# Homework #3

This homework is complete and will not be changed. The homework does not require a lot of writing, but may require a lot of thinking. It does not require a lot of processing power, but may require efficient programming. It accounts for 12.5% of the course grade. All questions and comments regarding the homework should be directed to Piazza.

## Submission details

This homework is due on **May 4th** at 2:00pm, while late days expire on **May 8th** at 1:00pm. The homework must be submitted as a hard-copy in the submission box in front of R 2.49 and also as an electronic version to eUcilnica. It can be prepared in either English or Slovene and either written by hand or typed on a computer. The hard-copy should include (1) this cover sheet with filled out time of the submission and signed honor code, (2) short answers to the questions, which can also demand proofs, tables, plots, diagrams and other, and (3) a printout of all the code required to complete the exercises. The electronic submission should include only (1) answers to the questions in a single file and (2) all the code in a format of the specific programming language. Note that hard-copies will be graded, while electronic submissions will be used for plagiarism detection. The homework is considered submitted only when both versions have been submitted. Failing to include this honor code in the submission will result in **10% deduction**. Failing to submit all the developed code to eUcilnica will result in **50% deduction**.

## Honor code

The students are strongly encouraged to discuss the homework with other classmates and form study groups. Yet, each student must then solve the homework by herself or himself without the help of others and should be able to redo the homework at a later time. In other words, the 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, 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 Graph Laplacian (0.75 points)

Let $n$ be the number of nodes in an undirected network and let $m$ be the number of links. Graph Laplacian $L$ is $n \times n$ matrix defined as $L = D - A$, where $A$ is the network adjacency matrix and $D$ the diagonal matrix with node degrees $\{k_i\}$ along its diagonal. Link incidence matrix $B$ is $m \times n$ matrix defined as $B_{ij} = 1$ if $j$ is the first endpoint of $i$-th link, $B_{ij} = -1$ if $j$ is the second endpoint of $i$-th link, and $B_{ij} = 0$ otherwise. *(Arbitrarily designate one endpoint to be the first one and the other to be the second one.)* First show that $L = B^T B$. Using this equality further show that all eigenvalues of $L$ are non-negative and that vector of all ones is an eigenvector of $L$. These results prove useful in spectral community detection [Fie73, New10].

### What to submit?

Show that the equality holds (0.25 points). Give proof of the non-negativity of the eigenvalues of $L$ (0.25 points) and show that vector of all ones is an eigenvector of $L$ (0.25 points).

# 2 Ring graph modularity (1 point)

Imagine a graph with $n$ nodes positioned on a ring thus each node is linked to its two neighbors (see 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 $c$.)* Compute modularity $Q$ [GN02] of such partition and express it in terms of $n_c$ and $n$. *(See lecture handouts for the definition of modularity.)* Find the size of clusters $n_c$ that optimizes modularity $Q$ and express it in terms of $n$.

