PRIPOSE: Predictive Routing in Presence of Scheduled Events

Brian Thompson (brthomp2), Chenhao Wu (cwu57), Yuntae Kim (ykim157), Anirudh Madhusudan (anirudh2)
University of Illinois, Urbana Champaign
Github Repo: https://github.com/bthomp2000/PRIPOSE

Abstract: With the emergence of new technologies, navigating and routing applications are getting smart enough to try mitigating traffic congestion by distributing traffic into uncongested roads. However, while they are successful in analyzing the current traffic, they are not fully capable of predicting how congestion will act out in the future. Planned Special Events (PSEs) are non-periodic events with an expected large attendance. [9] Our traffic rerouting framework takes into account PSEs that are likely to cause congestion due to its intermittence and randomness. Our framework predicts traffic speed of nearby road segments ahead of time and lets the drivers know the optimal route they should take to avoid these congested segments. PRIPOSE was shown to be successful in creating better route predictions than baselines models by giving an average 6% reduction in drive time.

1. Introduction

Traffic congestion in big cities is common, however, researchers have found that the "transition from free-flowing traffic to a jamming state occurs spontaneously when the average vehicle density exceeds a certain critical value" [10]. Thus, if we can predict the surge of traffic density along a planned route, we can choose to travel through less dense streets and consequently avoid the possibility of getting caught in traffic.

Existing route planning services such as Google Maps, Here Maps and Waze use optimized version of Dijkstra's algorithm or A* search algorithm to find the shortest or fastest path. These services create routes using the live traffic densities at the time of route-planning, without considering the traffic disruption caused by PSEs in the near future. This paper, Predictive Routing in Presence of Scheduled Events, or PRIPOSE in short, aims at using information obtained from social media on upcoming scheduled events to predict the change in traffic densities around the venue of these venues. While planning the route to his/her destination, PRIPOSE shall use its prediction models to inform the user about potential traffic disruptions that will occur during the travel time. The user will be suggested alternative re-routes that he/she can choose based on their time or route preferences.

Planned Scheduled Events (PSEs) are planned public events such as concerts or sports games, where a large amount of attendance could cause traffic congestion around these event venues [3]. A good source to find data on PSEs is the Eventful website. Eventful is an online calendar and events discovery service that allows users to search and track upcoming entertainment events in their area. Using the Eventful API, it is possible to retrieve information on location, date, time, duration, event type, popularity and other attributes, for past and future events. [7] To learn how traffic trends are affected by PSEs, live-traffic data around the event location, prior, during and post the event shall be analyzed in conjunction with information that can be obtained using the Eventful API.

Using PRIPOSEs traffic density prediction helps inform the driver ahead of time about sudden traffic disruptions (for example: large group of people leaving the stadium after a game) that might not be necessarily reflected using the historical traffic based prediction systems from these existing mapping services.

2. Related Works

Although there has been some past research carried out in predicting traffic congestion using the knowledge of scheduled events, these attempts suffer from inaccuracy as a result of limited information on these events. Additionally, some papers were only able to predict the traffic densities right after the event based on the traffic trends right before the event began [1]. Some attempts relied solely on historical data to regenerate and predict traffic congestion patterns. Lastly, none of these attempts offer a rerouting feature, which is very important, and is the obvious consequential step following predicting traffic densities. Our paper will be reinforcing these issues to provide more accurate traffic prediction and efficient reroutes. We address these shortcomings by exploiting social media (Eventful) as a tool for sensing incoming and outgoing waves of traffic accurately. We will also be able to predict traffic congestions that occur around venues without historical traffic data. Our paper will also suggest reroutes to avoid traffic dense areas.

3. Design

3.1 Historical floating car Data

At the beginning, we tried to use some existing historical floating car data. We found floating car data from multiple online sources and other researchers. However, all the data was inadequate to our project for a few reasons. We first considered historical traffic data from New York City. Even though the data was well spread all over the city, the time window of the data was so limited that it was very hard to find enough events that overlapped well with the data. More specifically, in order to see the reduction of the traffic speed throughout the time, we needed the data to cover the events for at least 6 hours (we will discuss more about this in the next section). We also found the traffic data for Austin, Chicago and San Francisco, but the data points around the venues were too sparse to analyze. Thus, after spending a lot of time on existing traffic data, we decided to move on collecting live floating car data on our own, which we discuss it in the next section.

