WHO YOU'LL BUMP INTO AT A HUBWAY STATION

Jeffrey Ericson | Group #58 | December 10, 2013
CS224W Project Final Paper

I. Introduction

In 2011, Boston became one of the first cities in the United States to launch a bike-sharing program. Because public transportation by bus and subway can be slow in Boston, and driving even worse at times, bikes were thought to be a good alternative. One of the fun parts about taking public transportation is seeing who will be riding with you. Oftentimes you'll learn more about some topic just from overhearing a conversation about it, or from making small talk with those around you. Sometimes you might bump into someone fairly frequently on your rides, being able to identify them as "the guy with the moustache" or "the yoga girl", or maybe you might bump into a familiar crowd, like "the freshmen at MIT" or "the young parents with their baby strollers". In fact, some people (let's call them A and B) have entirely different opinions about other riders simply because person A might always be stuck on busses with elderly people that take awhile to pay their far, whereas person B might always be riding with college kids. Is there really something to this idea – that a person might have a collection of people that they tend to bump into more than others? Or does it just seem that way, and despite the fact that you might be living, say, in an area with more college students, you actually are not more likely to bump into them than other types of people?

Ideally, subway or bus riding data would answer this question best because you could see which people bump into others *during* the course of your ride. But since this is not available, a decent substitute is data from the bike-sharing program in Boston, called Hubway. Instead of people bumping into each other on rides, people might bump into each other at stations. The question then is, do certain types of people (based on demographic information) actually see a certain demographic at a station more frequently than the usage at the station suggests, or not? If so, then two types of people who are not likely to see each other at that station will have different viewpoints as to what types of people are using the station. If not, then two people who use the station will likely have a similar viewpoint (as long as they use the station enough) as to what types of people are using the station.

Before I discuss the model and methods behind the project, let me first provide an overview of prior work on community detection in networks.

II. Literature Review

The papers I read all discussed communities, but from different angles. The first discussed a high-level overview and proposed a more efficient greedy algorithm to determine communities.

The second analyzed a trove of real-world datasets to compare and contrast what they observed with what previous studies or generative models had shown. The third was full of proofs and arguments to justify the difficulty of finding optimal communities, and how greedy approaches approximate successfully with real-world data despite having the potential to fall far short in worst-case scenarios. I will discuss each paper in turn and then follow with a synthesis.

[1] Finding Community Structure in Very Large Networks

Clauset et al. frames the discussion of networks in terms of *modularity*, which is a network-wide property and value that rewards within-community edges appearing more often than they would in a randomized network (that respects the vertex degrees, not just the number of nodes and edges present). This value is 0 if within-community edges do not appear more often, and in practice, a value of 0.3 denotes significant community structure. A detail that the paper does not address is what the baseline is for a randomized network. One could imagine different baselines that still respect the vertex degrees, depending on which generative graph you choose. I would imagine the assumption is a Gnm randomized generative graph, but one could also imagine a baseline against a small-world network or tree-inspired network.

In the explanation of their algorithm, they discuss how the sparse matrix (since they do not need to keep row/column combinations for two communities with no edge between them, since this will never increase Q, and thus never be chosen in the iterative algorithm) of ΔQ values (the change in Q values from joining two communities) has higher values for two nodes that are connected, but individually have small degrees. Intuitively, this makes sense as a way to build up larger communities from smaller communities, and indeed they discuss a hierarchy that can be represented with a dendrogram, though they do not include one because it would be so large. Still, it would be interesting to see this dendrogram for the communities that you eventually identify as the most important because they might reveal interesting substructure.

The authors describe how the value of Q steadily increases as you continue to join smaller communities into larger ones during the iterative algorithm. In their example, the maximum Q value for their Amazon dataset contains 1684 communities with an average size of 243. I wonder where the balance is between maximizing the Q value and maintaining communities that are small enough to individually analyze and verify as being meaningful according to additional contextual information in the data. Particularly in lieu of the next paper (that I am about to review) that argued that the best communities are usually 100 nodes large, I wonder if there is a better point to stop joining smaller communities together. A plot that I would have appreciated, to better depict community sizes, would have one that plotted the top k communities as a cumulative percentage of the network. The authors mentioned that the 10 largest communities accounted for 87% of the entire network, so I imagine the plot would have been one with a long tail.

