Transportation Research Part C 107 (2019) 444–462

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Predicting real-time surge pricing of ride-sourcing companies

Matthew Battifarano[a], Zhen (Sean) Qian[a,b,*]

[a] Department of Civil and Environmental Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, United States
[b] Heinz College, Carnegie Mellon University, Pittsburgh, PA 15213, United States

ARTICLE INFO

Keywords:
Data mining
Ride-sourcing service
Surge pricing
Clustering
Multi-source data

ABSTRACT

Ride-sourcing companies such as Uber and Lyft represent a popular and growing mode of transit in cites worldwide. These companies employ surge pricing in real time to balance the needs of both drivers and riders. The prediction of surge prices in the next few minutes to hours encapsulates the complex evolution of service fleets and service demand in the short term. Surge pricing, if effectively predicted and disseminated to both drivers and riders, can be used to more efficiently allocate vehicles, save users money and time, and provide profitable insight to drivers, which ultimately helps the efficiency and reliability of transportation networks. This paper explores the spatio-temporal correlations between the urban environment, traffic flow characteristics, and surge multipliers. We propose a general framework for predicting the short-term evolution of surge multipliers in real-time using a log-linear model with L_1 regularization, coupled with pattern clustering. This model is able to predict Uber surge multipliers in Pittsburgh up to two hours in advance using data from the previous hour out-performing the overall mean and the historical average in all but 3 of the 49 locations in Pittsburgh and outperforming three non-linear methods in 28 of the 49 locations. The model is able to out-perform the overall mean, historical mean, and non-linear methods on Lyft surge multipliers in Pittsburgh up to 20 min in advance. Cross-correlation of Uber and Lyft surge multipliers is also explored.

1. Introduction

Ride-sourcing companies, sometimes also known as Transportation Network Companies (TNCs) as a more general term, like Uber and Lyft have become a common presence in cities worldwide. Despite their growing popularity, much of their operational metrics remain opaque. In particular, little is known about *surge pricing*: the dynamic pricing mechanism employed to ensure sufficient vehicle supply during periods of high rider demand. Though it is unclear how surge pricing is updated at each time point, forecasting the surge prices in the next few minutes to hours would appear to benefit all parties. If effectively predicted and disseminated to both drivers and riders, future surge prices can be used to (1) understand the foreseen results of evolution of service vehicles and demand for transportation managers; (2) help ride-sourcing companies navigate the changing transportation landscape for real-time operation; (3) provide profitable insights to drivers; and (4) save riders' money and time.

The existence of surge price multiplier implies a spatio-temporal imbalance of supply and demand in a ride-sourcing system. With this in mind, surge pricing can be viewed as a market correction due to information not already priced into the cost of a ride (Gurley, 2014). In other words, it is a measure of unanticipated demand for ride-sourcing services and a proxy for unanticipated travel demand more broadly. Predicting surge multipliers is therefore a way of characterizing the nature of unanticipated demand within a ride sourcing system. Under the assumption of perfect information, surge pricing should be random and brief. If it can be predicted

* Corresponding author at: Department of Civil and Environmental Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, United States.
E-mail addresses: mbattifa@andrew.cmu.edu (M. Battifarano), seanqian@cmu.edu (Z.S. Qian).

https://doi.org/10.1016/j.trc.2019.08.019
Received 31 January 2019; Received in revised form 23 August 2019; Accepted 23 August 2019

then there is room for greater efficiency in the allocation of vehicle supply. This paper does not intend to reverse engineer the underlying mechanism of surge pricing offered by ride-sourcing companies. Rather, we use machine learning methods to predict surge prices minutes or hours ahead of time in selected locations in a city, regardless of surge pricing strategies. The prediction is performed with a data-driven approach. In fact, the surge prices become a holistic metric that measures the relation of vehicle supply and rider demand, both temporally and spatially. As much as riders and drivers would like to forecast surge prices for trip/business planning, it would also benefit ride-sourcing companies to pro-actively manage fleet operation with knowledge of the distribution of vehicles and riders in advance.

Specifically, a real-time predictive model of price surge multipliers over time and locations is of interest to several parties in the ride-sourcing ecosystem.

- **Drivers** may use the model to increase their earnings by acting on better forecasts of when and where surge pricing is likely to occur.
- **Riders** may use the model to plan their departures to avoid surge multipliers. In particular, riders could know whether the surge multiplier is soon likely to increase or how long they would likely need to wait for the surge multiplier to decrease.
- **Ride-sourcing companies or TNCs** may use the model to more intelligently dispatch their fleet or offer additional incentives to drivers to better balance supply and demand.
- **Third parties**, such as Gridwise Inc., who sell ride-sourcing intelligence to drivers, may use this model to improve their products. Public agencies may use this model to assess gaps in public transit operations or to implement targeted TNC policy.

A reliable, transparent model enables drivers to make choices with confidence. Drivers do not always trust market information provided by Uber since Uber's strategies may not be in alignment with individual drivers (Guda and Subramanian, 2017; Rosenblat and Stark, 2016). If drivers can not trust that the reported surge multiplier is reflective of the fare multiplier they will actually receive (or the profitable trip order they will receive), then surge pricing will not function as intended. In particular, by having an estimate of future surge multipliers, drivers would be better able to assess the opportunity cost of relocating. Predictions help drivers at a range of timescales. In the short-term, it may help drivers relocate in time to catch a surge. At longer horizons it may help drivers choose which trip to take now. For example, suppose a driver knows it will likely surge in 1 h at location B 20 miles away. In choosing their next trip, they may opt, all else equal, to take the one in the direction of location B so that they are set up to meet the surge.

If potential riders had access to near-term forecasts of surge multipliers, they would be able to better plan their departure times or their mode choice. When faced with a surge multiplier, many users who have opened the app choose to postpone their trip or seek an alternate mode. Broadly, this indicates that demand for ride-sourcing services is elastic (Hall et al., 2015). If riders have access to a surge price forecast they may decide to take a trip in time to avoid the surge price. In this way, surge prediction not only saves the rider money but offers the TNC a more efficient use of their vehicles.

Accurately forecasting surge multipliers is more valuable to TNCs than forecasting demand because it highlights precisely the travel demand that TNCs are not currently able to serve effectively. Because TNCs like Uber and Lyft do not employ drivers directly, they may only incentivize, rather than compel, drivers to relocate to areas of high demand. A predictive model of surge multiplier could then be utilized to help direct drivers, in real-time, to areas before the imbalance of supply and demand materializes. It may also be used, in tandem with their own proprietary data, to generate hypotheses as to why surge multipliers are predictable at all and what interventions may be necessary to better allocate the fleet.

A better understanding of the dynamics of demand and ride-sourcing service would enable the development of better tailored policies for regulating them. Government agencies have adopted a wide-range of policies for regulating these now-ubiquitous services. A 2017 report by the San Francisco County Transportation Authority found that Uber and Lyft accounted for 15% of all vehicles trips in San Francisco (San Francisco County Transportation Authority, 2017). London, Denmark and Germany among other European countries have banned them entirely (Satariano, 2017). In 2015, New York City mulled a licensing freeze (Lapowsky, 2015).

This paper introduces a real-time spatio-temporal predictive model of Uber and Lyft surge multipliers based on measurements of the urban environment and traffic characteristics in the recent past. We build a location-dependent log-linear model that takes a comprehensive set of spatio-temporal features extracted from multiple sources over the past hour to predict the future surge multiplier at a fixed time horizon.

The rest of the paper is organized as follows. Section 2 discusses prior research on predictive models and prior analysis of ride-sourcing companies—and of surge pricing in particular. Section 3 describes the data used in this paper. Section 4 presents the model formulation. Section 5 presents the quantitative and qualitative results of the predictive model on real-world data. Finally, Section 6 summarizes the results and contributions of this paper as well as potential extensions.

2. Prior work

Ride-sourcing companies like Uber and Lyft employ surge pricing to balance supply and demand. When the number of requests for service in a location exceeds the number of nearby vehicles, a so-called **surge multiplier** is introduced. By increasing the fare, TNCs intend to encourage riders to postpone their trip and reward drivers who relocate to serve the demand.

Initially introduced to Uber's platform in 2012, surge pricing was aimed at increasing the percentage of requests for rides that are fulfilled, known as the **completion rate** (Gurley, 2014). The magnitude of the surge multiplier is driven by the wait times experienced by the users, which in turn is driven by the rate at which demand is out-pacing supply (Gurley, 2014; Hall et al., 2015). Uber displays the surge multipliers to drivers on an in-app heatmap with the intent of "nudging" them from areas of low demand into areas

of high demand (Rosenblat and Stark, 2016). Cohen et al. (2016) find that for major cities in the United States Uber is surging between 14% and 28% of the time. Internal estimates from Uber place this figure at around 10% platform-wide (Gurley, 2014). We find that Uber is surging in Pittsburgh approximately 8% of the time. Chen and Sheldon (2016) characterize Uber surge multipliers; in particular, that Uber surge multipliers are bounded between 1.2 and a per-market maximum, and are discrete in increments of 0.1. Further, there is a maximum rate of change for surge multipliers.

Later analysis has suggested that in addition to direct measurements of their own system, Uber is using predictive analytics to anticipate future surges in demand and shortages of drivers (Guda and Subramanian, 2017; Rosenblat and Stark, 2016; Phillips, 2017). In addition to historical platform data, Uber's demand forecasting relies on environmental data such as weather, local events, and traffic conditions (Guda and Subramanian, 2017). While it is known that Uber uses forecasted demand to provide notifications to drivers, it is unclear whether the it is also used to compute the surge multiplier displayed to drivers (Guda and Subramanian, 2017; Rosenblat and Stark, 2016).

In order to function properly as a signal to re-allocate vehicle supply, surge multipliers provided to each driver should be predictive. That is, the surge multipliers provided to the driver should reflect the surge the driver should expect to see at a location when they arrive should they travel there from their current location. Drivers frequently report arriving to a surge area only to find that the surge has ended (Guda and Subramanian, 2017). In fact, one of the most prominent pieces of advice offered by experienced drivers to those new to the platform is "don't chase the surge" (Rosenblat and Stark, 2016). In fact, it has been shown that drivers routinely ignore the surge multiplier and other market information provided by Uber. In one study, surge pricing in certain areas of New York City and San Fransisco caused drivers to *avoid* the area entirely (Chen et al., 2015). This behavior is likely caused by driver's widespread mistrust of the market information provided by Uber (Lee et al., 2015; Rosenblat and Stark, 2016). This mistrust is not unfounded: in certain circumstances the platform operator has an incentive to mis-report the surge multiplier (Guda and Subramanian, 2017).

It should be noted that in contrast, several studies have shown that surge pricing is an effective mechanism to increase supply. Hall et al. (2015) examines a case study of Uber surging around a specific well-attended event. When surge is in effect, supply rises to meet demand and completion rate remained high. In comparison, when the surge pricing system experienced an outage on New Year's Eve, completion rate fell dramatically during the outage. Further, Chen and Sheldon (2016) show that drivers remain on the platform longer and accept more rides when surge pricing in is effect.

Uber has recently indicated that it may move away from explicit fare multipliers in favor of "Upfront Pricing": simply displaying the total cost of the ride to the user at the time of request (Uber Technologies, 2018). However, the displayed fares still *implicitly* contain a fare multiplier and thus the model presented in this paper would still apply.

