The Road Home: Auditing Google Maps during Heavy Traffic

Tushar Chandra

Abstract

Routing algorithms are incredibly important to many people's daily lives, as the task of how to get from one place to another is often difficult. Anecdotal experience suggests that commercial routing algorithms will route vehicles differently in heavy traffic, but it is difficult to verify this due to the proprietary nature of such algorithms and platforms. We provide a framework for auditing the Google Maps routing algorithms by using an open-source routing library, GraphHopper, as a basis for comparison, and begin a discussion about the externalities associated with doing so.

Introduction

Vehicle routing is a common part of everyday life; indeed, Google Maps boasted of having over 1 billion monthly active users in 2014 [1]. But the task of getting people from one place to another is deceptively simple, and transportation research has focused on how best to accomplish this at the individual and aggregate levels. We consider the extent to which Google Maps routes users differently in heavy traffic by comparing it to an open-source routing library, GraphHopper, and evaluating the routes produced by these platforms.

Related Work

In this section, we discuss the research and technologies that provide the foundation for our work by considering routing algorithms, traffic dynamics, and algorithmic auditing.

Routing Algorithms

We begin by reviewing literature from the field of vehicle routing. Reginald Golledge, at the University of California Routing Center, was among the first to study questions that arise during the path selection process. Traditional systems make use of "fastest path routing," which routes people down the path that is likely to be fastest [2], and one can also imagine algorithms that consider paths with the least walking or the fewest turns [3, 4]. But there also exists an exceedingly important *human* component to routing. For instance, Garling showed as early as 1986 that, for pedestrians, the routes that users choose to take are often *not* strongly aligned with those generated by shortest path algorithms [5].

Building upon Garling's work are what have come to be known as "alternative routing algorithms." At a high level, such algorithms find the "best" routes according to some other criteria for optimality [6]; in other words, they make some sacrifices in the speed of the route in order to improve, for instance, how scenic [7] or how safe [8] it is. Methods for estimating the beauty of areas include using volunteered geographic information from Twitter or Flickr [6, 9]. Other research in this space has focused on developing different criteria for alternative routing mechanisms, but it was not until recently that the externalities thereof were addressed [6].

Johnson et al. (2017) provided the first robust assessment of the externalities associated with alternative routing mechanisms. A significant contribution of their research was that, if widely deployed, criteria like beauty and safety might lead to negative impacts at the community level. In particular, scenic routing tends to redirect people away from highways to local roads, which they found could cause increased levels of traffic in those communities. They emphasized that the traditional methods for routing evaluations -- distance and travel time -- often leave out critical information about the externalities of those routes [6].

Routing in Traffic and Algorithmic Auditing

Standard routing platforms such as Google Maps will also often adjust their routes based on current traffic patterns [10]. While these are not considered to be alternative routing schemes (the goals thereof is still to find the quickest path), they might also have externalities associated with them. It might, for instance, be the case that a highway being congested leads to people being directed along county roads; this could lead to similar effects as discussed in Johnson et al [6], with heavier-than-usual traffic within communities. But rigorous methodology in this space (known as dynamic vehicle routing) has only recently began to emerge, as prior research has primarily focused on the static case [11].

Fleischmann considered the dynamic vehicle routing problem in the case of fulfilling delivery orders -- a well-studied problem -- but added online traffic information into their analysis [12]. Meanwhile, Ziliaskopoulos developed time-dependent shortest path algorithms for intelligent highway systems (in contrast with traditional methods that use traffic information from the current time, but do not attempt to predict it). Their approach was too computationally expensive to implement anywhere besides a supercomputer, but the theory was novel [13]. However, the algorithms used by commercial mapping platforms are proprietary and unknown to the public, so it is difficult to summarize the current state of the field.

Because these algorithms work as black boxes, there is a strong need for algorithmic auditing. The details of the Google Maps routing algorithm are clearly unknown, and attempts to audit them can also only rely on comparing them to existing, transparent systems. Johnson et al. discussed this when they measured the externalities of alternative routing criteria [6], and the open-source routing library GraphHopper has attempted to spur more transparency in such algorithms [14]. Our work provides an audit of traffic-based routing algorithms, addressing this gap in the literature.

