A Visual System for Managing Bike-Share Networks

Sam Longenbach & Minkun Liu

April 2019

Abstract

Bike sharing systems (BSS) have been becoming more popular in urban areas across the globe. For this mode of transportation to be viable for its users there must be sufficient surrounding infrastructure. Managers of docking BSS are always looking for all docking stations have bikes present as well as all docking stations must not be completely full or a user can not return his or her bike rental. This problem of bicycle re-positioning can be viewed as dynamic during the day and static during inactive hours as user demand is flat. We look to tackle the issue dynamic bicycle re-positioning by processing historical bike trip data to identify daily trends. We then combine these daily trends with real time station capacity information to present a interactive visualization to help BSS managers optimally manage and allocate bicycles in the network.

1 Introduction

The bicycle is an invention that often goes under appreciated. It 3-5x faster than walking and requires little surrounding infrastructure to able to use effectively in comparison to other modes of transportation. It is a invention that has been out shadowed by another invention, the automobile. Especially in a country like the United States, transportation infrastructure has been centered around the automobile for last century. For traveling long distances in rural environments, the automobile reigns supreme over the bicycle. However, in recent years as the flow people has been moving towards highly dense urban areas, the bicycle may be a solution to providing citizens efficient and economical mobility.

While citizens could purchase a their own bicycle, another option would to take advantage of bike sharing systems (BSS) that have been becoming more popular in urban areas across the globe. In general, users are able to checkout a bike from a docking station, ride the bike for a given amount of time, and return it to another docking station. While sharing platforms may vary in cost or max ride time, managers of docking BSS are always looking to meet at least two criteria. First there must always be bikes at all stations and secondly docking stations must not be completely full or a user can not return his or her bike rental.

For the automobile to be a successful mode of transportation, there had to be a surrounding infrastructure in the form of a quality network of roads and highways. The same idea holds true for implementing a well working bike share system. While the infrastructure for an effective BSS is multifaceted, we concern ourselves with the problem of bicycle re-positioning. There has been a decent amount operations research carried out in the area of tackling the problem of bicycle re-positioning problem [2]. The issue of overnight bicycle re-positioning is often viewed as static, while intraday bicycle re-positioning is dynamic as users real-time demand must be accounted when attempting to balance the system. Unsurprisingly the dynamic re-positioning problem is far more challenging than the static re-positioning as real time user demand has to be accounted for. In reality in high volume BSS, the re-positioning decisions are made by humans and is not completely automated.

To aid this human decision making process, we lay our idea and present a basic prototype for a visualization system to assist bike sharing system managers in handling the problem of dynamic bicycle re-positioning. We implement our prototype using data from Divvy, the bike sharing system in Chicago. Our visualization system utilizes historical user trip data from Divvy along with their live JSON data which provides real time updates of the current capacity of all the docking stations in their network. We use Divvy simply due to the fact their data is reliable and public. However, our visualization system is general in the sense it can accompany any bike sharing systems tracks this type of user data.

2 Related Work

Applying clustering techniques to discover patterns in user activity has been a popular technique used in literature. Vogel used data mining to gain insight into bike activity patterns by binning trip data [3]. His works inspired some of the techniques we have implemented in our own system presented below. Bargar explored the utility of building an interactive web-based visual analytic application for comparing usage patterns between bike sharing programs. The paper utilized the ST-DBSCAN algorithm to cluster trips as a way of categorizing flow patterns [1]. We also tried applying clustering techniques such a DB-Scan to extract patterns to show how different stations behave in regards to usage however after reviewing initial results we decided to move in another direction.

Recent research in the visualization domain looks to implement a dashboard to highlight bike activity patterns at various stations depending on time of day or types of attraction in close proximity [4].Solely using user trip data can sometimes not tell the whole story when it comes to discovering user activity patterns. This paper aims to explore the user activity patterns by combining different data sources, such as taxi data and social media data. Processing this data into tensors they are quickly able to process and render data on the visualization dashboard. Their proposed interactive visual analytic system is then used for exploring the latent user activity patterns of BSS data in and between cities.

3 Data Preprocessing

Divvys BSS data extends from mid 2013 until the end of 2018. As of 2019 there are 608 stations with 5,800 total bikes that Divvy operates throughout the city of Chicago. There are different types of trips a user can choose from when renting a Divvy bike. One is called a Single Ride that allows the user to rent the bike for 30 minutes. Another option for Divvy riders is called a Explorer pass which allows a user unlimited number of rides within a 24 hour period with the caveat being they must dock every 3 hours. The final type of ride is called Annual Membership which is for members who want to ride unlimited number of times throughout the year with the first 45 minutes per ride free.

