Uber Load Rebalancing for Predictable Demand Spikes

Brian Allen, Pit Kauffmann, Sihyun Lee, Keni Mou, Suikai Lin

May 1, 2018

1 Introduction

High fluctuations in rider demand present a unique problem for Uber. In these situations, riders that log onto the app are often met with long wait times, since the supply of cars in the proximate area are matched very quickly with the first riders to request an Uber. These long wait times discourage riders from using the service, who then opt for alternative modes of transportation. In order to resolve the discrepancy between supply and demand, Uber employs a surge pricing policy to match Uber drivers with riders that value the ride the most. This policy incentivizes Uber drivers to react to high fluctuations in demand, which eventually reduces wait times for potential new riders.

The main reason Uber hasn't developed a better policy for addressing demand fluctuations is because they often result from unpredictable events, such as a sudden rainstorm. However, demand increases can also occur from predictable events, such as flight arrivals at the airport or ending of a sporting event at a stadium. In these instances, Uber continues to operate under the status quo, which is evidenced by its use of surge pricing after demand fluctuating, predictable events. With prediction of demand fluctuations with some degree of accuracy, Uber could better manage its supply by preemptively moving drivers to areas that they anticipate will see a spike in demand. This can improve wait times for potential new drivers, thus increasing the number of riders.

This study aims to optimize Uber load management in a spatial ride-share model, where demand fluctuations can be predicted with a certain degree of accuracy. The event used in the study are concerts at Madison Square Garden during the summer months of 2014. Applying the improved load management policy, the study concludes that Uber can pick up additional riders and increase profits by approximately 7% to 15% given a 1 hour demand spike of around 50 customers.

2 Model Approach

Uber rider decision making can be modeled according to a stochastic choice model. Given a series of factors, the rider may decide between several transportation options, including Uber, taxis, public transportation or walking. In this study, we assume the only factors that influence a riders decision for choosing Uber over its alternative are price and wait times.

Further, we consider two spatial areas. The first area is defined as the epicenter where the demand spike is fully enclosed, defined as location 1. The second area is defined as the immediate area surrounding the epicenter, defined as location 2. Also, assume a location 3 that is everything within 15 minutes outside of location 2. In this area, there is an infinite supply of drivers that can service requests within these two areas, but at high cost.

Assume demand can be predicted with some degree of accuracy. Drivers can be relocated between areas with some cost in order to preempt demand spikes and service customers that value Uber with low wait times. Now, people that open the app to take an Uber ride are met with lower wait times on average and therefore more customers are picked up.

Each parameter in the model is based on Uber and NYC Taxi data. The methodology behind calculating each parameter and the actual values of parameter that were used in the model can be found in Appendix A. The context with which we estimated the parameters is the demand

spike occurring after a concert at Madison Square Garden from 10-11pm. The conclusions from each model should be read with this context and it should be recognized that with lower or higher demand spikes, the expected revenue increases may vary.

3 Model 1

In this model, we assume that demand can be predicted without error. Given a set of parameters, we develop a linear program that relocates drivers at some cost in order to pick up more customers and achieve higher revenues. The objective function is therefore the revenue of each pickup less the cost of relocating vehicles.

3.1 Optimization Variables

- m_{ij} := number of cars relocated from location i to location j
- y_{ij} := number of people who open app in location i matched with car from location j

3.2 Model Parameters

- D_j := true demand for Ubers from location j
- x_{ij} := probability customers in location i will select Uber given it is in location j
- N_{0j} := number of cars in location j prior to relocation
- $N_{1j} = N_{0j} + \sum_i m_{ij} - \sum_i m_{ji}$:= number of cars in location j after relocation
- c_{ij} := cost of relocating Uber from location i to location j
- r := revenue from picking up a customer

3.3 Linear Program

$$\max \sum_i \sum_j y_{ij} x_{ij} r - m_{ij} c_{ij}$$

$$\text{such that } N_{1j} = N_{0j} + \sum_i m_{ij} - \sum_i m_{ji} \quad \forall j$$

$$N_{0j} \geq \sum_i m_{ji} \quad \forall j$$

$$y_{ij} \geq 0 \quad \forall i, j$$

$$\sum_j y_{ij} \leq D_j \quad \forall i$$

$$\sum_i y_{ij} x_{ij} \leq N_{1j} \quad \forall j$$

3.4 Results

With this set of parameters, the model allowing for relocation compared to the model allowing no relocation $m_{ij} = 0$ results in an increase in total profit of 13.7%. In general, the higher demand increases, the more total revenue increases. This result is robust to changes in demand and revenue, which can be seen in Model 2.

