

Homework #3

This homework is complete and will not be changed. 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 10% of the course grade. Any questions and comments regarding the homework should be directed to [Piazza](#).

Submission details

This homework is due on **April 22nd** at 11:59pm, while late days expire on **April 25th** at 11:00am. The homework must be submitted through (1) [Gradescope](#) (entry code **6PDZD3**) and (2) [eUcilmica](#). (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 other. (2) Submission to [eUcilmica](#) should include at least this cover sheet with signed honor code and all the programming code used to complete the exercises. The homework is considered submitted only when *both* parts have been submitted. Failing to include this honor code in the submission will result in **10% deduction**. Failing to submit all the developed code will result in **50% deduction**.

Honor code

Students are strongly encouraged to discuss the homework with other classmates and form study groups. Yet, each student must then solve the homework by itself 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 or other material, is considered an honor code violation. Also, stating any part of the solutions in class or on [Piazza](#) is considered an honor code violation. Finally, failing to name the correct study group members, or filling out the wrong date or time of the submission, is also considered an honor code violation. Honor code violation will not be tolerated! Any student violating the honor code will be reported to [faculty disciplinary committee](#) and vice dean for education.

Name & SID: _____

Study group: _____

Date & time: _____

I acknowledge and accept the honor code.

Signature: _____

1 Ring graph modularity (0.75 points)

Consider a graph with n nodes positioned on a ring thus each node is linked to its two nearest neighbors (Figure 1). Let the graph be partitioned into c consecutive clusters with $n_c = n/c$ nodes each. (*You can assume that n is divisible by c .*)

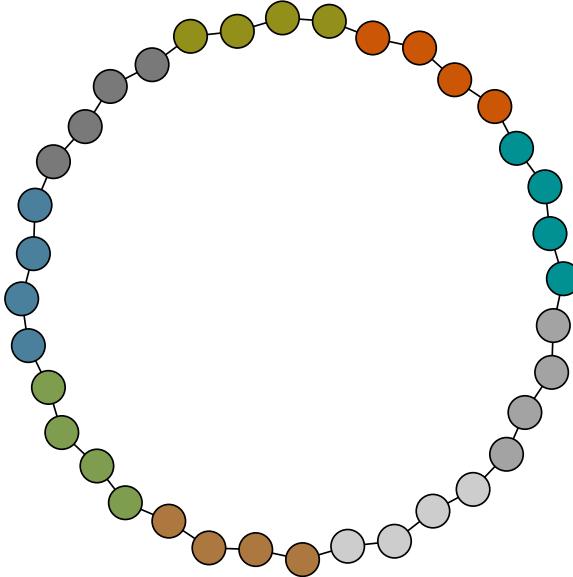


Figure 1: **Ring graph with $n = 36$, $n_c = 4$ and $Q = 0.64$**

First, derive modularity Q [GN02] of such partition of a ring graph and express it in terms of only n_c and n . (See lecture handouts for the definition of modularity.) Second, find the size of clusters n_c that optimizes modularity Q of a ring graph and express it in terms of n .

What to submit?

Derive the expression for modularity Q of a ring graph (0.5 points) and find the optimal size of clusters n_c according to modularity Q (0.25 points).

2 Who's the winner? (4.5 points)

Community detection is a popular research area of network science [New12]. Indeed, literally hundreds of community detection algorithms have been proposed in the literature in the last two decades [For10, FH16]. These include hierarchical clustering, spectral methods (e.g. [Grclus](#)), 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. [COPRA](#)), random walks (e.g. [Walktrap](#)), clique percolation (e.g. [SCP](#)) and many others (e.g. [BigClam](#), [DEMON](#)).

Your task is to compare accuracy, robustness, uncertainty, complexity and speed of three algorithms. These should include (1) [Leiden](#) or [Louvain](#), (2) [Infomap](#) or [\(DC\)SBM](#) and (3) one algorithm of your own choice. (*If you are unable to compile any of the required algorithms, search*

