Improving Traffic Flow Using Uber Movement Data

Mackenzie Pearson
pearson3@stanford.edu

Javier Sagastuy
jvrsgsty@stanford.edu

Sofía Samaniego
sofiasf@stanford.edu

Abstract

Utilizing the Movement data set recently released by Uber, we propose to analyze and discern traffic bottlenecks in multiple cities. Further, we plan to uncover general traffic movement patterns throughout these cities at various times in the day and identify the main differences in mobility behavior between them. Finally, we plan to come up with a solution to alleviate traffic on our identified bottlenecks and measure its impact through simulated data.

1 Introduction

Studying travel patterns and the structure of cities has long been a research topic of great interest in urban planning. In the past decade, this interest has spiked due to an increase in availability of GPS and mobile phone data. Further, open-data initiatives such the New York City Open Data Project provided researchers with taxi trajectory data that enabled them to investigate different aspects of traffic flow.

While there has been a lot of work in this area, GPS taxi trajectory data sets used in previous work usually only comprised small periods of time; hence, a temporal component was seldom included in traffic flow models. However, with the boom of the digital age, human mobility and location data has dramatically increased in volume. We now have data of over two billion Uber trips in seven cities around the world starting in 2016, which is significantly more data than any other study in this topic that we've encountered. We believe that analyzing this brand new and very powerful data set could help us learn a lot about the future of urban mobility. In particular, our goal is to uncover traffic bottlenecks within a given city at different hours of the day by using travel times between sources and destinations of over two billion Uber trips in the past two years.

We want to identify mobility patterns that exist in cities and compare them against each other. Further, we aim to construct a sequence of graphs built with Uber trip durations across different times of day and use this information to identify bottlenecks in traffic. We have reasons to believe the introduction of affordable ride-share services such as Uber has significantly increased the scope and volume of people that chose cabs as a primary mode of transportation; hence, coming up with solutions to these bottlenecks could have a great positive impact on millions of people!

2 Previous Work

In this section we summarize and critique three papers that use GPS taxicab data as a tool to approach urban dynamics from different perspectives. These papers cover a broad range of topics such as different approaches for estimating urban flow, methods for uncovering travel communities and patterns in a city's structure, and techniques for identifying locations of traffic-flow co-behavior and potential flaws in city planning.

2.1 Understanding traffic flow characteristics

Historically, several authors have claimed that the configuration of a city's street network plays an important role in vehicular flow and, hence, used centrality measures of a street graph to model and predict traffic. Specifically, authors such as Turner [5] proposed betweenness centrality as a good predictor of traffic flow. We focus on the work of Gao, et al. [2], who criticized this approach and proposed a new model of traffic flow based on the non-uniform distribution of human activity and the distance-decay law.

Gao, et at. argue that the betweenness centrality measure of a street network is static and thus can't be used to model the dynamic behavior of traffic demands. Further, the authors claim that this measure does not take into account the fact that travel demand (i.e. traffic flow) depends on the distance between origin and destination and, in particular, is decreasing as a function of trip length. To support their critique, the authors compute the weighted correlation (with weights given by street length) between "real" traffic flow, estimated through the line-density method using a one-week long GPS data set of 149 taxis in the core urban area of Jiaozhou Bay, and the betweenness centrality measure of the nodes of this city's street primal and dual networks. They find that this measure is not ideal by itself to predict urban traffic flow.

The alternative approach proposed by the authors is to construct a trip demand model that incorporates the heterogeneity in real human activities and the decay-distance law in trip demand. Specifically, they use the total call-traffic volume of base stations in Jiaozhou Bay in one hour, namely the Erlang values, to model the sample probabilities of origin and demand pairs (OD). Meanwhile, they model the probability of an edge existing between a sampled OD pair through a distribution that decreases exponentially as a function of the distance between the origin and distance nodes (power-law). Using this method they run Monte Carlo simulations to generate trip data and produce an estimate of traffic flow. Finally, they use weighted correlation to measure goodness of fit between their simulated and observed taxi trajectory data. They conclude that the proposed model can interpret urban traffic flow well.

2.2 Revealing travel patterns and city structure

In 2015, Liu, et al. [4] presented an analysis to infer travel patterns and city structure from data modeling traffic flow. By using taxi trip data from the city of Shanghai, they represented traffic flow as a directed graph and applied modern network analysis techniques to characterize it. Their approach revealed a two-level hierarchical structure of Shanghai based on the length of the taxi trips and contrasted the administrative boundaries of the city with the natural boundaries derived from the travel patterns. A modification of administrative and transportation planning boundaries is proposed to improve local mobility and current traffic analysis modeling to aid in urban planning.