3.2 Live Floating Car Data Collection

Before the process of data collection, the first step was to pick a city where the primary mode of transportation is personal cars. Also the city needs to have an abundance of venues. Meeting both constraints, Los Angeles, CA was our final choice. The second step was to pick venues with "bursty" events meaning that the these venues don't hold regularly scheduled events. In other words, we wanted to avoid venues that have constant influence of traffic on a regular basis. Otherwise the regular traffic congestion may be reflected on the average historical data. After hand picking events from the Eventful website, we selected 19 over a span of a month as listed in table 1. In order to get enough traffic information, we need the data to satisfy 2 constraints: 1) it covers the whole event for at least 6 hours with 2 hours before the event start and 4 hours after; 2) we will collect data on 2 days: Event Day(ED) and Non-Event Day (NED) which is the comparison day. And there should not be any event happening at the specific venue on the NED. We chose the NED to be exactly a week before or after the event time. We then collected live traffic speeds on the ED and the NED. Specifically, for each event, we generated 50 coordinates to be uniformly distributed within a 1 kilometer radius around the venue as shown in Fig 1. The TomTom API is a service that provides the current traffic speed closest to a provided location [4,5]. We use this API to output the current traffic speeds for these 50 generated coordinates. By querying Tomtom API every 30 minutes for 6 hours, we successfully sampled 12 traffic speeds for each coordinate for each event. We also collected the events information from the Eventful API.

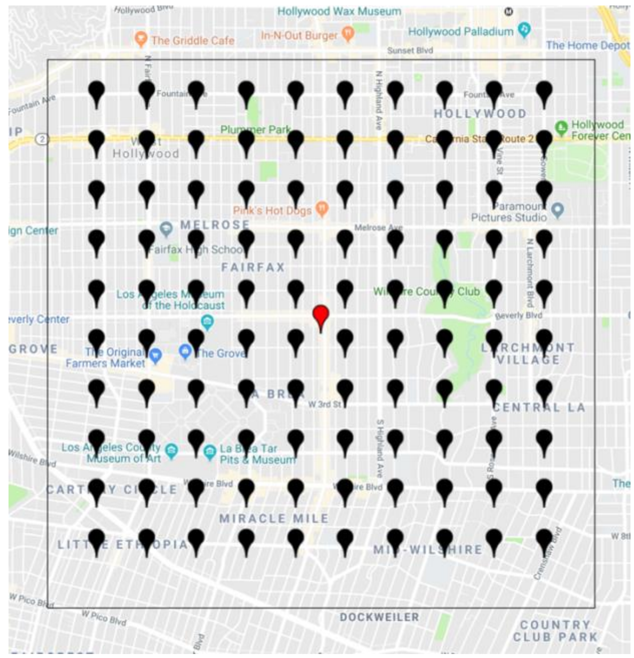


Fig1: Evenly distributed coordinates used in for tracking live floating car data

Given all of the available event information, we chose Event Time, Event Type and Popularity Score as our main parameters for analysis. According to the API documentation, "Popularity Score is a fuzzy number that gives an indication of an event popularity but is not an exact number. In general, as the number goes up the more popular the events become" [6]. We will discuss more about how we built up our model using these parameters in the next section.

Events Collected For Training

Event ID	Event Name	Category	
E0-001-103317913-9	Hamlet	Performing Arts	March 24, 2018
E0-001-103857672-6	Laugh Factory Long Beach	Nightlife	April 12, 2018
E0-001-104932362-5	Khullam Khulla - Rishi Kapoor	Performing Arts	March 24, 2018
E0-001-108286887-9	Los Angeles Dodgers vs. Arizona Diamondbacks	Sports	April 14, 2018
E0-001-108651019-2	Our Last Night	Concerts	April 7, 2018
E0-001-110558861-4	Gloria Trevi vs. Alejandra Guzman	Concerts	April 14, 2018
E0-001-110565032-8	Matt and Kim	Concerts	April 4, 2018
E0-001-110961095-7	Circa Survive - Hail the Sun & Foxing	Concerts	April 8, 2018
E0-001-110961124-8	Once in a Whale	Performing Arts	March 23, 2018
E0-001-111784050-8	Synapse Animation	Concerts	April 1, 2018
E0-001-111825012-8	The Dustbowl Revival - Shook Twins	Concerts	April 11, 2018
E0-001-112293407-8	Kurt Carr Live ft. Tasha Paige-Lockhart	Performing Arts	April 8, 2018
E0-001-112323987-3	Rotations: Shiva DJ Set	Concerts	March 23, 2018
E0-001-112436039-2	Jamrock Thursdays	Concerts	April 12, 2018
E0-001-112887201-7	LOL IRL: An Improv Comedy Show	Comedy	March 26, 2018
E0-001-112893006-5	Nollywood in Hollywood	Film	March 23, 2018
E0-001-113386241-2	Rakim	Concerts	April 12, 2018
E0-001-113414956-7	TigerHeat	Concerts	April 5, 2018
E0-001-113515176-3	"We Are One!" (LAUSD Benefit Concert)	Performing Arts	April 12, 2018

