CS 7280 Final Project Report

Yuanda Zhu

Abstract

In order to satisfy the needs of management from a public bicycle sharing company, Divvy, in Chicago, the student adopted network models to perform two separate analysis. The undirected network model plus HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) aims to provide communities of bicycle stations based on the "traffic distance" instead of physical distance: stations with more frequent bicycle traffic between are ideologically closer to each other and can be managed under the same team. If needed, new stations can be built on the possible routes between these popular stations for the "hotlines". The directed network and Q-learning aim to predict the incoming and outgoing traffic flows of popular stations so that the company is capable of predicting future shortage of bicycle at certain stations either in the morning or in the evening, or both. Decisions such as increasing capacity of storing bicycles at these stations and temporarily transferring bicycles from other stations can be made given the reliable prediction results.

Introduction

Nowadays people who live in metropolitan areas start riding bicycles. Being convenient within short distance, working out during transportation and behaving environment-friendly are the reasons why people choose bicycles. Divvy in Chicago has been recording data on bicycle sharing since 2013 [1]. Around 80,000-140,000 rides occurred in each half-year dataset.

There are over 560 bicycle stations of Divvy in Chicago. Surely not all stations are equally popular. Some stations and paths (ex. from Station A to Station B) are mostly used during certain period of time. With eight half-year dataset available online, I explored the most popular bicycle stations in Chicago. Some of them have formed communities that large number of traffic has been recorded between these subnets of the entire bicycle network.

Meanwhile, popular paths are formed between bicycle stations. People who ride bicycle to work and back home can create regular and frequent paths between pairs of two stations. An interesting fact is that, some people may ride bicycles from Station A to B in the morning, and ride back to Station A after work. Another group of people may do the contrary. In real bicycle

network, the situation is definitely more complicated. Inward and outward traffic of a certain station is essential to the company, since if one station runs out of bicycles due to sudden increase of demand, either in the morning or in the evening, customers cannot commute equally timely or costly. By incorporating machine learning algorithms, patterns of inward and outward traffic can be predicted at certain stations. Expanding the capacity of the popular stations, or temporarily transferring bicycles from less-popular stations, is one aspect of the daily maintenance.

This project has strong connection to network science topics covered in class. Community detection is the main idea. Popular stations will merge nearby unpopular stations; thereby a few communities will stand out whose cores are those mostly used stations. What's more, an additional directed network is constructed to analyze any shifts within the same day; with certain historical data, machine learning algorithms is implemented for prediction and validation.

Prior work

Unsupervised clustering algorithms is the main approach for community detection. K-means [2] is the easiest clustering algorithm but generally fail to achieve satisfying results. Spectral clustering [3] is efficient in frequency-related dataset. HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) [4] is a powerful clustering algorithm but relies on large number of data.

For machine learning part, reinforcement learning has shown its effectiveness in many applications [5]. Q learning [6] has its own strength in quick convergence and accuracy in prediction. It's also easy to implement.

Methodology

Part 1: The undirected network and HDBSCAN

The dataset is obtained fully from website Divvy Data. Roughly 80,000-140,000 rides are given in each excel file of the half-year data. As the nodes of the network are different stations, each

row of dataset representing a ride from Station A to Station B, one path was added from node A to node B. By the end of each excel file, a weighted network would be built. Then the weighted, undirected network has the following property: highly weighted paths tend to be shorter in distance while less frequently used paths become longer in distance.

HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) was implemented as the unsupervised clustering algorithm. Comparing with other unsupervised clustering algorithms, HDBSCAN has several advantages. First, no a priori knowledge on number of clusters is needed. Second, HDBSCAN can find any arbitrary shape due to single-linkage effect. Third, HDBSCAN can identify noise and robust to outliers. Comparing with traditional DBSCAN, HDBSCAN incorporates the idea of hierarchies. Thus, only one parameter is required, the minimum cluster size, which is used to condense the cluster hierarchy.

