algorithm 1 takes as input the number of synthetic points m to be generated, and associates each synthetic point to an initial node; this is determined from prior probability vector pr. Once the initial node is chosen, Algorithm 2 performs a random walk starting from initial node centroid such that the synthetic points are spread out in the geographical area (Figure 6). Since PoI data could be sparse, we use a kernel density estimation function K[5] over the geographical space to guide the random walk.

In Algorithm 1 method *randomChoice* generates m points with replacement using the prior probabilities vector pr; *geoCoords* returns the latitude and longitude associated with the node label; *perturb* performs a uniform random selection within the final node (blue area in Figure 6) to determine the final location of the newly generated point.

In Algorithm 2, method *randomStep* chooses a node amongst the neighbouring eight nodes (or less) of the current node, $curr$, by normalizing the probabilities returned by the kernel density estimates; it returns the new node new, and a reward r. We kept reward equal to the probability estimate for current node, $r = K(curr)$. Note that every node is defined by latitude, longitude coordinates of its centroid.

[3]OSM data is publicly available at http://download.geofabrik.de/

[4]http://wiki.openstreetmap.org/wiki/Node

[5]We used Gaussian Kernel Density Estimation library http://statsmodels.sourceforge.net/devel/generated/statsmodels.nonparametric.kernel_density.KDEMultivariate.html

Algorithm 1 To capture spatial properties. Inputs: Kernel density estimation function K, number of synthetic points $m \in \mathbb{N}$, prior probability vector $pr \in \mathbb{R}^n$

```
1: procedure SPATIALPROPGEN(K, m, pr)
2:     labels = randomChoice([0, ..., len(pr)], m, pr)
3:     pts = []
4:     for each point i ∈ [0, m) do
5:         l = labels[i]
6:         s = geoCoords(l)
7:         p = RANDOMWALK(K, s, max_r, max_s)
8:         add(pts, perturb(p))
9:     end for
10:    return pts
11: end procedure
```

Algorithm 2 Random Walk. Inputs: Kernel density estimation function K, start location s, maximum reward, and maximum number of steps max_r & max_s

```
1: procedure RANDOMWALK(K, s, max_r, max_s)
2:     let tot_r = 0
3:     let n_steps = 0
4:     let curr = s
5:     while tot_r ≤ max_r and n_steps ≤ max_s do
6:         curr , r = randomStep (curr, K)
7:         n_steps = n_steps + 1
8:         tot_r = tot_r + r
9:     end while
10:    return curr
11: end procedure
```

Densification Properties

In the previous subsection, we only distribute points spatially such that they are either source or destination locations. Our model to connect these points such that they become concrete ride requests is described in Algorithm DENSPROPGEN. This model allows us to capture the *densification* property observed in RRGs from real data. Algorithm 3 requires three parameters: (1) number of points to generate m; (2) the probability of choosing a point which has not been visited before p; (3) number of outlinks n_{edges} from a source point is defined by geometrically distributed random number with mean $1/q$. $1 - p$ is the probability of choosing a previously visited source point as destination (variation of the preferential attachment technique described in (Chakrabarti and Faloutsos 2012)) which captures the idea of *rich getting richer*.

In Algorithm 3, *uniformRandomChoice* uniformly at random selects a point from the set points. *geometricRandom* generates values from a geometrically distributed random variable with success probability q. *uniformRandom* generates a uniformly random value $\in [0, 1)$.

Synthesized RRG Model

Our complete model which embodies all aspects of the synthesized RRGs is as follows:

Step 1: Use OSM data to aggregate PoI count for each node.

Algorithm 3 To capture densification properties. Inputs: $m \in \mathbb{N}$, p & $q \in [0, 1]$

procedure DENSPROPGEN(m,p,q)
  let $M = [0, \ldots, m-1]$
  let $R = []$
  let $rides = []$
  while length(M) > 1 **do**
    s = *uniformRandomChoice*(M)
    remove(M, s)
    n_{edges} = *geometricRandom*(q)
    for each edge $e \in [0, n_{edges})$ **do**
      if *uniformRandom*$() < p$ **then**
        d = *uniformRandomChoice*(M)
        remove(M, d)
      else
        d = *uniformRandomChoice*(R)
      end if
      $r =$ connect(s, d)
      add$(rides, r)$
    end for
    add(R, s)
  end while
  return $rides$
end procedure

Step 2: Choose a subset of nodes of interest S [6].

Step 3: Calculate prior probabilities based on PoI count to get vector pr on the subset of nodes.

Step 4: Compute Gaussian kernel density function K using centroids of nodes in S.

Step 5: Use SPATIALPROPGEN with prior probabilities, number of points to be generated, and the kernel function K to generate synthetic points over space.