Table 1: List of Event collected for Model Training

3.3 Modeling

From the data that was collected using the TomTom API, we were able to retrieve the following information:

a. Location (lat-lon)
b. Current Speed
c. Free flow Speed
d. Time

As mentioned in the previous section the traffic data was collected for two days, namely, the day of the event i.e. Event Day (ED) and a comparable day the next week i.e. Non-Event Day, (NED). To illustrate, if the event took place on Monday, April 9, 2018 at 9pm, we would compare this traffic data with another Monday evening at the same locations a week later on Monday, April 16, 2018 at 9pm when there is no event. April 9 would be the ED, and April 16 would be the NED for this example.

3.4 Data Preparation

Using SQL, the traffic data from the ED and NED are joined for each corresponding event. The resulting dataset contains information on the current speed and free-flow speed at a fixed location and fixed time, on two different days namely - ED & NED. Given the intention of learning the reduction of traffic speeds near event sites, we remove data (roughly 10%) that show an opposite trend where traffic speeds are higher on EDs. This is done because these data-points do not align with the generalization of our model and may add bias and/or noise to our model. The remaining portion of the data-set is used to calculate the percentage reduction in traffic speed during ED with respect to a NED. The percentage reduction is chosen as a measure instead of difference of speed because it helps normalize the inconsistencies that can come from different speed limits on different road segments.

3.5 Feature Selection

Using Eventful we obtained information on the event's type, popularity and day of week. Using backward selection, we found that there was a strong correlation between spatio-temporal distance and the event i.e. closer we are to the start of the event, the higher the probability for traffic congestion. We also observed this trend with respect to distance from the venue. This is illustrated using the graphs below.

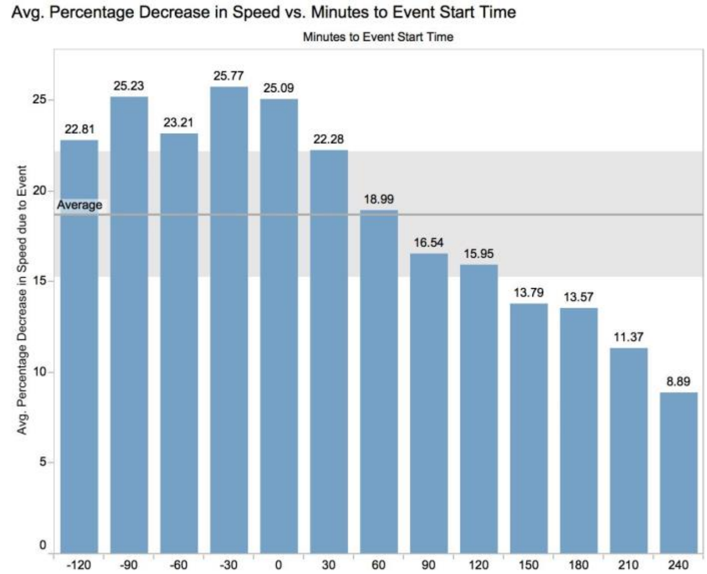
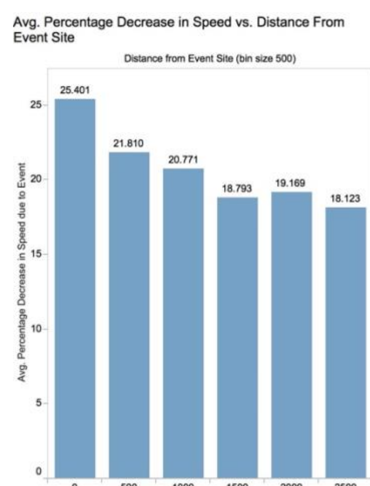


