Homework #2

The homework does not require a lot of writing but may require some thinking. It does not require a lot of processing power but may require efficient programming. It accounts for 12.5% of the course grade. Any questions or comments should be posted on Piazza.

Submission details

This homework is due on **May 3rd** at 11:59pm. It must be submitted through (1) Gradescope (entry code **JKX87N**) and (2) eUcilnica. (1) Submission to Gradescope should include answers to all questions, each on a separate page, which may also demand pseudocode, proofs, tables, plots, diagrams and/or other. (2) Submission to eUcilnica should include this cover sheet with signed honor code and all the programming code used to complete the exercises (preferably in `.py` format). The homework is considered submitted only when *both* (1) and (2) have been submitted. Failing to include the honor code in the submission will result in **10% deduction**. Failing to submit all the developed code will result in **10% deduction**.

Honor code

Students are strongly encouraged to discuss the homework with other classmates and form study groups. But each student must then solve the homework by herself/himself without the help of others and should be able to redo the homework at a later time. In other words, students are encouraged to collaborate, but should not copy from one another. Referring to any solutions obtained from classmates, course books, previous years, found online, AI tools or other is considered an honor code violation. Also, stating any part of the solutions in class or on Piazza is considered an honor code violation.

Any violation of the honor code will be reported to the **faculty disciplinary committee** and vice dean for education.

SID: ______________________________

Full name: ____

Study group: ____

I understand and accept the honor code.

Signature: ______________________________

1 Where is SN100? (7.5%)

Figure 1 shows social network of bottlenose dolphins famously studied by Lusseau [LSB+03]. After the disappearance of a particular dolphin named SN100 during the experiment [LN04], the rest split into two groups shown by different node colors. Your task is to study whether network analysis could be utilized to detect this important role of SN100.

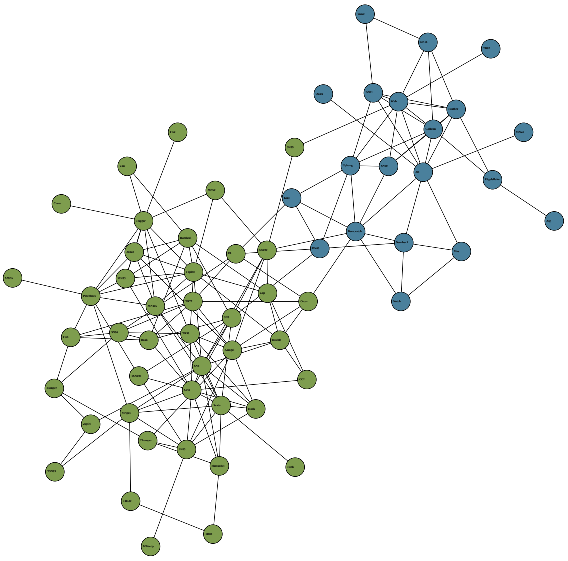


Figure 1: **Lusseau bottlenose dolphins network**

You can use any network analysis method, algorithm or technique. Describe how you measured the importance of SN100 and provide all the necessary results.

What to submit?

State your measure of importance and its motivation (2.5%). Give all the necessary results (2.5%). Provide a printout of all the code used to solve the exercise (2.5%).

2 HIV and network sampling (12.5%)

When a patient is diagnosed with HIV, in most Western countries, she/he will be questioned about past sexual contacts. The authorities would then make an effort to track down those contacts and test them for HIV. The process is repeated with anyone who also tests positive, tracing her/his contacts as well, until all leads have been exhausted. This process is called contact tracing.

Notice that contact tracing gives a (biased) sample of the underlying social network [LF06]. Assuming that one gets HIV from a random sexual contact, contact tracing can be approximated by a simple random walk. Simulate a random walk on small social network until you sample 15% of the nodes and take an *induced* graph on the sampled nodes for your sampled network.

Is the original social network small-world and/or seemingly scale-free? *(Does the network contain hubs?)* Is the sampled network small-world and/or seemingly scale-free? Could you reason why? Support your answers with the necessary computations.

What to submit?

Give brief answers to all three questions ($3 \times 2.5\%$). Support your reasoning with necessary results (2.5%). Provide a printout of all the code used to solve the exercise (2.5%).

