# Final exam

Fall 2021

**Analysis of Network Data** Wedensday 12/07

Please do not turn the page until instructed to do so.

This exam has 4 questions. You are expected to pick 3 to work on and submit.

Every question you pick will be equally weighted.
The exam is **a take-home exam**.

It is due by end-of-day on Friday 12/16.

You are **not** allowed to collaborate with other students in the class.

You are **though** allowed to ask me questions and request hints.

Answer every question you pick to the best of your knowledge.

My office hours for the exam this week will be:

- Online office hours by request on Zoom.
- Or during the following days on Zoom:
  - Friday 12/09, Monday 12/12, and Friday 12/16 between 1pm and 3pm.

Make sure to show your work for **partial credit**.

### Question 1: A sensor network protocol

A network $G(V, E)$ consists of sensors ($V$) that can serve as data collectors. Some of the sensors can communicate with other sensors in their vicinity ($E$). Every time a sensor collects data it passes the data on to the other sensors in the network. All sensors can send data they have collected to at least one neighbor. However, to minimize battery consumption, only a subset of the sensors is allowed to relay data that other sensors have collected. Let this set be called $D$: it has the following two properties:

1. any sensor in $D$ can send data to another sensor in $D$ using only intermediary nodes in $D$; and
2. any sensor in the network is either in $D$ or adjacent to a sensor in $D$.

Answer the following questions.

(a) Formulate the problem of finding the set of sensors $D$ with minimum cardinality $|D|$.

(b) A related problem can be stated as follows. Find a spanning tree $T$ with the maximum number of leafs. We define a leaf as a node that only has one edge incident to it in the tree. Formulate this spanning tree extension as an integer program.

(c) Let $|D|$ the cardinality of the optimal set of part (a) and $|L|$ be the number of leafs from the optimal solution of part (b). Show that $|D| + |L| = |V|$.

### Question 2: Conference games

You are given a list of universities and the games they have played against each other during the athletic season of 2019: this is provided in file "games.csv". To read the file, simply use `networkx`'s read_edgelist functionality as

`G=nx.read_edgelist('games.csv', data=False, delimiter=',').`

Then, create a *final* graph by iteratively trimming any university that appears in less than 10 games. Hence, in the end, your graph should have all nodes with a degree 10 or more. This will allow you to take all universities participating in a conference with other universities. Answer the following questions.

(a) Use `networkx` to identify clusters by:

1. iterative bipartitioning based on the Fiedler vector. That is, first find the number of expected clusters (recall the methodology to do so looking at the "jumps"), and then iteratively bipartition them until getting the number of clusters you are looking for.
2. the Girvan-Newman method. Again use the number of expected clusters from the previous approach. Then use the (simple) Girvan-Newman method to identify all clusters.
3. Louvain clustering (modularity maximization). For this one you do not need an expected number of clusters. Does the number from (a) and (b) match up with the number of clusters you get here?

Based on your answers and coding, compare your obtained clusters to the athletic conferences: you may find them here https://en.wikipedia.org/wiki/List_of_NCAA_conferences. Which one achieves a higher accuracy rate? Define accuracy as the number of correctly paired universities versus the number of total pairs.

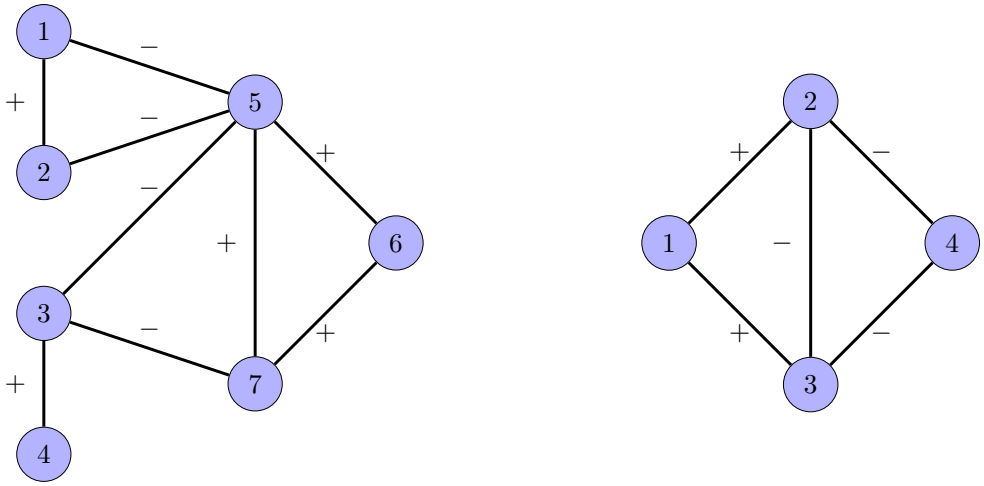
(b) Propose one method to identify **overlapping clusters**: that is, to get clusters that may share one or more nodes. Provide an application or example for this extension.