Methods

In this section, we discuss the framework we developed for analyzing routing algorithms, how we collected the data, and how we analyzed the data. We begin by stating our research questions:

RQ1**:** Do we see evidence of Google Maps using alternative routing during heavy traffic?

RQ2**:** How can we characterize these alternative routing schemes?

These empower us to engage in algorithmic auditing of the Google Maps routing algorithm -- first by seeing if it differs in heavy traffic, and secondly by understanding how.

Framework and Data Collection

Broadly speaking, we were interested in the routes throughout Chicago on different routing platforms in different traffic conditions. Ideally, we would consider a dataset of requests made to Google Maps, but this is clearly unavailable to the public. Instead, we follow the practice from the literature [15] by randomly generating 1000 pairs of origin and destination points.

We then extended the functionality of the open-source library GraphHopper to route people in heavy traffic. Since GraphHopper uses standard graph-based methods, this amounted to creating custom weightings for the edges, which represent road segments. This code is made freely available on Github (tuchandra/graphhopper). In order to choose the weightings, we turned to live data from the City of Chicago. They release a “Live Traffic Tracker” that allows users to view real-time traffic speed along arterial roads [16]. The road segments represented as latitude / longitude waypoints along polylines, which can then be matched to edges within GraphHopper. The data comes from positioning probes on buses and is updated every 10 minutes. It bins the average speed along the road segments into light, medium, and heavy traffic, instead of providing more granular information. Because of this, we assign each road segment the average value of its speed bin.

Armed with this custom weighting, we were able to generate the GraphHopper routes between our origin and destination points during heavy traffic. We collected data at 9:00 am on a Monday (during morning rush hour), and used this to generate the traffic-based routes. We also generated routes using GraphHopper's default, fastest path routing, which served as a basis for comparison. Likewise, we queried the Google Maps API at 1:00 am and 9:00 am. Once done, we had four sets of routes between 1000 start- and end-points within Chicago to analyze.

Data Analysis

Our analysis had two stages. The first was quantitative, in which we considered the extent to which traffic-based routing redirected people away from highways and onto local roads. We performed this analysis for both Google Maps routes and GraphHopper. The second stage of analysis involved considering qualitatively the types of roads that saw more and less activity in traffic-based routing. This was motivated by the discussion in Johnson et al. that alternative routing schemes may have hidden impacts on communities [6]; a reasonable hypothesis would be that traffic-based routing redirects people to local roads when main arteries are congested.

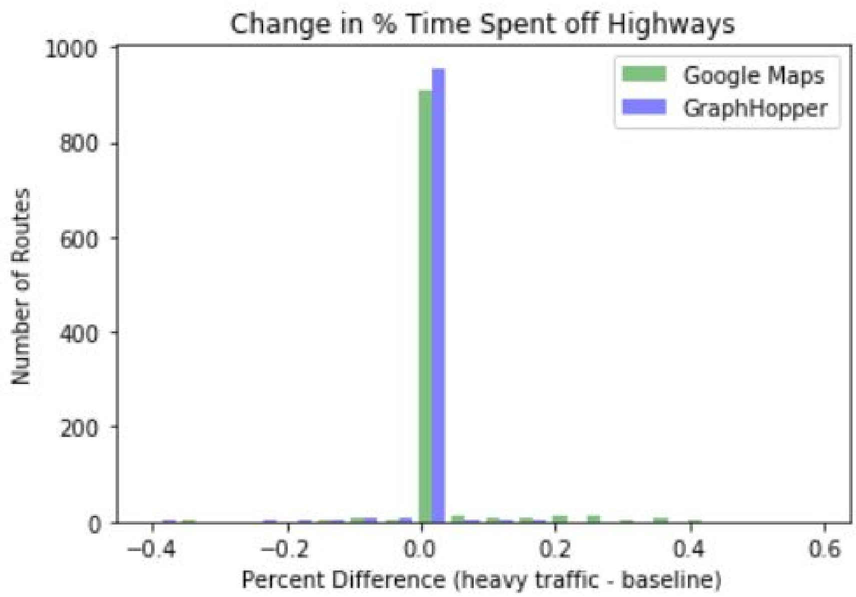
In the first stage, we considered the routes from all four sets of routes (Google Maps without traffic, Google Maps during heavy traffic, GraphHopper with no traffic, GraphHopper during heavy traffic). When generating the GraphHopper routes, we were able to calculate the percent time spent on (and off) highways, and likewise the percent time spent in neighborhoods (used as a proxy for local roads). The Google Maps API does not provide this functionality, but we were able to match the generated routes to GraphHopper edges and compute the same metrics.