Step 6: Use synthetic points to generate synthetic ride requests using DENSPROPGEN with parameters p and q.

Step 7: Create the Ride Request Graph using the synthetic ride requests returned by DENSPROPGEN.

One can repeat the steps above for consecutive time intervals.

Comparison of Graph Models

Figure 4 (bottom row) provides plots for the synthesized graph model with densification factors very similar to those from RRGs generated from real data in Figure 4 (top row). Figure 5 shows the spatial distribution produced by Algorithms SPATIALPROPGEN and RANDOMWALK for the city of San Francisco. The plots show nodes (aggregation of points) in an RRG for a single time interval for both the real RRG (left) and the synthesized RRG (right). The nodes are more closely packed in the synthetic plot which is due to bias induced by the prior probability distribution using PoI density. Certain geographical areas in Figure 5b have no nodes in comparison to Figure 5a, also due to the prior PoI density distribution being low in sparse areas.

6 Ride Request Poolability

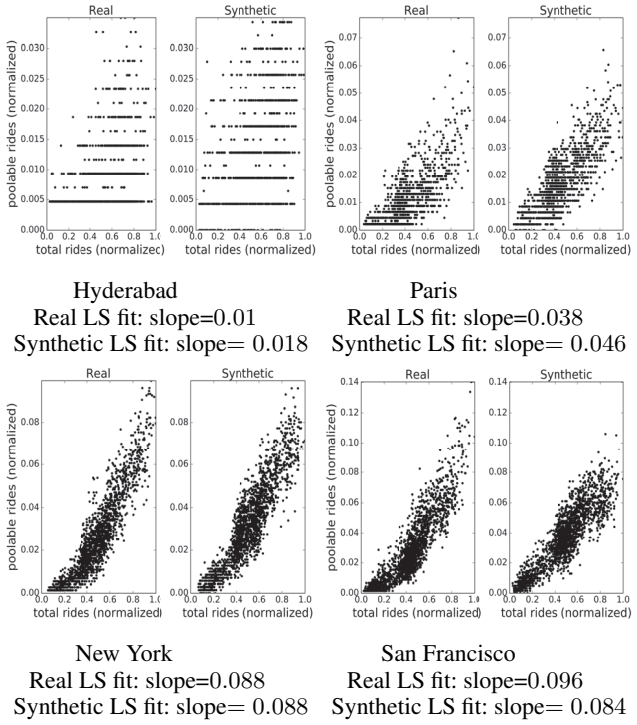
Recently, ride-sharing services have started offering the option of ride pooling by matching similar ride requests in real time. Such ride or rider matching problem is similar to the well studied combinatorial problem commonly referred to as the *Vehicle Routing Problem* (Laporte 1992). An interesting and powerful approach to maximize pooling and minimize costs is by advanced scheduling, wherein a rider provides a time period in the future for pick-up, and the ride-sharing service performs matching within the time period.

In this work, we instead consider the potential for on-demand ride pooling, i.e. pooling rides not scheduled in advance. We first focus on assessing the potential of ride pooling based on historical data. We examine all the ride requests in a city and attempt to bundle ride requests within certain proximity constraints in both space and time. For example, we can pool ride requests initiated within a 5 minute window, with requesting locations less than 100m apart.

Consider a set of rides P ordered by time of request, such that $|P| = p, p > 1$; the first ride to occur in time in P is referred to as the *master* ride. Then *master* is *poolable* with any request $\in P \setminus \{master\}$ if the following constraints are satisfied:

1. both ride requests are requested within Δt minutes.
2. source locations of both requests are within Δs meters radius.
3. destination locations of both requests are within Δd meters radius.

All rides in such proximity with *master*, and *master* itself are removed from P, and the above steps are repeated with a new *master* being the next earliest ride request in P. Any requests that remain unmatched are considered not *poolable*.



Hyderabad
Real LS fit: slope=0.01
Synthetic LS fit: slope= 0.018

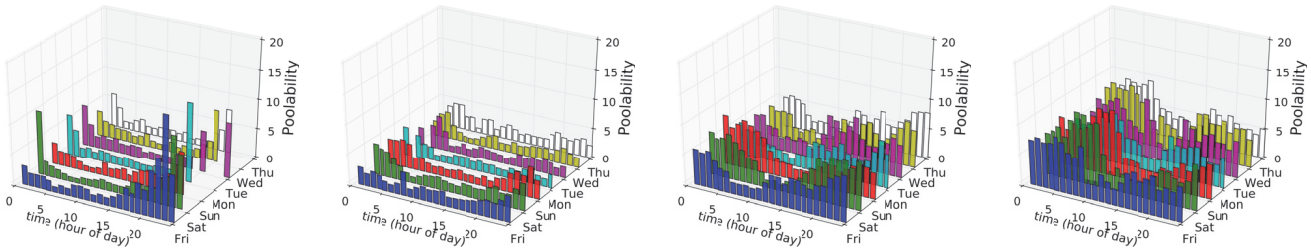
Paris
Real LS fit: slope=0.038
Synthetic LS fit: slope= 0.046

New York
Real LS fit: slope=0.088
Synthetic LS fit: slope= 0.088

San Francisco
Real LS fit: slope=0.096
Synthetic LS fit: slope= 0.084

Figure 7: Scatter plots of poolable rides vs. total rides for 2016 5-minute intervals in a week, based on the RRGs from real data (left) and the synthesized RRGs (right).