[2] Statistical Properties of Community Structure in Large Social and Information Networks

Leskovec et al. used the *network community profile plot* extensively throughout the paper as a visual to compare community quality with community size. By analyzing a large corpus of real-world datasets, the authors came to some surprising conclusions. Repeated many times throughout the paper, one takeaway was that the best communities have about 100 nodes, and these usually are isolated communities with only one or two edges connected them to the rest of the network. Beyond 100 nodes, the quality of communities lessens. As they state, communities tend to "blend into" each other.

Important to note is that the authors ground their concept of "community quality" with a different metric than the first paper. Here, the authors refer to *conductance*, which is not a network-wide value, but a community-specific value. The conductance of a set of nodes, which in our case is usually a community, rewards edges that connect members within the community and punishes edges that extend out of the community. The higher the conductance, the worse the community quality is. Interesting to note here is that if a community has 100 nodes and is not very tightly connected, then 1 edge from it to the outside world will have a much more severe effect on its conductance than if that community was very tightly connected. The size of the community is less important than the number of bonds between community members. This seems to indicate that smaller, more completely connected networks without edges to other communities are better. This is what I achieved in my Hubway network analysis. One criticism I have of the paper is that they interchangeably describe lines in the community profile plots, as well as conductance scores, with "high" or "low" and "good" or "bad" and "increases" or "decreases". Because there is no consistent adjective, some parts were difficult to follow.

A topic that the paper touches on, but could have expanded more on, was the notion of community coherence. When building up larger communities from smaller communities, if two communities are both isolated from the main graph, but share one link to each other, then they could be combined into a single community with a low conductance score. This community thus has a "good" score despite the fact that there is little coherence between the two communities except for a single link between their sub-communities.

[3] On Modularity Clustering

Brandes et al. went into depth about the mathematical underpinnings of algorithmically determining optimal communities in a network. They use the words "communities" and "clusters" interchangeably. This seems to make some sense since the first paper was the only to discuss dendrograms, which are usually used to describe hierarchical clustering. Despite the bevy of formulas, lemmas, and derivations that complicated the flow of the paper, one of the main takeaways was that modularity maximization is an NP-hard problem, even if you choose to

restrict the number of total clusters. Hence, greedy approximation algorithms must be used for networks of any reasonable size. The paper does a good job at explaining the potential pitfalls of greedy approaches. If the greedy approach chooses one path of a tiebreaker over another, the result could be a suboptimal clustering by a factor of two. But the paper goes on to point out that in practice, greedy approximations are quite good. In fact, even if greedy approximations result in a slightly lower modularity score for the network, it can be true that the maximum modularity solution is not as faithful of a community representation as the greedy modularity solution when interpreted with respect to truths about communities (which can only be verified when they are known). The discussion on this was well articulated, but one complaint I have is one the visual representations. In three examples, the authors alternate between how they decide to visually encode the maximal solution: first with enclosing blocks, then by node shape, then again with enclosing blocks. There was no good reason for this.

The paper hints at the following, but unfortunately does not expand on it. Since there are multiple ways to cluster the network (and even many different paths the same algorithm could take, depending on whether it chooses tiebreakers randomly or not), almost all of which in practice result in similar clusterings that seem to be a fairly faithful representation, then does it make sense to individually score nodes within a community? For instance, if you saw that one community was slightly different on different algorithmic runs, but it always had the same 90% of nodes, could you assign these nodes a weight near 1, and other nodes a weight in proportion to how likely they would appear in said cluster? This might help disambiguate clusters that are fundamentally the same between algorithmic runs. It also might help someone who is trying to interpret the community structure figure out whether a "borderline" node is justified in being a member of a particular community.

[4] Weighted Graph Cuts without Eigenvectors: A Multilevel Approach

This paper talks about clustering graphs without using eigenvectors, instead by looking at the cut of a graph. The cut measures how many edges are between distinct sets of a graph. Most methods of looking for sets that have the best conductance score in a graph are tractable, and thus do not scale well, but this paper proposes a method that they implemented in C and can be used via MATLAB to k-cluster undirected networks with weighted edges.

