Eurographics Conference on Visualization (EuroVis) 2015
H. Carr, K.-L. Ma, and G. Santucci
(Guest Editors)

Volume 34 (2015), Number 3

Exploring Traffic Dynamics in Urban Environments Using Vector-Valued Functions: Appendix

Jorge Poco[1], Harish Doraiswamy[1], Huy. T. Vo[1], João L. D. Comba[2], Juliana Freire[1], and Cláudio. T. Silva[1]

[1] New York University, USA [2] Instituto de Informática, UFRGS, Brazil

Appendix A: Performance of Yen's algorithm for computing closest paths.

Yen's ranking loopless shortest paths algorithm [Yen71] computes the k-shortest paths between a source and destination in a graph for a given k. It inductively computes the i^{th} shortest path between two nodes using the common subpaths of the $(i-1)$-shortest paths. The algorithm uses the result of the shortest path as starting point, followed by a relaxation procedure until the distance constraint is met.

This algorithm can be used to compute the closest path as follows: identify a set of k' shortest paths, for a large enough k' such that the k'^{th} shortest path is the closest path. Unfortunately, due to a grid-like structure of the road network in most part of NYC, the above algorithm doesn't work well. This is because it has to compute and discard a lot of suboptimal paths. For example, Fig. 1(b) illustrates the class of paths that Yen's algorithm produced for a taxi route shown in Fig. 1(a). Note that the algorithm would try every possible detours around any block along the shortest path, e.g. as denoted by the green ($d = 1.06$), orange ($d = 1.27$) and purple ($d = 1.34$) lines in Fig. 1(b), before changing the path structure. In this case, the algorithm needed to go through 40 paths before finding a path having the required length. In our experiments, this number even reached as high as 10000 in many cases. Fig. 2 plots a histogram of k' for a random set of 1000 trips that occurred over a day. We found that not only more than 380 trips had k' greater 1000, but over 280 of these trips had k' greater than 10000. Thus, Yen's approach was not a practical option for our model.

Appendix B: Accuracy of the closest path algorithm.

We received a small sample of recent taxi trips which included the actual route taken by the taxis. Note that this information is not available for the three years worth of historical data collected by the taxi agency. We use this data to validate the accuracy of the closest path algorithm for identifying taxi routes.

Given the number of way-points n to be used, we computed the k-closest paths for each of the trips. (As in the paper, we set $k = 20$). We then compute the intersection of these paths with the actual route. The maximum intersection percentage over all the k paths is used to measure the accuracy of the prediction for that trip. The route in Fig. 1(a) shows one such sample taxi trip for which the actual path taken by the taxi was available. Notice that the route taken by the taxi is different from the shortest path between the pickup and dropoff locations. Fig. 1(c) shows the $k = 4$ closest paths computed using our method. Note that one of paths correctly identifies the actual path. For this example, the accuracy is measured to be 100%.

Using $n = 1$, the average accuracy of our trip route prediction was 82.8%. That is, on an average, we could correctly predict 82.8% of the taxi route. When we increased the value of n to 2, the average accuracy increased by only 1.9%. However, the time taken to compute the k-closest paths became over 200 times slower. This is because, as mentioned in the paper, the time-complexity of our closest path algorithm is exponential in the number of way-points n ($O(|V|^n)$). When we increased the value of n to 3, this time further reduced by another two orders of magnitude. Therefore, we choose $n = 1$ in our closest path computation since it provides the best trade-off between accuracy and time.

Appendix C: Validation of the closest path traffic model.

In order to validate the closest path traffic model, we compare the results obtained from our model with those obtained using the shortest-path model [SRS*13], as well as with the actual data obtained using EZ-pass tag readers. NYC has a set of such readers placed at strategic points in order to collect traffic information. We had access to data for the month of November, 2011, corresponding to Madison Avenue between 49th and 57th streets and Lexington Avenue between 49th and 57th streets. Using the time-stamps of the different EZ-pass tag IDs along a street, we estimate the speed of traffic on that street. Since the tag readers are placed at

(a) (b) (c)

Figure 1: *Comparison of the k-closest path computation based on Yen's model [Yen71] and ours.* ***(a)*** *The red dots show the actual gps track of a taxi ride that did not follow the shortest route due to traffic conditions;* ***(b)*** *top k (k = 4) paths based on Yen's model, starting out as the shortest path and gradually growing until it matches the required distance; and* ***(c)*** *our approach started out with those paths that best match the required distance, thus, providing more accurate answers in much less time.*