To accomplish this, Shanghai was split into a 1×1 km cell grid; each of the resulting cells representing a node in the graph. Two nodes u and v were connected by a directed edge if a trip originating physically inside u and ending inside v existed. The edges were then weighted according to the number of existing trips between the same cells. Only data from Monday to Thursday was used, since this represents the most constant traffic flow due to an increased number in leisure and entertainment trips near the weekend.

The resulting network was then processed using community analysis to identify regions within which trips were common. Further, the detected communities were characterized by measuring graph density, node strength, closeness centrality and betweenness centrality for each of the nodes in a community. From this analysis, centers with a high degree of traffic flow (measured through node strength) were identified.

2.3 Urban Computing with Taxicabs

Zheng, et al. [6] provide an interesting framework for analyzing taxicab data, which could be relevant for our project. This framework consists of linking pairs of regions (i, j) to three key features:

1. The number of taxis going from region i to region j.
2. The average speed these taxi drives when commuting from region i to region j.
3. The ratio between the actual travel distance and the distance between the centroids of these two regions.

By mapping taxi trajectory data from 30,000 taxis driving in Beijing from March to May in 2009 and 2010 onto this framework Zheng et al. seek flaws in current urban planning.

Flaws are detected by finding obvious issues in these taxicab commutes. For example, if the flow from a region i to region j is high, but the average speed between these two regions is low and the actual distance traveled is high compared to the distance between the centroids of the two regions, then one could conclude that there is high traffic and the detours are slow. Zheng et al. compare and contrast these issues over two years to see if new roads or subways systems have had a clear impact on these problem areas.

3 Preliminaries

In this section we describe the travel time data we worked with, the temporal and spatial graphs we constructed from it, and provide an extension of some key graph definitions to the weighted case that will prove useful in our analysis.

3.1 The data set

This January, Uber unveiled “Uber Movement”, a tool intended for use by city planners and researchers looking into ways to improve urban mobility. The data set includes over two billion Uber trips in the cities of Bogotá, Boston, Johannesburg, Manila, Paris, Sydney, and Washington D.C., Specifically, it includes the arithmetic mean, geometric mean, and standard deviations for aggregated travel times over a selected date-range between every zone[1] pair in each of these cities. Uber Movement is open to the public and can be download in `.csv` format directly from [Uber Movement’s Website].

Uber provides a useful visualization tool to explore the data set which we show in Figure 1. They also provide a download tool on which one may filter the data on a given time range and specify the granularity of the computed aggregates. We performed some exploratory analysis on data from Washington DC during the first quarter of 2016, computing hourly aggregates. For our initial approach we’ll focus only on one one-hour time slice, although for our final project we’ll report how our calculated metrics change over time of day for different cities.

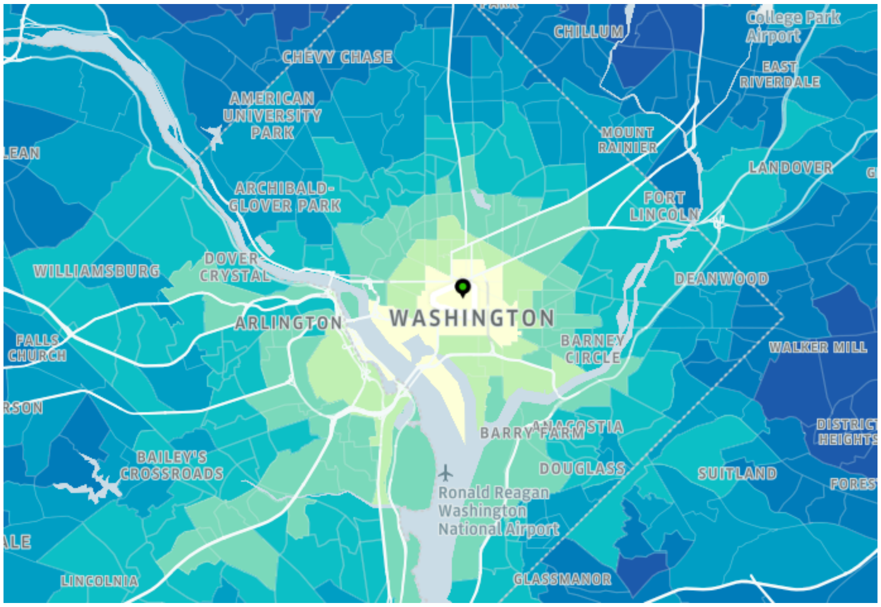

Figure 1: The Uber Web interface colors cells in the city grid based on the average travel time to them from the specified pin