Fig. 2: Congestion vs. Time to Start of Event

Although it is expected that a higher popularity score, would mean higher traffic congestion, during our Exploratory Data Analysis (EDA) we found that there is some noise or inconsistency in this trend. Since our data is sparse, with only 19 events to train with, we didn't have enough data to prove this observed trend otherwise. Hence, we account for the popularity score differently as explained in the section below. Also, during the EDA process, it was observed that the difference in traffic speeds between ED & NED was more during weekends by 2miles/hr on an average, however this data was a little noisy as well [2]. Here, weekends mean Friday, Saturday and Sunday; weekday refers to the other days of the week. The type of the event had not shown much significance in our analysis.



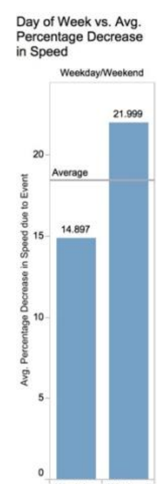


Fig3: Congestion vs. Distance from Event, **Fig 4**: Congestion vs. Weekend/Weekday

3.6 Machine Learning

$$S = 0.895 * Regressor(D_e, T_e) + 0.005 * Popularity + 0.1 * W + 1$$

From our feature selection process, we identify that *distance from the event*, D_E, and *time to event*, T_E, are the two main independent variables that affect the *percentage change in traffic speed*, S. We randomly set aside 20% of the dataset for testing and 80% of the data for training (~5000 data points). Upon trying Linear Regression, Polynomial Regression, SVM and Random Forest Regression (RFR), RFR gave the least error. Hence, the Random Forest Regressor was chosen for learning S. In addition to D_E and T_E, we also take popularity, P & weekday/weekend, W, into account (1 for weekend, 0 for weekday). Given their noisiness in our data, we add them heuristically in our model instead of incorporating them into the regression model.

3.7 Front End User Experience

So far, we have described all the tasks required to collect the data and build a prediction model. Now that we have these, we will next describe the pipeline that occurs when a user uses PRIPOSE to generate a route.

Pseudo-code

- Input City, OD pair, time of departure, T_D
- Create OSMnx graph for given City
- Extract current date from the computer, and extract events from Eventful API for current date. Event information will have Time of Event, T_E and Popularity, P.
- W = 1 if date shows it is a friday, saturday or sunday (weekend), 0 otherwise
- Use Google Maps API, obtain Time of Journey, T_J. Comparison Time $T_C = T_D + 0.5 * T_J$
- Loop over all events of the day

  Time Difference in minutes, $Diff = T_E - T_C$

  Select event IF (-120 min < Diff < 240 min):

  - ❑ Compute $S = 0.895*Regressor(D_E, T_E) + 0.005*P + 0.1*W + 1$, where D_E is computed for 100,200,300,400,500 m from event site.
  - ❑ Find the distance $D_{E,max}$ in the set (100,200,300,400,500) that has a maximum corresponding S value, S_{max}
  - ❑ For all distances smaller than $D_{E,max}$, set their corresponding S value to S_{max}
  - ❑ For all distances larger than $D_{E,max}$, iteratively set their corresponding S values to $0.9 * D_{E-1}$ (i.e S corresponding to 500 m is 0.9 * S corresponding to 400 m)
  - ❑ For distances D_E = 100 to 500:

    Define S_E as the S corresponding to D_E. For all street segments within D_E distance from the event site, update the corresponding length of street segment, i.e segment length = segment length* S_E
- Given that the city graph has been altered to reflect the congested traffic routes, compute the shortest distance path between origin and destination. The algorithm will intelligently circumnavigate through the traffic prone areas caused by scheduled events.

To describe the above pseudo-code briefly, we generate a bunch of candidate events based on the spatio-temporal proximity to the users location and time of departure. Our prediction model estimates the traffic congestions caused by these candidate events. Using OSMnx, a geo-spatial street network module, we penalize road segments near event sites by lengthening them. The factor we lengthen them by is based on event parameters as the spatio-temporal proximity to the event. We then calculate a new route based on this penalized graph. This algorithm intelligently avoids traffic prone areas and circumnavigates them thereby decreasing drive time.