To our best knowledge, surge multipliers have previously only been modeled from a theoretical perspective. Guda and Subramanian (2017) develop a two-region economic model to examine driver behavior under surge multipliers from a game theoretic perspective. Yang and Yang (2011) propose a "meeting function" to investigate the equilibrium properties of taxi markets. Similarly, Zha et al. (2016) propose a "matching function" to model the e-hailing market. Alonso-Mora et al. (2017) propose an optimization model for matching riders and drivers in a ride-sourcing fleet. Guha et al. (2018) model several aspects of the ride-sourcing platform including surge-pricing in a competitive setting as a differential game.

Several studies view surge pricing from an market equilibrium perspective. Wang et al. (2016, 2015) propose an equilibrium model to examine the effects of several pricing strategies on e-hailing platforms. Similarly, Bimpikis et al. (2019) propose an time-invariant equilibrium model of riders and drivers on a network, investigating the spatial effect of surge pricing. Zha et al. (2018) leverage a discrete-time geometric matching algorithm to examine the spatial effects of surge pricing under market equilibrium in time and propose a commission rate cap balancing platform revenue and consumer welfare. Sun et al. (2019) propose an econometric model of labor supply in ride-sourcing markets, using surge pricing as a natural experimental environment.

Predictive models of transportation driven by existing sources of environmental data are well-studied in previous literature (Yang et al., 2019; Yang and Qian, 2019), but not with a focus on surge pricing. Several researchers have built predictive models of travel characteristics other than surge multipliers. Each model uses data from a single system to predict its target. Zheng et al. (2016) develop a real-time predictive model of traffic flow from the CO2 sensors in a large office building in Hong Kong. Zhang and Qian (2018) use household energy consumption to predict the start and end of morning traffic congestion in Austin, Texas. In contrast, Liu et al. (2019) synthesizes taxi GPS data and license plate recognition data on the road network to infer vehicle volume and fleet composition at high geospatial resolution. The synthesis of multiple data sources is conceptually aligned with our approach in this work.

In regard to ride-sourcing systems, Ke et al. (2017) develop a deep learning based approach to short-term demand forecasting in ride-sourcing systems. Similar to our work, they use both historical measurements and exogenous features such as time-of-day, day-of-week, and weather. They also use a spatially aggregated random forest for feature selection. Similarly, Wei and Chen (2012, 2017) develop neural networks to predict short-term demand in metro systems based on historical data and temporal factors. Noursalehi et al. (2018) predict passenger arrivals at metro stations in real time using dynamic factor models.

Our approach is firmly rooted in the latter approach. Our goal is to investigate the extent to which the evolution of the imbalance of supply and demand in ride-sourcing systems in the immediate future could be recovered from features of the urban environment at a specific time, so that it helps inform both riders and TNC drivers for their better decisions without necessarily engaging TNCs. This goal should be contrasted with those of theoretical models which aim to explain the decision making process of individuals or groups by identifying variables which cause them to, on average, change their behavior. This is an important vein of research that can help economists and policymakers analyze the ride-sourcing market. However, a keen understanding of driver and/or rider behavior on average is of limited use in forecasting how surge pricing will actually change over the next few hours. The trade-off between a causal

model of on-average surge multipliers and our data-driven approach is analogous to the difference between traffic equilibrium models and traffic state forecasting. The former models congestion as the result of a decision making process of when to leave and how to travel on average, but cannot be directly used to predict congestion later this evening. Traffic state forecasting on the other hand, can accurately predict the evolution of traffic flow but does not yield insight into the decision making processes of individual drivers in traffic.

Our work departs from prior work in five ways:

1. We build a data-driven model of surge multipliers, an operational characteristic of ride-sourcing service with its associated demand. Other work either builds theoretical models of surge pricing or builds data-driven models of characteristics of general traffic, such as demand or traffic flow.
2. Our model uses a broadly collected, multi-source data set, combining data from multiple disparate systems across the public and private sectors, such as traffic speed, events, road closures, and weather conditions. We employ L_1 regularization to utilize the entire feature set without over-fitting. Most existing work uses data from a single source.
3. Our model uses spatio-temporal features available in real-time from a regional network. The feature set contains measurements at multiple locations and time points, fully exploring the spatio-temporal relations among those features and surge prices.
4. Our model is capable of being run in real-time and is able to predict surge multipliers up to two hours in advance—a relatively broad prediction horizon compared to other real-time predictive models.
5. This paper offers a head-to-head comparison of Uber and Lyft surge pricing.

In terms of prediction horizon, there are generally two types of real-time prediction. The first aims to predict only the next time step and updates itself in real-time, often in the Bayesian sense. Such models are usually referred to as short-term forecasting. Fei et al. (2011) build a real-time short term travel time model capable of forecasting 5 min ahead of time. Hamed et al. (1995) predict the short-term evolution of traffic volume predictive over the next minute. Zheng et al. (2006) predict the short term evolution of traffic flow, again over the next few minutes. The second category which offers predictions over longer time horizons. Relatively few studies fall into this category. Min and Wynter (2011) build a real-time predictive model of traffic up to 1 h in advance. Vanajakshi and Rilett (2004) compare the performance of neural networks and support vector machines in real-time traffic speed prediction up to 1 h in advance. Our model falls into this second category but is able to perform well at a wide range of prediction horizons (10 min up to 2 h).

This paper predicts the surge multiplier up to two hours in the future based on the last hour of observed surge multipliers as well as the last hour of observed environmental features. To our knowledge, real-time predictive models of surge pricing has not been explored in transportation literature. Moreover, synthesis of data across real-time APIs including Uber, weather services, and municipal services to serve as features in a predictive model is also unexplored.

3. Descriptive analysis

Surge multipliers are a rare occurrence; over 90% of 10-min windows over all locations during the study period had no surge (e.g. had a surge multiplier of 1.0). The reported surge multiplier is discretized from an underlying continuous process (Chen and Sheldon, 2016). Three artefacts of the discretization process are visible in the histogram in Fig. 1.

First, Uber and Lyft have different rounding preferences. Uber prefers to round to the first decimal place whereas Lyft prefers to round to increments of 0.5 although both use a finer discretization for small surge multipliers. Second, despite their respective preferences, other values of surge multipliers do occur, but much less frequently than otherwise expected. For example Uber surge multipliers of 1.2 and 1.3 occur approximately two orders of magnitude more frequently than a multiplier of 1.25. Lastly, both Uber and Lyft surge multipliers are capped: Uber at 4.9 (with two specific exceptions) and Lyft at 8.

Uber surge multipliers greater than 4.9 occurred only twice: once in the early hours of Sunday October 30 2016 and once in the

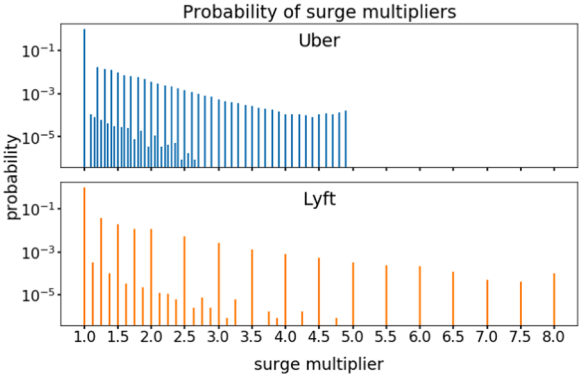
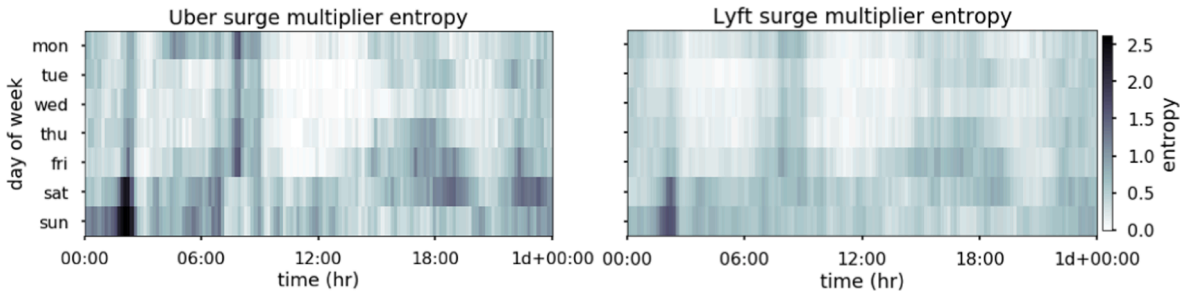


Fig. 1. Probability of surge multiplier occurrence by surge multiplier value in log-scale.



(a) Entropy of Uber surge multipliers.

(b) Entropy of Lyft surge multipliers.

Fig. 2. Entropy of the empirical surge multipliers distributions over day of week and time of day over all locations. Daily patterns in the early morning and during the mid-day during the week are clearly visible.

early hours of New Year's Day 2017 (also a Sunday). In both instances nearly half of the 49 locations experienced a surge greater than 4.9. Interestingly, the locations experiencing large surges were nearly the same on both days. This suggests that there is a maximum surge value for Pittsburgh that may be lifted in certain circumstances. The data from these two instances were omitted from the training data.

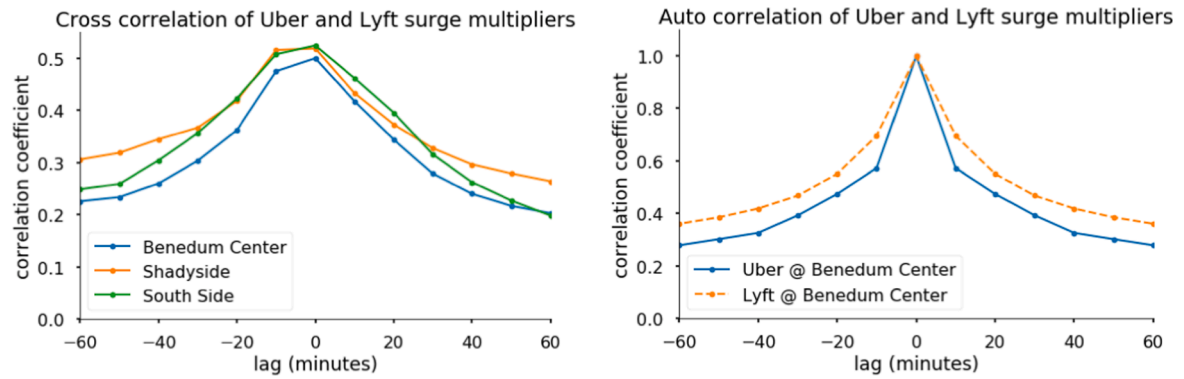
Surge multipliers exhibit some daily and weekly temporal periodicity. Entropy is a measure of spread in a distribution which is used in this setting to identify times where large surge multipliers are more likely. In this case, low values of entropy correspond to periods of low surge, whereas high values correspond to times when larger surge values are more likely. Several patterns can be seen the entropy of the surge multipliers shown in Fig. 2. First, the mid-day during the week (Monday through Friday) experiences the least amount of surging. Second, the AM peak surge around 8:30 AM is visible on weekdays. Third, surge multipliers increase around 2:00 AM, particularly on the weekend. Fourth, the PM peak is visible during the week, but not as pronounced as the AM peak. This is consistent with traditional characterizations of commuter demand in literature, for example in Cain et al. (2001). Lastly, Friday and Saturday evenings are marked with a higher propensity to surge.

