Class Network - Network Science

Gabriel Luciano Gomes (265673)

November 23, 2020

Database description

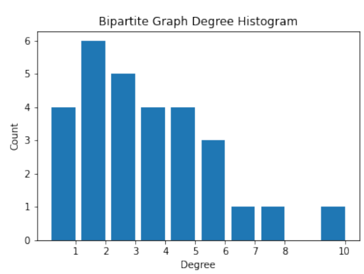
We asked the 15 students in the class to tell us their hobbies. A total of 14 distinct hobbies were mentioned, and led to the network given in the accompanying file class-network.tsv. This is a bipartite network, because each link has a student in one extremity and a hobbie in the other. Both students and hobbies were anonymized, so students are identified by codes S1 to S15, whereas hobbies are identified by codes H1 to H14. The file has not been sorted.

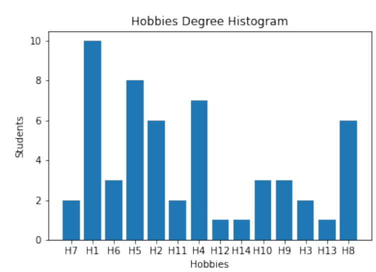
Analysis

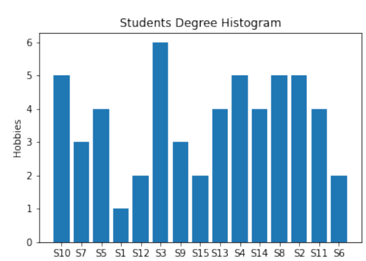
As described at the description, a bipartite graph was created, where students form a set and hobbies the other one. Since each student has one or more hobbies and each hobby is related to one or more students, an undirected network is formed. Additionally, the whole graph has 29 nodes and 55 links, the hobbies projection has 14 nodes and 50 links, and the students projection has 15 nodes and 83 links. Based on this, we're going to analyse the following items.

Analysing graph as a whole:

(A) Degree distribution: Since the graph has two sets, we need to consider both of them to get the degree distribution. Moreover, the distribution is based on node's degree of each set. So we have the following plot.







(B) Average Degree: To compute the average degree, we can use the undirected network formula, defined by:

$$\langle k \rangle = \frac{2L}{N}$$

We also know that L is the number of links of this network and N is the number of nodes. Thus, we have:

$$\langle k \rangle = \frac{2 \times 55}{29} = \frac{110}{29} \approx 3.79$$

The same result is obtained with the code:

```
[IN]  print(nx.info(bipartiteGraph))
[OUT] Name:
[OUT] Type: Graph
[OUT] Number of nodes: 29
[OUT] Number of edges: 55
[OUT] Average degree: 3.7931
```

(C) Connected Components: Connected components are related for the number of subgraph within the network. Moreover, the number of components was analysed and only one was found. It is because all the links in the network is connected to a node that has at least one connection with a component. As a proof, we can calculate the size of this component, and if its size is equal to the number of nodes, so the premise is correct.
With the following piece of code we can verify it.

```
[IN]  cc = nx.connected_components(G)
[IN]  ncc = nx.number_connected_components(G)
[IN]  sortedComponents = sorted(cc, key=len, reverse=True)
[IN]  gc = G.subgraph(sortedComponents[0])
[IN]  print("Number of connected components:", ncc)
[IN]  print("Size of the giant component:", gc.number_of_nodes())
[OUT] Number of connected components: 1
[OUT] Size of the giant component: 29
```

(D) Average Distance: From the previous item, we got that the network is formed by one component. Furthermore, we can compute the average distance among nodes in the graph.
The following code was used to compute it.

```
[IN]  ad = nx.average_shortest_path_length(G)
[IN]  print("Average distance within the graph:", ad)
[OUT] Average distance within the graph: 2.6995073891625614
```

(E) Clustering coefficient: As described, we're working with a bipartite networks. So hobbies nodes do not link among themselves, likewise for the students nodes. Therefore, the cluster coefficient will always be zero. Furthermore, the following formula can compute the clustering coefficient for the whole graph:

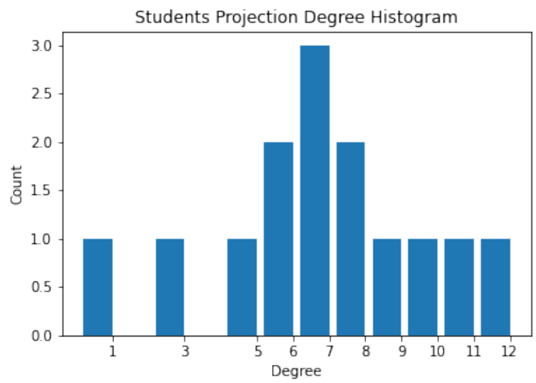
$$\begin{aligned} \langle C \rangle &= \frac{1}{N} \sum_{v \in G} c_v \\ &= \frac{1}{N} \times 0 = 0 \end{aligned}$$

The following piece of code also confirms it:

```
[IN]  cc = nx.average_clustering(G)
[IN]  print("Graph Average clustering coef:", cc)
[OUT] Graph Average clustering coef: 0.0
```

Analysing the students projection

(A) Degree distribution: Considering only the students projection, we have the following distribution:



(B) Average Degree: To analyse the average degree for this set, we will base on the same formula described at item B for the graph as a whole.

$$\langle k \rangle = \frac{2L}{N}$$

We also know that L is the number of links of this projection and N is the number of nodes. Thus, we have:

$$\langle k_{students} \rangle = \frac{L}{N} = \frac{2 \times 83}{15} \approx 11.0667$$

The same result is obtained with the code:

```
[IN]  print(nx.info(bipartiteGraph))
[OUT] Name:
[OUT] Type: Graph
[OUT] Number of nodes: 15
[OUT] Number of edges: 83
[OUT] Average degree: 11.0667
```

(C) Connected components and Average Distance: As defined previously we have:

Number of components: 1
Size of the giant component: 15
Average distance: 1.2095238095238094

(D) Clustering coefficient: Similar to previous definition, but considering only one projection (X). To do so, we divide by the size (number of nodes) of the analysed projection, not the size of the whole graph. Thus, we have:

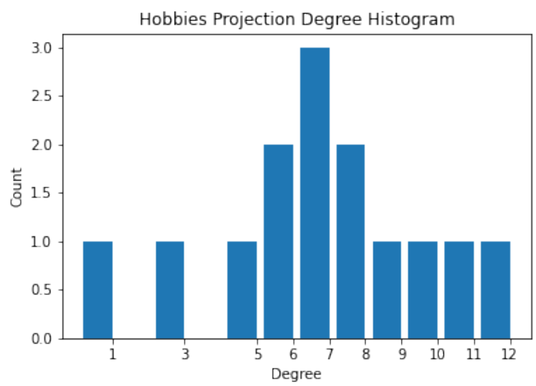
$$\langle C_X \rangle = \frac{1}{|X|} \sum_{v \in G} c_v$$

The following code computes the coeficient for us:

```
[IN]  cc = nx.average_clustering(G)
[IN]  print("Graph Average clustering coef:", cc)
[OUT] Graph Average clustering coef: 0.8891094091094093
```

Analysing the hobbies projection

(A) Degree distribution: Considering only the hobbie projection, we have the following distribution:



(B) Average Degree: To analyse the average degree for this set, we will base on the same formula described at item B for the graph as a whole.

$$\langle k \rangle = \frac{2L}{N}$$

We also know that L is the number of links of this projection and N is the number of nodes. Thus, we have:

$$\langle k_{hobbies} \rangle = \frac{L}{N} = \frac{2 \times 50}{14} = 7.1429$$

The same result is obtained with the code:

```
[IN]  print(nx.info(bipartiteGraph))
[OUT] Name:
[OUT] Type: Graph
[OUT] Number of nodes: 14
[OUT] Number of edges: 50
[OUT] Average degree: 7.1429
```

(C) Connected components and Average Distance: As defined previously we have:

Number of components: 1
Size of the giant component: 14
Average distance: 1.5054945054945055

(D) Clustering coefficient: Similar to previous definition, but considering only one projection (X). To do so, we divide by the size (number of nodes) of the analysed projection, not the size of the whole graph. Thus, we have:

$$\langle C_X \rangle = \frac{1}{|X|} \sum_{v \in G} c_v$$

The following code computes the coeficient for us:

```
[IN]  cc = nx.average_clustering(G)
[IN]  print("Graph Average clustering coef:", cc)
[OUT] Graph Average clustering coef: 0.7197381983096268
```

Plots

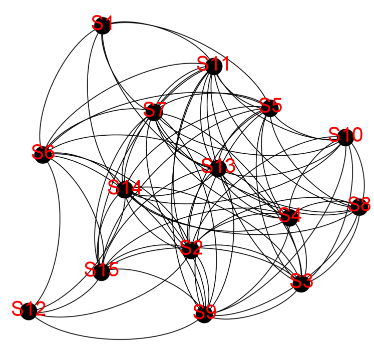


Figure 1: Students Projection

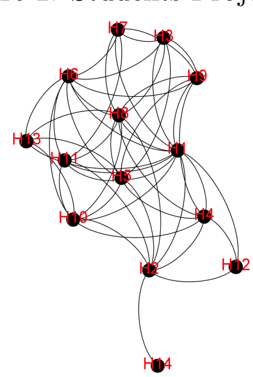


Figure 2: Hobbies Projection

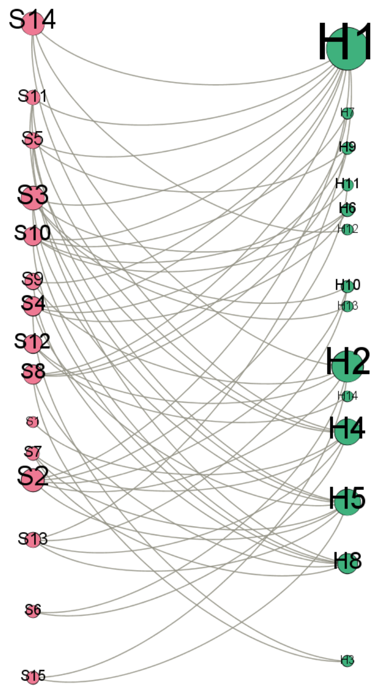


Figure 3: Graph plotted with Gephi