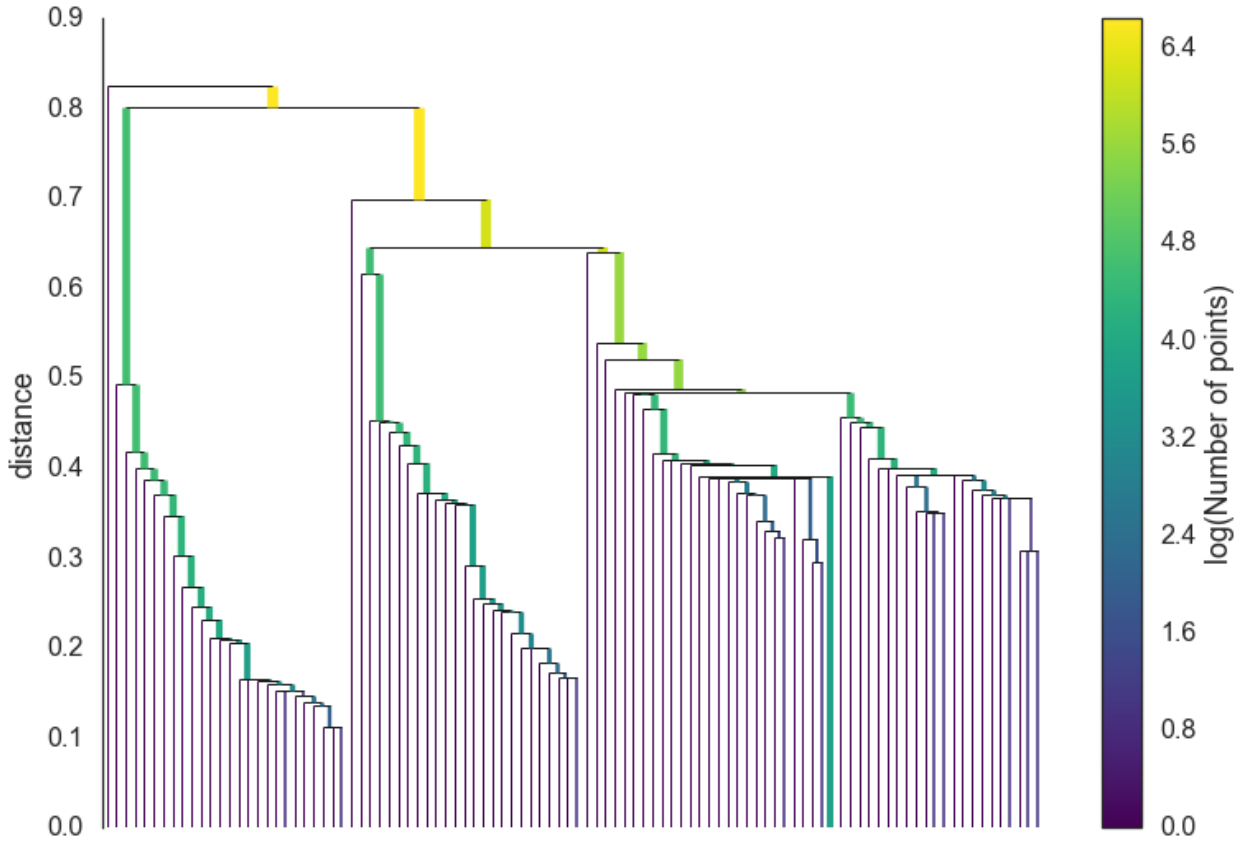

Figure 1. Hierarchies of nodes based on their relative distance to the closest neighboring cluster

Part 2: The directed network and Q-learning

Besides the undirected network, a directed network is also constructed. Different from the undirected network, which considers traffic all day long, the directed network is built based on morning rides (arrive between 7am and 10am) as well as evening rides (leave between 4pm and 8pm). Two columns of data on time from the dataset are used for filtering out unqualified data.

Since machine learning algorithms, including Q-learning, requires a large number of data, for training and testing, a monthly based network was built to provide features. However, unlike cities such as Atlanta, Chicago has a long and freezing winter. The severe weather and harsh wind significantly reduce the number of rides from November to March. Thus, even though the morning and evening rides are divided into monthly arrays, for the smoothing purpose, a three month duration of traffic is considered when building the directed network. Each time by sliding one month, 92 smoother "monthly"-based networks are constructed out of 48 months of raw data (morning has 46 networks, evening has 46).