For each route, we considered the difference in time spent off highways and in neighborhoods, then looked at whether or not this was significantly different from zero (which would indicate no difference) using a paired samples t-test. This was done for each routing platform independently, and we also compared the platforms to each other. Again, the goal of this analysis was to understand if different types of roads experienced changes in activity during heavy traffic.

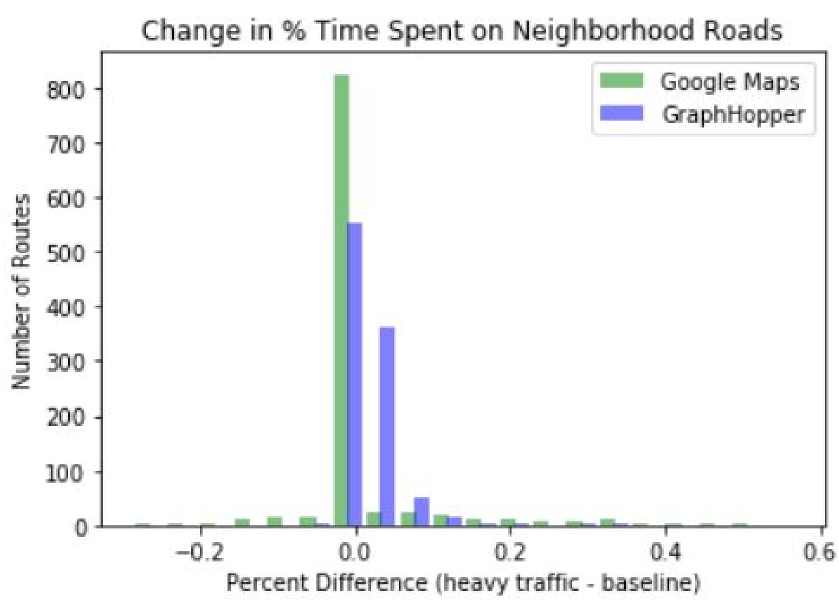
The second stage of the analysis was qualitative. We sampled the road segments with replacement (also known as bootstrap resampling), in accordance with the methods of Johnson et al. We then computed the mean change in activity on each road segment over all trials, then visually examined the regions where activity changed during times of heavy traffic. This allowed us to understand, at a community level, the impacts of traffic on the local road network.