[6]For our experiments we selected the nodes from historical data where ride requests happened.



Hyderabad Paris New York San Francisco

Figure 8: Poolability metric for four cities for a week of data with $\Delta t = 5min, \Delta s = 100m, \Delta d = 1000m$. Time is in GMT.

City	*Mean*	*Minimum*	*Maximum*
Hyderabad	2.23	0.84	7.41
Paris	2.39	0.79	4.22
New York	4.48	1.70	7.84
San Francisco	5.48	2.50	9.16

Table 1: Overall mean, minimum, and maximum poolability for four cities for a week of data with $\Delta t = 5min, \Delta s = 100m, \Delta d = 1000m$

We define *poolability* as the percentage of rides that can be pooled. In Figure 8 we plot the *poolability* (z-axis) for four cities. The poolability data is shown for each day of the week. Summary of poolability metric for four cities is provided in Table 1. Poolability in Hyderabad shows maximum variability with minimum of 0.84, and maximum of 7.41.

Paris and San Francisco exhibit quite different degrees of poolability. San Francisco consistently exhibits higher poolability, with a consistent daily pattern for weekdays. For each day there are two time periods, matching the morning and evening rush hours, that exhibit significantly higher poolability. We suspect the key difference between Paris and San Francisco is due to the topology and terrain of the two cities. This is a very interesting area for future research.

Ride Poolability Attributes

We also observe that the poolability of a city is directly correlated with its densification factor. Cities with higher α always exhibit higher poolability. Comparing Figure 8 with the top row of Figure 4 we see that $\alpha = 1.031, 1.054, 1.098, 1.104$, for Hyderabad, Paris, New York, and San Francisco, respectively. This ordering matches exactly the ordering of poolability in Figure 8 and Table 1 (mean poolability).

Synthesized RRG Poolability

Figure 7 provides a comparison of *poolability* obtained using real data and randomly generated data. The slope of the straight line fitted to real and synthetic plots suggests that the synthesized graph model is a relatively good fit to the real *poolability*. There are instances where the synthesized version over or under predicts poolability. This is most likely due to the lack of spatial information of ride requests distributed over time. Node density information from OSM is not dynamic relative to the time of day, hence there is a bias towards generating points in high PoI density regions which may not hold for consecutive time intervals.

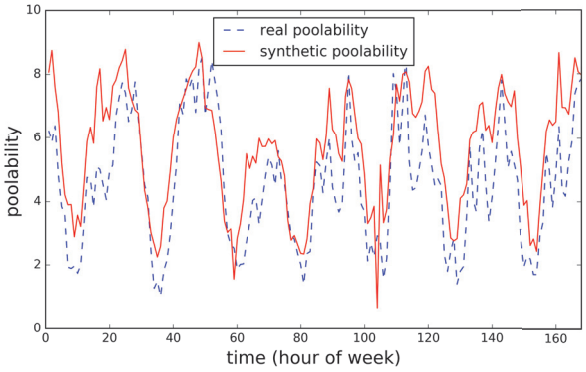


Figure 9: Comparison of *Poolability* generated by synthetic (red line) and real (dotted blue line) models for New York. RMSE: 1.54, abs. delta min=0.02, abs. delta max=3.17

Figure 9 shows how the *poolability* metric varies over 168 hours (starting with 8pm on Friday) of a typical week. The synthetic model captures the temporal variations and matches well with the *poolability* from the real ride request data. For the synthetic model, the total number of ride requests were kept equal to the number of ride requests in real data. The first three peaks in the plot depict evening hours for Friday, Saturday, and Sunday which on average are higher than the remaining four peaks in the plot.

7 Conclusion

The emergence of ride sharing services and the availability of extensive data from such services is creating unprecedented opportunities for: doing large scale data analytics on urban transportation; gaining new insights on human mobility; and facilitating new public services for societal benefit. This work is an initial attempt at this. The key contributions of this paper include:

- Based on extensive real world data, we introduce a space-

time framework for modeling ride requests in a city, and the notion and analysis of ride poolability.

