Louisville Scooters: Morning Setup

Eric Chang, Zhuangzhuang Jia, HaoDi Liu, Madeline Temares, Sijie Wang

May 2019

1 Introduction

Electric scooters are rising as a means of transportation throughout many cities throughout the country and world. They aim to help with the problems of pollution, congestion, and gridlock caused by cars, trucks, buses, and other vehicles that are on the road. In addition, they provide a level of convenience with their current dockless systems. Riders have the ability to find a scooter in their proximity and leave it exactly where they want to go, without searching for a nearby station with available docks. Electric scooters from the companies of Bird and Lime were released for operation in the city of Louisville in August of 2018. The data for the rides of the two companies combined is available for public access from the city. Throughout our analysis, we treated them as one company. However, separate data would be key to each company's true measures of success. We assumed that each work in similar enough manners that they can be grouped together for initial analysis.

These companies rely on customer rides to generate profit. Both companies charge a fixed rate to unlock a vehicle and then charge per minute as you ride. It is very important for these companies to generate as many rides as possible. In addition, scooters have a high fixed cost so it is important that they are being put to use and are not brought to an area that they will not be moved from. We want to investigate this problem and provide a good initial morning setup so that these companies are generating as many rides as they can and avoiding as much lost demand as possible. We decided to simulate the movement of scooters and the demand captured and lost given an initial morning setup.

2 Dataset and Current Problems

In this dataset, we have attributes such as a Trip ID, the start and end date and time (rounded to the nearest fifteen minutes), the trip duration in minutes, the drip distance in miles, the start and end latitude and longitude (rounded to three decimal places), and the day of week and hour. There are over 134,000 rides that we are able to analyze. Few outliers exist with errant end latitudes/longitudes far out of the range of the city of Louisville. We removed these outliers from our analysis. To account for this, we looked at a rectangular area of Louisville that would capture the middle 98% of latitude and longitudes. A visual of the rides can be seen below (Figure 1 : Visualization of Rides). The density of the rides closely follow the more populous areas in Louisville.

The current operations are such that the scooters move naturally throughout the day, with users picking up and dropping off scooters at their convenience. At night, Bird and Lime hire Chargers and Juicers, respectively, to pick up the scooters, charge them in their homes, and release them the next day. A problem that we see is possible is that scooters can be released in the morning in a setup that is not suitable to the demand pattern. For example, it is possible that ten scooters are dropped off in an area with very little demand, evidenced by few rides starting there. In this case, scooters would go unused in these areas. On the other hand, it is possible that two scooters are dropped off in an area with very high demand. Here, demand would be lost because people would be looking for scooters and unable to find them. We simulate the movement of the scooters throughout the day and provide measures of demand captured and lost, as well as provide a reasonable morning setup.