III. Methods

The raw data was downloaded from hubwaydatachallenge [dot] org. The raw data provides among other variables: trip start time, start station, end time, end station, rider zip code, rider year of birth, and rider gender. To model people seeing each other at a particular station, I used a weighted, undirected graph for each station, and with 96 stations total, I wound up with 96 distinct graphs. Since Hubway aimed to make the data more anonymous, there is no rider ID

variable, but I modeled each distinct demographic by a unique tuple of rider zip code, year of birth, and gender (3291 unique tuples). Thus, each node represents a "type of person" and each edge represents when two different types of riders were at the same station within 10 minutes of each other. Note that this means self-loops are possible because while you can't meet yourself, you can surely bump into someone at a station of your same age, gender, and residence zipcode. Also, this means that each edge has a weight assigned to it, which is the count of times person type A bumped into person type B. I chose 10 minutes because the data records when a user has scanned their credit card to take out a bike, or when a user locks their bike back to a stand to return a bike, and since a user is at a station with some buffer time both before and after each of those events, it is very likely that a user will see another user if their timestamps were within 10 minutes at a station.

The data was pre-processed to remove irregular zip codes, strange trip duration times, and other common, abnormal artifacts. This left 349,016 trips: nearly 700,000 distinct time points where users could overlap. The main tenants of my approach include representing a null model, clustering, and scoring clusters by conductance and density. I will briefly discuss each and then show results.

One issue is that if a particular station has a skewed user population, for example, if 90% of the users were college-aged and only 10% were older, then of course you would expect there to be a strong cluster of college-aged user types in the network using the station at the same times. This would not entail anything about two different users seeing two different types of people at a station because a college-aged user and a non-college-aged user might on average still see the same demographic breakdown when they use the station: 90% college-aged, 10% not. However, if the cluster of college-aged user types was strong enough that there were fewer edges than expected connecting them to non-college-aged users, such that a college-aged user bumped into 98% of college-aged students at the station, and non-college-aged users bumped into 50%, then something beyond the proportion of underlying users is happening.

Thus, the null model I chose was rewiring of edges in proportion to the node frequencies at a station. Note that this null model is specific to a station and does not use any census data. The reason census data would not work is because census data is not detailed enough. There are often many Hubway stations within the same zip code and a zip code often will have parts that are more residential or sections with a university. Also, census data probably does not reflect the proportion of users across Hubway. There are few users under the age of 18 or over the age of 70, and each station has a different breakdown of common users. Thus, the safest way to normalize the data is by comparing networks using the real data to networks with edge rewiring, which was a method used by Leskovec, et. al[2].

To cluster each station network, I used a method that clusters based on minimum cut across the network. The implementation I used was Graclus, from the paper by Dhillon, et. al[4]. Since the goal is to determine if two different user types actually see different types of users at a station, it suffices to find two clusters in the network (with certain properties to be explained). If

two or more clusters exist that are separate from one another, and the same is not found in the null model, then there is evidence to answer our original question as yes. However, setting the requested number of clusters to 2 does not always yield the best clusters. To judge each cluster, I employed conductance, density, and connectivity:

$$Conductance(S) = \frac{\sum_{(i,j)\in E; i\in S \text{ and } j\notin S} weight(i,j)}{\sum_{(i,j)\in E; i\in S \text{ or } j\in S} weight(i,j)}$$

$$Density(S) = \frac{\sum_{(i,j)\in E; i\in S \text{ and } j\in S} weight(i,j)}{\sum_{i\in E; i\in S} 1}$$