- We introduce a space-time graph model for modeling ride requests in a city and show that these graphs exhibit power law densification as they evolve in time.
- Based on the densification power law, we show that the pattern of ride requests and ride poolability for a city can be concisely characterized by the densification factor of its ride request graphs.
- We further show that the degree of ride poolability of a city is directly correlated to the densification factor of its ride request graphs.
- Using previous work, we show the space-time ride request graph model for a city can be automatically generated.
- We further show the attributes of the generated synthetic graphs match quite well the attributes of graphs extracted from real ride request data.

We have only scratched the surface in this paper. There are many promising avenues for further research. Some open research questions include:

1. If the ride pooling proximity constraints, both temporal and spatial can be relaxed, is it possible to significantly improve ride poolability?
2. Can the temporal and spatial variation of ride poolability be leveraged to create intelligent ride pooling algorithms?
3. Is it possible to significantly reduce the number of vehicles needed on the road through aggressive ride pooling?
4. Can we rigorously characterize the relationship between the degree of ride poolability and the densification factor of ride request graphs?
5. Is it possible to use insights from historical ride request data for real-time traffic congestion prediction and potentially alleviation?
6. Comparison of RRG generation with existing graph generators over space and time.

Acknowledgements: We are grateful to Peter Frazier and Jon Petersen for many useful discussions. We also thank the anonymous referees for their helpful comments.

References

Bicocchi, N., and Mamei, M. 2014. Investigating ride sharing opportunities through mobility data analysis. *Pervasive and Mobile Computing* 14:83–94.

Burns, L.; Jordan, W.; and Scarborough, B. 2013. Transforming personal mobility. the earth institute, columbia university, new york.

Buzzfeed. 2016. People in Los Angeles Are Getting Rid of Their Cars. https://www.buzzfeed.com/priya/people-in-los-angeles-are-getting-rid-of-their-cars. [Online; accessed 9-September-2016].

Cao, Z.; Guo, H.; Zhang, J.; and Fastenrath, U. 2016. Multiagent-based route guidance for increasing the chance of arrival on time.

Chakrabarti, D., and Faloutsos, C. 2012. Graph mining: laws, tools, and case studies. *Synthesis Lectures on Data Mining and Knowledge Discovery* 7(1):1–207.

Fortunato, S. 2010. Community detection in graphs. *Physics reports* 486(3):75–174.

Fortune. 2015. Uber Completes 1 billion rides. http://fortune.com/2015/12/30/uber-completes-1-billion-rides/. [Online; accessed 26-August-2016].

Huang, Y.; Bastani, F.; Jin, R.; and Wang, X. S. 2014. Large scale real-time ridesharing with service guarantee on road networks. *Proceedings of the VLDB Endowment* 7(14):2017–2028.

Kamar, E., and Horvitz, E. 2009. Collaboration and shared plans in the open world: Studies of ridesharing.

Knapen, L.; Hartman, I. B.-A.; Keren, D.; Cho, S.; Bellemans, T.; Janssens, D.; Wets, G.; et al. 2015. Scalability issues in optimal assignment for carpooling. *Journal of Computer and System Sciences* 81(3):568–584.

Laporte, G. 1992. The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research* 59(3):345–358.

Leskovec, J.; Kleinberg, J.; and Faloutsos, C. 2005. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, 177–187. ACM.

Leskovec, J.; Kleinberg, J.; and Faloutsos, C. 2007. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 1(1):2.

Lu, W. 2014. Optimization and mechanism design for ridesharing services. Technical report.

Newman, M. E. 2005. Power laws, pareto distributions and zipf's law. *Contemporary physics* 46(5):323–351.

OpenStreetMap. 2016. OpenStreetMap contributors. (2016) Planet dump. http://http://wiki.openstreetmap.org/wiki/Main_Page. [Online; Data file accessed till 26-August-2016].

Shmueli, E.; Mazeh, I.; Radaelli, L.; Pentland, A. S.; and Altshuler, Y. 2015. Ride sharing: a network perspective. In *International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction*, 434–439. Springer.

Stiglic, M.; Agatz, N.; Savelsbergh, M.; and Gradisar, M. 2015. The benefits of meeting points in ride-sharing system/less. *Transportation Research Part B: Methodological* 82:36–53.

Wang, X. 2013. Optimizing ride matches for dynamic ridesharing systems.

Watts, D. J., and Strogatz, S. H. 1998. Collective dynamics of small-worldnetworks. *nature* 393(6684):440–442.

Wired. 2016. Didi Kuaidi Announces 1.43 billions Rides. http://www.wired.com/2016/01/didi-kuaidi-announces-1-43-billion-rides-in-challenge-to-uber/. [Online; accessed 26-August-2016].