Some locations are simply less likely to experience surges than others: for both Uber and Lyft the top six highest surging places will surge around twice as often as average. In general, Lyft and Uber are likely to surge at similar locations and at similar times. For both Uber and Lyft, surge multipliers tend to occur in densely populated areas near the urban core. Both services are between 2 and 3 times as likely to surge in urban areas than in suburban areas. The surge multipliers of two services are also well-correlated in time (Fig. 3).

Compared to Uber, Lyft is slightly more likely to surge in the central business district (CBD) and slightly less likely to surge in populous areas just outside the CBD. Moreover, Lyft surges are well correlated at small negative lag times, implying that Uber surges are generally a leading indicator of Lyft surges.

We summarize our comparison of surge pricing between Uber and Lyft below.

- The surge multipliers on the Uber and Lyft platforms are spatially and temporally well-correlated. Uber surge multipliers are a leading indicator of Lyft surge mutlipliers over short time horizons (<10 min).
- Uber and Lyft prefer different discretization. Uber prefers to round their surge multipliers to the nearest 0.1 whereas Lyft prefers to round to the nearest 0.5. As discussed in Section 5, we hypothesize that this choice has a large impact on model performance.
- Both Uber and Lyft appear to have a maximum surge price in the Pittsburgh market: Uber at 4.9 and Lyft at 8.



(a) Cross-correlation of Uber and lyft surge multiplier at three locations. The surge multipliers are best correlated at 0 lag. However, the asymmetry indicates that Lyft surges more often lag Uber surge than not.

(b) Auto-correlation of Uber and Lyft surge multiplier at the Benedum Center. In general, lyft surge multipliers are more linearly predictive of future surge multipliers than Uber surge multipliers are.

Fig. 3. Auto and cross correlations of Uber and Lyft surge multipliers.

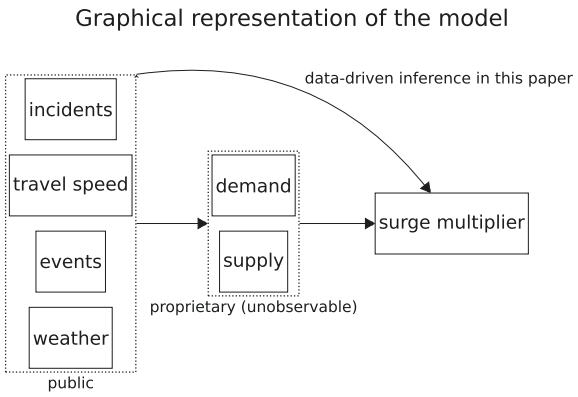


Fig. 4. Conceptual graphical model of our predictive surge multiplier model. Solid arrows represent dependence of variables. The dashed arrow represents the data-driven model we build in this paper.

Road closures as reported by the PennDOT Road Condition Reporting System (RCRS) contain regularly scheduled road work as well as unexpected incidents such as collisions. For the purposes of this study, all closures are treated equally. On average, road closures are expected during the mid-day during weekdays. Moreover, certain tracts are far more likely than others to experience road closures. The urban core, relative to much of the surrounding area is slightly more likely to experience closures. Roadways near the airport are particularly prone to road closures. Similarly, certain tracts are far more likely than others to contain events. This is not unexpected, as events venues tend to concentrate geographically.

4. Methodology

Regardless of whether ride-sourcing companies use predictive models in their surge pricing algorithm, surge multipliers are fundamentally a spatio-temporal function of vehicle supply and user demand. However, vehicle supply and user demand in ride-sourcing systems are currently considered trade secrets and are not made available publicly in any form. Therefore, we may not consider them directly, nor train a model to estimate them. Succinctly, this model represents a choice made in the face of a trade-off between estimating the direct factors of surge multipliers and measuring indirect ones. The principal drawback of estimating the supply and demand individually is that the errors in estimating each compound in estimating the surge multiplier. By skipping the intermediate modeling step, we remove the source of compound error which leads to more accurate predictions. Fig. 4 provides a graphical representation of the predictive model introduced in this paper. Our model leverages the environmental features which have been shown to influence supply of and demand for travel in ride-sourcing systems.

Of course, supply and demand in ride-sourcing systems also likely depend on socio-demographic, land-use, and other location-specific features. Our methodology implicitly models location-dependent features by training a separate model for each location at which data has been collected. Any spatial feature which is not also time-varying would thus be a constant in the training set and could not meaningfully contribute to the model.

As a result of the fitting a separate model for each location, the model trained at a location generalizes to new data collected at that location. In order to predict well on a new location the model must be trained with data collected about that location. It may not simply use the fitted parameters from another location. There are two main reasons for this. First, many features used in the model are not stationary over long periods of time. The distributions of weather features, for example, are not stationary over the year. Events are similarly non-stationary: sports have on and off seasons and outdoor events are less common in winter. On short time scales however, the assumption of stationarity is reasonable for many features including weather measurments. By limiting the training set to only data collected to a finite time period prior to the held-out test data, we are able to apply a model which assumes stationarity. Second, locations may exhibit idiosyncratic relationships among environmental features and ride-sourcing supply and demand. By limiting the training set to only one location per model, we implicitly model the complex and multi-faceted relationship between location-specific characteristics (e.g. land use, demographics).

Because the surge multipliers are bounded below by 1, linearity is likely an unrealistic assumption. Specifically, the transition between no-surge and surge is likely not linear. We address this issue in three ways. First, we predict the log of the surge multiplier. Even though surge multiplier are bounded below by 1, predicted surge multipliers between zero and one can be thought of as a discount when supply far outstrips demand. In practice, instead of setting a discount to boost demand when supply is too high, Uber and Lyft rely on drivers to either relocate or log off of the system when they are having a hard time finding passengers. Predicted surge multipliers less than zero, however, have no meaningful interpretation. By using a log-linear model predicted surge multipliers are always positive. This has the additional effect of reducing the impact of training error on those samples for which the model predicts surge multipliers less than one.

Second, we compare the performance of the log-linear model to non-linear methods. We find that the log-linear model generally out-performs the non-linear methods. In this case the linearity constraint combined with the L_1 regularization yields a model that generalizes better than more complex non-linear models despite the fact that surge multipliers likely do not depend linearly on the

features.

Third, we cluster the surge multiplier timeseries at each location and use the distance to cluster centroid, based on the last hour, as a variable in the model. This helps differentiate no-surge from surge in the relatively common case when there is simply no surge multiplier for most of the day at a location.

In short, we claim that our methodology is able to ingest five month's worth of data about a specific location, and produce a fitted model capable of reliably predicting the short-term evolution of the surge multiplier at that location given data collected in the following week.

We outline the full procedure for training the model below, followed by full descriptions in each subsection.

1. *Data collection.* All data used in this model is available in real time from web APIs. Data is collected using a straightforward python script[1] which requests data at regular intervals.
2. *Data processing.* The API response data must first be parsed then re-indexed to a common timeseries—in our case, even 10 min intervals. Spatial data is aggregated to its containing US Census Tract.
3. *Temporal segmentation.* To account for time-of-day heterogeneity, data is segmented into heuristically defined time-of-day windows.
4. *Clustering.* The timeseries of surge multipliers is clustered within each time of day window. Clusters capture natural modes (or patterns) in the data over longer time periods. In particular, clustering obviates the need to include day-of-week variables. Distance to each of the cluster centroids is then included as a feature in the model.
5. *Training.* A linear model with L_1 regularization is trained on Uber and Lyft data, respectively, using the scikit-learn Python package (Pedregosa et al., 2011). A separate linear model is trained for each time-of-day window, location, and prediction horizon (Δt). Two-level cross validation is used to select the L_1 penalty and evaluate the model. Each training set is comprised of five consecutive month of data. Each training example contains the last hour of data (from time $t - 1$h to time t) and is used to predict the surge at time $t + \Delta t$. The following week is held out for validation. Mean-squared percentage error is used to measure the performance of the fitted model on the validation data.
6. *Prediction.* Predictions for a full day are made by concatenating the predictions of each time-of-day model.

4.1. Data collection

The Uber ride pricing data contains data on 5,769,456 requests on the Uber platform from September 2016 through the end of March 2017 from 49 locations in the greater Pittsburgh area. Those requests were queried from real-time Uber APIs.[2]

Event data contains 19,052 public events within a 20-mile radius of Pittsburgh from September 2016 through the end of March 2017 collected from four event aggregation websites (Eventbrite,[3] Eventful,[4] LotaData,[5] and, SeatGeek[6]). Some cleaning was applied to the public event data to remove erroneous or irrelevant entries. For each time point, events were then aggregated as counts to census tracts within the study area by geographic inclusion.

Pennsylvania Department of Transportation road condition reporting system[7] (RCRS) data contains road conditions reports from 89,399 incidents in the State of Pennsylvania from August 2015 through mid-May 2017. Only road closure incidents are used. Counts of road closures are aggregated to the tract level.

The INRIX road segment speed data contains 133,958,103 speed measurements from 1,923 traffic management channels (TMCs) in Allegheny County from September 2016 through the end of April 2017. The data was obtained through the Pennsylvania Department of Transportation (PennDOT). Speed measurements were averaged to every five minutes. All records have either a speed measurement or an average speed and nearly 99.5% of records have both.

Weather Underground is an online weather forecasting service which provides a web-based application programming interface (API) to historical weather conditions.[8] Hourly measurements of temperature, dew point, humidity, wind speed, wind gust speed, visibility, pressure, windchill, heat index, precipitation, fog, rainfall, snowfall, and qualitative weather condition were collected within the study period. The weather data is measured hourly from a single station so that the measurements do not vary spatially in this study.

4.2. Data processing

First, the study period was defined as the six-month period from October 2016 through the end of March 2017. The Uber and Lyft surge responses within the study period were smoothed to even 10-min increments by rounding the timestamp to the nearest 10-min.

[1] e.g. https://github.com/mbattifarano/mac-data
[2] https://developer.Uber.com/docs/riders/references/api.
[3] https://www.eventbrite.com/developer/v3/endpoints/events/.
[4] http://api.eventful.com/docs.
[5] https://docs.lotadata.com/apis.html.
[6] http://platform.seatgeek.com/.
[7] https://www.penndot.gov/Doing-Business/OnlineServices/Pages/Developer-Resources.aspx.
[8] https://www.wunderground.com/weather/api/.

Samples that were rounded to the same timestamp were resolved by taking the maximum of the surge multipliers. Timestamps that had no samples rounded to them were interpolated by taking the maximum of any samples in a 20-min window centered at the timestamp. The convex hull of the locations of the INRIX Traffic Management Channels (TMCs) was computed and determined the study region; only those surge locations inside this convex hull were considered. The precipitation, wind speed, and temperature data were collected hourly, then joined to the surge data that occurred during the same hour.

The Road Condition Reporting System (RCRS) data contains several records for each incident, describing the progression of the road incident from the initial closure through the road or lane's re-opening. For each timepoint in the study period, counts of ongoing road closure events were aggregated to the census tract level. For each timepoint in the study period, counts of ongoing public events were aggregated to the tract-level.

The traffic speed data contains speed measurements from TMCs throughout the Pittsburgh region. Speed data was first rounded to the nearest 10 min. For each timepoint in the study period, TMCs were aggregated to the tract level and the mean speed was computed over the TMCs in each tract.

For each timepoint in the study period, counts of ongoing public events were aggregated to the tract-level.

