Data

This is my egocentric Facebook network. The data were downloaded with the NameGenWeb application. Each node represents a friend, and each link represents a connection between two of my friends. The network doesn’t show my own node as it is already known that it would be connected to every other node. Thus this network is of a level 1.5, without my personal node. The network contains 226 nodes and 2326 unweighted edges without self-loops, resulting in a graph density of 9.14%.

I should connect the disconnected friends

I started by importing the graphml file into NodeXL. Then I chose to group by the Clauset-Newman-Moore clustering algorithm. I then selected the Fruchterman-Reingo layout with each group in its own box. This resulted in 3 big clusters, 1 medium, 2 small and a box of disconnected nodes. This initial clustering makes sense. In fact, it is shown that there is a cluster of friends that I made while I was pursuing my studies and then working in Rabat (Morocco). Those are connected to a cluster of classmates that I met in my home town Tetouan. The connection between these two clusters is not surprising given that most engineering students from Tetouan pursue their studies in Rabat. These two clusters are also connected to a group consisting of my big family, which resides in Tetouan. In several cases, the outliers of these 3 groups are friends who moved to a different city or country, which weakened their connections with mutual friends. The medium-size cluster represents a network of Fulbright alumni that I met during a 3 weeks camp.

What I find interesting, though, is the group of 11 independent nodes. In fact I know 6 of them as volunteers in a cultural website that I maintain. This is the first time that it comes to my attention that I’ve never introduced them to each other. Even more, 3 of them are from the same country. I should definitely make these connections in the sake of the success of the website. In addition, there is a gap between the cluster of friends, and that of Fulbright alumni. In fact I know several colleagues who should be connecting them. But I just don’t have them as friends in Facebook yet.

-------------

1 https://apps.facebook.com/namegenweb/
It's all about my wife, roommate and cousin

As I live far away from my family and friends, I rely mainly on virtual connections to keep in touch with them. Nevertheless, sometimes a physical connection is needed. In this case I would like to find the minimum number of people that I can contact virtually, but who can interact with most of my relatives physically. A simple idea, which I refer to as Method 1, is to select a node with the highest degree, then select the next one, and so on. However, it can be the case that the first selected nodes share about the same mutual friends. Thus, I may not gain additional contacts in an early stage while navigating through an ordered list of high degree nodes. An alternative way (Method 2) is to find high degree nodes, but only one in every cluster. Here I’m only interested in my subnetwork of Moroccans, which is already clustered into three groups. NodeXL computes the degrees with the graph metrics. Then I use dynamic filters and set the minimum degree to the maximum possible value. This results in only one node shown in the graph, which I select. Next I keep decreasing the minimum degree until one node in a different cluster appears. I repeat this process until selecting three nodes, one from each cluster (Figure 2).
Once the seed of 3 nodes is chosen, I select the subgraphs connected to them, with a level of adjacent vertices equal to 1.0. Figure 3 shows, for each of the two methods, the seeding nodes as black triangles, and the connected nodes as red disks. It is clear that Method 2 is selecting much more nodes (most of the nodes in the three clusters I’m interested in). Interestingly, the seeds represent my wife (K.B.), a one 1-year roommate who was also a classmate for 5 years (A.C.), and a cousin (A.M.) with whom I share many friends. I note that since my wife lives with me in the US, I looked for the following high degree node in the same cluster, which was her cousin (S.B.). The reachable nodes did not change much.
Eigenvector centrality is just node degree

The textbook in section 3.5.2 compares eigenvector to degree centralities. While the former measures “strategically connected people”, the latter doesn’t differentiate a key player from a “high school dropout”. In order to verify if this difference holds in this particular dataset, I auto-fill the Vertices X and Y by the degree and eigenvector centralities respectively. The layout, after hiding the links, takes the shape of a scatterplot, with the sizes of the disks being proportional to the total friend count of a particular node (including friends that do not appear in my network). The colors reflect the clusters introduced previously. Figure 4 shows the graph resulting from this manipulation.

![Figure 4 - Degree and eigenvector centralities](image)

It appears from this plot that, within a given cluster, there is a high correlation between the degree and eigenvector centralities. It is also worth mentioning that the cluster of classmates (dark green) has a small slope. This can be explained by the fact that this cluster is less connected to the two other ones. Thus, nodes belonging to it have lower strategic impact on the overall network. I also note that most of the big nodes (i.e. those who have a high number of friends) belong to the cluster of friends residing in Rabat (blue). Not surprisingly, people living in the capital city Rabat are more likely to have more connections than those living in a smaller town (Tetouan).
Critique of NodeXL

During this experience of network analysis, NodeXL has shown several positive aspects. The main strength is the availability and accessibility of a large “bag of tricks”. The clustering algorithms, the different layout possibilities, the data aggregators from popular social networks, and the ability to auto-fill numerous attributes are useful features that enriched the experience. In addition, the integration of the software within Excel accelerated the learning curve. User feedback based on highlighting selected nodes and/or edges was intuitive and helped to concentrate on the tasks.

On the other hand, there is still need to overcome some problems. In fact, I tried to get some tweets related to 4 different hashtags (#FirstTimeVoter, #Question6, #Question7 and #Sandy). After waiting for several hours I’ve got less than 200 tweets for each hashtag, with a sparse network. I had enough time to wait for few additional days and get more interesting data, but NodeXL just didn’t return enough tweets that I could make good use of. In addition, a reproducible crash happens whenever I try to change the shape or the color of a selected vertex (Figure 5). By looking at last year’s reports, I see that this is an old bug that hasn’t been solved yet.

Some additional features may increase the number of insights that one can get within the initial interaction with the data. It will be helpful to provide statistics per cluster, such as the density of a group of nodes. Right now the automated computations take place only at the vertex or at the network level. Some in-between metrics at the group level will be interesting to see. In the Vertex Color Options within the Autofill Columns functionality, it is not possible to specify desirable colors for categorical data. By adding such a feature, one can easily color males with blue and females with pink for instance.

Finally, I find that I learned a lot about the data in a reasonable amount of time. Addressing the issues raised in this section should improve both the user experience and performance.