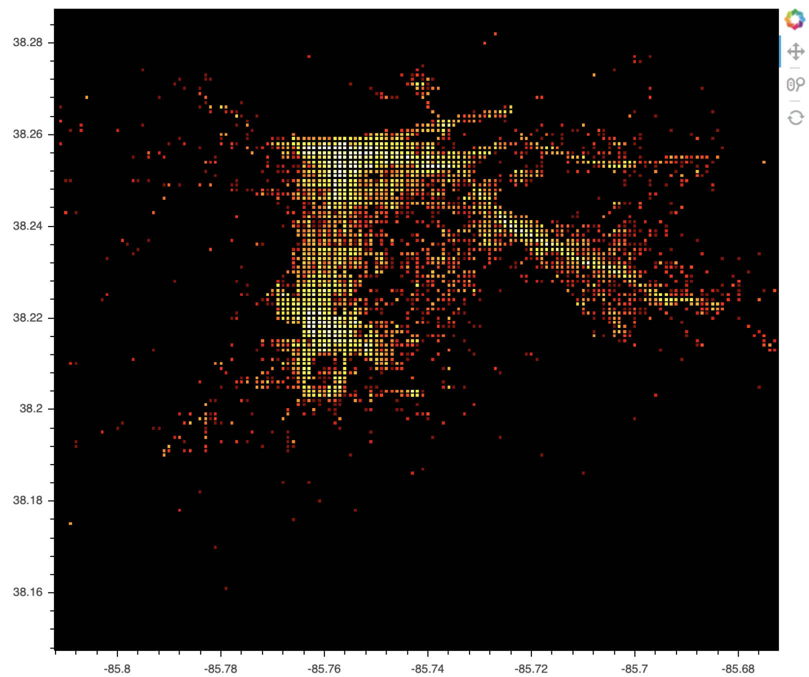

Figure 1: Visualization of Rides

3 Our work

3.1 Split the area and get the transition matrix

We first broke the city of Louisville into sixteen zones, split by latitude and longitude. Each zone was its own rectangle. We assigned every ride a start and end area, a number between one and sixteen. Our zones measured 1.29 miles by 1.5 miles which we think was a good size of a zone because the average trip distance was 1.17 miles. As this is a dockless system and riders can essentially leave scooters wherever they want, dividing into zones was essential for data analysis. We then studied the flow of rides in and out of these sixteen different zones per hour. Using this, we computed 24 transition matrices to see the typical flow per hour. Here, we used a mathematical model in which we assumed stationary probabilities for each hour. This was a key factor in our simulation, as we could then simulate movement of used scooters according to these transition probabilities.

3.2 Simulation

Using both these transition matrices and the average demand per zone per hour, we simulated the movement of the scooters throughout a day based on a given initial setup. In this simulation, we aim to count the captured and lost demand. The naive approach would be to assign every morning an equal number of scooters to each zone. We compared this with our approach which would be to assign every morning each zone a fraction of the total number of scooters proportional to the fraction of total rides leaving those zones. A third comparison we made was with a setup that had the scooters assigned proportional to the demand of a typical morning. We simulated the situation with these three starting setups and were able to calculate the captured demand in each situation.

```
SIMULATION ALGORITHM(morning setup, lost demands)
for each hour
    for each zone
        simulate demand (~ Poisson(lambda) + lost demand factor)
        if # scooters in zone ≥ simulated demand
            increment demand captured
        else
            check flow in of scooters
            increment demand captured
            increment demand lost
        if scooters in zone
            scooters moving = min(simulated demand, scooters in zone)
            for each scooters moving
                simulate random number
                move according to transition matrix probability
```

A key factor in our simulation is the number of scooters in the system. In each simulation, we assumed 250 total scooters in the city. This was based on research. For the setup in which each zone has an equal number of scooters, this number dictated 16 scooters in each zone (with the exception of 14 in zones with very little demand to bring our sum to 250).

3.2.1 Simulation Algorithm

Our code for the simulation of the movement of the scooters and the demand captured and lost can be found in the appendix. The approach we took was to go one hour at a time. Within each hour, we looked at each zone. In each zone, we simulated a poisson random variable to represent the simulated demand in that hour in that zone. The parameter for this poisson random variable was the average daily flow out of that zone for that hour. If there were sufficient number of scooters for that demand, we knew we could capture all of that demand in the zone. If there were fewer scooters than this simulated demand, we saw if the number of scooters flowing into the zone in that hour would allow us to capture the demand. Our reasoning for this was that within an hour, scooters will flow into a zone and can then be used later in the same hour. The demand captured for that hour would then be whichever was smaller: the demand calculated or the sum of the number of scooters there plus the flow of scooters in. If there was still excess demand unsatisfied even with accounting for the additional flow of scooters in, this is what we deemed would be lost demand.

After we deemed what amount of demand would be satisfied versus lost for the coming hour, we moved the scooters according to the probability matrix for that hour in that zone. Again, this probability matrix was created from the transition matrix according to the stationary probability mathematical model for each hour. As long as there were bikes in the zone, we moved scooters from the zone. If the demand for the hour was smaller than the number of scooters present, we only moved the number of scooters of demand. To do so, we simulated a random variable and moved it according to the distribution of the probability matrix. The scooter has the ability to stay in its current zone if that is where the simulation falls.