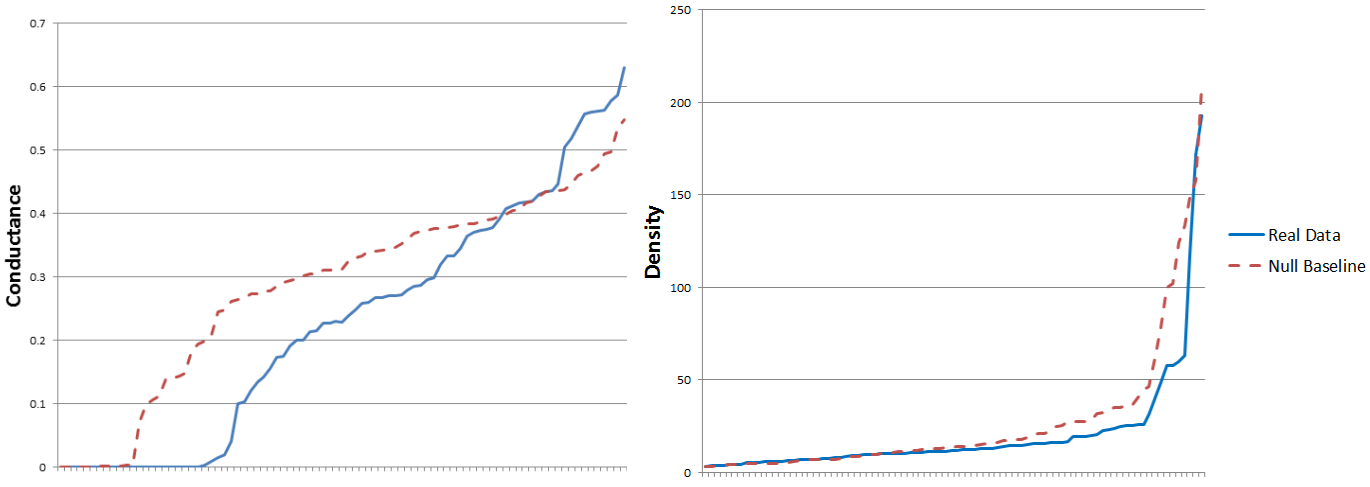
Connectivity was simply whether the cluster was a connected component or not.

Thus, I ran the clustering technique for k=2 to k=7 clusters (empirical testing suggested that k>7 did not yield nodes that scored as highly), resulting in 27 (2+3+4+5+6+7) sets of nodes. From these 27 sets of node, the best cluster chosen had to be a connected component and was one with the lowest conductance, but if below a conductance threshold of 0.3, the one with the highest density. Again, empirical testing suggested that 0.3 was a fair threshold to start prioritizing instead by highest density because a cluster with conductance < 0.3 looked fairly segregated from other nodes after visual inspection of the drawn graph.

The end result was the best cluster (by the above definition, which of course is not the only way to define what a "best" cluster is) for each of the 96 stations' null model network and real data network. How often did strong clusters show up, and how often did they show up for each model? The answer to those questions helps answer the original question.

IV. Results

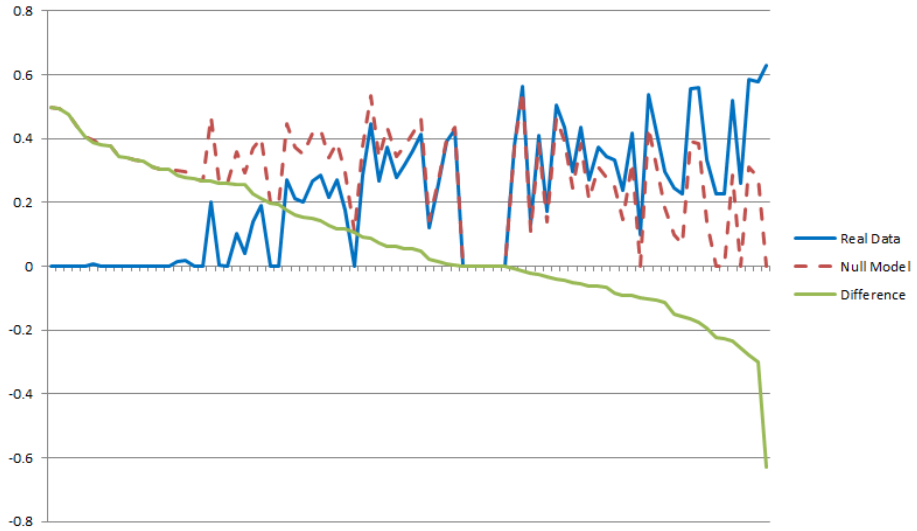
First, a comparison of the real model to the null model for each of the 96 Hubway stations is in order:



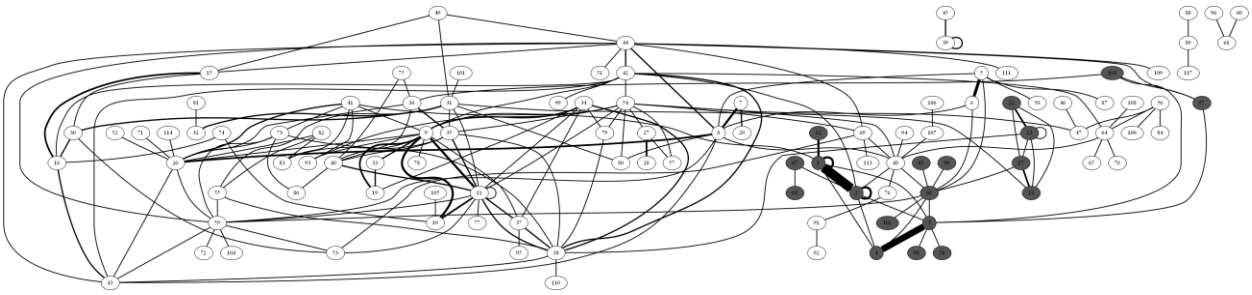
Overall, the real data produced best clusters that had lower conductance scores and lower density. The lower conductance scores indicates that the real data best clusters are more self-contained because the ratio of edge weights inside the cluster versus those than leave the cluster is lower. The lower density scores (plotted with the y-axis logged) indicate that nodes within the real data best clusters collectively have a lower out degree (or out weight) than those within clusters for the null model.

Notice also that many more stations in the real model had a best cluster with conductance score under 0.3, which is the cutoff I empirically chose where the algorithm transitions to then finding a cluster to maximize density. Thus, an overall sense is that the clusters that the real data is producing are clusters that are segregated from the rest of the network more so than clusters from the null model.

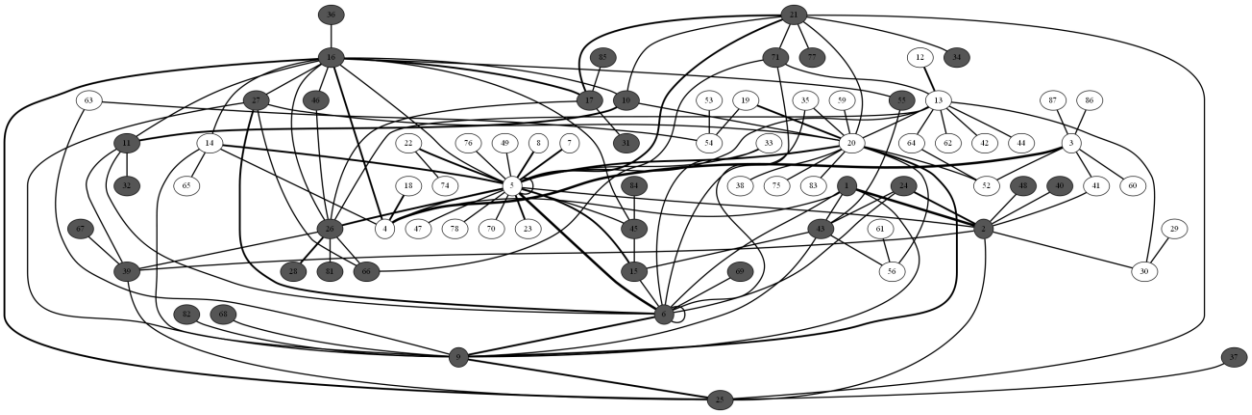
However, at a per-station level, the answer to our original question is "yes" for some stations and "no" for more stations. There are relatively few stations whose best cluster in the real data has a low conductance score that is also much lower than the best cluster in the null model. For each station, it is difficult to draw a line in the sand and determine what exactly makes a cluster from the real data "better enough" than the cluster from the null model that we can answer "yes" for the station, but we can rank the stations by difference in conductance scores to give a general idea of how much better the best clusters are in the real data. First, a plot that ranks stations by their conductance score differences (positive value means the real data has a lower score), and then some networks drawn that rank highly.