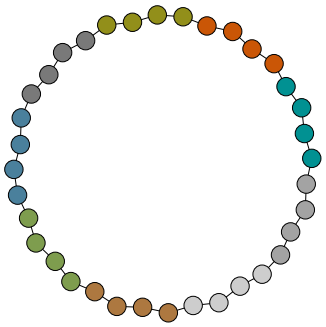


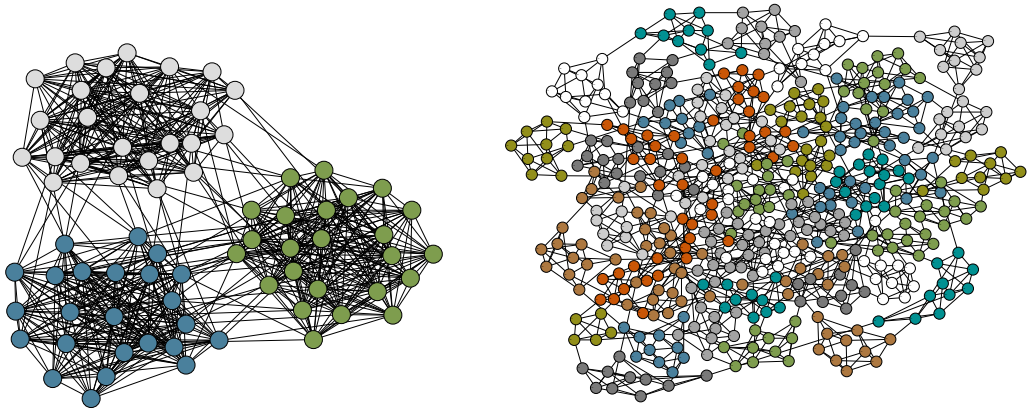
Figure 1: **Ring graph with $n = 36$, $n_c = 4$ and $Q = 0.64$**

### What to submit?

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

# 3 Who's the winner? (4.5 points)

Community detection is one of the most popular research areas of network science [New12]. Indeed, literary 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. Graclus), modularity optimization (e.g. Leiden and Louvain), map equation algorithms (e.g. Infomap), stochastic block models (e.g. SBM), statistical methods (e.g. OSLOM),



(a) **Girvan-Newman** benchmark graphs (b) **Lancichinetti** benchmark graphs

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

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. DEMON). Your task is to compare the accuracy, robustness and uncertainty of selected three algorithms. These should include at least Leiden or Louvain and Infomap or SBM, and also an algorithm of your own choice which should not be from the same class of methods as the mentioned two. *(If you are unable to compile any of the required algorithm implementations on your machine, and there is no equivalent implementation within your or e.g. CDlib programming library, you should write to Piazza and ask for an appropriate alternative. 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 synthetic benchmark graphs with planted partition [GN02]. The graphs consist of three groups of 24 nodes each, while the expected degree of each node is 20 (see Figure 2a). The group structure is controlled by a mixing parameter $\mu$. 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 $\mu$ equal to 0, 0.1, 0.2, 0.3, 0.4 and 0.5. For each algorithm and each value of $\mu$, compute normalized mutual information between the planted partitions and the 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 $\mu$ on the horizontal axis and the average 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 a more realistic Lancichinetti synthetic benchmark graphs with planted partition [LFR08]. *(See lecture handouts for the description of benchmark graphs.)* The graphs consist of 2500 nodes (see Figure 2b), while the group structure is again controlled by a mixing parameter $\mu$. Apply all three community detection algorithms to 25 benchmark graph realizations with $\mu$ 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 $\mu$ on the horizontal axis and the average 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 single 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 degree on the horizontal axis and the average normalized variation of information on the vertical axis. *(See lecture handouts for the definition of normalized variation of information.)* Which algorithms appear 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 the average normalized variation of information for all three algorithms and briefly discuss the results. Which algorithms appear most deterministic?

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

### What to submit?

**(i)** Give a printout of the benchmark graph implementation (0.25 points). Plot community detection accuracy of all three algorithms ($3 \times 0.25$ points). Give an answer to the question and briefly comment on the results (0.25 points).

**(ii)** Plot community detection accuracy of all three algorithms ($3 \times 0.25$ points). Give an answer to the question and briefly comment on the results (0.25 points).

**(iii)** Plot community detection robustness of all three algorithms ($3 \times 0.25$ points). Give an answer to the question and briefly comment on the results (0.25 points).

**(iv)** State community detection uncertainty of all three algorithms ($3 \times 0.25$ points). Give an answer to the question and briefly comment on the results (0.25 points).

**(v)** State the weaknesses of each algorithm and give a brief answer to the question (0.25 points).

## 4 Peers, ties and the Internet (3.25 points)

Link prediction is probably the most common application of network analysis techniques. For given nodes $i$ and $j$, link prediction methods compute an index $s_{ij}$ that should be high for $i$ and $j$ that are likely to connect, 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 of real networks is believed to be the consequence of preferential attachment [BA99] that 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 $i$.