We note these specifications of the Divvy system to make you the readers aware that different bike share systems may vary in membership structure. With this said, from a managing prospective we dont concern ourselves with these nuances when analyzing the data. Instead we view all trips equally in the sense that the result is the same. The result being a bike is no longer at a particular station and will be returned at some other station at a later date. Moreover, summary statistics on Divvy data show that the overwhelming majority trips are less than hour in duration. Therefore people are using this particular system to get from point a to point b.

Each row of data contains the trip start and end time as well as the start and stop station. After cleaning and eliminating trips that are less than 60 seconds meaning the user did NOT actually ride the rented bike, we have around 17 million trips. Additionally, the real time station information of the 608 stations is provided by the live JSON feed. For each station we know the longitude and latitude location as well as have access to how many bikes are available at each station. Through testing we found Divvys live JSON feed is quite stable as it updates station information about every 5 minutes.

As for historical trip data which contains the 17 million user trips, we process this data by counting the number of trips from and to each station. We consider trips from a station as a out bike trip and trips to a station as a in bike trip. For example, if a user took a trip from station 2 to station 6 then we count this as a out bike trip for station 2 and a in bike trip for station 6. Additionally, we can bin the out and in trips for each station by each hour of the day. The result is two matrices each with the 608 rows and 24 columns. Each row represents a station and each column represents the total number of in/out trips for that hour of the day. Figure 1 displays the distribution of number of in and out trips for each station.

We see that unsurprisingly the total number of in trips equals the total number of out trips. More importantly the lower quartile of the box plot indicates that 75% of stations have seen more than $e^{7.5} \approx 1{,}700$ trips in or out trips. While the median of the box plot indicates that 50% of stations have seen more than $e^{9.5} \approx 13{,}000$ in trips or out trips. This variation can be attributed stations near the city center experience more use as well as a lot of stations have been built fairly recently.

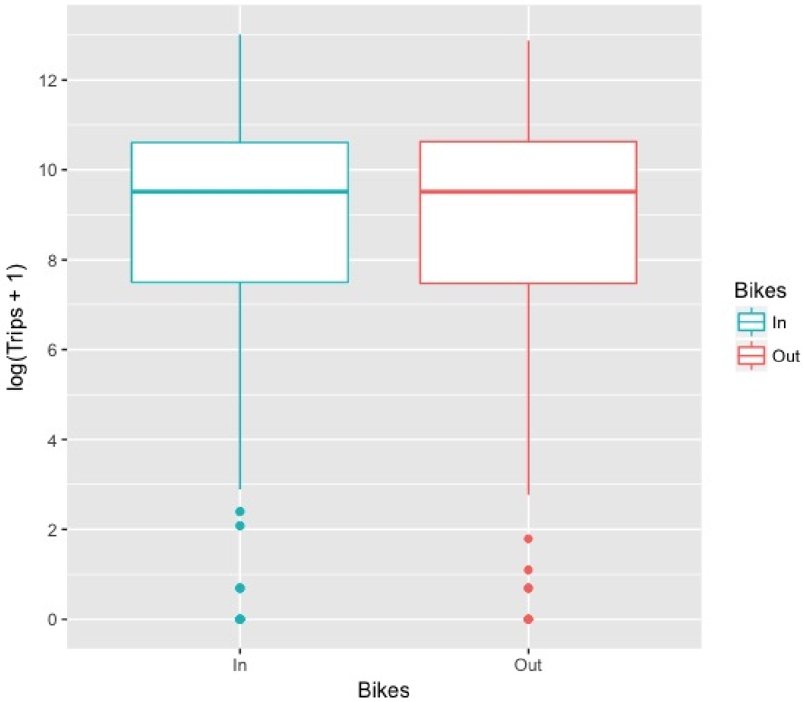

Figure 1: Variance in the number in/out trips among the 608 Divvy stations.

Despite the imbalance of in/out trips across stations, we look to normalize the station data. For each station row we divide by the total number of in/out trips. Therefore each entry in the out trip matrix displays for a given station the proportion of out trips that occurred for that hour of the day. The same logic applies for the in trip matrix. We do this so that we can recognize station trends. Our vision is that overtime as trip data accumulates for all stations, eventually there could be multiple subsets of these in/out matrices. For example, we could have one for holidays, one for weekdays, ect. However, for now we implement our system with one historical aggregate. Below shows examples of what these trends can look like across different stations.

Each subplot in figure 2 shows a given stations proportion of in and out trips over a day. Station 35 is an example of a station that has fairly constant and equal number of bikes coming in and out. In contrast, station 192 is an example of a station that experiences a huge outflow of bikes in the morning hours from 6am to 9am and huge inflows of bikes from 3pm to 6pm. For station 192 the local maximum point shows that almost 30% of the daily bike that flow into this station occur between 4pm and 5pm. Lastly, we can also take the difference between these two distributions to see that the net in/out flow of bikes for a particular station. Figure 3 below displays this difference.