3 Ring graph modularity (10%)

Consider a graph with n nodes positioned on a ring where each node is linked to its two nearest neighbors (Figure 2). Let the graph be partitioned into c consecutive clusters with $n_c = n/c$ nodes each. *(Assume n is divisible by c.)*

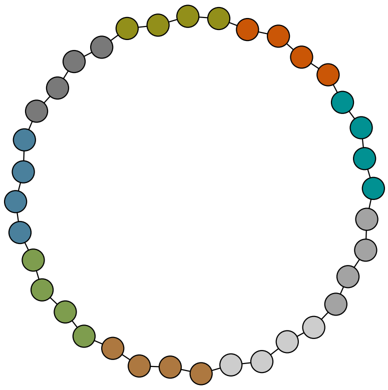


Figure 2: **Ring graph with $n = 36$, $n_c = 4$ and $Q = 0.64$**

First, derive the modularity Q [GN02] of such partition of a ring graph and express it in terms of only n and n_c. Next, find the size of clusters n_c that maximizes modularity Q of a ring graph and express it in terms of only n. Does it make sense to apply modularity optimization to a ring graph? For example, is the resulting partition unique?

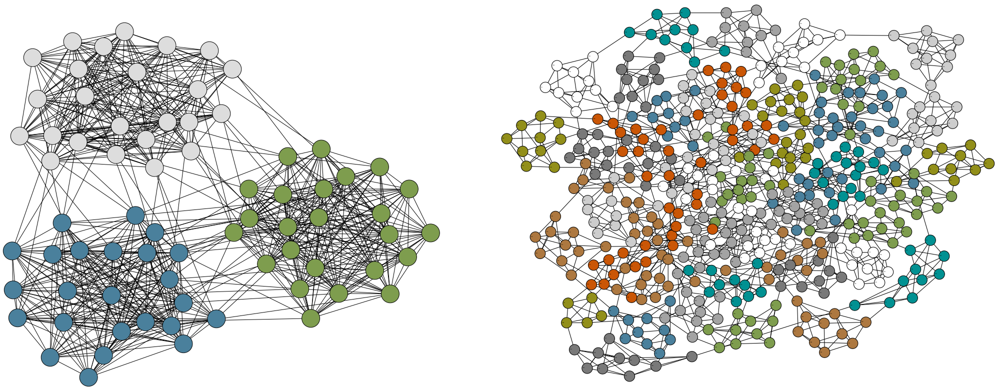
What to submit?

Derive an expression of Q in terms of n and n_c (4%). Find the optimal size of clusters n_c in terms of n (4%). Give brief answers to both questions (2%).

4 Who's the winner? (25%)

Community detection is a popular research area of network science [New12]. Indeed, literary hundreds of community detection algorithms have been proposed in the literature in the last two decades [FH16]. These include hierarchical clustering, spectral methods (e.g., Graclus), modularity optimization (e.g., Leiden and Louvain), map equation algorithms (e.g., Infomap), stochastic block models (e.g., (DC)SBM), statistical methods (e.g., OSLOM), link clustering (e.g., Links), label propagation (e.g., FLPA), random walks (e.g., Walktrap), clique percolation (e.g., SCP) and many others (e.g., BigClam, DEMON).

Your task is to compare the accuracy and robustness of three algorithms. These should include either Leiden or Louvain, FLPA and another algorithm of your own choice. *(If you are unable to compile the selected algorithms, search for an equivalent implementation within CDlib library.)*



(a) **Girvan-Newman graphs with $\mu = 0.1$** (b) **Lancichinetti graphs with $\mu = 0.1$**

Figure 3: **Benchmark graphs with planted partition**

(i) Implement a variant of Girvan-Newman benchmark graphs with planted partition [GN02]. The graphs should consist of three groups of 24 nodes each and the expected degree of each node should be 20 (Figure 3a). The group structure is controlled by a mixing parameter μ. For $\mu = 0$, all links are placed within the groups, while for $\mu = 1$, all links are placed between the groups.

Apply the community detection algorithms to 25 benchmark graph realizations with μ equal to 0, 0.1, 0.2, 0.3, 0.4 and 0.5. For each algorithm and each value of μ, compute normalized mutual information between the planted partition and detected community structure, and average the results. Plot community detection accuracy of the algorithms on a single plot with μ on the horizontal axis and normalized mutual information (NMI) on the vertical axis.

Which algorithm comes out on top? Briefly discuss the results by comparing the performance of the algorithms.