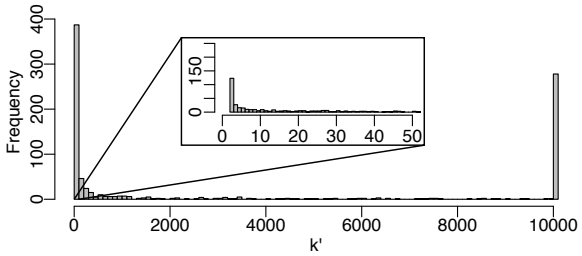


Figure 2: *Histogram of the frequency of number of shortest paths k′ to be computed to obtain the closest path using Ren's algorithm.*

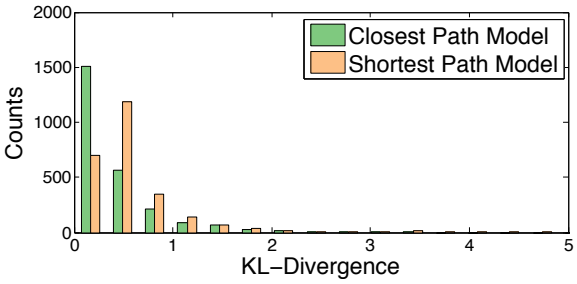


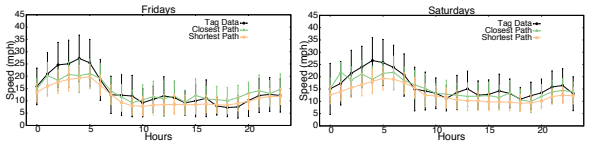
Figure 3: *Histogram of the KL divergence measures obtained when comparing the closest path and shortest path models with the distributions computed using data from EZ pass tag readers. Note that values of $D_{closest}$ from more time periods are closer to zero than the values of $D_{shortest}$ indicating that the closest path model better approximates the observed traffic speeds.*

frequent points along a street, we expect the speeds computed to be representative of the actual traffic. The computed speeds are then used to compute the mean and standard deviation. The number of samples available for the tag reader data per road segment was small (only around 100 samples per hour). This not only decreases the precision of the traffic speed that we are comparing against, but it also restricts us from using small time intervals for the validation. Therefore, the speeds from both the models, as well as from tag reader data was aggregated for hourly intervals.

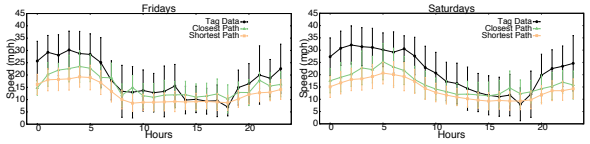
We use the Kullback-Leibler (KL) divergence metric [BA02, KL51] to compare the two traffic models with the actual data. For two distributions P and Q, the KL divergence measure $D_{KL}(P||Q)$ computes the amount of information lost when approximating P using Q. Given the traffic derived from the closest path model $M_{closest}$, we first compute a set of divergence measures $D_{closest} = D_{KL}(tag||M_{closest})$ for different time periods. Here, *tag* represents the distribution obtained using the EZ-pass tag readers. A lower divergence value reveals a better approximation of the observed traffic distribution. We repeat this process using the shortest path to infer traffic speeds to obtain $D_{shortest}$.

Fig. 3 plots the histogram of $D_{closest}$ and $D_{shortest}$. Note that most of the divergence values of the closest path model are close to zero, implying that this model is a good approximation of the observed speed distribution. Moreover, the closest path model has more divergence values close to zero than the shortest path model, implying that this model better approximates the observed speed distribution when compared to the shortest path model. As mentioned earlier, this is due to the fact that many of the taxi trips do not take the shortest path from its pick-up location to its drop-off location, and therefore, the closest path model does a better job of identifying the most probable routes.

We can also compare the actual predicted speed values across the different models, by plotting the speed distribution over time for different streets. Fig. 4 compares the speeds computed using tag readers, with the closest path model, and shortest path model on two road segments along Lexington Avenue and Madison Avenue. Note that the mean speed of the road segments computed using both the models lie within the speed range as computed using the tag readers. Also, for most of the time periods, the mean speed computed



(a) Lexington Ave., between 50th – 51th Street



(b) Madison Ave., between 50th – 51th Street

Figure 4: *Comparison of the speed of traffic on Fridays and Saturdays in November 2011 computed using tag readers, the closest path model, and the shortest path model.*

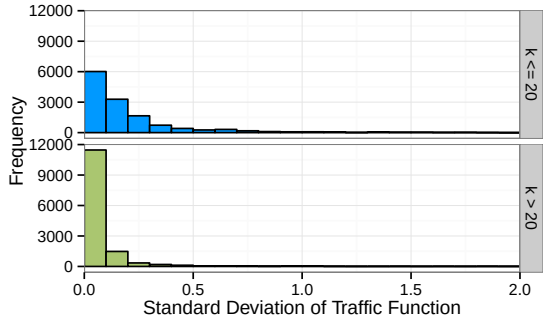


Figure 5: *Variation of traffic function for different values of k.*

using the closest path model is closer to that of the tag reader than the mean speeds of the shortest path. We noted this behavior along other streets as well.