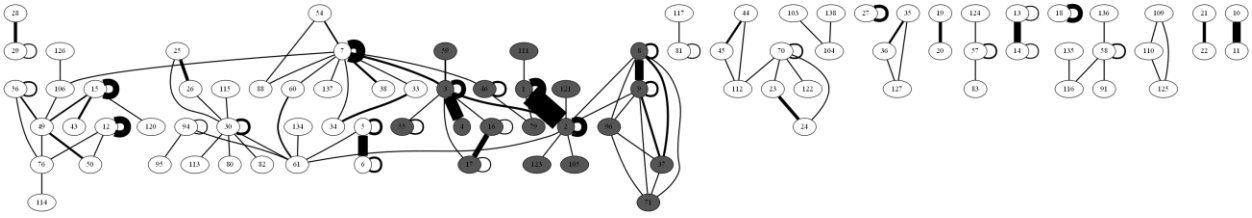


The following are network comparisons between the real data and null model for two different stations. The first is for station "One Broadway / Kendall Sq at Main St / 3rd St" and the second is for station "Harvard Real Estate - Brighton Mills - 370 Western Ave". Nodes shaded are members of the best cluster and edges have line thickness in proportion to their edge weight. Note that the node labels for the graphs are distinct for each.

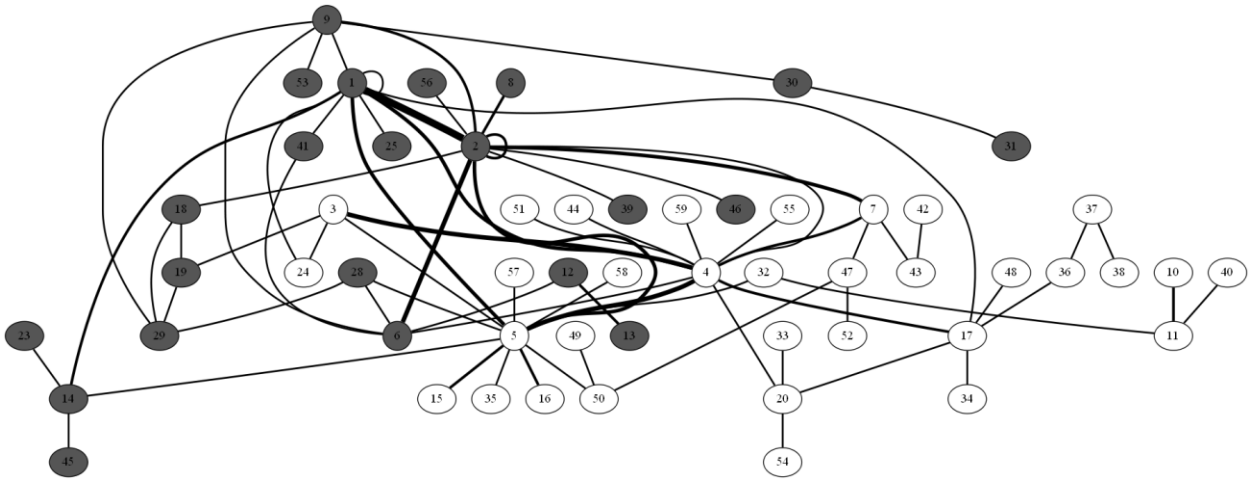


real network above, null baseline below

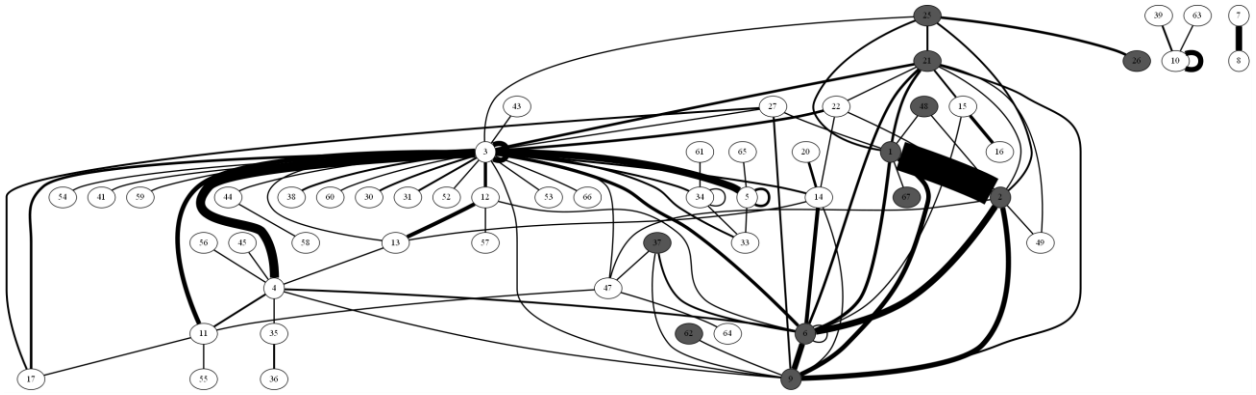




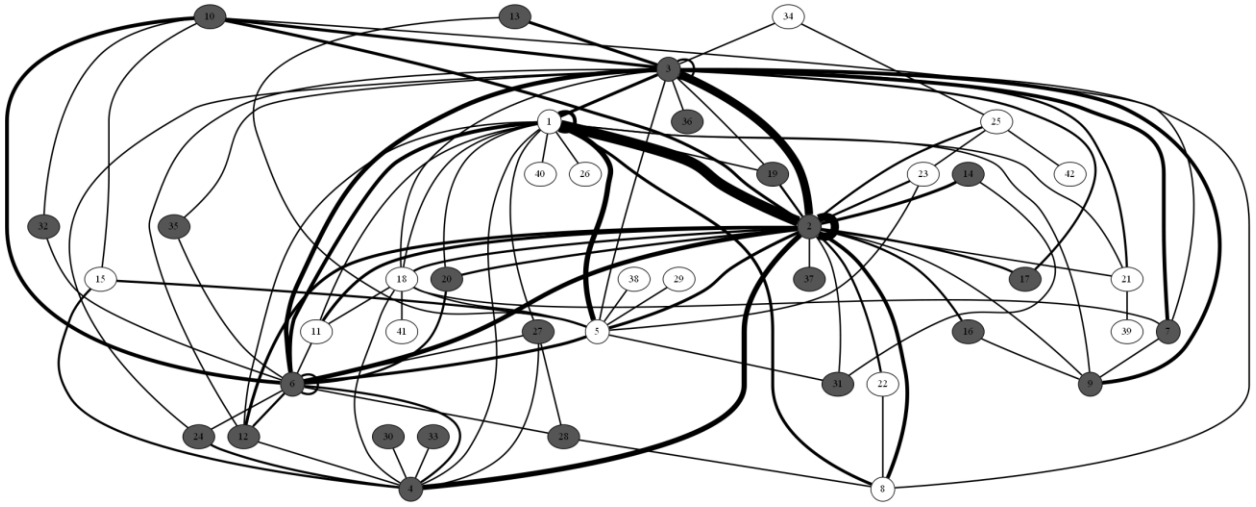
real network above, null baseline below



From inspecting each, it does seem like the network metrics of conductance, density, and connectivity have yielded the best clusters, and it also seems like the best clusters in the real networks are better than the best clusters in the null baseline networks. This suggests that for some of the stations, there are types of riders who are more likely to see a certain demographic of rider more so than other types of riders, even after accounting for the frequency of riders that use that station. However, for most stations, statistics don't differentiate the best cluster from the real data network enough from the null model to be confident in a "yes" answer. For example, here is a station where the real network is only slightly better statistics-wise:



real network above, null baseline below



Finally, let's compare which types of people are in each cluster for the real data versus the null model. For station "One Broadway / Kendall Sq at Main St / 3rd St," the left is the real data and the right is the null model:

```
02139 -- 19 -- M
02139 -- 20 -- M
02139 -- 21 -- F
01982 -- 21 -- M
02139 -- 23 -- M
02142 -- 23 -- M
```

```
02139 -- 19 -- M
02215 -- 19 -- M
02139 -- 20 -- M
02143 -- 22 -- M
02139 -- 22 -- F
02215 -- 23 -- M
```

```
02142 -- 24 -- F
02142 -- 26 -- M
02139 -- 27 -- F
02116 -- 27 -- M
02142 -- 27 -- M
02116 -- 28 -- F
02142 -- 28 -- F
02113 -- 38 -- F
02478 -- 39 -- M
02155 -- 41 -- M
01773 -- 43 -- M
01803 -- 44 -- M
02050 -- 59 -- M
```

```
02215 -- 24 -- F
02116 -- 24 -- F
02142 -- 24 -- F
02114 -- 24 -- F
02108 -- 25 -- M
02142 -- 26 -- M
02139 -- 26 -- M
02143 -- 26 -- M
02113 -- 27 -- M
02116 -- 27 -- M
02118 -- 27 -- M
02108 -- 28 -- M
02139 -- 29 -- M
01239 -- 31 -- F
02114 -- 31 -- F
02138 -- 31 -- M
02108 -- 32 -- M
02143 -- 33 -- M
02143 -- 34 -- M
02118 -- 38 -- M
02171 -- 39 -- M
01890 -- 39 -- F
02210 -- 43 -- M
02210 -- 46 -- F
02142 -- 46 -- M
02110 -- 47 -- M
01923 -- 48 -- M
01832 -- 50 -- F
02114 -- 52 -- F
02142 -- 54 -- M
01810 -- 58 -- M
```

And here is the same for station “Harvard Real Estate - Brighton Mills - 370 Western Ave”:

```
02115 -- 26 -- F
02115 -- 27 -- F
02134 -- 27 -- F
02138 -- 27 -- M
02134 -- 28 -- F
02134 -- 28 -- M
02115 -- 28 -- M
02163 -- 29 -- M
02129 -- 30 -- F
02134 -- 30 -- M
02129 -- 31 -- M
02215 -- 32 -- M
02446 -- 32 -- M
02116 -- 35 -- M
02138 -- 35 -- M
02043 -- 41 -- M
02144 -- 45 -- F
02238 -- 51 -- F
02116 -- 56 -- F
```

```
02135 -- 22 -- F
02139 -- 25 -- M
02215 -- 26 -- M
02109 -- 26 -- F
02115 -- 27 -- F
02163 -- 27 -- M
02478 -- 28 -- M
02135 -- 29 -- M
02114 -- 30 -- M
02114 -- 30 -- F
02134 -- 30 -- M
02148 -- 30 -- M
02138 -- 32 -- F
02446 -- 32 -- M
02215 -- 32 -- F
02127 -- 34 -- M
02116 -- 35 -- M
02138 -- 35 -- M
02238 -- 39 -- M
02108 -- 40 -- M
02468 -- 43 -- F
02118 -- 52 – F
```

The demographics shows that there is an overrepresentation of some demographic tuples in the best cluster for the real data than one would expect from simply rewiring the edges. In

general, there is evidence that for some stations, the answer to our original question is “yes”. Namely, certain types of people (based on demographic information) do see a certain demographic at a station more frequently than the usage at the station suggests, but we can only be reasonably confident that this happens at a few of the 96 stations.

V. References

[1] A. Clauset, M.E.J. Newman, C. Moore. Finding community structure in very large networks. Phys. Rev. E 70, 066111, 2004

[2] J. Leskovec, K. Lang, A. Dasgupta, M. Mahoney. Statistical Properties of Community Structure in Large Social and Information Networks. In Proc. WWW, 2008.

[3] U. Brandes, D. Delling, M. Gaertler, R. Goerke, M. Hoefer, Z. Nikoloski, D. Wagner. On Modularity Clustering. IEEE TKDE, 2007.

[4] I. Dhillon, Y. Guan, B. Kulis. Weighted Graph Cuts without Eigenvectors: A Multilevel Approach. IEEE PAMI, 2007.