Results

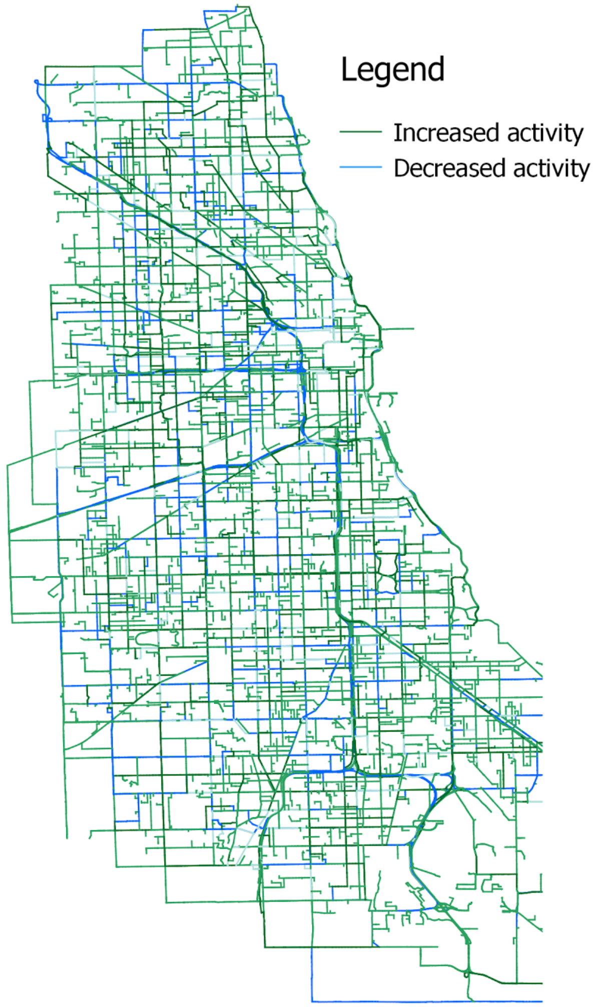
We first present graphs for the first stage of the analysis. The following figures show the distributions of the differences in the percent time that routes spent off highways; refer to *Methods* for detailed descriptions of how these were calculated. We see that, for both Google Maps and GraphHopper routes, paths in heavy traffic spent slightly more time off highways. While the difference in time spent off highways was not found to be significant for GraphHopper routes ($p = 0.08$), it was highly significant for Google Maps routes ($p = 10^{-8}$), and the difference from each other was also highly significant ($p = 10^{-9}$). This indicates that, in heavy traffic, Google Maps tends to direct people away from highways.

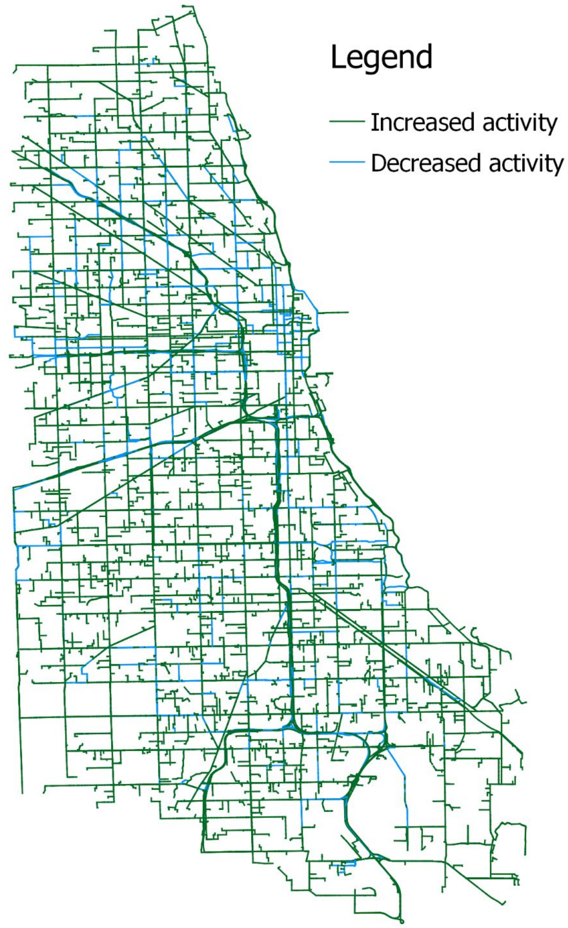


We see a slightly different story for the percent time spent on neighborhood roads. Here, both Google Maps and GraphHopper show an increase in the time spent on such roads that is statistically significant ($p = 10^{-52}$ for Google Maps, $p = 10^{-9}$ for GraphHopper). However, the two are not significantly different than each other ($p = 0.38$), indicating that here Google Maps performs similarly to GraphHopper.



Finally, we visualize the roads that received more or less activity in conditions of heavy traffic, relative to baseline conditions with no traffic. The left figure below shows the data from Google Maps routes, and the right figure from GraphHopper routes. While it is difficult to draw conclusions from these visualizations, it is apparent that the two models appear to have significant differences between them -- certain road segments appear to have more activity on one platform, but less on the other. This can almost certainly be attributed to the difference in the traffic model used between Google Maps and GraphHopper; in particular, the model used in the latter was very simplistic (recall that we assigned each edge a constant speed) and could likely be improved to be more realistic. Further research in traffic modeling would allow us to further refine our models and likely obtain clearer results.