In our simulation, we thought it was important to account for real-world lost demand. The data obviously only captures the rides that were completed in which people were able to find a scooter to start a ride. In reality, there are likely people who would like to start a ride but cannot obtain a scooter. To account for this in our simulation, we took in a parameter of the typical proportion of lost demand for an hour. We struggled with computing real world lost demand by finding a sudden change in the demand rate because the time stamps are rounded to the nearest fifteen minutes. In addition, the time of day largely impacts the demand rate so there will be naturally occurring large rates of change in the demand rate. Therefore, our conclusion was to have a parameter that could be changed based on consumer insights data that the companies may have. We used a low estimate in our simulations, but this could be changed with more information.

Having the scooter IDs would largely help have a realistic view of this. We would be able to track the movement of individual scooters. This would allow us to see when areas have no scooters and would be able to say there is a higher chance for lost demand here. The void of this attribute significantly reduces the knowledge we have about the scooters, but proves our simulation much more useful. We think that the knowledge of scooter IDs could have opened up a lot of doors for data analysis but understand that it was anonymized for the privacy of customers.

3.2.2 Result

After simulating our three scenarios of equal scooters in each zone and the number of scooters in each zone being proportional to the flow out both over the course of the day and just for the morning demand, we found that although they provide similar metrics for the number of demand captured (651 and 655 and 668, respectively), there is a larger difference in the amount of demand lost. The setups proportional to the daily demand and to the morning demand were quite similar, and we think that with the current number rides, they are pretty comparable. In the scenario in which scooters are positioned equally in each zone, we lose on average 16 rides. However, when we position the scooters proportional to the flow out of each zone, we only lose on average 4 rides. Similarly, when we position the scooters proportional to the morning flow out of each zone, we only lose on average 8 rides. We lose a much smaller number of rides when the morning setup is more proportional to the flow out. Positioning more scooters in areas of high demand would help the company.

4 Additional Insight

One key piece of information we picked up on was that the number of scooters seems to be quite large for the number of rides that happen in a typical day. The average number of rides in a day is approximately 431. With our assumption of 250 scooters, which was on the lower end of our research, each scooter would on average provide less than 2 rides per day. In addition, this metric over the course of 6 months is not really growing, evidenced by the figure below (Figure 2: Rides Per Day). As we can see, the number of rides per day is not increasing as time goes on. This shows that there is a lot of room for improvement for more rides and for scooters to be utilized more each day.

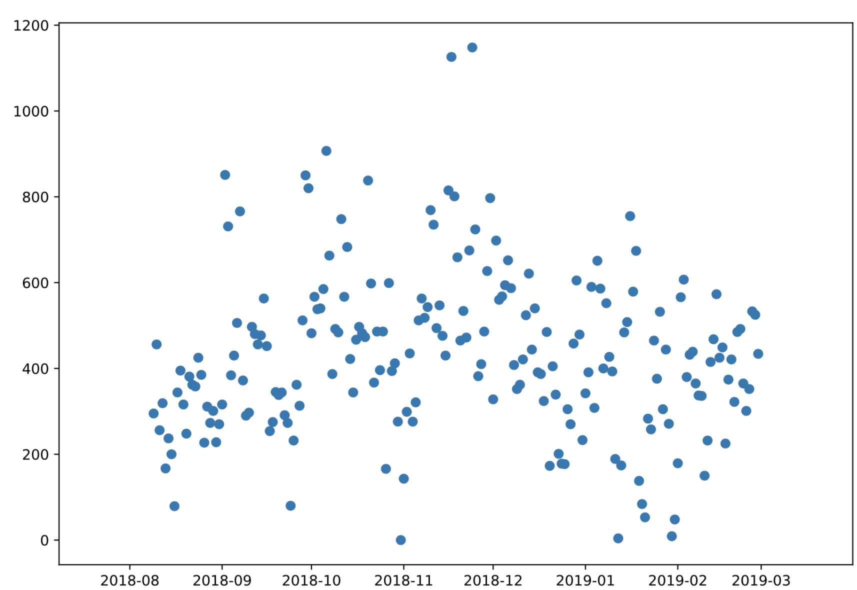

Figure 2: Rides Per Day

Another insight that we took away from this was that the arrivals and departures from each zone flow very closely together. They are highly correlated. This makes intuitive sense. Because this is a dockless system, when a scooter flows somewhere, this is what allows for a departure later. A scooter is necessary for a ride so as one arrives, a departure is possible. For each zone, we can see a similar pattern of arrivals per day and departures per day. One example is given below, with departures in blue and arrivals in orange. (Figure 3: Zone 3 Arrivals and Departures)

