Making Sense of Social Graphs: My Facebook Graph De-construction with Gephi

Introduction and Import
When we think of large networks we sometimes think of the following figure:

The picture above is my Facebook friend connections grabbed from https://tools.digitalmethods.net/netvizz/facebook/netvizz/ and visualized in Gephi and it does not have a particularly huge amount of nodes. The graph is represented by 432 nodes and 5066 edges, making it pretty sparsely connected overall, with a density=2E/V(V-1)=.05. The nodes represent my friends in my Facebook network and the edges represent which nodes are friends with other nodes. Even if I ventured to tell you that the size and color of the nodes in the above figure represent the degree of the nodes, you probably will not be able to tell me anything of interest about this network. All hope is not lost however, after doing a few steps we can start to identify interesting characteristics in the graph structure. These steps include import, layout, and partition data followed by filtering and analysis.

Layout
Since we have accomplished the import stage we move to the layout stage. This is an important step in order to get a global view of our data. First we apply a force directed layout to identify communities, in which we can see three distinct groups:

The three distinct groups represent the Army, High School, and Undergraduate College. This is no surprise to me since these were the groups that I was most interested in at the peak of my Facebook use.

Next we space out the nodes a little with Fruchterman-Reingold (FR). I found this technique to be useful in Gephi since most of the layouts would update over a period of time. This allowed me to halt the algorithm when the screen was the best for me. Below you can see the result is a combination of FR and Force-Atlas:

As I mentioned earlier, I noticed three distinct connected networks forming. The one that I will further investigate in this paper is my Army network.
Partition

The partition step filters the data, though I make the distinction between filtering here we are interested in isolating a group for further analysis. The filter that I explain later pertains to each group individually instead of over all groups collectively, such as filtering by edge connectivity. Below I delete nodes from the other groups, high school and college, in favor of the Army community:

Ok, I will admit, at this point we still have an unintelligible graph, but now we can start our exploration that leads to three headlines immediately.
**Filter and Analysis**

In the military the hierarchy that we will be concerned with is platoon, battery, and battalion. The platoon being the smallest entity and the battalion being the largest entity in our case. In Air Defense, every battery has one or two warrant officers. Technically, these officers are considered to be lower rank than general incoming officers, but higher rank than Soldiers (enlisted and Non-commissioned officers). They are highly specialized experts and trainers in their career fields and they generally have had at least ten years of military experience.

Once we filter the Army graph by the nodes which are the most connected we are left with eight people, seven of which are Warrant Officers and the eight of which served as the battalion S3 (involved in all military operations for the battalion) and the battalion XO (second in command).
Headline 1: Warrant Officers are Very Well Connected and May be the Key to Operational Success

By filtering the graph by edge connectivity we are left with the most well connected individuals in my Army network. This immediately identifies the highest ranking person in my network and warrant officers. Coming from the military, this analysis seems spot on. Warrant Officers are the first to work long hours and help others out, which increases their network and potentially their influence. Being close to the Warrant Officers may enhance your capability as an Officer due to their connections and further lead to successful operations.

Headline 2: Warrant Officers have as much influence as a Major

Sometimes influence in the Army is more important than the rank you have. Warrant officers can typically have more years in the military than Majors, Colonels, and Generals. When someone has been in the military for that period of time they usher a great deal of respect. After filtering by edge weight we now know that Warrant Officers are not only highly respected, but they are also highly connected. Concluding that the Warant Officer commands as much influence if not more than that of a Major in the US Army.

Headline 3: Exploring Social Communities in Army networks may reveal structural details of the Organization

One of the nice features of Gephi is that many of the graph algorithms that they provide are adapted from papers. Specifically, I was able to define communities in my network with an adapted paper *Fast unfolding of communities in large networks* published in the Journal of Statistical Mechanics. Below you can see that the metric discovers four different modularity classes, only three of which apply to my current graph. The blue modularity class is the most striking in this analysis. Out of the entire group the blue class successfully captures 100% of Bravo Battery Soldiers, with one data point being a wife of one of the Soldiers.
Headline 4: Army Batteries are very well connected

On a scale of zero to one, the graph density of this network is .69, which is in direct contrast of our .05 that we obtained in our first graph. In a group of 21 people the average connectivity is more than four people per node. Patriot Batteries spend a lot of time in the field together and their success is largely determined by their cohesiveness. This confirms my hypothesis that the close connectivity will also be recognizable in the social graph.
**Gephi Review**

I have mentioned a few things already that make Gephi a breeze to work with. The features that I discussed include the ability to stop a graph mid-layout in order to use a combination of the layout algorithms and built in Graph algorithms that were adapted from published works. Gephi did, however, suffer some serious drawbacks, most notably constant crashing when trying to view large networks or perform computations on the graph. I did also notice that NodeXL has many import options, where Gephi assumes that you have a graph network ready to analyze.

Gephi’s interface reminds me of an attempt at re-creating a Spotfire-Like tool. My suggestions would be to continue along the path as many of the Spotfire features would be useful in the graph context. Most notably the ability to click a node and highlight all of its connections is nice, but would be much improved with a details on demand section. Furthermore, Gephi would benefit from node hiding options to reduce the size of a network instead of having to delete the data all together.