4. Evaluation

We split this section into two parts. First, we describe our method of statistically evaluating our method and show the corresponding results. Then, we show a real-life case study of how our method of future route prediction performs against Google Maps future route prediction.

4.1 Statistical Analysis

Here, we compare our PRIPOSE predicted route with a naive predicted route in presence of PSEs. The naive route is defined as the shortest route from origin to destination computed on a graph using unmodified road segment lengths as the edge weights. As a side note, we found it was often the case that when requesting a route from google at a future time, they returned this shortest route. The PRIPOSE route is defined as the shortest route computed on a weighted graph that our model outputs based on PSEs. We begin our experiment by taking the map of a city (Chicago) and fetching the most popular events that will affect traffic given the current time. We then input these events into our model which gives us a weighted graph. G0 is defined as the unmodified Chicago graph, and G1 is the modified one from our model. Next, we generate 150 random origin-destination (OD) pairs that were around 5 km apart. For each OD pair, we first calculate the time it takes to travel on the naive route on G0 and use the distance and time to calculate the average speed. We then overlay both the PRIPOSE and naive routes onto G1 and find the total penalized distance it takes to traverse both of these. We justify these new total distances by the fact that we simulate increase congestion with stretching the road segments. This is done in a proportional way, i.e. if we predict the average speed decreases by half, we keep the same average speed but double the length of the road segment. Once we have both of the new distances for these two routes along with the original average speed, we estimate the time it takes to traverse both of them. For our results, we only consider routes that cross through areas affected by these PSEs. There will be nothing interesting about comparing two routes that go through the same unpenalized section. On average, we found that the PRIPOSE route reduced travel time by around 1 min and 5 secs for 5 km routes, which ended up being a 6% decrease in travel time.

4.2 Case Study

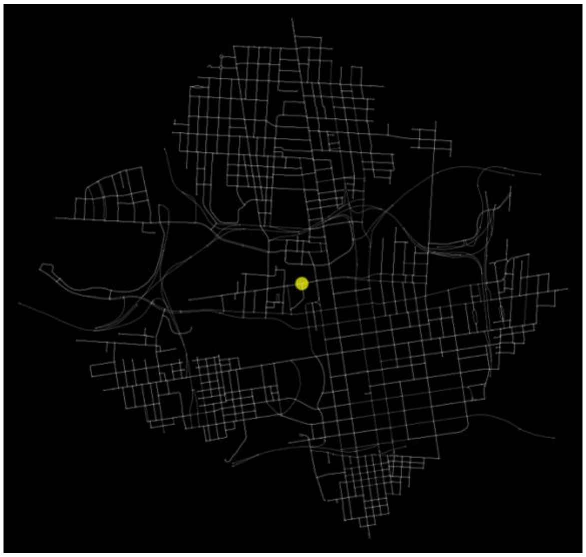


Fig5: Event Site of Concert

In this section, we explore a case study of PRIPOSE routing versus a popular routing application, Google Maps. The event we track is a Justin Timberlake concert in Columbus Ohio on May 7th at 7:30 pm. Its location is shown above in fig 5. The purpose of this study is to show that PRIPOSE *predictive* routing can outperform Google Maps *predictive* routing. PRIPOSE predicts with the knowledge of future events that will occur, Google Maps predicts with the knowledge of average historical traffic patterns at this location and time. We do this by computing routes at 7:30 pm for a time of departure of 10:30 pm and compare the results.

Fig 6: OD pair and shortest route

We create a route from a selected OD pair that passes nearby the event venue. Above is the shortest path only considering road distance from the start (bottom) to the end (top).

Fig 7: PRIPOSE route for given OD pair

We then query this event and penalize road segment lengths according to our model. We set the time of departure to 10:30 pm (recall the current time is 7:30 pm). A new route is calculated on the weighted graph, and the results are shown above. As you can see, the route is adjusted to bypass the predicted increase in traffic density.