4 Model 2

The profitability of this policy depends on the relationship between the cost of relocating a car and the expected revenue increase from picking up an incremental customer. Therefore, the model is highly dependent on the revenue and cost parameters within the objective function. In order to test the sensitivity of these parameters, model 2 incorporates a weight parameter $\alpha \in [0, 1]$, where the weight on revenue is α and the weight on cost is $(1 - \alpha)$. A value of $\alpha = 0$ implies Uber only cares about reducing cost, which can be interpreted as Uber not wanting to make drivers relocate, giving them full autonomy on how they operate. A value of $\alpha = 1$ implies Uber only cares about maximizing revenue, which means Uber wants to pick up as many customers as possible at any cost to the driver. A value of $\alpha = .5$ reduces back to model 1, where Uber simply maximizes total profit according to the parameters.

4.1 Linear Program

$$\begin{aligned}
\max & \sum_i \sum_j \alpha \cdot y_{ij} x_{ij} r - (1-\alpha) \cdot m_{ij} c_{ij} \\
\text{such that } & N_{1j} = N_{0j} + \sum_i m_{ij} - \sum_i m_{ji} \quad \forall j \\
& N_{0j} \geq \sum_i m_{ji} \quad \forall j \\
& y_{ij} \geq 0 \quad \forall i, j \\
& \sum_j y_{ij} \leq D_j \quad \forall i \\
& \sum_i y_{ij} x_{ij} \leq N_{1j} \quad \forall j
\end{aligned}$$

4.2 Results

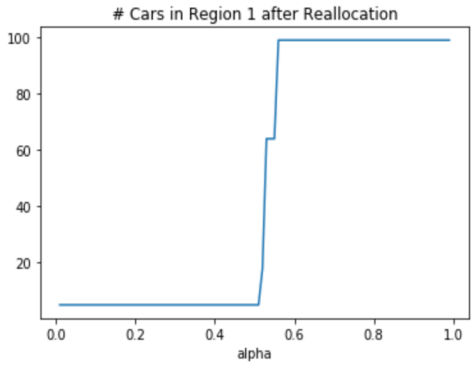

Figure 1: Plot of N_{1j} against α

This model increments from $\alpha = 0$ to $\alpha = 1$ with a step-size of 0.1 and solves for the optimal relocation. The chart above plots the post-relocation number of cars to location 1 with a true demand of 100 customers, for different values of α. As expected, when $\alpha = 0$, Uber does not want to incur any relocation costs and so maintains the normal operation. When $\alpha = 1$, Uber wants to pick up as many customers as possible, so floods the area with cars to meet the demand in location 1.

5 Model 3

In the third model, we approach the load rebalancing problem using simulation. In locations 1 and 2, we assume customers arrive according to a Poisson process. That is, the interarrival times are distributed according to $Expo(\frac{1}{D_i})$ for location i. Initially there are N_i cars sitting in location i. When there is an arrival in location i, vehicles are matched to that customer under some defined priority as long as there are still cars available in the prioritized location. Arrivals in location 1 are prioritized by cars in location 1, 2 then 3. Arrivals in location 2 are prioritized by cars in locations 2, then 3. Cars in location 1 are not matched to arrivals in location 2. When a customer is matched to a car, the rider in location i accepts the ride from location j with probability x_{ij}. If the customer does not match with the car, the car is still available for picking up another customer. If the customer does accept the car, the customer and car leave the system.

5.1 Results

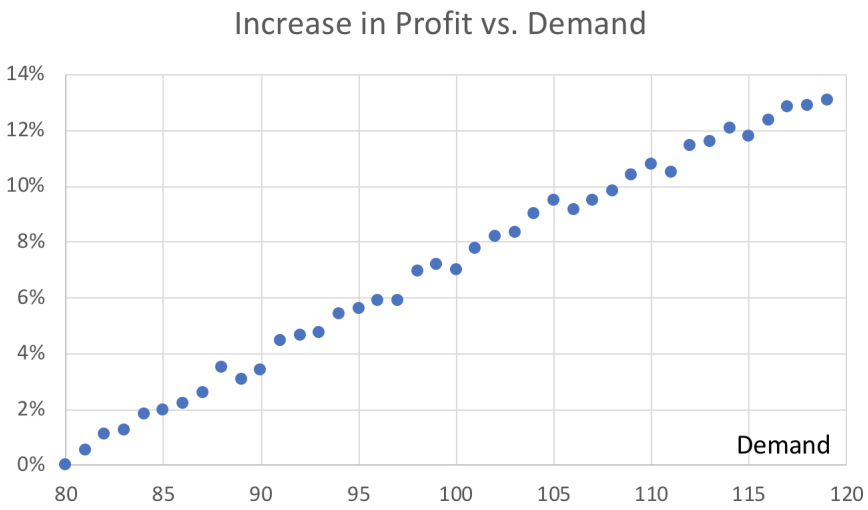

Figure 2: Plot of profitability against $\hat{D_1}$

The profits were computed against the baseline profit calculated in Model 1. We can see from the simulation that, for bigger values of the demand spike, we are gaining more profit. It would be unrealistic if our model could be used for any value of demand, and as we can see from the graph our model can be profitable only when there is a demand spike big enough. Under the current set of parameters, a demand spike of anything less 80 customers would be unprofitable for Uber to address with this relocation policy.

