Aalto University, School of Science
CS-E5740 Complex Networks, Fall 2022
Saramäki, Hiraoka
Exercise set #8

Exercise set #8 (17 pts)

- The deadline for handing in your solutions is November 15th 2022 20:00.
- Return your solutions (one `.pdf` file and one `.zip` file containing Python code) in MyCourses (Assignments tab). Additionally, submit your pdf file also to the Turnitin plagiarism checker in MyCourses.
- Check also the course practicalities page in MyCourses for more details on writing your report.

1. Network sampling (9 pts)

Many network data sets are samples of some underlying graphs that we are actually interested in. That is, nodes and edges of these empirical networks have been sampled in a way that we only observe parts of the network. This can severely bias even the most simple network measures so that the sampled graph has quantitatively different properties when compared to the underlying graph. In this exercise we will see the effect of three different sampling schemes on network transitivity C (also known as the global clustering coefficient) and derive estimators that can be used to correct for these biases.

Let us first recall the definition of transitivity C:

$$C = \frac{\tau_\triangle}{\tau_\angle} = \frac{\sum_i E_i}{\sum_i \binom{k_i}{2}}\,, \tag{1}$$

where $\tau_\triangle$ is three times the nuber of triangles [1] (three nodes that are fully connected), $\tau_\angle$ is the number of two-stars (a node and two of its neighbors, regardless of whether they are triangles), i is a node, k_i is the degree of i, and E_i is the number of triangles centered on i (in other words, how many triangles pass by i).

Now let's get to work. Start by generating a network from a random model. Use the command `nx.relaxed_caveman_graph` to generate a graph with 55 communities of 12 people each, where each link is then rewired with probability 0.1. This is the "true" underlying graph from which we obtain samples.

a) (3 pts) First, let's implement the three sampling schemes. Program three functions that perform:

1. Bernoulli sampling of nodes: iterate over nodes, and sample each one with probability p. We observe an edge if and only if we have sampled its two constituting nodes.
2. Bernoulli sampling of edges: iterate over edges, and sample each one with probability p. We observe a node if and only if we have sampled at least one of its edges.
3. Star sampling: iterate over nodes, and sample each one with probability p. If you have directly sampled a node, you also observe all of its neighbors (a real-life example

[1] This is three times the number of triangles as we are going through all the nodes - and thus counting each triangle three times. We could define an alternative estimator that corrects for this overestimation, but for transitivity $\tau_\triangle = \sum_i E_i$ suffices.

would be a data set obtained by crawling through friendship lists of randomly selected users in a social networking website). Note: here we will have nodes that are sampled a) directly with probability p and b) indirectly via sampling a neighbor. It is useful to keep a list of nodes you sampled directly.

Obtain samples of the network using the three sampling schemes with probability $p = 0.22$. Then, use either the code we provided or your own to obtain empirical estimates of the number of triangles, the number of two-stars, and transitivity. **Report** your results on a table where the columns represent:

- sampled number of triangles,
- sampled number of two-stars
- transitivity in sampled network
- fraction of sampled triangles over triangles in original network
- fraction of sampled two-stars over two-stars in original network

and the rows represent the different sampling schemes (plus an extra row with the values of the original network). **Answer** the following questions: how do sampling schemes compare in the fraction of triangles/two-stars they preseve? Do sampling schemes affect two-stars/triangles in the same way? Transitivity via node sampling should be similar to the real value, what could be the reason for this?

Hints:

- For star sampling, there are at least two ways of counting the sampled two-stars: we can either focus on the directly sampled nodes and obtain their full degrees, or we can count all the two-stars regardless of whether the center node itself was selected (so if two nodes with the same neighbor are selected, we observe a two-star centered on a node not directly sampled). Both answers are correct, but using the directly-sampled nodes will make calculations easier on the next excercise.

b) (2 pts, pen and paper) Different sampling schemes alter the observed number of structures on the sampled networks. Luckily, knowing the sampling probability p, we can estimate how much these numbers vary. As an example, for Bernoulli sampling of edges, the probability of sampling a two-star from the original network is $p^e_\angle = p^2$ (we need to observe two edges), while a triangle is sampled with probability $p^e_\triangle = p^3$ (we need to observe three edges). **Derive and explain how to obtain** the i) probabilities of sampling a two-star and a triangle using Bernoulli sampling of nodes, $p^n_\angle$ and $p^n_\triangle$; and ii) the probabilities of sampling a two-star and a triangle using star sampling, $p^s_\angle$ and $p^s_\triangle$.