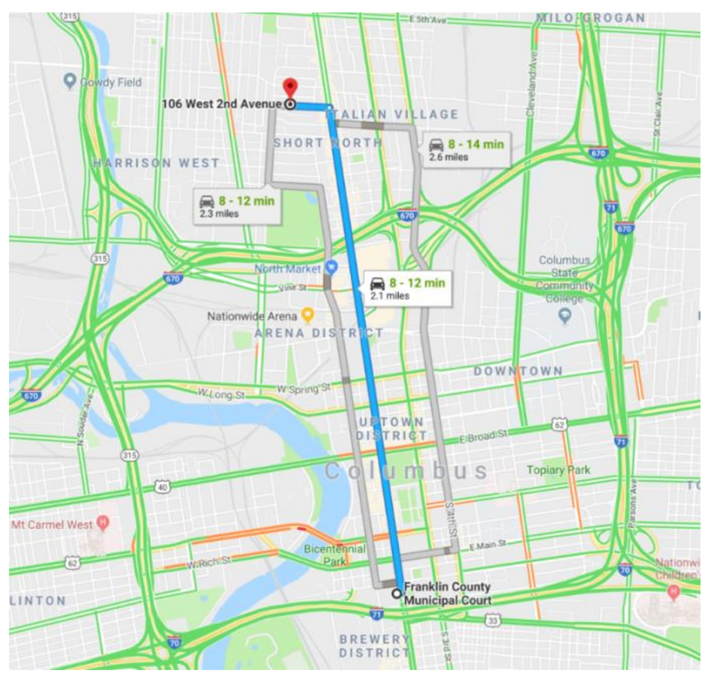


Fig 8: Google Maps Predicted Route | Real Time: 7:30pm, Time of Depature: 10:30pm

We also request a route from Google at 7:30 pm, setting the time of departure to 10:30 pm. It suggests the shortest path shown in the first graph as it has no reason to believe there will be an increase in congestion three hours from now.

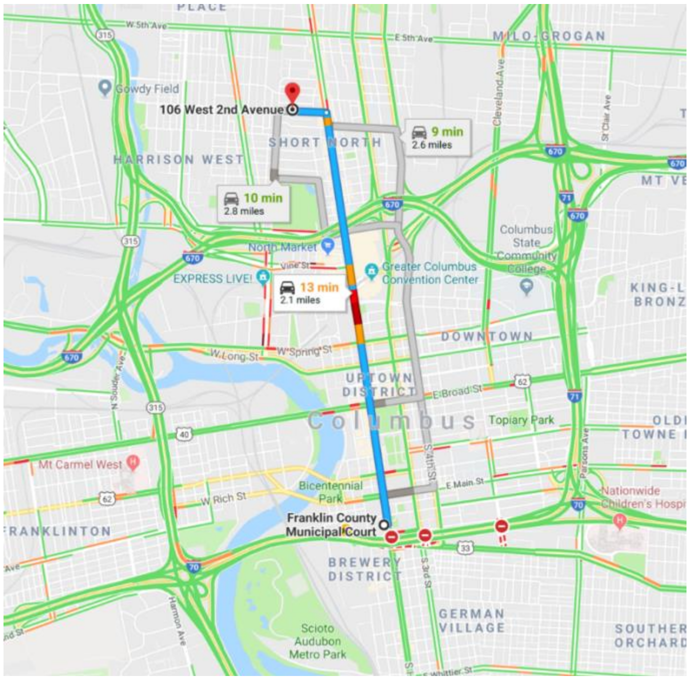


Fig 9: Google Maps Predicted Route through Live Traffic | Route predicted at 7:30pm | Overlayed on live 10:30 map

The time is now 10:30 pm. Above we show the route originally suggested by Google at this time. As you can see, there is a noticeable increase in congestion that we hypothesize is caused by event attendees leaving the concert. According to live traffic models, this route will take 13 minutes.