(ii) Consider more realistic Lancichinetti benchmark graphs with planted partition [LFR08]. The graphs consist of 2 500 nodes (Figure 3b), while the group structure is again controlled by a mixing parameter μ.

Apply the community detection algorithms to 25 benchmark graph realizations with μ equal to 0, 0.2, 0.4, 0.6 and 0.8. Plot community detection accuracy of the algorithms on a single plot with μ on the horizontal axis and normalized mutual information (NMI) on the vertical axis.

Which algorithm comes out on top now? Briefly discuss the results by comparing the performance of the algorithms.

(iii) Consider an Erdős-Rényi random graph that lacks community structure. Community detection algorithms should be robust enough to detect this and output each connected component of the graph as a separate community.

Apply the community detection algorithms to 25 random graph realizations with 1 000 nodes and the average node degree equal to 8, 16, 24, 32 and 40. Plot community detection robustness of the algorithms on a single plot with the average node degree on the horizontal axis and normalized variation of information (NVI) on the vertical axis.

Which algorithms are robust to random structure? Briefly discuss the results by comparing the robustness of the algorithms.

What to submit?

(i) Provide a printout of benchmark graph implementation (2.5%). Plot community detection accuracy of all three algorithms (3 × 2%). Briefly defend your answer to the question (1.5%).

(ii) Plot community detection accuracy of all three algorithms (3 × 2%). Briefly defend your answer to the question (1.5%).

(iii) Plot community detection robustness of all three algorithms (3 × 2%). Briefly defend your answer to the question (1.5%).

5 Get at least 70% right! (20%)

You are given a citation network between scientific papers published by the American Physical Society between the years 2008 and 2013. The papers were published in 10 different journals (e.g., *Physical Review E*) which represent the information you would like to infer from the structure of the citation network.

Your task is to predict the correct journal of all papers published in the year 2013 based on their citations and the journal information of papers published between the years 2008 and 2012. Predicting the paper's journal to be the most frequent journal in the neighborhood of the corresponding node gives $\approx 67\%$ classification accuracy, whereas your task is to propose a strategy that gives $\geq 70\%$ classification accuracy.

Your strategy can use any network analysis technique or other approach.

What to submit?

Describe your strategy and briefly explain its motivation (2 × 3%). State the average classification accuracy over ≥ 10 runs (8%). Compare your performance with the baseline $\approx 67\%$ (2%). Provide a printout of all the code used to solve the exercise (4%).

6 Peers, ties and the Internet (25%)

Link prediction is a common application of network analysis techniques. For given unlinked nodes i and j, link prediction methods try to compute an index s_{ij} that is high for i and j that are likely to link in the future, and low for other pairs of i and j. You will be investigating three link prediction methods that are based on different structural properties of real networks.

1. Scale-free degree distribution is believed to be a consequence of preferential attachment, which states that nodes are more likely to connect to high-degree nodes. The preferential attachment index [LNK07] is thus defined as $s_{ij} = k_i k_j$, where k_i is the degree of node i.

2. Small-world networks are characterized by an abundance of triangles, which can be explained by triadic closure in social networks. Therefore, nodes are more likely to connect if they share many neighbors. The Adamic-Adar index [AA03] also takes into account that it is more likely to share a high-degree neighbor. It is defined as $s_{ij} = \sum_{x \in \Gamma_i \cap \Gamma_j} \frac{1}{\log k_x}$, where Γ_i is the neighborhood of node i.