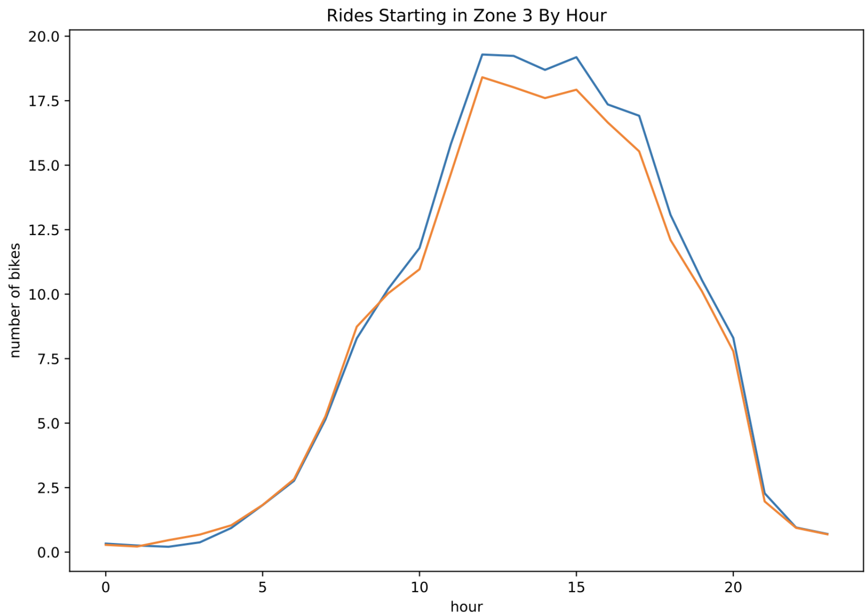

Figure 3: Zone 3 Arrivals and Departures

One idea we had when working on this data was that perhaps these companies should prohibit customers from riding somewhere extremely far. As the rides are centered around the populous areas of Louisville, it is likely that these areas will continue to have high demand. Companies have a discrete number of scooters and if some of these are just used for one ride that then puts the scooter far out of place, it is not maximizing its potential.

5 Conclusion

Overall, we believe that our work could be valuable information to a company such as Bird or Lime. Our simulation could provide very useful for a company that knows how their morning setup is going to look. For example, if a company knows that their morning is likely to follow closely from the night time, they will be able to see what the following day holds for them in terms of the natural movement of scooters and the captured and lost demand. They can make decisions with more information to aid in moving scooters for the morning setup if they desire. Additionally, we believe that we have shown that it is valuable to situate the scooters in the morning in a way such that the amount of scooters in each zone is proportional to the flow out of the zone. We think there is room for improvement with this suggestion, but we believe that the ability to simulate is helpful and over time will allow us to capture a morning setup that allows for good balancing. In conclusion, we believe that there is a lot of room for growth in this e-scooter market and we think that with our simulation model, companies can track the movement of their scooters based on their previous ride data to ensure their morning scooter placements will allow them to capture demand and upset as few customers as possible.

Appendices: Mathematical Model

A Initial Assumptions

For any scooter x, we define its state X_t at time t to be the zone it is located at at time t. We divide the city of Louisville into a total of 16 zones. We model the variation of state of x using a **discrete-time Markov Chain** because the assumption of "future is independent of past" is reasonable. The state of scooter x in the future only depends on the present state of x (where it is now) and the destinations of the users who will use it in the future. Where has x been in the past has no impact. We discretize the time for convenience of making use of the data. The state space S of the Markov Chain is $\{1, 2, 3, \ldots, 16\}$.