For further purpose of pre-processing, non-frequent rides are eliminated from the directed network. I would expect people who ride bicycles to work shall maintain a minimum number of ride per quarter of year; thus, I manually set a threshold of 50 rides so that some random rides between two unpopular stations are removed from analysis. Only the frequent rides contribute their paths into those 92 networks.

Considering the fact that the number of the stations grew from around 350 to 560 within the past four years, the inward and outward traffic of only a handful stations can be monitored. Thus, I manually picked the 10 different stations with the highest degrees in the first network. These nodes are considered for further machine learning analysis due to their popularity, especially the "hub" influence.

The next step is to extract features for Q learning. In degree centrality and out degree centrality are the features. The reason is also related to weather. Despite the fact that the total number of rides differ significantly from summer to winter, specific stations and routes have always been more popular than others. However, some suburban or distant stations might be popular in summer, but have almost no riders stopping by in winter. Such relative popularity of certain stations in the network can be compared between summer months and winter months. Thus, the in degree centrality, corresponding to how many bicycles are received by one station

comparing with all bicycles received at that time, and the out degree centrality, corresponding to how many bicycles are taken away by cyclers at one stations comparing with all bicycles taken away, are calculated for all 10 stations, morning and evening. Thus a total of four .csv files of features are created.

At last, Q learning is adopted. One single parameter, five-month SMA (smooth moving average), is calculated for the centrality. Each .csv file will train one Q-table and has it tested. I manually choose the top thirties months of data for training the Q-table, and the separated bottom 15 months of data for testing. No cross validation can be performed since I am supposed to predict future; thus, no future data could be used for training purpose.

Here are the details on how to build the Q-table. First, the rows of Q table are divided into 10 parts, and each equal number of rows are for states of the same station. Next, SMA data array is discretized. For example, I divided the SMA data into five equal subsets between minimum and maximum. Data within the smallest subset belongs to state 0, within the next subset belongs to state 1, etc. Note that only the top 30 data of one single column shall be considered into discretization, not the entire column. The third step is to define “actions” of the table. Reinforcement learning emphasizes how the action can influence the performance and maximize expected interests; however, we don’t really expect a real action. All we need is the prediction of a rise (+25%), a drop (-25%), or a level (no big change) of the centrality. Thus, three visual actions are adopted.

To finalize the Q learning, reward has to be included when updating the Q table. For a successful match, a positive 50 is given; otherwise, a negative 50 is assigned as penalty.

When testing the bottom 15 month of data, we no longer need to update the Q-table. Accuracy of prediction is calculated for each station as well as all stations in the same .csv file.

Results

Part 1: The undirected network and HDBSCAN

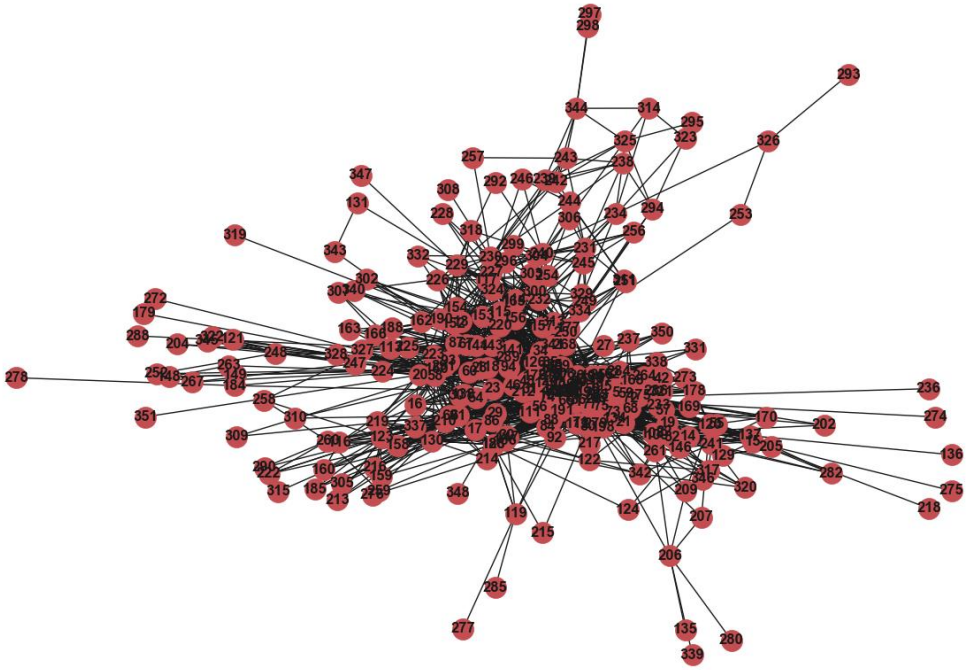

Figure 2. Weighted, undirected network in edges and nodes from year 2013 data.