Because we train one model per location, the feature dataset is indexed by place label l and timestamp t. Each feature is then scaled to zero mean and unit variance. Δt represents the prediction horizon (such as 10 min, 1 h, or 2 h). Each record contains the following features,

- The historical mean of the surge multiplier at each of the 49 places at the hour and minute of time t taken over all prior days.
- A vector containing surge multipliers at each of the 49 places at time t. (49 variables)
- A vector containing the average speed over the TMCs in each census tract at time t. (503 variables)
- A vector containing the number of ongoing road closures over the segments in each census tract at time t. (503 variables)
- A vector containing the number of ongoing public events over the venues in each census tract at time t. (503 variables)
- The temperature in Allegheny county at time t. (scalar)
- The wind speed in Allegheny county at time t. (scalar)
- A boolean indicating whether or not is it precipitating in Allegheny county at time t (scalar)
- A one-hot encoding of the weather condition in Allegheny county at time t. (10 variables)

Note that none of these features actually depend on the place label, l, at which we are trying to predict the surge multipliers. In other words the same exact data are used to train the models at each location. This is advantageous. Since we are fitting the parameters for data collected at each place separately, we can directly compare the values of the parameters to gain a more robust understanding of how surge multipliers behave at the network level.

4.3. Temporal segmentation

Surge multipliers exhibit a highly non-linear relationship with respect to time of day. To prevent this non-linearity from degrading the performance of the linear model, time-of-day windows were defined to segment the day into behaviorally distinct periods based. The boundaries of each segment were based on the entropy of surge multipliers by time of day. Entropy is a particularly useful statistical property of surge multiplier distributions: larger values of entropy correspond to longer-tailed surge multiplier distributions. As a result, entropy is a proxy measure for surge activity and window boundaries were defined at the approximate times at which the entropy of the surge multiplier distribution increases or decreases rapidly. Fig. 5 shows the entropy of the surge multipliers at each time point over all locations and all days as well as the chosen time of day window breakpoints. This process resulted in six time of day windows: Early Morning (03 h-06 h), AM Peak (06 h-09 h), Mid-day (09 h-16 h), PM Peak (16 h-18 h), Evening (18 h-21 h), and Late Night (21 h-03 h + 1 day).

When fitting the model, the training set is first split by the time of day of target. Concretely, suppose we have features for a

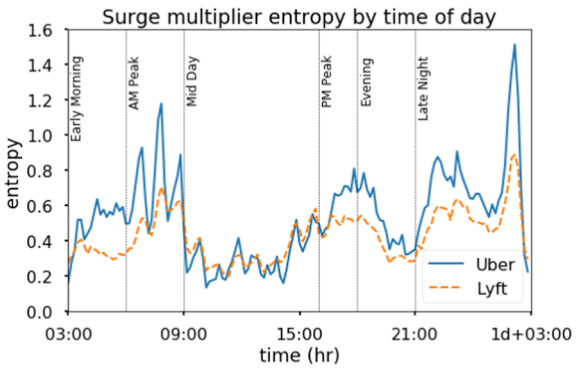


Fig. 5. Daily surge multiplier entropy labeled by time of day segments. Segments were chosen to divide the day into segments of distinct patterns in entropy.

(a) Uber surge multiplier clusters

(b) Lyft surge multiplier clusters

Fig. 6. K-means clustering ($k = 4$) performed on each time windows. Clusters of daily Uber surge multipliers are shown in (a) from largest cluster (top subfigure) to smallest (bottom subfigure). Similarly, (b) shows clusters of daily Lyft surge multipliers from largest (top subfigure) to smallest (bottom subfigure). Blue line shows the mean of each cluster, red line shows median, dashed lines show 10-th and 90-th percentiles. We can assess the relative quality of clusters by noting where the 10 and 90 percentile lines differ from the no-surge event represented in each time window by the largest cluster (top subfigure).

location l indexed by time t. We are trying to predict the surge multiplier at location l at a horizon of Δt. Our training set is then given by $\{(y_{l,t+\Delta t}, X_t)\}_t$. This set of (target, feature) pairs is partitioned by the time of day window in which $t + \Delta t$ resides. For example, if we wish to make predictions 2 h in advance, then a prediction for the surge at 18:30 would use the Evening model, even though the features would come from the hour between 15:30 and 16:30.

4.4. Clustering

Surge multipliers exhibit non-linear behavior with respect to day caused by changes in underlying activity patterns. In particular, distinct modes exist for weekdays and weekends. Instead of modeling day of week explicitly, we use unsupervised learning to partition the time course of surge multipliers into behaviorally distinct modes. The resulting clusters will implicitly model the day-of-week modes as well as their exceptions (e.g. holidays and special events). K-means clustering was performed on the time-course of surge multipliers separately for each time window using the scikit-learn Python package (Pedregosa et al., 2011). The clustering reliably recovers the no-surge mode as the largest cluster in each time window. Clustering within time windows allows time-window specific modal behavior to be recovered. For example, the clusters in the late night time window (the right-most time window in Fig 6) recover three distinct behaviors that are wholly different from behaviors observed at any other time of day. By incorporating cluster information into the model, non-linear longer-term temporal surge behavior can be exploited to refine linear predictions.

Interestingly, cluster membership is nearly independent across time of day bins in the same day. In other words, knowing the cluster of today's surge multipliers during the mid day tells us almost nothing about which cluster this evening's surge multipliers are likely to fall in. We establish this result by noting that adjusted mutual information between cluster labels of different time windows on the same day is small (<0.3). This implies that predictions can be made independently across time of day windows and offers a

heuristic justification for this particular temporal segmentation.

4.5. A log-linear model with L_1 regularization

Eq. (1) defines the linear model for predicting the log of the surge multiplier at location l with prediction horizon Δt where t indexes time. Cluster distance measures the euclidean distance between each cluster centroid and the time course of the surge multiplier up until time t. Window elapsed measures the fraction of the time-of-day window that will have elapsed at time $t + \Delta t$. The historical mean is the average surge multiplier at each location at the hour and minute of $t + \Delta t$ over all previous days. The final term can be interpreted as the state of the urban environment over the last hour. It includes, for each of the last six timepoints (i.e. the last hour), the surge multipliers at all locations, the weather conditions, the traffic speed measurements, and the ongoing events aggregated to the tract-level.

$$\begin{aligned}\log(\text{surge}_{l,t+\Delta t}) \sim & \sum_{i=1}^{4} [\text{cluster distance}_{t,i}] + \text{window elapsed}_{t+\Delta t} + \sum_{l'=1}^{49} \text{historical mean}_{l',t+\Delta t} \\ & + \sum_{t'=t-1\,\text{h}}^{t} \left[\sum_{l'=1}^{49} \text{surge}_{l',t'} + \sum_{i=1}^{503} (\text{mean speed}_{t',i} + \text{n events}_{t',i} + \text{n closures}_{t',i}) \right. \\ & + \text{temperature}_{t'} + \text{wind speed}_{t'} + \text{is precipitating}_{t'} \\ & \left. + \sum_{j=1}^{10} \text{weather condition}_{j,t'} \right] \end{aligned} \tag{1}$$

In total, there are 8898 features. In order to avoid overfitting, L_1 regularization is employed during training (Park and Hastie, 2007). L_1 regularization encourages the model to concentrate the weights on only a small number of the most relevant features. The number of selected features can be tuned by adjusting the L_1 penalty hyper-parameter.

Two-level timeseries cross validation is employed during training. The outer level evaluates the model performance by training the model on moving windows of twenty consecutive weeks and evaluating the trained model on the (held out) subsequent week. The inner level selects the best L_1 penalty weight by cross validation on the training set (Baraniuk, 2007). The two-level cross validation ensures that the model is evaluated on data that was held out from both training *and* L_1 penalty tuning.

Timeseries cross validation differs from other methods of cross validation in that it ensures that the held out data occurs *after* the training data. This is a particularly important consideration for this data set as the features contain past surge multipliers. In short, training a model on the future and evaluating it on the past is not fair. Moreover, we found that timeseries cross-validation was critical in the inner level to select a L_1 penalty term that was large enough to generalize to the validation data.

The performance of the log-linear model is then compared to two naïve methods: the overall mean and the historical mean. The overall mean is simply the average surge multiplier at the location being predicted over all previous days. The historical mean is the average surge multiplier at the location being predicted at the hour and minute of the time being predicted taken over all previous days.

5. Results

We train and evaluate the log linear model on Uber and Lyft surge data for all 49 locations in the greater Pittsburgh area. We focus our discussion of the results on three locations: the Benedum Center, South Side, and Shadyside. These locations were selected because they are likely to surge and are geographically representative of different parts of the urban areas of Pittsburgh. Predictions under the three models are compared to actual surge multipliers in Fig. 9. At these locations we also fit the following three non-linear models and show that the lasso model generalizes the best to unseen data.

1. Random Forest trained on the features selected by Lasso.
2. Random Decision stumps trained on all of the features with the maximum tree depth hyper parameter chosen by cross validation.
3. Neural Network with two ReLu hidden layers of width 32 and 16 respectively and with L2 regularization parameter chosen by cross validation trained on the features selected by Lasso.

We find that the proposed methodology generally out performs both the naïve and non-linear methods.

5.1. Model performance on Uber surge multipliers

When trained on Uber data, lasso regression is able to out-perform both the historical mean and overall mean for 46 of the 49 locations in the greater Pittsburgh area when predicting surge multipliers two hours in advance. Further, it out-performs all three non-linear methods in 28 of the 49 locations. We evaluate the performance of each model on unseen data by the *mean squared percentage error* defined in (2) below. The model out-performs naïve methods in all locations in the urban areas of Pittsburgh and out-performs naïve methods in all but 3 of the 18 suburban locations as shown in Fig. 7. To quantify the relative performance of two methods we compute the *model advantage*, defined in (3), of the linear model with respect to the naïve predictor as the difference

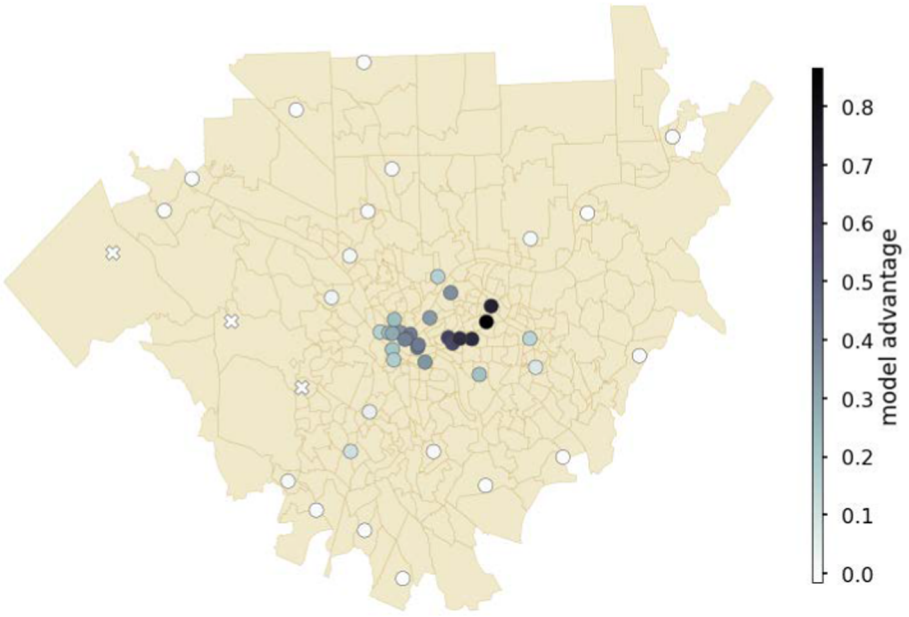


Fig. 7. Model advantage by location at a 120 min prediction horizon. Model advantage is defined as the difference between the mean squared percentage error (mspe) of the overall mean and the mspe of the linear model. Positive values mean that the model out-performed the overall method. X shaped markers denote locations where the model strictly under-performs the overall mean.