The next assumption we make is that the Markov Chain modeling the state of x is **irreducible**. It is clear that all possible states of x communicate. Arbitrarily pick two zones (states) i and j. Given that x is in zone i now, it is always possible for x to go to zone j after finite amount of times and vice versa if x is in zone j now.

The Markov Chain in our model is **recurrent** because this is a **finite state** Markov Chain (there are finite number of zones in the city after all). Let i be an arbitrary zone and let τ_{ii} =min$\{n \geq 1 : X_n = i | X_0 = i\}$ be the time until the scooter x get back to zone i given that it is zone i now. In short, we call it *return time*. It makes sense for us to assume that $E[\tau_{ii}] < \infty$ in that on average the scooter x will be back to zone i in a finite amount of time given that it is in zone i now as long as x is in the system of city Louisville. (Here we ignore the few scooters that might be removed from the city Louisville.) Therefore, we conclude that the Markov Chain is **positive recurrent**.

Now it is tempting for us to think of the following theorem: For a irreducible Markov Chain, it is positive recurrent if and only if there exists a probability solution π to the set of linear equations $\pi = \pi P$. We can apply this theorem to say that the π exists because we just showed that the Markov Chain modeling the states of an arbitrary scooter x is irreducible and positive recurrent. Here, P is the one-step probability transition matrix and $\pi = (\pi_1, \pi_2, \ldots, \pi_{16})$ is a vector where for all $i \in S = \{1, 2, \ldots, 16\}$,

$$\pi_i = \lim_{N \to \infty} \frac{1}{N} \sum_{n=1}^{N} I\{X_n = j | X_0 = i\}$$

is the long run proportion of time the scooter x is in zone j. (Note: It doesn't matter where x start. So the i can be arbitrarily chosen here.) Formally, π is called the limiting or stationary distribution of the Markov Chain on the state space S. Of course, $\pi_j \geq 0$ for all j and $\sum_{j \in S} \pi_j = 1$. The linear equations $\pi = \pi P$ **seem to be** the core of our mathematical model and we should do simulations and data analysis based on that.

B Issues

Unfortunately, there is an issue with the model described in section 1 because the system is not **stationary**.

Issue (1): Suppose that we adhere to the model in section 1 and allocate the total 250 scooters to 16 zones according to the computed **stationary distribution** π in the morning (6am). (For each $j \in S$, we allocate $250 \times \pi_j$ scooters in zone j.) Then by the property of stationary distribution, the distribution of the scooters will still be the same at any later time. Basically, there will still be $250 \times \pi_j$ scooters in zone j for all j at any future time. The number of scooters in a zone will be the same forever. Of course, that is NOT realistic.

Issue (2): In the model in section 1, we assume that the transition of the state of the scooter x will follow the same probability matrix P for each step (hour). Nevertheless, this is not true in reality. For example, zone i is a residential area and zone j is an office area. Then of course, P_{ij} is larger during 8:00am - 9:00am than during 5:00pm - 6:00pm as more people would need to use scooters to go to work in the morning. Thus, we can't use the same transition matrix throughout.

C Modifications

We decide to break the time of a day into 24 hours to discretize the time. So we will be talking about the state of a scooter during an hour and the transition of its state from hour to the next hour. And our analysis will be based on each hour separately. For each hour, we will estimate from data the associated transition matrix P that governs the movement of scooters during the hour. The entries of the transition matrix P are estimated by

$$P_{ij} = \frac{\text{number of flows from zone i to zone j during the hour under consideration}}{\text{number of flows out of i during the hour under consideration}}$$

. There may be different transition matrix P for different hours. Now we model the change of the state of a scooter using a **pseudo discrete-time Markov Chain** where the memoryless property, irreducibility and positive recurrent property still hold but the transition matrix P will change from hour to hour.(From step to step)

These estimated transition matrices for all hours of a day play a crucial role in our simulation work. We allocate the total of, say 250, scooters into 16 zones in a certain manner each morning at 6am. Then for each hour later on, the movement of scooters will be governed by the transition matrix P for the hour. (For each hour, a scooter in a zone i will go to zone j with probability P_{ij} where P_{ij} is the entry in the associated transition matrix at row i and column j. If the scooter really goes from zone i to zone j, then zone j will be the location of the scooter for the next hour and its next movement will again to governed by the new associated transition matrix and so on and so forth.)