3.2 Data processing

The data as provided by Uber cannot be directly used to build a graph on which we can detect traffic congestion. Instead, we need to build a graph that models the underlying city structure on which trips take place. To do so, we built a spatial graph to represent the adjacency of the zones on which Uber aggregates data. A GeoJSON file describing the polygons which delimit the zones in a city is provided with the data. Using the `igraph` and `rgeos` package for R we were able to load the geometry and compute an adjacency matrix for when any two given polygons were touching each other. The adjacency matrix could then easily be exported as an edge list to be imported into Snap. Figure 2 shows how the adjacency graph is built from the set of Polygons.

Once we have a graph representing the structure of the city, we define two sets of weights which will result in two different interpretations of the resulting graph. First, we may think of the graph as being undirected and the weights being the distances between cells in the grid. To compute this distances we rely once more on the geospatial packages available for R. We computed the geographic centroid of each cell and then measure pairwise distances between centroids using Haversine distance[2]. We’ll further refer to this as the **spatial graph** $G_s = (V_s, E_s, w_s)$.

[1] A zone is a predefined region within the city, and each city consists of hundreds of zones.

[2] The Haversine distance is the shortest distance between two points on the surface of a sphere and has proved to be a good enough approximation for distances between (latitude, longitude) pairs.

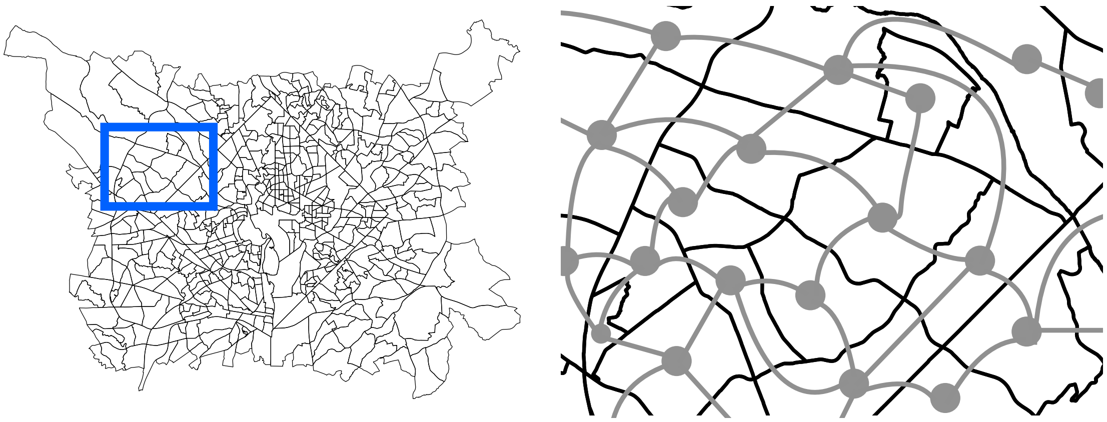

Figure 2: The GeoJSON file contains a set of polygons on which we'll zoom in (left). Looking closely at the region in the blue rectangle, we define a graph where every node represents a region and two regions are linked if they are adjacent (right)

Second, we may wish to assign weights for each of the four statistics on trip length. In this interpretation of the graph, there exists a link between node u and v if there was at least one trip originating in cell u and ending in cell v. Since we may not have data for all possible node pairs in the graph for a given time-frame, the nodes in this graph will be a subset of V_s. Also note that the average time of travel from node u to node v may differ from the time of travel from node v to u. Thus this graph is directed and since it may be weighted with any of the 4 aggregate measures on trip duration provided by Uber, we'll further refer to it as the **temporal graph** $G_t = (V_t, E_t, w_t)$.

3.3 Mathematical background

As described in Section 3.2, we will be working with two weighted graphs: a spatial graph (undirected) and a temporal graph (directed). In this section we define some notions and matrices associated to weighted graphs that we will need for our analysis. We will denote a weighted graph by a triple $G = (V, E, w)$, where (V, E) is the associated unweighted graph, and w is a function from E to the real numbers.