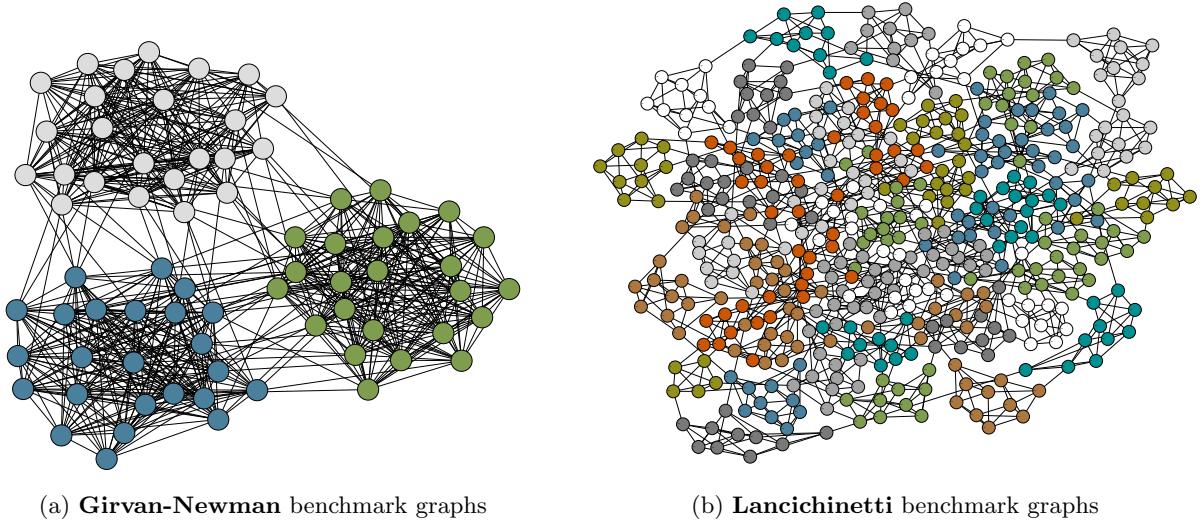


Figure 2: Synthetic benchmark graphs with planted partition for $\mu = 0.1$

for an equivalent implementation within [CDlib library](#). Code required to solve this exercise will likely consist of several ad hoc scripts that will have to be run sequentially.)

- (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 2a). The group structure should be 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 all three 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 partitions and detected community structures, and average the results. (*See lecture handouts for the definition of normalized mutual information.*) Plot community detection accuracy of all three algorithms on a single plot with μ on the horizontal axis and normalized mutual information on the vertical axis.

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

- (ii) Consider more realistic [Lancichinetti benchmark graphs](#) with planted partition [LFR08]. The graphs consist of 2500 nodes (Figure 2b), while the group structure is again controlled by a mixing parameter μ . (*See lecture handouts for the description of benchmark graphs.*) Apply all three 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 all three algorithms on a single plot with μ on the horizontal axis and normalized mutual information on the vertical axis.

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

- (iii) Consider an Erdős-Rényi random graph [ER59] 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 all three community detection algorithms to 25 random graph realizations with 1000 nodes and the average degree equal to 8, 16, 24, 32 and 40. Plot community detection robustness of all three algorithms on a single plot with the average node degree on the horizontal axis and normalized variation of information on the vertical axis. (*See lecture handouts for the definition of normalized variation of information.*)

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

- (iv) Consider Lusseau bottlenose dolphins network [LSB⁺03] with a known sociological division into two groups.

Apply each community detection algorithm 25 times and analyze community detection uncertainty. More precisely, compute pair-wise normalized variation of information between the detected community structures, and average the results.

State normalized variation of information for all three algorithms and briefly discuss the results. Which algorithm is most deterministic?

- (v) Given all knowledge gained above (e.g. accuracy, robustness, uncertainty, transparency, complexity, speed), which algorithm would you choose for your course project if needed? State the weaknesses of each algorithm and finally select a winner.

What to submit?

- (i) Give a printout of benchmark graph implementation (**0.25 points**). Plot community detection accuracy of all three algorithms (3×0.25 points). Briefly defend your answer to the question (**0.25 points**).
- (ii) Plot community detection accuracy of all three algorithms (3×0.25 points). Briefly defend your answer to the question (**0.25 points**).
- (iii) Plot community detection robustness of all three algorithms (3×0.25 points). Briefly defend your answer to the question (**0.25 points**).
- (iv) State community detection uncertainty of all three algorithms (3×0.25 points). Briefly defend your answer to the question (**0.25 points**).
- (v) State the weaknesses of each algorithm and give a brief answer to the question (**0.25 points**).