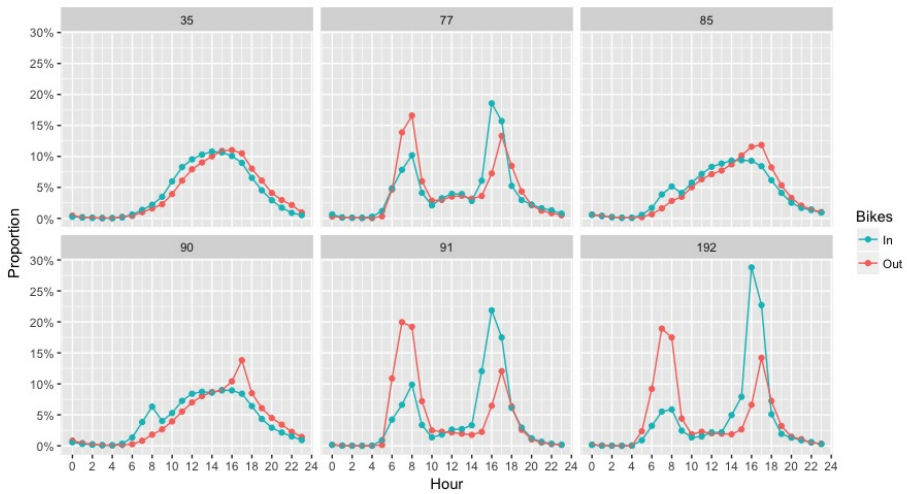

Figure 2: The historical proportion of in/out trips over a day for Divvy stations 35,77,85,90,91,192.

4 Managing Algorithm

As discussed above, from the live JSON feed we have the figures for the real time quantity of bikes at each station. For each station we could have a domain expert decide on the minimum and maximum number of bikes that should be at each station at any point of time. However, for our current version of our visualization system we code in a global alpha and beta to represent the upper and lower bound percentage thresholds. As default we have alpha and beta equal to 5%. Therefore for each station at a any given point in time the algorithm will label the station as the following:

- Low (Red*): Total Docks*Alpha > current bike quantity
- Balanced (Green) : Total Docks*Alpha $\leq$ current bikes $\leq$ Total Docks - Total Docks*Beta
- Full (Blue*): Total Docks - Total Docks*Beta < current bikes

Additionally, from figure 4 we have developed this idea of historical net bike flow. We can use these historical trends of the proportion of bike flows to add a second labeling rule. For stations currently labeled low we change the label to yellow and for stations currently labeled full we change to purple if the following is true at the current time:

- Low, expecting inflow (Yellow): Historical Inflow - Historical Outflow > 0
- Full, expecting outflow (Purple): Historical Inflow - Historical Outflow < 0

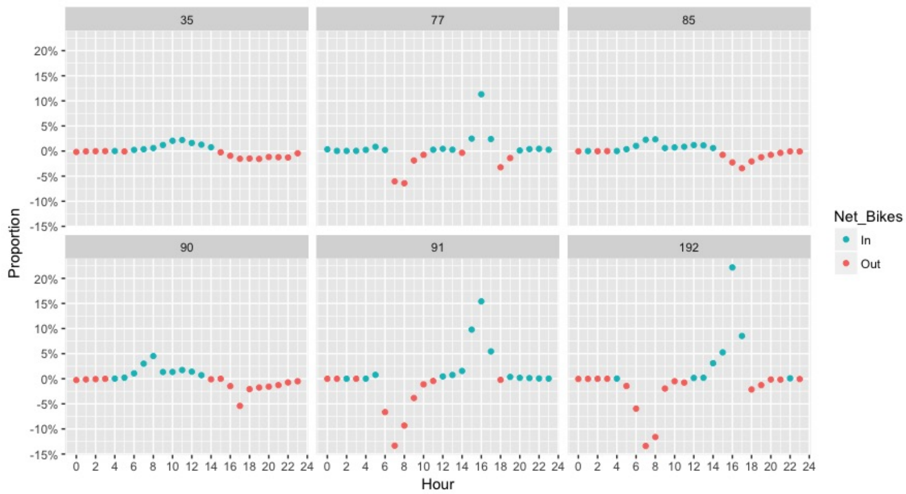

Figure 3: The difference in historical proportion of in/out trips over a day for Divvy stations 35,77,85,90,91,192.