6 Model 4

Of course, predictions are not made with 100% accuracy. The fourth model addresses model prediction error and specifically at what error rates the policy is no longer profitable. Now, assume as before that there is a true demand D_i of customers in location i. However, Uber predicts there to be $\hat{D}_i$ demand in location i. Since D_i is unknown, Uber will operate under the assumption that $\hat{D}_i$ is the true and relocate the cars accordingly, incurring the relocation costs. The fourth model runs a simulation where the cars are located optimally according to satisfy $\hat{D}_i = 100$, but the customers arrive according to $Expo(\frac{1}{D_i})$. In this model, we add an additional cost of excess cars in the system to avoid sending overly many cars. Uber will incur a cost of \$25 per excess car, which is the cost to Uber for an hour's worth of work.

6.1 Results

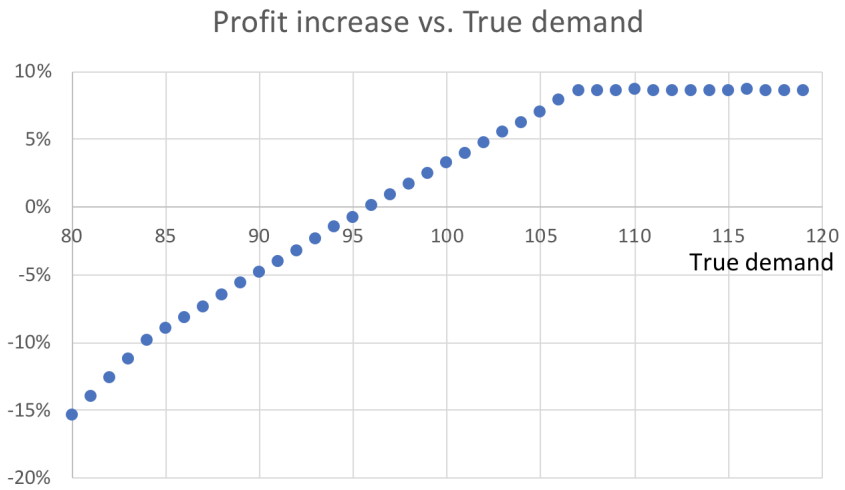

Figure 3: Plot of profitability against $\hat{D}_1$

In this model, we try out values of D_i from 80 to 120 while $\hat{D}_i = 100$ and plot percent profit increase relative to the baseline profit with no reallocation. These results can be seen in the chart above, where the profit was computed against the baseline profit computed in Model 1. As we can see from the graph, if the true demand is below 96, we are better off not changing our strategy. That is, as long as Uber doesn't over-predict by 4% the true demand, Uber can expect profit improvements by implementing this policy.

This assumes that Uber makes the decision statically; however, Uber can always change its decision dynamically by not sending the cars all at once to improve. Therefore, 4% is a very conservative lower bound for the tolerance of our estimation; in reality, we can adjust based on our observations and better account for overestimations.

7 Conclusion

This paper offers a new policy that would allow Uber to handle predictable demand spikes, such as the outflow of people coming from concerts and sporting events. This policy is a load management strategy where Uber cars preemptively locate themselves in areas with expected demand increases in order to service more people. According to the model, by applying the improved load management policy, Uber can pick up additional riders and increase profits by approximately 6%. In order to actually assist Uber in implementing this policy, part of that profit increase could be re-distributed to drivers as part of an incentive structure to motivate drivers to relocate in advance of demand spkies.

There are several additional studies our team considered and should be explored to ensure robustness of the results. This includes finding the optimal surge price since there is more balance between supply and demand, identifying the minimum number of additional customers in the demand spike to make the policy profitable and developing an incentives program that make drivers want to conform to the new policy.

Outside of revenue increases, there are additional benefits from incorporating this policy: Uber drivers can react better to demand fluctuations, customers are not turned away by high surge pricing, and overall more customers ride with Uber, which helps improve loyalty. So from a profit perspective and a larger economic benefit perspective, Uber can stand to gain from this proposed new policy.

8 Appendix A

8.1 Model Parameters

8.2 Location

The data used for estimating our model parameters came form Uber as well as NYC Yellow Cab. Since our project focuses on predictable, demand spiking events, such as concerts, we used a particular concert as the reference: Lady Gaga at Madison Square Garden on May 13th 2014, a Tuesday night. We took the time window of 10PM - 11PM (i.e. the the time when we expect the concert to end and the Uber demand to spike) as a reference window. Our effect date was hence May 13th, 2014 from 10PM - 11PM, and our control group was comprised of the same times for the remaining Tuesdays of May 2014; May 6th, May 20th and May 27th.