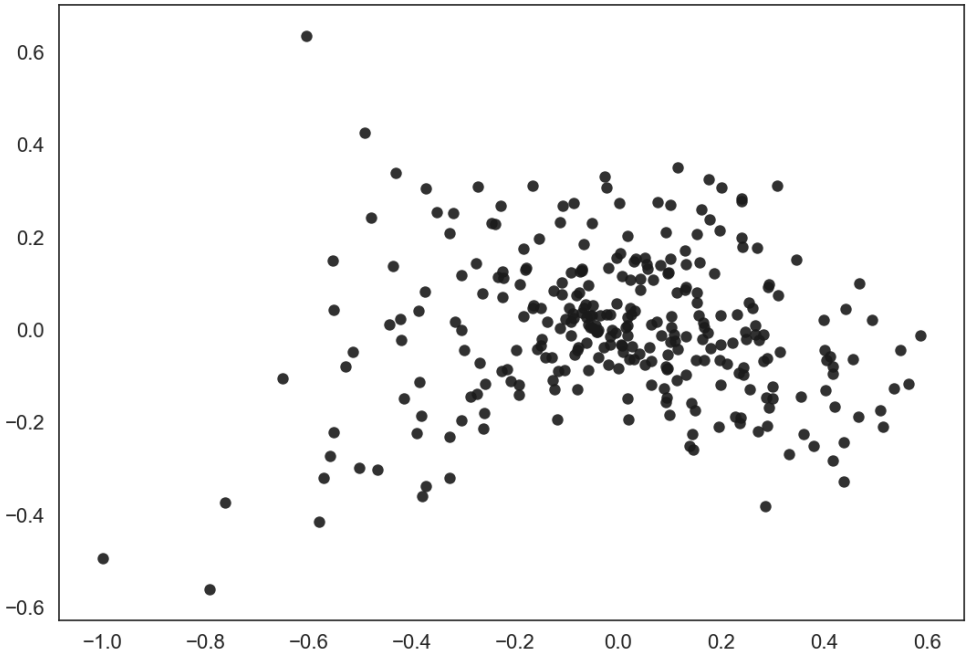

Figure 3. Unclustered, undirected and weighted network from year 2013 data.