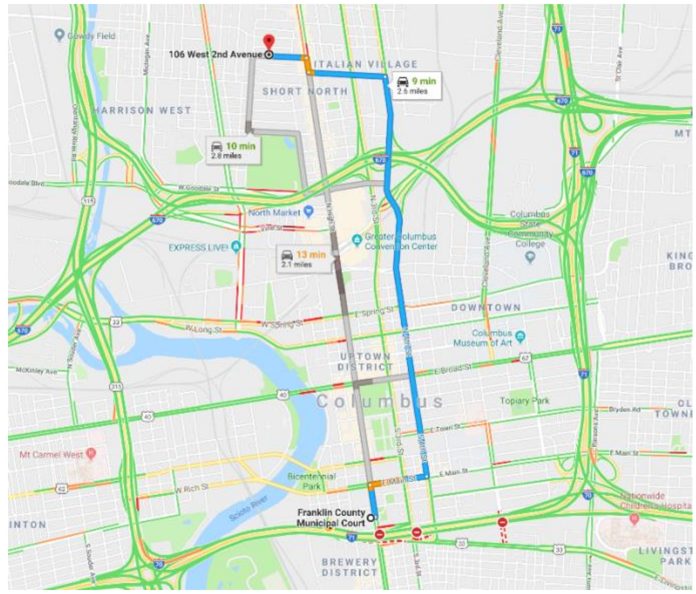


Fig 10: Google Maps Suggested Route through live traffic | At 10:30pm

Here is the route Google suggests taking at 10:30 with the knowledge of live traffic data. It bypasses the traffic by taking the road to the East, and the predicted time is 9 minutes.

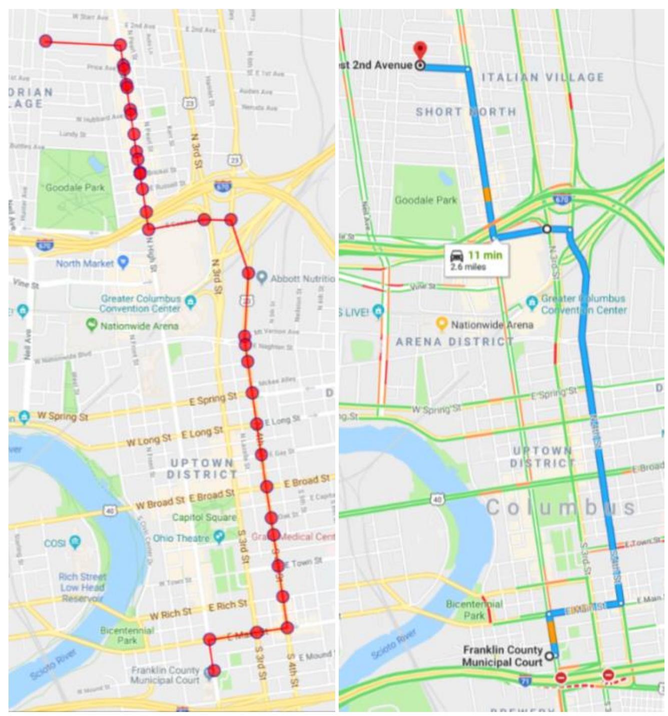


Fig 11. PRIPOSE predicted route in Google Maps

Finally, we show the route PRIPOSE suggested at 7:30 with knowledge of this event. We have imported it into Google Maps, which tells us that this route would have taken 11 minutes in the presence of live traffic at 10:30.

Here we have shown that PRIPOSE understood that the best route was to bypass traffic by taking the road to the East. While it was slightly different than Google's best route calculated at 10:30, it still would have taken 11 minutes compared to Google's original suggestion of 13 minutes. Thus, PRIPOSE's predicted route beat Google's predicted route by 2 minutes. In this particular case study, it is obvious that if the user requested this route from Google at 10:30 pm, it would've given them the best route as the live traffic was showing congestion. We argue, however, that there are situations where traffic can be normal at the time of route calculation but can quickly deteriorate while driving along the route. At this point, the driver may be too far along in the trip to take an effective detour. Whereas if this future traffic increase was predicted at the time of route calculation, an optimal route can be chosen from the beginning. Knowing more information earlier will always lead to a more informed decision.

4. Challenges

Several challenges were identified during different phases in our project.

4.1 Model Learning

To train the regression model for prediction of congestion, our framework requires event information, along with traffic data of spatiotemporal proximity. Additionally, it requires the baseline traffic data for each road segment when there is no event at the nearby event venue, to be used for comparison.

Our team searched for open sourced traffic data and tried to maximize the usefulness of such data by analyzing them and gathering event information for the corresponding venues that are covered by the traffic data. Nevertheless, it was not very successful because open sourced traffic data was very sparse in coverage, low in volume, and limited timewise. Thus, we decided to gather the most recent data by scraping it from existing real-time traffic monitoring applications. Our team was able to gather significant amount of traffic data that corresponds to events that we scraped earlier in the progress.

4.2 Evaluation

There was no practically effective means to evaluate by how much our routing model outperforms other routing algorithms, because such algorithms can only be tested in real life, to provide promising results. Existing traffic simulators result in a biased result because such simulations are based on usual traffic behavior that can be predicted, while our model aims to enhance prediction of unpredictable congestions.