3 Peers, ties and the Internet (**3 points**)

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 all 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 [BA99], which states that nodes are more likely to connect with 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 [WS98], which can be explained by triadic closure in social networks. Therefore, nodes are more likely to connect if they share many common neighbors. The Adamic-Adar index [AA03] takes into account also 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 .
3. Many real networks consist of communities of densely linked nodes with only few links between the communities [GN02]. Links are thus more likely to appear within communities, rather than between. Let $\{c\}$ be community structure revealed by either Leiden [TWVE19] or Louvain [BGLL08] modularity optimization algorithm and let c_i be the community label of node i . (*If you are unable to compile any of the required algorithms, 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 can be defined as $s_{ij} = m_c / \binom{n_c}{2}$ for $c = c_i = c_j$, whereas $s_{ij} = 0$ for $c_i \neq c_j$.

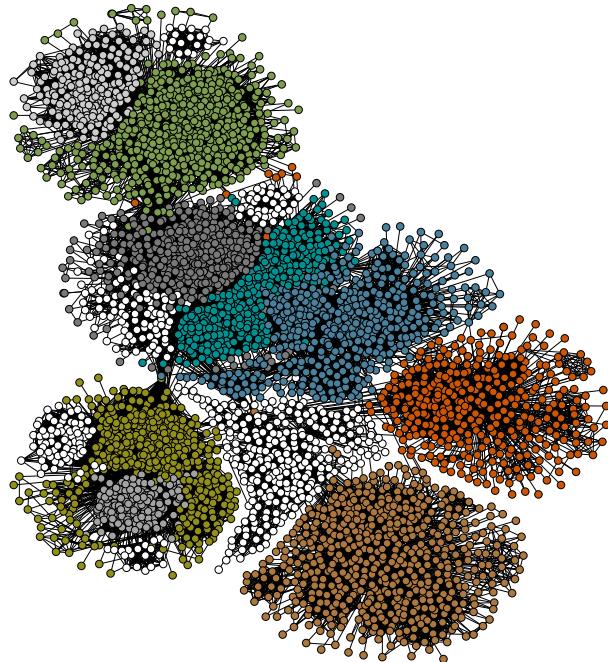


Figure 3: Communities in Facebook social circles revealed with Louvain method

- (x) Assume you apply a link prediction method to all unlinked pairs of nodes of a real network and later evaluate between which pairs of nodes the links actually 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 yet linked 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 in L_P has higher value of s than a randomly chosen pair of nodes in 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 both 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 [ER59] with $n = 25\,000$ nodes and average node degree $\langle k \rangle = 10$, and three real networks. These are [Gnutella peer-to-peer file sharing network](#), small sample of [Facebook social circles network](#) (Figure 3) and [nec overlay map of the Internet](#). (*Although some networks are directed, represent them with undirected graphs.*)

Which method comes out on top for each network? Why? Briefly discuss the performance of methods on real networks and random graphs.

What to submit?

- (x) Briefly defend your answer to the question (0.25 points).
- (y) Give a printout of the framework implementation (0.5 points).
- (z) State AUC over ≥ 10 runs for each link prediction method and graph or network (3 \times 0.5 points). Answer both questions for each network and briefly comment on the results (3 \times 0.25 points).

4 Get at least 70% right! (extra points)

You are given a [citation network](#) of scientific papers published by the American Physical Society between the years 2008 and 2013. The papers were published in ten journals, which represent the information you would like to infer from the structure of citation network.

More precisely, you would like to predict the correct journal of all papers published in the year 2013 based on their citation pattern 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 $\gtrsim 65\%$ classification accuracy, whereas your task is to propose a strategy that gives at least $\geq 70\%$ classification accuracy.

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

What to submit?

Describe your strategy and briefly explain its rationale (**0.25 points**). State classification accuracy over ≥ 10 runs (**0.5 points**) and compare your results with the baseline $\gtrsim 65\%$ (**0.25 points**). Print out any code you might have used or describe how you solved the exercise (**0.25 points**).

References

- [AA03] Lada A Adamic and Eytan Adar. Friends and neighbors on the Web. *Soc. Networks*, 25(3):211–230, 2003.
- [BA99] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.
- [BGLL08] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *J. Stat. Mech.*, P10008, 2008.
- [ER59] P. Erdős and A. Rényi. On random graphs I. *Publ. Math. Debrecen*, 6:290–297, 1959.
- [FH16] Santo Fortunato and Darko Hric. Community detection in networks: A user guide. *Phys. Rep.*, 659:1–44, 2016.
- [For10] Santo Fortunato. Community detection in graphs. *Phys. Rep.*, 486(3-5):75–174, 2010.
- [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.
- [LFR08] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms. *Phys. Rev. E*, 78(4):046110, 2008.
- [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.
- [WS98] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, 1998.