between the mean squared percentage errors of the two predictors on held out data. However, suburban locations both are far less likely to surge and far less likely to ever see large surges than locations in urban areas. Specifically, the model strictly out-performs naïve methods for all locations at which there is greater than a 2.5% overall chance of a surge. From the perspective of potential applications, this is acceptable; being able to predict larger, more likely surge multipliers is more valuable to drivers and riders than being able to predict smaller, less likely surges.

$$\text{mspe}(y_{\text{predicted}}, y_{\text{true}}) = \frac{100}{n}\sum_{i=1}^{n}\left(\frac{y_{\text{predicted}} - y_{\text{true}}}{y_{\text{true}}}\right)^2 \tag{2}$$

$$\text{model advantage} = \text{mspe}_{\text{naïve}} - \text{mspe}_{\text{lasso}} \tag{3}$$

To characterize how well we expect our methodology to perform on a new location, socio-demographic data from the US Census and Land Cover data[9] were used in a linear model to explain the spatial variation of model advantage. We find that the composition of land use in the surrounding area determines how well the model will perform. Our model will tend to perform better than naïve methods for locations with a large percentage of commercial land and worse on locations with a larger percentage of low-density residential land. We found that population density, per capita income, and the racial composition of the tract containing each location had a statistically insignificant impact on model advantage. Intuitively, this makes sense as we have no reason to believe most Uber trips originate near the user's home. Moreover, several locations are in tracts in which very few people actually live. For example, the tract containing Heinz field in the north side of Pittsburgh has no residents according to the US Census. In contrast, as Table 1 reports, 78% of the variance in model advantage was explained, with statistical significance at the 0.001 level, by just two land cover variables: commerical land area and low-density residential area, both as fractions of the tract containing the location. In line with our intuition, fraction of low density residential land negatively impacts model advantage while the fraction of commercial land positively impacts model advantage.

Model performance on unseen data varies by location and prediction horizon, as shown in Fig. 8. Intuitively, mean squared percentage error tends to increase with prediction horizon. In each of the shown locations—each in a different part of the urban core of Pittsburgh, the error remains below those of both naïve methods. Interestingly, the overall mean out-performs the historical mean for these locations. This is due to the overwhelming predominance of times at which there is no surge multiplier and emphasizes the lack of strong daily periodicity in surge multipliers.

The proposed methodology also out-performs the three non-linear methods for 28 of the 49 locations. However, lasso remains the best choice overall: when non-linear methods out-perform lasso, they do so with small margins. On average, when non-linear methods out-perform lasso, they reduce mean-squared percentage error by an average of 0.018%. In contrast, when lasso out-performs any of the non-linear methods it reduces mean squared percentage error by 0.08%. We hypothesize that the relatively poor

[9] https://data.wprdc.org/dataset/allegheny-county-land-cover-areas.

Table 1
Results of the linear regression of land use data onto model advantage.

	R^2	0.782		Adjusted R^2	0.772	
	F-statistic	80.52		Prob. F-statistic	1.36e−15	
	coefficient	std err	t	$P \geqslant \|t\|$	[0.025	0.975]
Low-Density Residential	−0.1848	0.015	−12.328	0.000	−0.215	−0.155
Commercial	0.2218	0.023	9.458	0.000	0.175	0.269

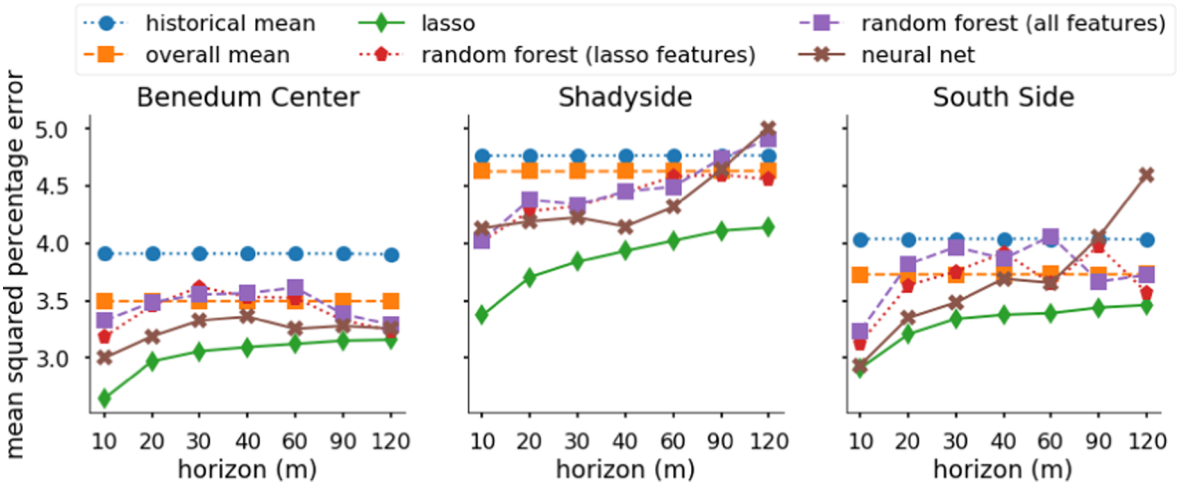


Fig. 8. Comparison of the mean squared percentage errors of Uber models on held-out data in three active surge locations in Pittsburgh over seven prediction horizons. For Uber surge multipliers, the model predictions outperform the historical and overall mean as well as the non-linear methods.

performance of the non-linear methods on held-out data is due to over-fitting. The increased expressiveness of the non-linear methods, even when regularized, ends up learning noise in the training set. In certain cases, the non-linear model fails to out-perform naïve methods. The strict regularization of the L_1 penalty paired with the lower complexity of a linear model lends this method to more robust predictions.

Model performance also varies by time window. Because the surge multiplier distribution changes over time windows, the performance of the naïve methods vary as well. This is particularly evident in the mid day time window, where the overall mean achieves a mean squared prediction error of nearly 2%. The same prediction yields around 3.75% error during the AM peak window preceding it. The historical mean robustly out performs the overall mean in the AM peak, PM peak, and Late night, when surges are most active, but under-performs the overall mean during the mid day and evening windows when surge pricing is less likely.

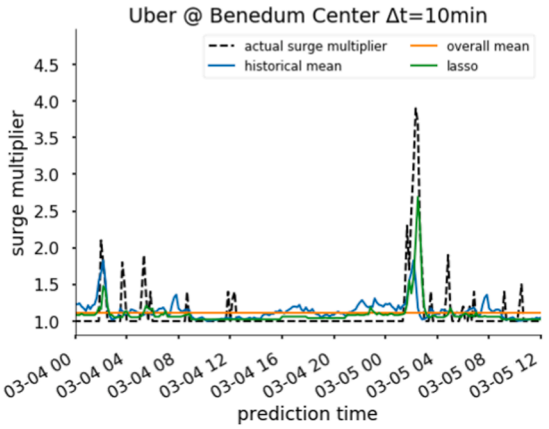
5.2. Model performance on Lyft surge multipliers

In contrast, when trained on Lyft data, the model only slightly out-performs the naïve methods. This is surprising considering that Uber and Lyft surge multipliers are well-correlated. The proximal cause is that the Lyft model has a higher prediction error compared to naïve methods on low surge multipliers (less than 1.5) than the Uber model does. Since both Uber and Lyft are not surging in over 90% of samples, poor performance on low surge multipliers is devastating to prediction error overall. In fact, the Lyft model performs as well or *better* than the Uber model for larger surge multipliers. This suggests that the non-linearity in the transition between no-surge and surge is more pronounced in Lyft than in Uber. In practice, this could be caused by differences in the size the two services' driver or user pools, different promotional practices, or any number of operational or business decisions.

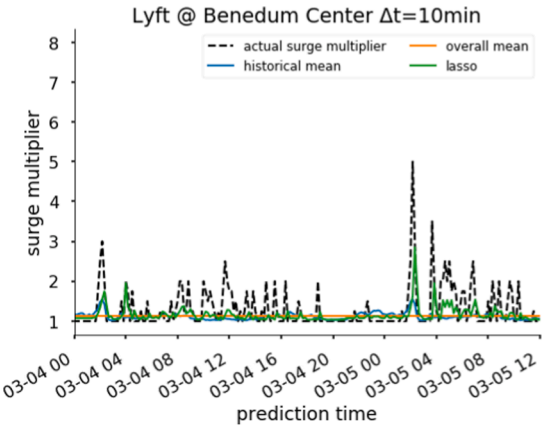
We also hypothesize that the coarser discretization employed by Lyft makes it harder to predict. Intuitively, this makes sense: the error one gets when fitting a line to a step function increases with step size. The coarser the discretization, the more difficult it is to recover a linear relationship. To validate this claim, we re-discretized the recorded Uber surge multiplier to match the discretization used by Lyft and then re-trained the model. When trained on the re-discretized data, the model performance degrades in both absolute value and relative to naïve methods. Overall, prediction error increases by around 1%. In particular, the model performance degrades most on surge multipliers less than 1.5. Although one might expect that a random forest model would be more capable in this regard, our experiments show otherwise.

5.3. Selected features

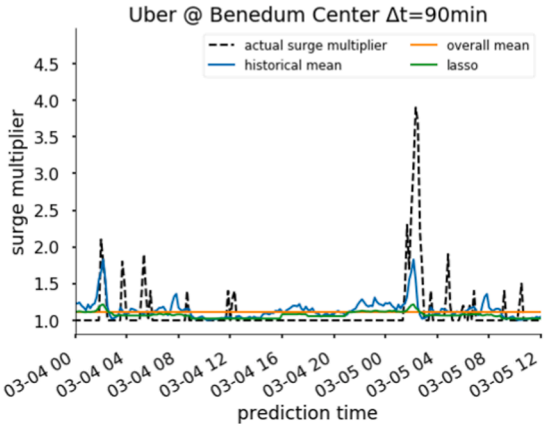
On average, L_1 regularization selects just 6 of the 8898 features. Within each fold of the cross-validation and within each time-of-



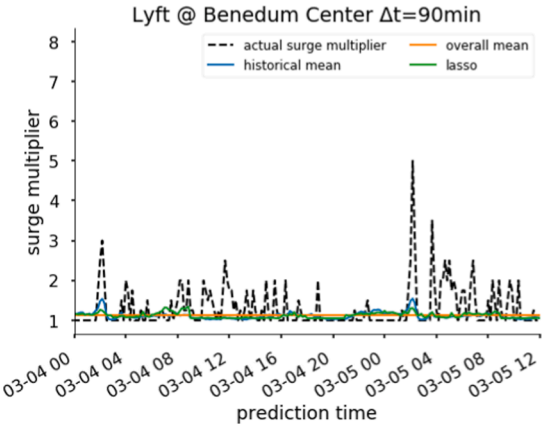
(a) Predicted Uber surge multipliers at the Benedum Center 10 minutes in advance.