Depending on the time of day and location of a station it may be okay to have it run close to empty or full. The classic example is stations located near residential areas may need to start out close to full in the morning while stations located near the business district may run close to empty. It would be a waste of resources to move bicycles midday is users are naturally going to rebalance the system for you. You can imagine situations where intervention could in some cases make the system worse off for users.

5 Visualization System

To show how a BBS manager could begin to make use of our analysis we create a simple visual system using D3.js and Flask. We use Flask, a Python web framework, to process our data in the back end before being visualized. This framework allows use to make use of Divvys real time JSON feed to update information on station capacities. As for the visualization is is comprised of a few components. The main part is a zoomable map of in this case Chicago. The coloring of the stations on the map is determined by the algorithm discussed above. Once zoomed in on the map you are able to click on stations to see more information about particular stations. For a particular station there is a trend line like in figure 3 displaying if the net flow is positive or negative. Note the vertical black line displays the current time. Since there are 608 stations in the Divvy bike network, there are a lot of stations in close proximity to each other. Currently, it would be up to the manager to decide to move bikes from one station to another if necessary. As mentioned above additional parameters such as alpha and beta could be tuned to meet the needs of a specific bike share system.

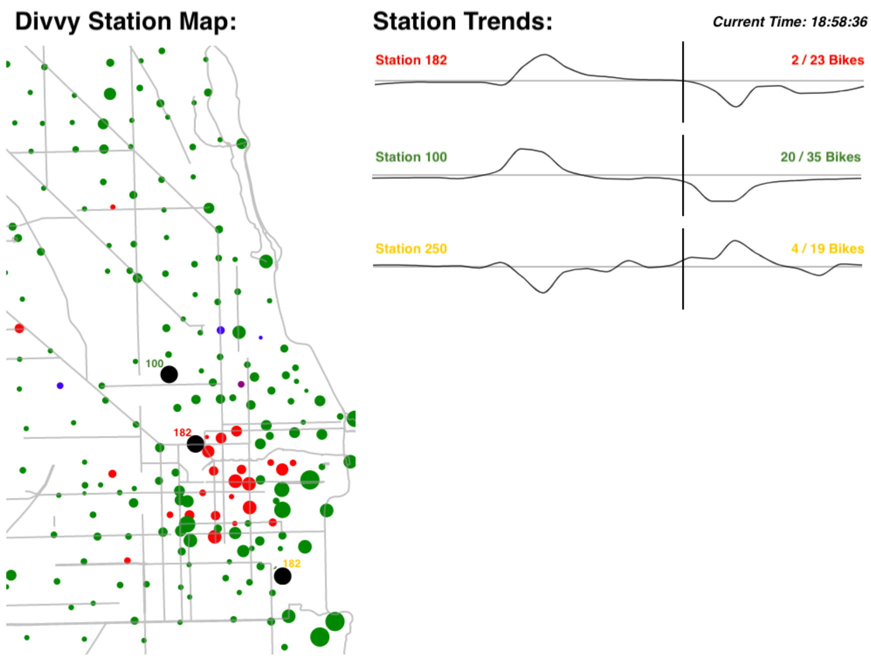

Figure 4: Visual System with stations 182,100,250 selected.

6 Conclusion

Bike sharing systems are becoming increasingly popular and require intelligent supporting infrastructure in order to meet user's demands. We propose a method to process historical trip data to highlight daily patterns across docking stations in a bike sharing system's network. Combining this analysis with real time station data we are able to create a visual system for BSS mangers to better tackle the problem of bicycle re-positioning. We initially highlight each station in the network either low, balanced or full. Furthermore we change the color of low and full stations if daily trends anticipate users will re balance the system. As a particular BSS network collects more data, future work would entail stronger trend prediction based on factors such as day of the week. Additionally, developing algorithms to suggest how to balance low/full stations may be useful to managers.

References

[1] A. Bargar, A. Gupta, S. Gupta, and D. Ma, "Interactive visual analytics for multi-city bikeshare data analysis," in *The 3rd International Workshop on Urban Computing (UrbComp 2014), New York, USA*, vol. 45, 2014.

[2] H. M. Espegren, J. Kristianslund, H. Andersson, and K. Fagerholt, "The static bicycle repositioning problem-literature survey and new formulation," in *International Conference on Computational Logistics*. Springer, 2016, pp. 337–351.

[3] P. Vogel, T. Greiser, and D. C. Mattfeld, “Understanding bike-sharing systems using data mining: Exploring activity patterns,” *Procedia-Social and Behavioral Sciences*, vol. 20, pp. 514–523, 2011.

[4] Y. Yan, Y. Tao, J. Xu, S. Ren, and H. Lin, “Visual analytics of bike-sharing data based on tensor factorization,” *Journal of Visualization*, vol. 21, no. 3, pp. 495–509, 2018.