Thus, we have decided to do brute forced real life comparison between the route create by our algorithm, and the route created by the baseline routing algorithm that is used by Google Maps. Considering the practical limitations, this evaluating method is enough to evaluate and determine that our framework does provide routes that are faster than those from other methods as a proof of concept.

4 Future work

Given more time and resources, we have a variety of tasks that would improve the overall quality of our framework. Here, we discuss what improvements we would like in regards to our data, our model generation, and our routing. In terms traffic data, we would have benefited from collecting denser data at more locations over a longer period of time. This provides us with data that is easier to remove noise from as well as give us the ability to have more complex models with more parameters. Furthermore, finding a good source that can confidently provide us with more event data would be very useful. One of the key assumptions at the beginning of our project was that we would have access to all sorts of event parameters such as number of attendees, venue size, etc. which we weren't able to get our hands on. Finding more explanatory variables would also be helpful for our model. Next, there are other techniques we can try for fitting a model to our data. We can experiment with first clustering based on a parameter such as event type, and then creating a separate model for each category. Techniques like this have proven to be useful in other social sensing papers. However, we would need more data first in order to effectively utilize this option. We also can try taking the idea from [1] and explore the addition of collecting live traffic data before the event to help improve our model for congestion after the event. We could also compare the performance of our approach with theirs. Last, the most useful application of our new traffic data would be to merge it with existing historical average traffic data such as Google's. Having both average historical traffic data for a given location and time along with our updated traffic data with knowledge of future events would provide optimal routing for the user (instead of just one or the other).

5 Conclusions

Our framework successfully predicts potential future traffic congestion caused by planned special events that are intermittent and unpredictable. Compared to the routes created by a baseline routing application, routes from our framework improved temporal efficiency by 6%. We have also shown its potential for routing around congestion with a real life case study. By performing this rerouting, we can alleviate overall congestion by diverting traffic around the potential future areas of congestion. Other benefits include lowering CO_2 emissions exhausted during idling and lowering stress amongst drivers. We believe we have shown good potential for the idea of adding knowledge of PSEs to existing traffic models. There is still future progress that needs to be made but incorporating PRIPOSE into routing can be a very impactful tool.

6 References:

[1] Kwoczek, S., Di Martino, S., & Nejdl, W. (2014). Predicting Traffic Congestion in Presence of Planned Special Events. In *DMS* (pp. 357-364).
[2] Chrobok, R., Kaumann, O., Wahle, J., & Schreckenberg, M. (2004). Different methods of traffic forecast based on real data. *European Journal of Operational Research*, *155*(3), 558-568.
[3] Kwon, J., Mauch, M., & Varaiya, P. (2006). Components of congestion: Delay from incidents, special events, lane closures, weather, potential ramp metering gain, and excess demand. *Transportation Research Record: Journal of the Transportation Research Board*, (1959), 84-91.
[4] TomTom International. White paper - how tomtoms hd traffic and iq routes data provides the very best routing. Technical report, TomTom International, 2009.
[5] TomTom. 2017. TomTom Maps APIs for Developers/Online Traffic. [ONLINE] Available at: https://developer.tomtom.com/online-traffic.
[6] "Eventful API." *Events Feed, Concert & Event API - Eventful API*, api.eventful.com/.
[7] gboeing. 2017. OSMnx: Python for street networks. Retrieve, construct, analyze, and visualize street networks from OpenStreetMap. [ONLINE] Available at: https://github.com/gboeing/osmnx.
[8] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. H. Byers. Big data: The next frontier for innovation, competition, and productivity. Technical report, McKinsey Global Institute, 2011.
[9] B. Pan, U. Demiryurek, C. Shahabi, and C. Gupta. Forecasting spatiotemporal impact of traffic incidents on road networks. In Data Mining (ICDM), 2013 IEEE 13th International Conference on, 2013.
[10] Sugiyama, Yuki & Fukui, Minoru & Kikuchi, Macoto & Hasebe, Katsuya & Nakayama, Akihiro & Nishinari, Katsuhiro & Tadaki, Shin-ichi & Yukawa, Satoshi. (2008). Traffic jams without bottlenecks - experimental evidence for the physical mechanism of the formation of a jam. New Journal of Physics. 10. 33001. 10.1088/1367-2630/10/3/033001.