(b) Predicted Lyft surge multipliers at the Benedum Center 10 minutes in advance.



(c) Predicted Uber surge multipliers at the Benedum Center 90 minutes in advance.



(d) Predicted Lyft surge multipliers at the Benedum Center 90 minutes in advance.

Fig. 9. Predicted Uber surge multipliers over three days at the Benedum Center with $\Delta t = 10$ min (a) and $\Delta t = 90$ min (c). Predicted Lyft surge multipliers over three days at the Benedum Center with $\Delta t = 10$ min (b) and $\Delta t = 90$ min (d) All models tend to over-estimate low surges and under-estimate large surges; a phenomenon which becomes more pronounced when the prediction horizon increases to 90 min.

day window however, L_1 regularization selects a different set of features. There are 7 folds of the cross validation, 6 time-of-day windows, 7 prediction horizon, and 49 locations. This means there are 14406 ($= 7 \times 6 \times 7 \times 49$) fits performed and evaluated in our experiments.

To measure the significance of each variable, we compute the p values of the variables selected by each fit. There is a small body of work dedicated to statistical inference based on the Lasso estimator (Lee et al., 2016; Lockhart et al., 2014; Loftus and Taylor, 2014). However, there is no well-established procedure for significance testing on the variables selected by Lasso. Because the data itself has been used to select the features, performing the usual tests for significance on the training data will result in anti-conservative p-values. (Loftus and Taylor, 2014) In this paper we take a conservative approach and test the significance of the features selected by Lasso with an ordinary linear model trained on the held-out data. We consider any feature with a p-value less than 0.05 to be significant. We caution that statistical significance in this setting is less meaningful than performance. Because location-specific relationships are implicitly modeled in this setting, it's less important that we identify those variables whose influence on surge multipliers is statistically significant and more important that on average our method is able to predict surge multipliers better than other methods. That said, significant variables may give us insight into potentially reliable location-specific pre-indicators of surge multipliers.

To measure overall influence of each variable, we compute *feature importance* as the average absolute weight for each feature over all folds.

In general, surge multipliers exhibit location-specific dependence on the features. As a result, a model trained in one location should not be expected to perform well in another. That each location requires a different model is an advantage of this approach because it abstracts away harder-to-measure interactions between ride-sourcing supply and demand, as well as socio-economic, demographic, and land use variables all of which contribute to the dynamics of surge multipliers. As such, we can not generalize the dependence revealed in the fitted models to other cities or even other locations within Pittsburgh. Rather, we may use these models to reveal hyper-local dynamics of the ride-sourcing market.

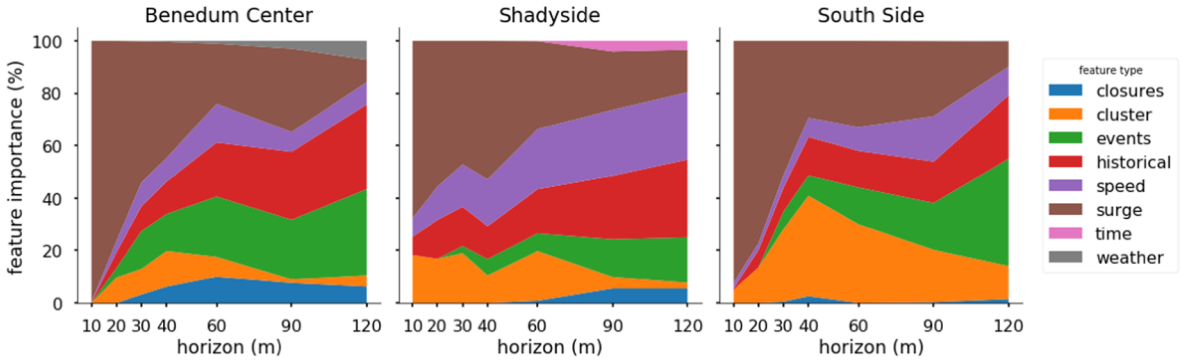


Fig. 10. Relative feature importance in the Uber model by feature type and prediction horizon. Overall, the surge multipliers was the most selected feature across all three locations. However, as the prediction horizon increases, current surge multipliers tend to become less important, and more weight tends to be placed on traffic speed, events, and historical average surge multipliers. Feature importance for each feature is computed as the mean of the absolute value of the feature weight over folds. Feature importance of a feature type is computed as the maximum of the feature importances of all features of that type.

Across locations, L_1 regularization consistently selects from the current surge multiplier variables and the weight placed on the current surge decreases with the prediction horizon. This makes sense given the strong autocorrelation observed in the surge multiplier timeseries at small lags. However, other features types are selected with different importance at different locations, re-affirming the need to fit them separately. For example, Fig. 10 shows that at the Benedum Center and South Side, event variables are consistently selected and their weight increases with prediction horizon. In contrast, the Shadyside model places less weight on events and more on historical averages. The patterns in feature type importance across locations is consistent with high-level neighborhood characteristics: the Benedum Center is an events center located in Downtown Pittsburgh in proximity to several other events centers. The South Side is home to bars, clubs, concert venues, and a play house. Shadyside has many restaurants and bars but fewer event venues.

Of the cluster distance features, the distance to the first (largest) cluster was selected with the most weight. The first cluster was the “no-surge” cluster and is helpful in distinguishing periods in which there is a lot of activity from periods of no surge. Taken together with the weight placed on historical averages, this would seem consistent with the pattern of nightlife characteristic of areas like Shadyside: somewhat regular bursts of activity unrelated to events that may be going on.

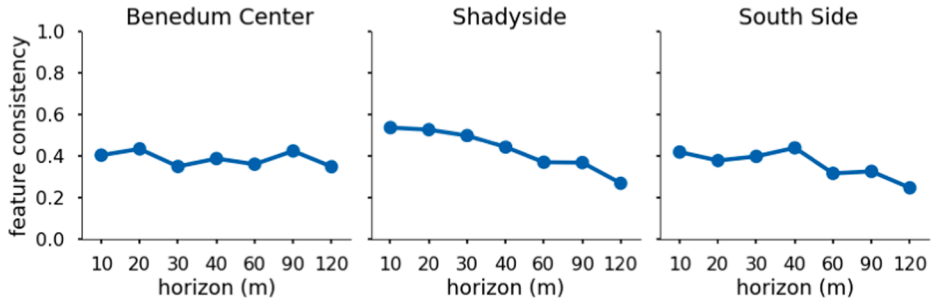
To understand how robust the feature selection is, we can look at how consistent the set of selected features is across the folds of the cross validation. First, we define consistency as the mean fraction of folds in which a feature will be selected given that it has been selected in at least one fold. A consistency score of 1 means that the same features were selected in each fold, and 0 means that no feature was selected more than once. Consistency is undefined when no features are selected in any fold. Second, we define feature drift to understand the extent to which selected features are changing over time. We use the hamming distance between two feature sets to measure how different any pair of selected features are. By computing the correlation coefficient between hamming distance and time we can understand the extent to which selected features are changing over time. Because the underlying data is temporal, the folds of the cross validation use chronologically sequential subsets of the data. As a result, we can meaningfully determine the amount of time between the data used in each fold. A high correlation coefficient means that the further apart in time two subsets of the data are, the more different the two sets of selected features will be.

Although there is no theoretical relationship between statistical significance and whether or not a variable is selected by Lasso, in our experiments, we find a statistical relationship between consistency and statistical significance. Features that are significant in one fold are slightly (but statistically significantly) more likely than non-significant features to be selected in at least one other fold. Around half of non-significant features are selected in a different fold compared to between 55% and 65% of significant features, depending on location.

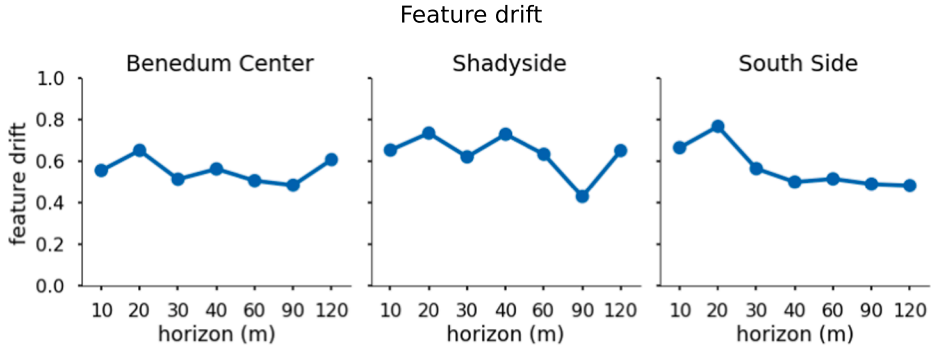
In general, two trends are clear. First, consistency is low and tends to decrease slightly with prediction horizon as in Fig. 11a. Second, the feature drift is high across all prediction horizons as shown in Fig. 11b. That feature drift is large may suggest that the importance of certain features is likely to change over long time periods (i.e. months).

The weight placed on each feature type changes slightly over folds. Event and surge variables slightly gain importance over folds whereas cluster and closure variables tend to lose weight. Within each feature type, however, feature weights shift between individual features. Within a type, this shift could be due to redundant information: for example, two tracts may have similar patterns of events with one being slightly more useful to the regression in one fold than the other.

The statistical significance of features varies by location and changes over prediction horizon. Current and historical surge multiplier features however, were consistently found to be significant across locations at short prediction horizons. The significance of current surge multipliers decreases over prediction horizon. Other features vary substantially. For example, nearly 60% of traffic speed features were found to be significant for the Shadyside location, whereas less than 10% of traffic speed features were significant at both the Benedum Center and South side. These findings support the idea that there are latent, complex interactions that vary by location. Some interactions may be idiosyncratic to a location. By fitting each location separately, these complex interactions are implicitly modeled, enabling relatively simple techniques to characterize non-trivial behavior.



(a) ***Feature consistency.*** Frequency with which features are selected in the Uber model over folds in the cross validation. The consistency tends to decrease with prediction horizon.



(b) ***Feature drift.*** Correlation coefficient between the temporal distance between two subsets of the data and the hamming distance between the features selected by models trained on those subsets.

Fig. 11. Feature consistency and feature drift at three locations.

Here we analyze the spatial distribution of dependencies for the three highlighted locations to demonstrate how location-specific models may be used to better understand the dynamics of the ride-sourcing market at a hyper-local level. In general, spatial dependencies tend to be geographically disparate, highlighting the importance of including non-local spatial data in the model.

Overall, each of the three location models place weight on events in similar, geographically disparate areas of the city. As previously noted, both the Benedum Center and South Side leverage event variables to predict surge multipliers. Interestingly while both models place weight on the tract containing South Side, the Benedum Center model does not place weight on its own tract, preferring instead geographically disparate tracts which are less likely to host events as shown in Fig. 12. These locations might particularly useful indicators precisely because they are *not* event centers. In other words, events occurring downtown are rarely a surprise: drivers may even assume there is always some event going on. This view is supported by the data. The tract containing the Benedum Center is also host to several other event centers which has the effect of there almost always being at least one ongoing event making the variable completely useless, a problem that finer spatial aggregation might help to solve. In contrast, an event in East Hills (the tract at the center right of Fig. 12) might be completely unanticipated.