In order to identify the structurally important nodes in our graph, we need to extend the centrality measures we learned in class to weighted directed graphs. We will initially consider four metrics to find the most central nodes in our graph: in-degree, out-degree, betweenness centrality, and closeness centrality. These measures are defined in terms of node degrees and shortest paths, so we need to extend these definitions to the weighted case to be able to use these metrics in our analysis.

Further, we will use Page Rank and HITS algorithms as an alternative way to compute measures of centrality of our nodes. The HITS algorithm uses the adjacency matrix of a graph to identify hubs and authorities; meanwhile, the Page Rank algorithm uses the stochastic adjacency matrix of a graph, which is defined in terms of the out-degree of its nodes. In this section we provide the definition of these matrices in the weighted case.

3.3.1 Degree

In the case of undirected graphs, the weighted degree of a vertex is defined as the sum of the weights of its attached edges. Formally, the degree of a vertex v of a graph, d_v, is defined as

$$d_v = \sum_u w(u, v).$$

Similarly, in the case of directed graphs we define the weighted in-degree $d_v^{(\text{in})}$ and the weighted out-degree $d_v^{(\text{out})}$ of a vertex v as the sum of the weights of the edges with source v and the sum of the weights of the edges with destination v, respectively. These degrees can also be denoted by node in- and out-strength.

3.3.2 Shortest path

The length of a path P in a weighted graph is the sum of the weights of the edges of P. That is, if P consists of edges $e_0, e_1, \dots e_{k-1}$, then the length of P, denoted $w(P)$ is defined as:

$$w(P) = \sum_{i=0}^{k-1} w(e_i).$$

The distance from a vertex u to a vertex v in G, denoted by $d(u, v)$ is the length of the shortest path from u to v, if such path exists.

3.3.3 Adjacency matrix

The adjacency matrix of a graph with n nodes is a matrix with rows and columns labeled by graph vertices, with the weight of an edge or a zero in position according to whether the vertices are adjacent or not. Formally, the adjacency matrix is $W = (w_{uv}), \ \ u, v \in \{1, 2, ..., n\}$ where

$$W_{uv} = \begin{cases} w(u, v), & uv \in E \\ 0, & uv \notin E. \end{cases}$$

3.3.4 Stochastic Adjacency matrix

Recall that the column stochastic adjacency matrix M used in the page rank algorithm is defined in the following way. Let v have $d_v^{\text{(out)}}$ out links. Then,

$$v \to u \implies M_{uv} = \frac{1}{d_v^{\text{(out)}}}.$$

We can extend this matrix to the weighted case by simply considering the weighted out-degree as defined in Section 3.3.1.

4 Algorithms, Techniques and Models

We start our analysis by reviewing some key centrality measures of each of the graphs and applying a community detection algorithm to detect clusters. We will then use this information to pinpoint specific stress points of the global city network and eventually simulate a solution designed to alleviate the stress on that regions.

4.1 Centrality

Following previous literature on traffic flow networks, we will start by exploring some centrality measures in our graph. In particular, we will measure betweenness of our nodes to identify structurally important locations and closeness centrality to pinpoint which sources and destinations live in the core and periphery of our travel network. It is important to note that, even if our nodes represent geographical locations in a map, this doesn't mean that nodes that are together in space will be "close" in our network. Nonetheless, we hope that analyzing centrality in our network will reveal some structural characteristics of urban dynamic traffic flow and spatial human activity.

4.1.1 Node degree

We start by exploring our most central nodes by weighted in- and out-degree, as defined in Section 3.3.1. The top nodes by degree are shown in Figure 3.

By out-degree	By in-degree
8600 Brook Road, McLean	7800 Montvale Way, McLean
1400 Montague Street NW, NW Washington	1800 Upshur Street NW
4500 Ohio Drive Southwest, SW Washington	Westbranch Drive, Tysons
Meadow Road NE, NE Washington	1600 Tuckerman Street NW, NW Washington
Perimeter East Road SW, SW Washington	4500 Ohio Drive SW, SW Washington

Figure 3: Top nodes by in- and out-degree.

4.1.2 Betweenness and Closeness Centrality

The betweenness centrality of a node v is the number of shortest paths in the graph that pass through it. This metric highlights the "gate keeper" nodes which are structurally important to the graph. Meanwhile, the closeness centrality of a node v is defined as the reciprocal of the mean average shortest path from node v to all other nodes in the graph. Intuitively, nodes in the core of our network will have high closeness centrality and nodes in the periphery should have low closeness centrality. Figures 4 and 5 show all nodes of our graph colored by betweenness and closeness centrality, respectively. Darker tones of blue denote higher centrality; meanwhile, the coordinates of the nodes in the graph correspond to their true latitude and longitude.