<u>Conclusion</u>

We see evidence that Google Maps routes people differently in heavy traffic, and can say with relative confidence that it tends to route people away from highways and towards local roads. We verified this through comparison with an open-source routing library, GraphHopper, combined with live traffic data from the city of Chicago. However, it is important to note several limitations. We made many simplifying assumptions, including assigning each “medium traffic” or “heavy traffic” road segment the same speed, rather than spatially modeling them. Our models are only as good as the data underlying them -- the positioning systems on the city buses, the time lag on updating the traffic tracker, and the quality of maps on OpenStreetMap.

Finally, a great deal of technical difficulties were encountered with building and running GraphHopper, and we were at several times limited in the amount of data we could process and analyze. Despite all this, however, we find evidence for Google Maps routing users differently in heavy traffic, and invite further research on the potential externalities associated with this.

References

[1] Google Maps: 1 Billion Monthly Users: 2014. https://www.gpsbusinessnews.com/Google-Maps-1-Billion-Monthly-Users_a4964.html. Accessed: 2017-12-06.

[2] Reginald G. Golledge. 1995. Defining the criteria used in path selection. University of California Transportation Center

[3] Matt Duckham and Lars Kulik. 2003. “Simplest” Paths: Automated Route Selection for Navigation. In International Conference on Spatial Information Theory, 169–185.

[4] Ivan J Tashev, Jeffrey D Couckuyt, Neil W Black, John C Krumm, Ruston Panabaker, and Michael Lewis Seltzer. 2012. Pedestrian route production. Google Patents.

[5] Gärling, T., Böök, A., & Lindberg, E. (1986). Spatial orientation and wayfinding in the designed environment: A conceptual analysis and some suggestions for postoccupancy evaluation. *Journal of Architectural and Planning Research*, 55-64.

[6] I. Johnson, J. Henderson, C. Perry, J. Schöning, and B. Hecht. 2017. Beautiful…but at What Cost?: An Examination of Externalities in Geographic Vehicle Routing. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 1, 2, Article 15 (June 2017), 21 pages. DOI: https://doi.org/10.1145/3090080

[7] Daniele Quercia, Rossano Schifanella, and Luca Maria Aiello. 2014. The shortest path to happiness: recommending beautiful, quiet, and happy routes in the city. In HT, 116–125.

[8] Samuel Elsmore, Irwan Fario Subastian, Flora Dilys Salim, and Margaret Hamilton. 2014. VDIM: Vector-based Diffusion and Interpolation Matrix for Computing Region-based Crowdsourced Ratings: Towards Safe Route Selection for Human Navigation. In MUM (MUM ’14), 212–215.

[10] See Traffic or Search for Places Along the Way. Google Maps. Retrieved from https://support.google.com/maps/answer/6337401?hl=en.

[11] Psaraftis, H. N. (1995). Dynamic vehicle routing: Status and prospects. *Annals of operations research*, *61*(1), 143-164.

[12] Fleischmann, B., Gnutzmann, S., & Sandvoß, E. (2004). Dynamic vehicle routing based on online traffic information. *Transportation science*, *38*(4), 420-433.

[13] Hall, E. C. J. (1993). Time-dependent, shortest-path algorithm for real-time intelligent vehicle highway system applications.

[14] https://www.graphhopper.com/

[16] http://webapps1.cityofchicago.org/traffic/

[15] Shazia Haque, Lars Kulik, and Alexander Klippel. 2006. Algorithms for reliable navigation and wayfinding. In International Conference on Spatial Cognition, 308–326.

[9] Abdallah El Ali, Sicco NA Van Sas, and Frank Nack. 2013. Photographer paths: sequence alignment of geotagged photos for exploration-based route planning. In Proceedings of the 2013 conference on Computer supported cooperative work, 985–994. Retrieved April 4, 2017 from http://dl.acm.org/citation.cfm?id=2441888