Using geohashing, we split the area around Madison Square Garden into 2 layered zones that were the focus of our analysis. For the purposes of this project, we assumed that the area outside zone 2 that is within a 15 minute driving radius of area 2 to be area 3. Our reasoning was that a customer would never get paired with an Uber that is more than 15 minutes away. This "outside-the-system" area was solely defined for the purpose of serving any remaining Uber requests (i.e. from a queuing perspective, we assumed that an infinite queue is possible and that all customers will be served)

8.2.1 Actual supply of Ubers at Location j: N_{0j}

We estimated this value by a simple average of the count of the number of trips requested during the relevant time window during the control days. Doing so, we obtained the following numbers:

Location	Uber supply
Area 1 (MSG area)	5.33
Area 2	43.67
Area 3	∞ (i.e. large enough to fulfill leftover requests)

8.2.2 True Demand for Ubers at Location j: D_j

Let D_j be the true demand for Uber rides during the demand spike given surge pricing and normal wait times. We reasoned that there is a true underlying demand for Uber, yet some customers are turned away by long wait times or high surge pricing, thereby constituting a foregone revenue to Uber. For example, during the relevant time period on May 13th, we observed 62 Uber pickups, yet the actual demand was likely higher. Hence, for model 1, we assumed $D_1 = 100$ and $D_2 = 20$. For simulations, we let the true demand fluctuate to see how robust our model would be to changes in true demand.

8.2.3 Probability Rider Chooses Uber by Location: x_{ij}

Let x_{ij} be the probability according to the choice model that a rider in location i will choose an Uber given the car needs to travel from location j. In our choice model, we consider 2 options, Uber and subway, and two parameters, wait time (W) and cost (P). Given the low price of a subway ticket, we assumed no sensitivity to it, i.e. $\beta_{price} = 0$. For Uber, we modeled an additional sensitivity to surge pricing multiples (S), as shown below:

$$U_{uber} = 14 - 1.2 * W_{uber} - 9 * (S_{uber} - 1)$$

$$U_{subway} = 6 - 0.9 * W_{subway}$$

Our assumption is that customers value Uber about 2.5 times over taking the subway. Note that here, a possible input for S_{uber} would be 1.5x. Our estimates for the choice model parameters were as follows:

Parameter	Estimate	Source
W_{uber}	6.86 mins from area 2 to area 1	Yellow Cab trip durations*
	10.78 mins from area 3 to area 1	Yellow Cab trip durations*
	7.89 mins from area 3 to area 2	Yellow Cab trip durations*
W_{subway}	7 mins	Personal Experience
S_{uber}	1.5x	Observation during Billy Joel concert

**Note: In order to estimate these quantities, we removed all trips with a travel time > 15 minutes, given that we assumed no customers will be matched with drivers that are more than 15 minutes away.*

8.2.4 Cost of Relocating Uber Cars: c_{ij}

Let c_{ij} be the cost of moving a car from location i to location j. The cost assumes an expected revenue per minute of an Uber driver, which is lost due to relocating, multiplied by the expected travel time in minutes to get from location i to location j, multiplied by P(driver is idle), since only then will a re-allocation occur, multiplied by a multiplication factor (M_3) for moving cars from location 3 into the system. Hence, c_{ij} can be decomposed as follows:

$$c_{ij} = E[Rev/minute] * E[Traveltime] * P(Driverisidle) * M_3$$

As mentioned above, the expected travel times between the areas were estimated using Yellow Cab trip durations. Expected revenue was estimated based on research on New York Uber drivers to be about \$0.86/minute (the average revenue for an uber driver is \$1554 per week, assuming a 30 hour work week). The idle probability was assigned at 70% for locations 1 and 2, and 80% for location 3 (due to the much larger size of location 3 as well as the time of night) given Uber driver reports. M_3 was assigned to be 1.1x for all relocations from location 3 and 0 for all other relocations.

8.2.5 Expected Revenue: r

Let r be the expected revenue from picking up a car in this location. This was calculated by applying our estimated surge premium of 1.5x to the average price of a Yellow Cab trip out of area 1. Note that we subtracted out driver tips, since Uber customers generally do not tip, as well as the base fare of \$3 that NYC cabs apply to each ride. In our case, the expected revenue per ride came out to be $9.03 * 1.5 = \$13.54$.

8.2.6 Relocation of Cars: m_{ij}

Let the decision variable m_{ij} be the number of cars that should be moved from location i to location j in order to maximize revenue. We can then define $N_{1j} = N_{0j} + \sum_i m_{ij} - \sum_i m_{ji}$ as the number of cars in location j after the relocation.