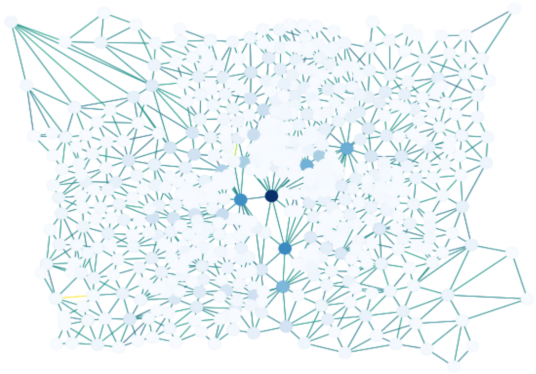

Figure 4: Betweenness Centrality

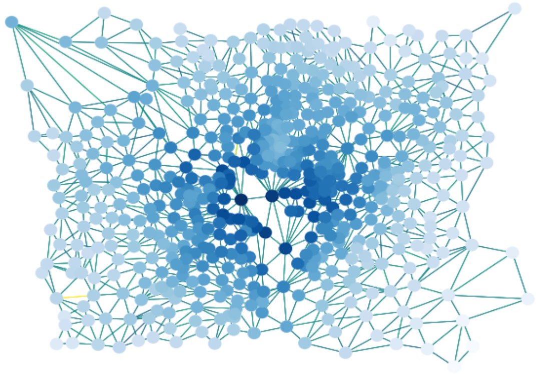

Figure 5: Closeness Centrality

4.1.3 PageRank and HITS

An alternative measure of centrality of a node is its PageRank score. This algorithm is based on the idea that links from important nodes count more; that is, that importances flow across the directed edges of a graph. We compute the PageRank scores for all nodes in our temporal graph using the `GetWeightedPageRank()` of snap. The results are shown in Figure 6. Darker tones of blue denote higher PageRank score; meanwhile, the coordinates of the nodes in the graph correspond to their true latitude and longitude.

To compute the Hubs and Authorities in our graphs we used the `hub_score()` and `authority_score()` functions of the `igraph` [1] package in R. The authority scores of the vertices are defined as the principal eigenvector of $W^\top W$, where is the adjacency matrix of the graph. Meanwhile, the hub scores are the principal eigenvector of $WW^\top$. The top five hubs and authorities are shown in Table 7.

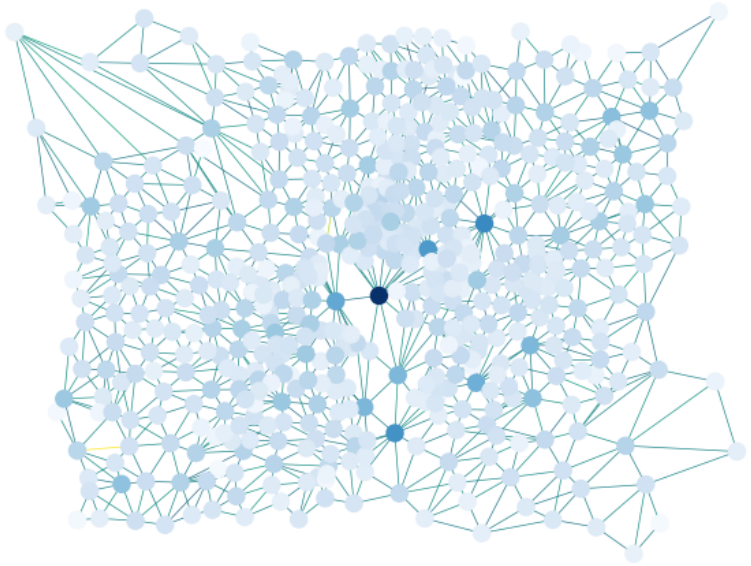

Figure 6: Page Rank scores

Top Hubs	Top Authorities
8600 Brook Road, McLean	57800 Montvale Way, McLean
6500 Bellamine Court, McLean	Westbranch Drive, Tysons
6800 Churchill Road, McLean	9400 Pamlico Lane, Great Falls
Westbranch Drive, Tysons	8600 Westwood Center Drive, Vienna
Dolley Madison Boulevard, Tysons	8300 Greensboro Drive, McLean

Figure 7: Top hubs and authorities by node id.