Note that commercial map tools, like Google maps, do not provide historical traffic data, even for the main roads. It was therefore not possible to compare our results with these tools.

Appendix D: Traffic function computation: choice of k

In order to identify a good value for k, we picked a time period, and computed the traffic function for varying values of k. Given the different traffic functions, we compute the standard deviation of the speeds for each edge of the road network. We computed this standard deviation for different ranges of k. In particular, we found that for ranges including values of k larger than 20, the variance did not vary by much, and was close to zero. Note that a zero variance indicates that changing k has no effect on the traffic function. This is because, as k increases, more paths in the resulting set have lengths significantly different from the actual length thus decreasing the weight assigned to these paths.

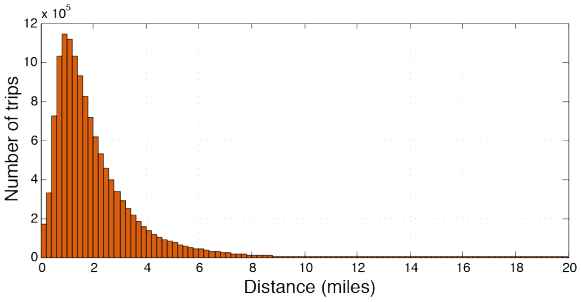


Figure 6: *Histogram showing the frequency of trip distances of the taxi trips within Manhattan that happened over a period of one month.*

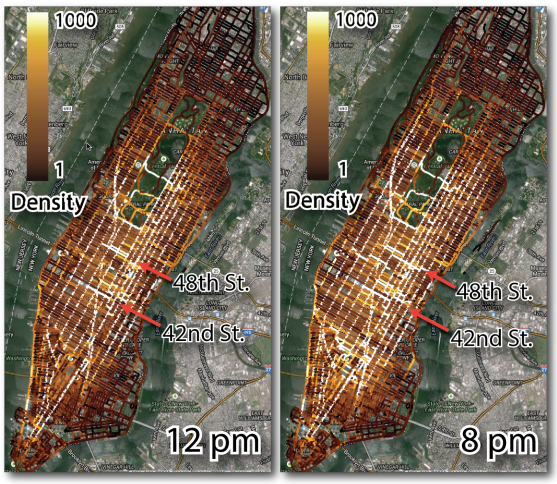


Figure 7: *Taxi distribution at 12 pm and 8 pm on Fridays. Note that taxis tend to move along avenues (colored black) compared to streets.*

Such paths therefore have very little influence in the traffic function.

Fig. 5 plots the frequency of this variance of speeds among the edges. When $k \leq 20$, there is a high variance between the traffic functions for different values of k. However, when $k > 20$, the variance is mostly close to zero.

Appendix E: Taxi trip distance distribution

A majority of the trips that occur within Manhattan are short distance trips. This is illustrated in Fig. 6, which shows the histogram of the trips that happen over a month.

Appendix F: Case Study: Taxi patterns

An overview of the general positions of taxis can be obtained by visualizing the density distribution of taxis using a color map. Fig. 7 shows this distribution at two different time periods on Fridays. The coloring varies from dark to white

in different shades of yellow, a darker color meaning lower density. Notice that most of the taxi movement is along the avenues (roads running north to south) compared to streets (roads running east to west), with a few exceptions, e.g., 42nd St. near Bryant Park and Times Square, and 48th St. near Rockefeller center, which are well-known tourist spots.

References

[BA02] BURNHAM K. P., ANDERSON D. R.: *Model Selection and Multi-Model Inference: A Practical Information-Theoretic Approach*. Springer, 2002. 2

[KL51] KULLBACK S., LEIBLER R. A.: On information and sufficiency. *Ann. Math. Statistics 22* (1951), 79–86. 2

[SRS*13] SANTI P., RESTA G., SZELL M., SOBOLEVSKY S., STROGATZ S. H., RATTI C.: Taxi pooling in new york city: a network-based approach to social sharing problems. *CoRR abs/1310.2963* (2013). 1

[Yen71] YEN J. Y.: Finding the k shortest loopless paths in a network. *Management Science 17*, 11 (1971), pp. 712–716. 1, 2