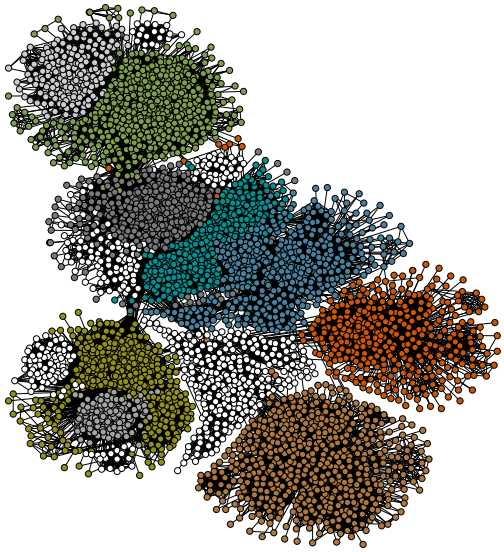


Figure 3: **Communities in Facebook social circles revealed with Louvain method**

2. Small-world networks are characterized by an abundance of triangles [WS98], which can be explained by triadic closure. Hence, 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 $\Gamma_i$ is the set of neighbors 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 than between. Let $\{C\}$ be the communities revealed by Leiden or Louvain modularity optimization [TWVE18, BGLL08] and let $c_i$ be the community label of node $i$. Furthermore, let $n_c$ and $m_c$ be the number of nodes and links within the community $C$. Then, the community index is defined as $s_{ij} = m_c / \binom{n_c}{2}$ for $c_i = c_j$, whereas $s_{ij} = 0$ for $c_i \neq c_j$. *(If you are unable to compile the required algorithm implementation on your machine, and there is no equivalent implementation within your or e.g. CDlib programming library, you should write to Piazza and ask for an appropriate alternative.)*

**(x)** Assume that you apply a link prediction method to all unlinked pairs of nodes in 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 is usually evaluated using area under the ROC curve (AUC), which can be 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 $\frac{1}{2}$. 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 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 the average AUC over several runs for all three link prediction methods above applied to an Erdős-Rényi random graph [ER59] with 25000 nodes and the average degree 10, and to three real networks. These are Gnutella peer-to-peer file sharing network, a small sample of Facebook social circles network (see Figure 3) and nec overlay map of the Internet. *(Although some networks are directed, treat them as undirected.)* Which method comes out on top for each individual network? Why? Briefly discuss the results and compare the performance of methods on random graphs and real networks.

### What to submit?

**(x)** Give a brief answer to the question (0.25 points).

**(y)** Give a printout of the framework implementation (0.75 points).

**(z)** For each link prediction method, state the average AUC obtained for random graphs and real networks ($3 \times 0.5$ points). Answer both questions for each individual network and briefly comment on the results ($3 \times 0.25$ points).

## 5 Get at least 70% right! (1.5 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 metadata 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 patterns and the journal information of all 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 $\approx 65\%$ classification accuracy, while your task is to propose a strategy that gives at least $\approx 70\%$ classification accuracy. The strategy can use any network analysis method or other approach as long as it scales better than $\mathcal{O}(n^2)$ in real networks. *(To get full credit your strategy should be at least slightly different from the methods seen in lectures.)*

### What to submit?

Describe your strategy and briefly explain its rationale (0.25 points). State the average classification accuracy obtained over several runs and compare results with the baseline (0.75 points). Print out any code you might have used or describe how you solved the exercise (0.5 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.

[Fie73] M. Fiedler. Algebraic connectivity of graphs. *Czech. Math. J.*, 23:298–305, 1973.

[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.

[New10] Mark E. J. Newman. *Networks: An Introduction*. Oxford University Press, Oxford, 2010.

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

[TWVE18] V. A. Traag, Ludo Waltman, and Nees Jan Van Eck. From Louvain to Leiden: Guaranteeing well-connected communities. *e-print arXiv:1810084 73v1*, pages 1–25, 2018.

[WS98] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, 1998.