### Question 3: Signed graphs and social balance

"The enemy of my enemy is my friend"

The above phrase can be viewed as the fundamental idea behind **signed graphs**. Let us define a signed graph as one where the edges have a weight $w_{ij} \in \{-1, +1\}$. You may view this as $(i, j) \in E$ have a positive relationship when weight $w_{ij} = +1$, or a negative relationship when weight $w_{ij} = -1$. We say that a signed graph is **balanced** if all cycles have an even number of negative weights – equivalently, if all cycles have a product of weights that is positive. An example is provided in Figure 2.

Figure 1: An example of a balanced signed graph (left) and an imbalanced signed graph (right).




(a) Can an all-negative graph (that is, a signed graph where all edges have a $-1$ weight) be balanced? What type of graph can it be?

(b) Prove the following statement: a signed graph is balanced if and only if there exists a bipartition of the graph into two sets $A$ and $B$ such that nodes in $A$ are connected only by positive edges, nodes in $B$ are connected only by positive edges, and every edge connecting nodes in $A$ to nodes in $B$ is negative.

(c) Assume that we have been given some signed graph and we are interested in making it balanced. We can do that by deleting edges: for example, deleting $(2, 3)$ in the right graph of Figure 2 will make the graph balanced. Formulate mathematically the problem of finding the minimum number of edges to remove to render any signed graph balanced.

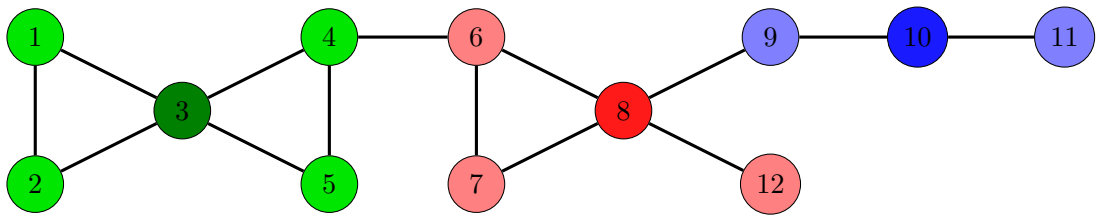
### Question 4: Spectral clustering and special structures

(a) In a social network, we would like to identify "leaders". Here, we define a leader as an entity with the highest betweenness among its peers. More specifically, let $b_i$ be the highest betweenness in the network (corresponding to node $i$) and let $b_j$ be the second highest betweenness (corresponding to some node $j$): then, if $b_i - b_j \geq 0.4$, we say that node $i$ is the leader of the network.

Consider the following approach. First, check whether a leader exists in the whole network. If not, then use spectral clustering to obtain a bipartition, and check whether a leader exists in the two induced subgraphs of the bipartition. If there exists a leader, report it; otherwise perform spectral clustering on each of the partitions and continue. We terminate when all leaders have been found or whether an induced subgraph has only 2 nodes.

If a leader exists in the whole network, then report that node as the leader; otherwise, if there is a leader in each of the two partitions obtained by spectral clustering, report these two nodes as the leaders. However, these are not the only possibilities. If a leader does not exist in either partition, you take that partition and you use spectral clustering again, leading to more leaders. Hence, a network could potentially have multiple leaders. As an example consider the network of Figure 3.

Figure 2: The leaders of the network below would be 3, 8, and 10. First, we partition, the network using spectral clustering clustering in two sets: $\{1, 2, 3, 4, 5\}, \{6, 7, 8, 9, 10, 11, 12\}$. The first partition has a leader in node 3; the second does not, so it is further partitioned into $\{6, 7, 8, 12\}, \{9, 10, 11\}$. The first of these partitions has a leader in node 8 and the second in node 10.




**Implement the algorithm described above using `networkx`. What are the leaders of the les misérables network and what are the leaders for the karate club network?**

(b) In class, we discussed numerous clique and star relaxations, like the quasi-clique (where a fraction $\gamma \in [0, 1]$ of all edges in a structure needs to exist), the $k$-club (where all nodes in a structure need to be within a distance of $k$ from one another), etc. In a clique, all nodes need to be adjacent to every other node; what if we relaxed this property? What if we allowed all nodes to be connected to every other node but $k-1$ of them? That is, what if for any structure $S$, we want each node to be adjacent to at least $|S| - k$ nodes in the same structure?
**Formulate the problem of finding the maximum cardinality such structure as an integer program and solve it for the les misérables network. What is the structure of maximum cardinality you find?**

Good luck!