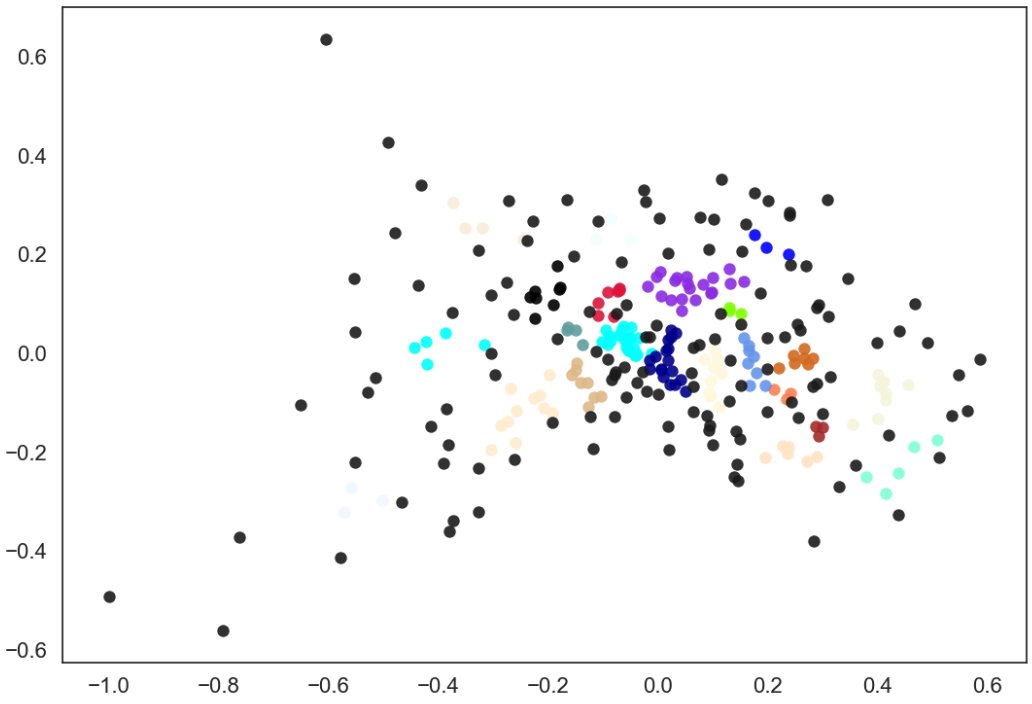

Figure 4. Undirected and weighted network from year 2013 data after HDBSCAN clustering. Nodes in the same color belong to the same community.

Figure 2-4 shows the undirected network constructed from the year 2013 data. In figure 2, stations tend to locate closer to each other provided more traffic between them. In figure 4, black nodes are the outliers, or noise. These nodes are considered too far from nearby clusters so that they are isolated. In real world these stations have relative fewer traffic towards another station.

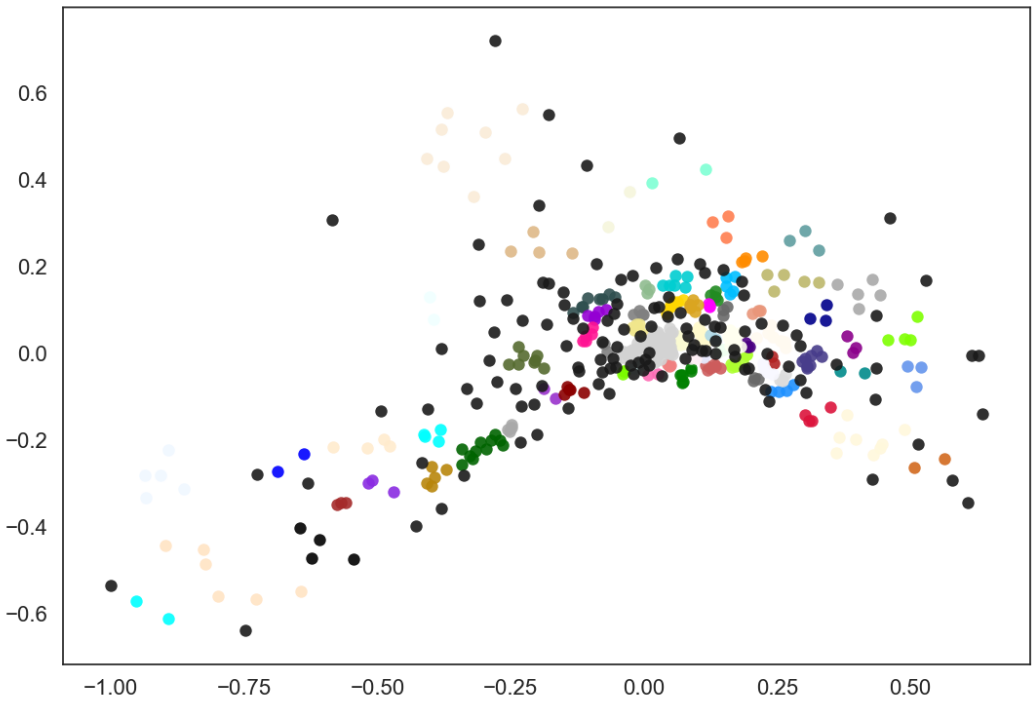

Figure 5. Undirected and weighted network from year 2015, Q1&Q2 data after HDBSCAN clustering. Nodes in the same color belong to the same community.