D Poisson Process Assumption

Besides the movement of scooters, the arrivals of demands(users) during a certain hour are also simulated. Here, we assume that the arrivals of demands in a zone i for an hour t are according to a **Poisson Process** with parameter (the arrival rate per hour) λ be the average number of daily flows out of zone i in hour t (this quantity can be obtained from data analysis).

The property characterizing Poisson Process is the **Memoryless Property**. As we know, a Poisson Process has **Exponential** interarrival times. We have two reasons why this assumption is not far-fetched.

Reason(1): "Future is independent of past" still applies here. The arrival of future demand only depends on the future user's own will. It has nothing to do with what happened in the zone in the past. The user simply wants to use a scooter some time in future for his/her own purpose. He/she doesn't care about who have used scooters in this zone in the past/how many people have wanted to use a scooter in this zone.

Reason(2): "Time until the next arrival can be assumed to be memoryless because of ignorance." From the future user's perspective, he/she might think that "I will need a scooter to go somewhere at 5:30pm today". So from that perspective, the time until he/she wants to use a scooter(the actual arrival of demand" is definitely NOT memoryless. Nevertheless, an outside observer has no idea about what a future user is thinking about. Basically, the observer has no knowledge about what will happen in the future. Telling the observer how long ago the last user wanted to use a scooter gives the observer no information about when the next potential user will want a scooter. Therefore, it makes sense for us to assume that the time until the next arrival of demand is **memoryless**, that is exponential.

In our simulation, we use a **Poisson** random variable with parameter λ to model the number of demands arriving in zone i during the hour t. Remember that for a Poisson Process with arrival rate λ, the number of arrivals in a time window of length t is a Poisson random variable $Poisson(\lambda t)$. Here length of time window is 1. So we use the random variable $Poisson(\lambda)$ to model the number of demands arriving in zone i during the hour t. The number obtained will be compare with the actual number of scooters in zone i during hour t to acquire the **demand captured** and **demand lost**, which are important measures of how good our morning allocation of scooters is.

E Simulation Code

```python
def function(initial_setup,lost_demand):
    print("start setup", initial_setup)
    demand_captured=0
    demand_lost=0
    bikes_not_used=0
    hourly_zone_demand=[zone_demand_day(i) for i in range(1,17)]
    hourly_zone_arrival=[zone_arrival_day(i) for i in range(1,17)]
    for i in range(24):
        hourly_transition_matrix=get_transition_matrix(i)
        for j in range(16):
            hour_zone_demand=hourly_zone_demand[j][i]* (100/(100-lost_demand[i]))
            hour_zone_demand_sim=round(np.random.poisson(hour_zone_demand))
            num_bikes_in_j=initial_setup[j]
            zone_arrival_day_sim=int(round(np.random.poisson(hourly_zone_arrival[j][i]))/2)
            if num_bikes_in_j>=hour_zone_demand_sim:
                demand_captured=demand_captured+hour_zone_demand_sim
                bikes_not_used=bikes_not_used+(num_bikes_in_j-hour_zone_demand_sim)
            else:
                demand_captured=demand_captured+min((num_bikes_in_j+zone_arrival_day_sim),hour_zone_demand_sim)
                demand_lost=demand_lost+max((hour_zone_demand_sim-(num_bikes_in_j+zone_arrival_day_sim)),0)
            if (num_bikes_in_j>0):
                num_bikes_moving=min(hour_zone_demand_sim, num_bikes_in_j)
                for k in range(num_bikes_moving):
                    zone_j_probs=np.cumsum(np.array(hourly_transition_matrix[j]))
                    u=np.random.rand()
                    x=0
                    while (u>zone_j_probs[x]):
                        x=x+1
                    if x!=j:
                        initial_setup[j]-=1
                        initial_setup[x]+=1
        print("after hour",i, initial_setup)
    print("demand captured", demand_captured)
    print("demand lost", demand_lost)
    print("bikes not used", bikes_not_used)
    return demand_captured,demand_lost, bikes_not_used
```