Similar to events, the models for each of the three locations place weight on traffic speed in geographically disparate areas of the city, as shown in Fig. 13. Each of the models for the representative locations place importance on traffic speed on the north side. This could indicate that traffic in the north side is a useful proxy for traffic conditions more broadly. Surge multipliers in Shadyside, whose model is more sensitive to traffic speed than the other two models, depend on traffic speed in a larger, geographically disparate set of tracts.

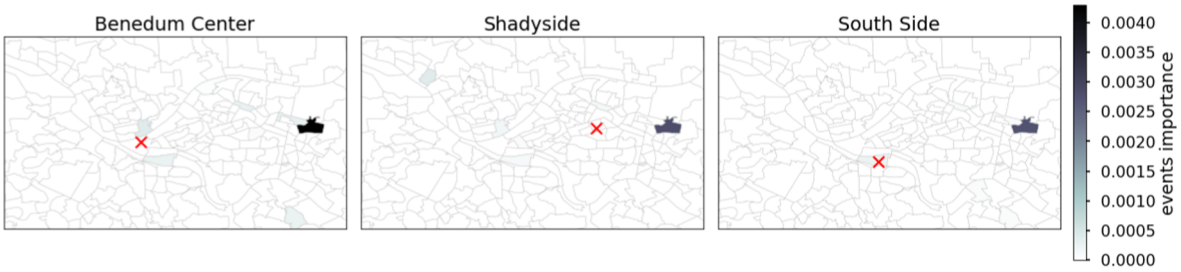


Fig. 12. Importance of event variables in each of the models for the three selected locations at a 120-min prediction horizon. The red 'x' indicates the geographic position of the modeled location. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

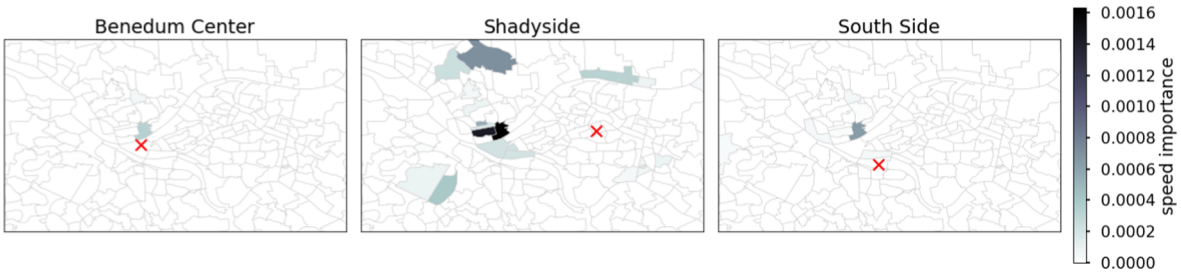


Fig. 13. Importance of speed variables in each of the models for the three selected locations at a 120-min prediction horizon. The red 'x' indicates the geographic position of the modeled location. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The dependence of surge multipliers at one location on past surge multipliers in other locations varies the most by the location being predicted. More so than the importance of speed and events, the importance of surge variables tends to concentrate geographically in areas near to the location being predicted. However models also place weight on farther-flung locations in many directions. The three selected locations exhibit markedly different spatial dependence patterns as can be seen in Fig. 14.

In short, because our proposed model abstracts harder-to-measure features impacting surge multipliers by training the model separately for each location using simple features, each fitted model does not generalize to other locations. However, the fitted parameters of each model reveal local relationships between surge multipliers at that location and real-time urban features across the city. As a result, the values of the fitted parameters, and their significance, may be used to better understand what can lead to a mismatch of supply and demand in ride-sourcing systems in a specific area of a city.

5.4. Parameter clusters

Having analyzed the selected features at three specific locations, we draw more general conclusion by exploring the structure of the fitted parameters within the parameter space. Structure in the parameter space indicates that the models are related and may be represented more succinctly than one model per location. As previously discussed, the fitted parameters differ between locations indicating location-specific relationships between features and surge multipliers. The nature of these relationships may be mediated by a variety of location variables including land use, socio-economic, and demographic features. Characterizing these relationships is no doubt interesting but outside the scope of this paper. Fitting each location separately serves to improve prediction accuracy, but this choice ignores two important facts. First, surge multipliers are correlated in space, meaning that we should expect locations that are near to one another to have similar models. Second, if surge multipliers are in fact responding to location-specific characteristics, then locations of similar character should have similar models. Taken together, this perspective illuminates a possible extension to this work in which the similarity of locations in some latent space is utilized to jointly fit all locations. In particular, a multi-task learning or mixture model approach might work well.

In light of this, we cluster the separately fitted parameters of all location models in order to examine first the structure of the fitted parameters within the parameter space and second the potential of parameter sharing in this setting. From a probabilistic perspective, we are now treating the fitted parameters of each location model as random vectors drawn from a distribution in the space of parameters. Structure in the parameter space means that the parameter vectors are drawn from a distribution conditional on a smaller number of latent factors. Agglomerative clustering was applied to the fitted parameter values for each location and each time of day (Pedregosa et al., 2011). To improve clustering performance, principal component analysis was first applied to reduce the dimensionality of the parameter space while preserving at least 99% of the variance (Pedregosa et al., 2011).

When rendered geographically, the clusters labels are intuitive and largely group locations by geography and land use patterns. In Fig. 15a the green cluster can be seen to represent suburban areas while orange represents the business, entertainment, and

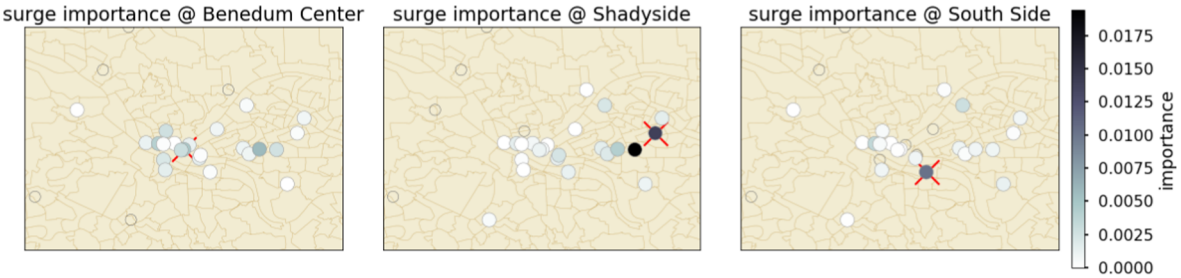
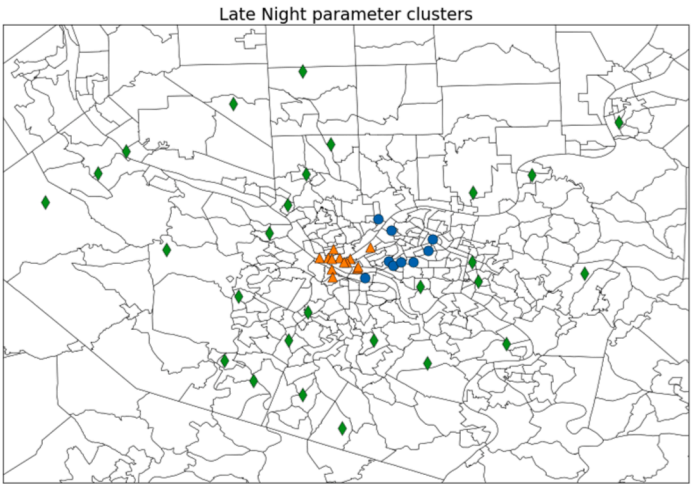
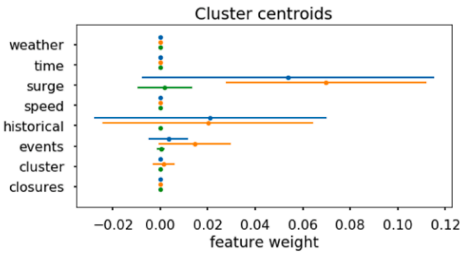


Fig. 14. Importance of surge variables in each of the models for the three selected locations at a 120 prediction horizon. The red 'x' indicates the geographic position of the modeled location. Surge variables with 0 importance are represented as hollow circles. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) Cluster labels of fitted parameters in the late night model.



(b) Total weight of each feature type in each cluster centroid. Errorbars show total standard deviation.

Fig. 15. Parameter clusters of the late night lasso models over all locations.

commercial districts while the blue cluster contains mostly dense residential areas with commercial main streets. This characterization can also be seen in the cluster centroids. In Fig. 15b, the centroid in each cluster is represented by the total feature weight within each feature type. Note that the centroid of the orange cluster places weight on events more so than the blue cluster centroid. The green cluster places almost no weight at all on events. This representation is necessary visually to avoid displaying a large number of features at the expense of some nuance. In particular that the total weights in the centroids for the blue and orange clusters are similar does not mean that the centroids are close together. In fact the centroids of the blue and orange clusters respectively place surge and historical surge weights on different geographic areas. This indicates that both clusters represent a different spatial relationship between past and future surges of similar magnitude.

Parameter clusters of the other time of day windows also largely group locations by geography and land use. However, because land use patterns also contribute to activity temporally, the clusters and their centroids differ between time of day windows. For examples there are clusters in the AM and PM peak, in contrast to the Late Night cluster centroids shown in Fig. 15b, that weight speed but not events. Cluster centroids in the early morning weight past surge activity almost exclusively as do clusters in the mid day.

The changing relationships also have geographic implications. Suburban locations tend to always be in the same cluster, but depending on the time of day, certain locations start to look like suburbs from the surge prediction perspective. In the mid day for example, nearly every location looks suburban except for the areas populated by several large universities, principally the University of Pittsburgh and Carnegie Mellon University. In contrast to the late night clusters, in the AM and PM peak times the business district separates itself into its own cluster while the commercial and entertainment districts join the suburban cluster. These observations suggest that the relationship between features of the urban environment is mediated by land use and in the temporal patterns of activity it produces.

From a probabilistic perspective, the clustering results show that the parameters of each of the linear models are drawn from a distribution conditioned on a low dimensional latent space representing a mixture of clusters. Ultimately what this means is that approximately 9,000 features spanning 49 locations can be described by to a handful of linear models each using on average 6 features. These linear models not only forecast surge multipliers across the city but also reveal what useful signals the driver network is failing to incorporate into its collective decision making process. Taken together this suggests that a mixture model approach or multi-task learning might work well here. From a multi-task perspective the separate lasso models could be trained jointly by asserting that locations that close in land-use space (or geographic space) should have similar parameters. This additional source of regularization could additionally improve generalization error. Similarly, from a mixture model approach, we consider the parameters to be drawn from a distribution conditioned on the location's cluster membership probabilities in the latent space association with locations.

6. Conclusion

Data describing the current state of the several urban systems are collected from web APIs and used as features of a log-linear model to predict surge multipliers in Pittsburgh. L_1 regularization is used to allow a small number of important features to be selected in a data-driven way from the large number of spatio-temporal features describing the urban state. To allow the linear model to describe non-linear behavior, temporal segmentation and clustering are employed. Days are segmented into time-of-day windows and separate models are trained for each. Clustering extracts temporal modes from the data and distance to the centroids of these clusters are included as features in the model.