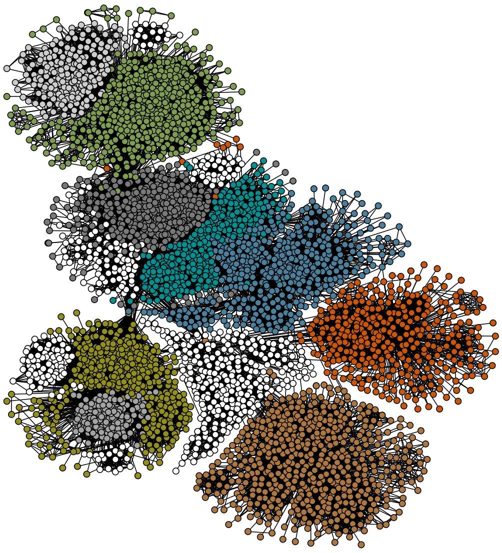


Figure 4: **Communities in Facebook social circles network**

3. Many real networks consist of communities of densely linked nodes with only a few links between the communities. Links are thus more likely to appear within communities rather than between communities. Let $\{c\}$ be the community structure revealed by Leiden modularity optimization algorithm [TWVE19] and let c_i be the community label of node i. *(If you are unable to compile the algorithm, search for an equivalent implementation within CDlib library.)* Furthermore, let n_c and m_c be the number of nodes and links within community c. Then, the community index is defined as $s_{ij} = m_c/\binom{n_c}{2}$ for $c_i = c_j = c$, whereas $s_{ij} = 0$ for $c_i \neq c_j$.

(x) Assume that you apply a link prediction method to all unlinked pairs of nodes of a large real network and later evaluate between which pairs of nodes the links occurred. Considering the density of real networks, what would be the expected classification accuracy of a method that simply predicts that no links will occur?

(y) Implement the following framework for evaluating link prediction methods. For a given network and link prediction index s, randomly sample $\frac{m}{10}$ pairs of nodes that are not linked, where m is the number of links in a network, and store them into L_N. These will serve as negative examples for the prediction. Next, randomly remove $\frac{m}{10}$ links from the network and store them into L_P. These will serve as positive examples for the prediction. Finally, compute the link prediction index s for all pairs of nodes in $L_N \cup L_P$.

Link prediction can be evaluated using the Area Under the ROC curve (AUC), which is defined as the probability that a randomly chosen pair of nodes from L_P has higher value of s than a randomly chosen pair of nodes from L_N. Note that random guessing gives 50%. To compute AUC, randomly sample $\frac{m}{10}$ pairs of nodes from L_P and $\frac{m}{10}$ pairs from L_N with repetitions, and compare their indices s. Let m' be the number of times when the value of s for the pair of nodes from L_P is larger than the value of s for the pair of nodes from L_N, and let m'' be the number of times when the values are equal. Then, AUC $= \frac{m'+m''/2}{m/10}$.

(z) Compute AUC over ≥ 10 runs for all three link prediction methods above applied to an Erdős-Rényi random graph with $n = 25\,000$ nodes and the average node degree $\langle k \rangle = 10$, and three real networks. These are Gnutella peer-to-peer file sharing network, a small sample of Facebook social circles network (Figure 4) and nec overlay map of the Internet. *(Represent all networks with an undirected graph.)*

Which method comes out on top for each graph/network? Could you reason why? Briefly discuss the performance of different methods by considering the structure of each graph/network.

What to submit?

(x) Give brief answer to the question (1%).

(y) Provide a printout of the framework implementation (4%).

(z) State AUC over ≥ 10 runs for each link prediction method and graph/network ($4 \times 3\%$). For each graph/network give answers to both questions and briefly comment on the results ($4 \times 2\%$).

References

[AA03] Lada A Adamic and Eytan Adar. Friends and neighbors on the Web. *Soc. Networks*, 25(3):211–230, 2003.

[FH16] Santo Fortunato and Darko Hric. Community detection in networks: A user guide. *Phys. Rep.*, 659:1–44, 2016.

[GN02] M. Girvan and M. E. J Newman. Community structure in social and biological networks. *P. Natl. Acad. Sci. USA*, 99(12):7821–7826, 2002.

[LF06] Jure Leskovec and Christos Faloutsos. Sampling from large graphs. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 631–636, Philadelphia, PA, USA, 2006.

[LFR08] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms. *Phys. Rev. E*, 78(4):046110, 2008.

[LN04] David Lusseau and M. E. J. Newman. Identifying the role that animals play in their social networks. *Proc. Biol. Sci.*, 271:S477–S481, 2004.

[LNK07] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *J. Am. Soc. Inf. Sci. Tec.*, 58(7):1019–1031, 2007.

[LSB$^+$03] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson. The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations. Can geographic isolation explain this unique trait? *Behav. Ecol. Sociobiol.*, 54(4):396–405, 2003.

[New12] M. E. J. Newman. Communities, modules and large-scale structure in networks. *Nat. Phys.*, 8(1):25–31, 2012.

[TWVE19] V. A. Traag, Ludo Waltman, and Nees Jan Van Eck. From Louvain to Leiden: Guaranteeing well-connected communities. *Sci. Rep.*, 9:5233, 2019.