Note that the top hubs and authorities are highly correlated to the top nodes by out- and in-degree, respectively.

4.2 Community Detection

One of our goals is to identify communities in our network in order to shed light on the structure of our traffic flow data. This could allow us to pinpoint locations within or across our cities that exhibit co-behavior at a certain time of day or at a certain week of year. In order to accomplish this, we will use a modification to the Girvan-Newmann Strength of Weak Ties algorithm for weighted directed graphs.

4.2.1 Girvan-Newmann Algorithm

This community structure detection algorithm, invented by M. Girvan and M. Newman in 2002 [3], is based on the betweenness of the edges in the network. The idea is that the betweenness of the edges connecting two communities is typically high, as many of the shortest paths between nodes in separate communities go through them. So we gradually remove the edge with highest betweenness from the network, and recalculate edge betweenness after every removal. This way sooner or later the network falls off to two components, then after a while one of these components falls off to two smaller components, etc. until all edges are removed. This is a divisive hierarchical approach and the result is a dendrogram. We use the `igraph` package in R to find communities in our temporal graph and show the results in Figure 8.

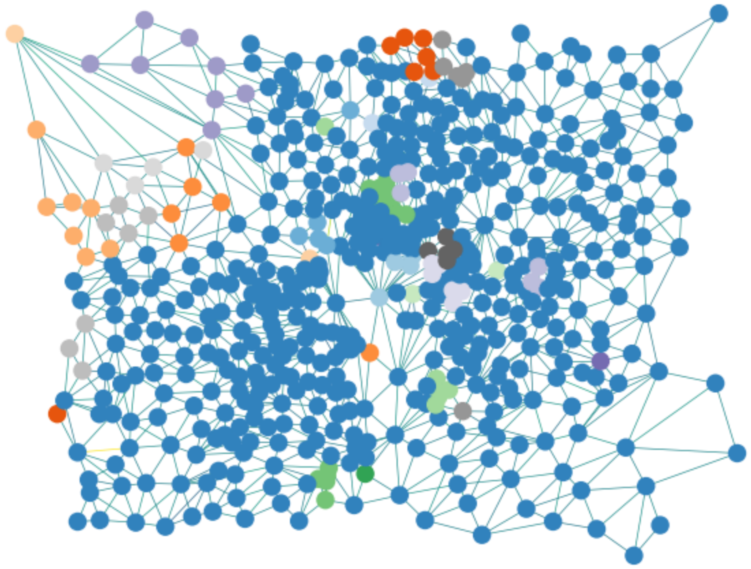

Figure 8: Communities detected by the Girvan-Newmann Algorithm

5 Next steps

We now have a detailed analysis of a variety of centrality measures in our temporal graph. The immediate next step is to analyze the structurally important nodes and compare them to the central nodes in our spatial graph in the hopes that this will help us identify points of conflict for traffic flow in cities.

Additionally, we are yet to implement a temporal component to study the evolution of our traffic flow network by observing snapshots of it taken at regularly spaced points in time. The nodes of the graph, which represent the sources and destinations of our trips, will remain fixed across time periods; however, the weights of the edges will be dynamic, as average travel times change throughout the day. We hope that this analysis of progressing travel times will help us identify the traffic bottlenecks and rush hours in the different zones of our cities and shed some light into what is originating them.

Once we identify these points of conflict, the final step will be proposing a solution to alleviate the problem in the identified bottlenecks and measuring its impact in traffic flow through simulated data.

6 References

[1] Gabor Csardi and Tamas Nepusz. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695, 2006.

[2] Song Gao, Yaoli Wang, Yong Gao, and Yu Liu. Understanding urban traffic-flow characteristics: a rethinking of betweenness centrality. *Environment and Planning B: Planning and Design*, 40(1):135–153, 2013.

[3] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

[4] Xi Liu, Li Gong, Yongxi Gong, and Yu Liu. Revealing travel patterns and city structure with taxi trip data. *Journal of Transport Geography*, 43:78–90, 2015.

[5] Alasdair Turner. From axial to road-centre lines: a new representation for space syntax and a new model of route choice for transport network analysis. *Environment and Planning B: Planning and Design*, 34(3):539–555, 2007.

[6] Yu Zheng, Yanchi Liu, Jing Yuan, and Xing Xie. Urban computing with taxicabs. In *Proceedings of the 13th international conference on Ubiquitous computing*, pages 89–98. ACM, 2011.