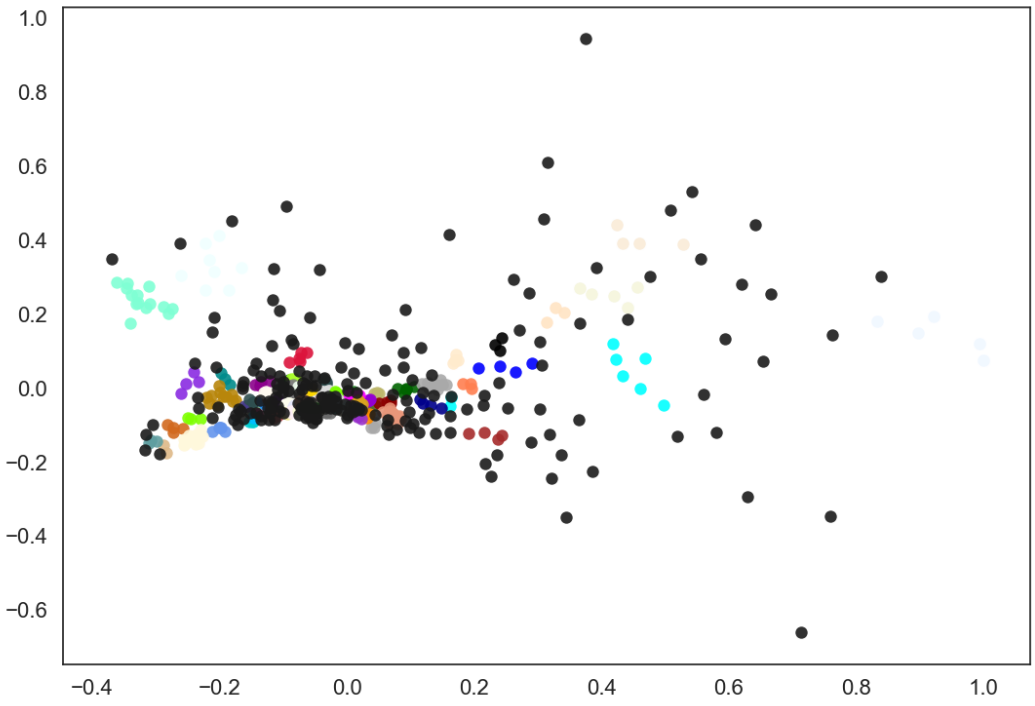

Figure 6. Undirected and weighted network from year 2017, Q1&Q2 data after HDBSCAN clustering. Nodes in the same color belong to the same community.

Surely the communities don't stay the same. With the emergence of new stations, as well as the rise and fall in popularity of existing stations, partially influenced by weather, communities can from half a year to another half a year. For a given period of time, the community clustering is stable and deterministic. By keeping track of these communities of bicycle stations, the Divvy Company can have a better approach to manage the stations, other than simply divide them based on physical locations.

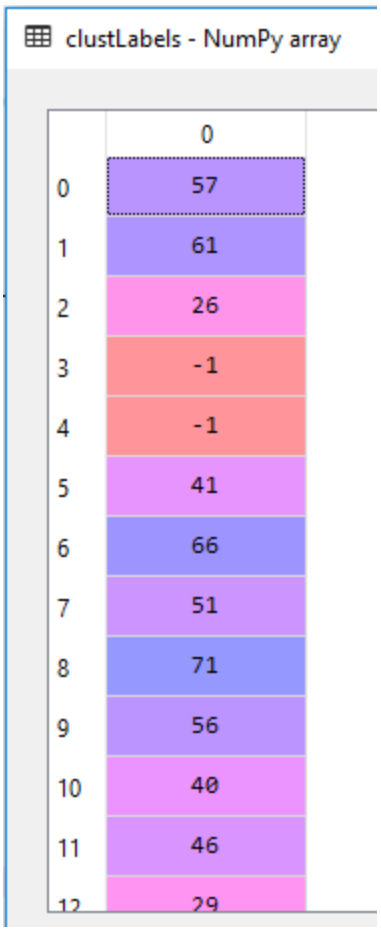

Table 1. HDBSCAN cluster labels. Outliers are clustered as -1. The index on the left are the node (station) index. All nodes are assigned a value.

Part 2: The directed network and Q-learning

Station	91	75	174	48	192	36	77	51	49	81
Correct	4	6	8	13	9	10	7	4	7	9
Incorrect	11	9	7	2	6	5	8	11	8	6

Table 2. Accuracy for morning data, in degree centrality. The overall accuracy is 51.3%.