Overall, log-linear regression with L_1 regularization is able to predict Uber surge multipliers up to 2 h in advance with greater precision than naïve and non-linear methods using only on average 6, and at most 25, of the 8898 measurements from the current and recent past urban environment. In each of the time windows the model is able to outperform naïve methods when predicting surge multipliers up to 120 min in advance. The fact that it is a linear model, of course, lends straight-forward interpretations to the fitted parameters. In addition to offering greater insight into surge multipliers, this model, without any modification could be deployed to produce real-time predictions of Uber surge multipliers. Each of the features used in this model is available in real-time from various web APIs.

A linear model with L_1 regularization, commonly known as "LASSO" is indeed a classical methodology, but one that, in combination with feature engineering and unsupervised learning, out-performs more complex methods in many applications. In this case, LASSO was the parsimonious choice. Moreover, more complex learning methods lend themselves less easily to analysis of their features than LASSO does. Taken together, LASSO enabled us to achieve better and more interpretable results than the more complex methods we applied.

We find that the model performs best to predict Uber surge multipliers in urban areas. For both Uber and Lyft the model tends to under-perform naïve methods for surge multipliers between 1.1 and 1.5 and out-perform naïve methods for larger values. This effect is particularly pronounced in the Lyft model, whose poor performance on low surge multipliers substantially degrades its overall performance. Lyft's coarser surge discretization is at least partially to blame, but other operational differences between the two organizations likely cause most of the discrepancy.

The model consistently places weight on the variables representing the current and recent surge multipliers; this makes sense given the strong auto-correlation observed in the surge multipliers. Outside of surge multiplier variables, feature selection varies by location consistent with high-level neighborhood characteristics. The spatial distribution of feature weights also varies by location but tends to be geographically disparate, highlighting the importance of including non-local spatial data in the model. The model sometimes selects seemingly counter-intuitive features, for example, the traffic speed in a distant location, or the number of events in a relatively un-eventful tract. However, since surge multipliers are a proxy for unanticipated demand, the occurrence of event that is not widely known might be more likely to influence a surge multiplier than a well-known event.

If this model were employed as a decision tool, awareness of the predictions would affect driver and rider behavior, which would affect observed surge multipliers. However, real-time prediction is critical to ride-sourcing system operation and management. Many of the features used in this paper do not meet the stationarity assumptions required by linear regression. It is for this reason that we train the model on a fixed period (approximately 5 months) and evaluate the fitted model on the subsequent week. Behavioral changes similarly violate stationarity assumptions by changing the relationship between the features and surge multipliers over time. Using a fixed trailing training period solves this problem as well. In short, if such a model were employed in a real-time application, current data would be incorporated and past data dropped on a weekly basis, slowly capturing changing behavior over time. We expect the model would remain effective even under changing surge multiplier dynamics caused by the dissemination of its predictions.

There are three broad classes of extensions to this model. First, a multi-task or mixture model approach could be applied to this modeling framework to both improve general insights into the behavior of surge multipliers at a city scale and to improve the generalization error through additional regularization. Second, the feature set could be extended to include more real-time measurements of the urban environment, including public transit data from GTFS, the wait time from Uber or Lyft's API, or more specific attributes of location, for example the number of open businesses. Third, the same feature set used in this model might be used to predict other characteristics of urban mobility, such as public transit availability.

Acknowledgments

This research is funded in part by National Science Foundation Award CMMI-1751448 and Pennsylvania Department of Community and Economic Development (DCED). The authors would like to thank Gridwise Inc. for providing consultative resources for this research. The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein.

References

Alonso-Mora, J., Samaranayake, S., Wallar, A., Frazzoli, E., Rus, D., 2017. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. Proc. Nat. Acad. Sci. 114 (3), 462–467.

Baraniuk, R.G., 2007. Compressive sensing [lecture notes]. IEEE Signal Process. Magaz. 24 (4), 118–121.

Bimpikis, K., Candogan, O., Saban, D., 2019. Spatial pricing in ride-sharing networks. Oper. Res.

Cain, A., Burris, M., Pendyala, R., 2001. Impact of variable pricing on temporal distribution of travel demand. Transport. Res. Rec.: J. Transport. Res. Board 1747 (01), 36–43.

Chen, L., Mislove, A., Wilson, C., 2015. Peeking Beneath the Hood of Uber. In: Proceedings of the 2015 Internet Measurement Conference, IMC '15. ACM, New York, NY, USA, pp. 495–508.

Chen, M.K., Sheldon, M., 2016. Dynamic Pricing in a Labor Market: Surge Pricing and Flexible Work on the Uber Platform. In: Proceedings of the 2016 Acm Conference on Economics and Computation, pp. 1–19.

Cohen, P., Hahn, R., Hall, J., Levitt, S., Metcalfe, R., 2016. Using Big Data to Estimate Consumer Surplus: The Case of Uber. Technical report. National Bureau of Economic Research.

Fei, X., Lu, C.-C., Liu, K., 2011. A bayesian dynamic linear model approach for real-time short-term freeway travel time prediction. Transp. Res. Part C 19 (6), 1306–1318.

Guda, H., Subramanian, U., 2017. 'Your Uber is Arriving: Managing On-Demand Workers through Surge Pricing, Forecast Communication and Worker Incentives',

Management Science.
Guha, S., Demirezen, E.M., Kumar, S., 2018. 'Dynamics of competition in on-demand economy: A differential games approach', Available at SSRN 3263152.
Gurley, B., 2014. 'A Deeper Look at Uber's Dynamic Pricing Model', UBER Newsroom. <https://www.uber.com/newsroom/guest-post-a-deeper-look-at-ubers-dynamic-pricing-model/>.
Hall, J., Kendrick, C., Nosko, C., 2015. 'The Effects of Uber's Surge Pricing: A Case Study', Uber Under The Hood, pp. 1–8.
He, F., Shen, Z.J.M., 2015. Modeling taxi services with smartphone-based e-hailing applications. Transport. Res. Part C: Emerg. Technol. 58, 93–106.
Ke, J., Zheng, H., Yang, H., Chen, X.M., 2017. Short-term forecasting of passenger demand under on-demand ride services: a spatio-temporal deep learning approach. Transport. Res. Part C: Emerg. Technol. 85 (October), 591–608.
Lapowsky, I., 2015. 'Uber Wins Its Battle Against NYC's Mayor—For Now'. URL: <https://www.wired.com/2015/07/uber-wins-battle-nyc-mayor-now/>.
Lee, J.D., Sun, D.L., Sun, Y., Taylor, J.E., et al., 2016. Exact post-selection inference, with application to the lasso. Annals Stat. 44 (3), 907–927.
Lee, M.K., Kusbit, D., Metsky, E., Dabbish, L., 2015. Working with machines: the impact of algorithmic and data-driven management on human workers. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI '15. ACM, New York, NY, USA, pp. 1603–1612.
Li, Y., Wang, X., Sun, S., Ma, X., Lu, G., 2017. Forecasting short-term subway passenger flow under special events scenarios using multiscale radial basis function networks. Transport. Res. Part C: Emerg. Technol. 77, 306–328.
Liu, J., Han, K., Chen, X.M., Ong, G.P., 2019. Spatial-temporal inference of urban traffic emissions based on taxi trajectories and multi-source urban data. Transport. Res. Part C: Emerg. Technol. 106, 145–165.
Lockhart, R., Taylor, J., Tibshirani, R.J., Tibshirani, R., 2014. A significance test for the lasso. Annals Stat. 42 (2), 413.
Loftus, J.R., Taylor, J.E., 2014. 'A significance test for forward stepwise model selection', arXiv preprint arXiv:1405.3920.
Min, W., Wynter, L., 2011. Real-time road traffic prediction with spatio-temporal correlations. Transp. Res. Part C 19 (4), 606–616.
Hamed, Mohammad M., Al-Masaeid, H.R., Said, A.M.B., 1995. Short-term prediction of traffic volume in urban arterials. J. Transport. Eng. 121 (3), 249–254.
Noursalehi, P., Koutsopoulos, H.N., Zhao, J., 2018. Real time transit demand prediction capturing station interactions and impact of special events. Transport. Res. Part C: Emerg. Technol. 97, 277–300.
Park, M.Y., Hastie, T., 2007. L1-regularization path algorithm for generalized linear models. J. Roy. Stat. Soc.: Ser. B (Stat. Methodol.) 69 (4), 659–677.
Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: machine learning in Python. J. Mach. Learn. Res. 12, 2825–2830.
Phillips, B., 2017. Balancing supply and demand in a two-sided marketplace. Plenary Talk.
Rosenblat, A., Stark, L., 2016. Algorithmic labor and information asymmetries: a case study of Uber's drivers. Int. J. Commun. 10, 27.
San Francisco County Transportation Authority, 2017. 'TNCs Today: A Profile of San Francisco Transportation Network Company Activity'.
Satariano, A., 2017. 'Uber Losing Battle in London After Regulator Revokes License'. URL: <https://www.bloomberg.com/news/articles/2017-09-22/london-authority-revokes-uber-s-private-hire-license>.
Sun, H., Wang, H., Wan, Z., 2019. Model and analysis of labor supply for ride-sharing platforms in the presence of sample self-selection and endogeneity. Transport. Res. Part B: Methodol. 125, 76–93.
Uber Technologies, 2018. 'Riding with Uber: Upfront Pricing'. URL: <https://www.uber.com/ride/how-uber-works/upfront-pricing/>.
Vanajakshi, L., Rilett, L.R., 2004. A Comparison Of The Performance Of Artificial. Neural Networks And Support Vector Machines For The Prediction Of Traffic Speed. In: IEEE Intelligent Vehicles Symposium, Parma, Italy, pp. 194–199.
Wang, X., He, F., Yang, H., Oliver Gao, H., 2016. Pricing strategies for a taxi-hailing platform. Transport. Res. Part E: Logist. Transport. Rev. 93, 212–231.
Wei, Y., Chen, M.C., 2012. Forecasting the short-term metro passenger flow with empirical mode decomposition and neural networks. Transport. Res. Part C: Emerg. Technol. 21 (1), 148–162.
Yang, H., Yang, T., 2011. Equilibrium properties of taxi markets with search frictions. Transport. Res. Part B: Methodol. 45 (4), 696–713.
Yang, S., Ma, W., Pi, X., Qian, S., 2019. A deep learning approach to real-time parking occupancy prediction in spatio-temporal networks incorporating multiple spatio-temporal data sources. Transport. Res. Part C: Emerg. Technol. 107, 248–265.
Yang, S., Qian, S., 2019. Understanding and predicting travel time with spatio-temporal features of network traffic flow, weather and incidents. IEEE Intell. Transp. Syst. Mag. 11 (3), 12–28.
Zha, L., Yin, Y., Xu, Z., 2018. Geometric matching and spatial pricing in ride-sourcing markets. Transport. Res. Part C: Emerg. Technol. 92, 58–75.
Zha, L., Yin, Y., Yang, H., 2016. Economic analysis of ride-sourcing markets. Transport. Res. Part C: Emerg. Technol. 71, 249–266.
Zhang, P., Qian, Z.S., 2018. User-centric interdependent urban systems: using time-of-day electricity usage data to predict morning roadway congestion. Transport. Res. Part C: Emerg. Technol. 92, 392–411.
Zheng, W., Lee, D.-H., Asce, M., Shi, Q., 2006. Short-term freeway traffic flow prediction: Bayesian combined neural network approach. J. Transport. Eng. 132 (February), 114–121.
Zheng, Z., Wang, D., Pei, J., Yuan, Y., Fan, C., Xiao, F., 2016. In: the 25th ACM International on Conference on Information and Knowledge Management - CIKM, pp. 1363–1372.