Station	91	75	174	48	192	36	77	51	49	81
Correct	15	2	13	5	14	8	6	5	14	7
Incorrect	0	13	2	10	1	7	9	10	1	8

Table 3. Accuracy for morning data, out degree centrality. The overall accuracy is 59.3%.

Station	91	35	90	75	26	174	76	85	97	59
Correct	3	8	8	9	5	7	8	5	4	7
Incorrect	11	6	6	5	9	7	6	9	10	7

Table 4. Accuracy for evening data, in degree centrality. The overall accuracy is 45.7%.

Station	91	35	90	75	26	174	76	85	97	59
Correct	4	9	3	5	5	6	10	3	7	4
Incorrect	10	5	11	9	9	8	4	11	7	10

Table 5. Accuracy for evening data, out degree centrality. The overall accuracy is 40.0%.

Table 2-5 show the accuracy on prediction using Q-table. The accuracy varies largely from station to station. The overall accuracy for each table is disappointing. With such poor accuracy, it's impossible to predict the rise or fall of relative incoming and outgoing traffic to these popular stations.

Discussion

The results from undirected network and HDBSCAN are satisfying. By analyzing the data from the past quarter, the company can re-draw the communities of stations, based on the "traffic distance", instead of physical distance. Since HDBSCAN is reliable and deterministic, such clustering results are stable and good to use.

On the other hand, the results from directed network and Q-learning are catastrophic. There are several reasons that this part fails.

First, weather. Chicago is a northern city that snows heavily in winter. People can hardly ride bicycles on those windy and snowy mornings (for four years data, evening rides are almost

three times as many as morning rides!). Consequently, the number of rides differ significantly from season to season, even sometimes from month to month. Thus, it is quite challenging to predict a pattern of rides based on historical data. Simply the weather is highly unpredictable and public bicycle services are easily impacted.

Second, limit number of data. The public sharing of bicycles began only four years ago; not enough data have been collected for machine learning training and prediction, especially on the network basis. Had there been 40 years of use and data collected, such a project could be improve a lot.

Third, when considering the importance of a node in the network, the expansion of network must be considered. As the business of Divvy Company grows so fast during the past four years, the number of stations increases from 350 to 560. Large number of new nodes and edges are added: new stations may have large capacity of storing bicycles due to sufficient funding; some new stations may purposely be used as hubs to accommodate the sudden needs of bicycles from nearby stations. Thus, tracking and monitoring the in and out degree centrality of old nodes while ignoring the new nodes and edges would make prediction extremely inaccurate. When the company maintains a stable business in Chicago, the analysis and prediction might be more accurate.

Last, reinforcement learning, to be more specific, Q-learning, might not be an optimal solution for training and testing on this dataset. On one hand, as I mentioned before, no real action is needed to influence the actual behavior. We simply monitor and predict. Thus, the choice of actions is not ideal. On the other hand, the reward can be better designed. For stock market simulator, by buying and selling stocks, the algorithm looks for the approach to maximize the profit. For path search applications on a map, reaching the destination will automatically be given a huge reward. However, for this part of the project, the ultimate goal is unclear for the algorithm itself.

Future work can start in either switching to a different machine learning algorithm for training and prediction, or designing a better reward for the existing Q-learning.

Reference

[1] Divvy Data, Historical trip data available to the public. https://www.divvybikes.com/system-data

[2] T. Kanungo. “An efficient k-means clustering algorithm: analysis and implementation”. IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume: 24 Issue: 7. Jul 2002

[3] Andrew Y. Ng et al. “On Spectral Clustering: Analysis and an algorithm” Proceeding NIPS'01 Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic. December 03 - 08, 2001

[4] L. McInnes, J. Healy, S. Astels. “HDBSCAN: Hierarchical density based clustering In: Journal of Open Source Software” The Open Journal, volume 2, number 11. 2017

[5] Sutton, R.S. and Barto, A.G. “Reinforcement Learning: An Introduction” Trends in Cognitive Sciences, Volume 3, Issue 9, 1 September 1999, Page 360

[6] Christopher J. C. H. WatkinsPeter Dayan. “P. Mach Learn (1992) 8: 279. https://doi.org/10.1007/BF00992698”