Hints:

- You can check the validity of your calculations by comparing the fraction of two-stars and triangles sampled from the totals in the original network obtained in a), and your answers when $p = 0.25$.

c) (1 pt, pen and paper) The Horvitz-Thompson (HT) estimator is a simple way of correcting for the bias induced by sampling. Let $\hat{\tau}$ be the emprical count of a structure found in a sampled network, such as your results from a). If p_τ is the probability of observing these structures, then the HT estimator for the total counts is simply:

$$\hat{\tau}^{HT} = \frac{1}{p_\tau}\hat{\tau} \tag{2}$$

Explain why the HT estimator corrects for sampling bias. **Write and simplify** the HT estimators for the number of two-stars, triangles and transitivity for the three sampling schemes we explored.

Hints: For transitivity, we may simply use a "plug-in" estimator by substituting the empirical estimators for the HT estimators; that is, use $\hat{\tau}_{\triangle}^{HT}$ and $\hat{\tau}_{\angle}^{HT}$ instead of $\hat{\tau}_{\triangle}$ and $\hat{\tau}_{\angle}$ in the transitivity formula.

d) (3 pts) Implement the HT estimator for the three sampling schemes. Given two selection probabilities p, we will sample at least $n = 150$ times to obtain distributions of some of our HT estimators, and compare it with the measurements from the sampled networks. In other words, for each $p = 0.35, 0.5$, and for each sampling scheme, obtain $n = 150$ samples from the original network, calculate the HT estimator for the number triangles and for transitivity. For $p = 0.5$ include also the empirical estimator (the counts without the HT correction), and for all plots include a vertical line depicticing the value of the original network. As a summary, you will **report your results in six plots**:

- Histograms of estimator $\hat{\tau}_{\triangle}^{HT,n}$ for $p = 0.35, 0.5$ and $\hat{\tau}_{\triangle}$ (empirical counts) for $p = 0.5$ and true value of $\tau_{\triangle}$.
- Histograms of estimator $\hat{\tau}_{C}^{HT,n}$ for $p = 0.35, 0.5$ and $\hat{\tau}_{C}$ (empirical value) for $p = 0.5$ and true value of C.
- Histograms of estimator $\hat{\tau}_{\triangle}^{HT,e}$ for $p = 0.35, 0.5$ and $\hat{\tau}_{\triangle}$ (empirical counts) for $p = 0.5$ and true value of $\tau_{\triangle}$.
- Histograms of estimator $\hat{\tau}_{C}^{HT,e}$ for $p = 0.35, 0.5$ and $\hat{\tau}_{C}$ (empirical value) for $p = 0.5$ and true value of C.
- Histograms of estimator $\hat{\tau}_{\triangle}^{HT,s}$ for $p = 0.35, 0.5$ and $\hat{\tau}_{\triangle}$ (empirical counts) for $p = 0.5$ and true value of $\tau_{\triangle}$.
- Histograms of estimator $\hat{\tau}_{C}^{HT,s}$ for $p = 0.35, 0.5$ and $\hat{\tau}^{C}$ (empirical value) for $p = 0.5$ and true value of C.

In your plots, the distribution of HT estimators should lie around the real value. **Answer** the following questions: what is the effect of the sampling probability p on your estimators? How the HT distribuitions differ between sampling schemes? In the last plot, the empirical estimators (without HT correction) should be centered around the true value, why could this be?

Hints:

- Since we want to observe how different samples may arise from the same network, we need to take a large number of samples ($n = 150$, for instance). However, while you are coding and testing it may be wise to use a smaller n. Keep in mind that if your code takes too much time to run on your computer, you can report results for a smaller n or a smaller network.
- In case you were not able to obtain the theoretical probabilitites in excercise b), you can substitute the theoretical probabilities ($p_{\triangle}^{s}$, for example) with the fraction of sampled structures over the totals in the original network. If this is the case, please state so in your report and explain why this substitution is possible.

Aalto University, School of Science
CS-E5740 Complex Networks, Fall 2022
Saramäki, Hiraoka
Exercise set #8

2. Modularity (8 pts, pen and paper)

When nodes of a network are partitioned into communities (or clusters), *modularity* can be used to measure how well the given partition catches network's community structure. The modularity Q of a partition into communities can be written as

$$Q = \sum_{c \in \mathcal{P}} \left[\frac{l_c}{L} - \left(\frac{d_c}{2L} \right)^2 \right] , \tag{3}$$

where the sum runs over all clusters/modules/communities, L is the number of links in the whole network, l_c is the number of links internal to cluster c (internal = both endpoint nodes are in c), and d_c is the total degree of nodes in cluster c (sum of the degrees of nodes in the cluster, such that the degrees count all links attached to the nodes irrespective of if they go inside or outside of the community). In other words, modularity is defined as the difference between the relative number of links inside the communities and the expected relative number of links inside the communities in a randomized network (usually configuration model) without clear community structure.

a) (2 pts) **Calculate** the value of modularity for the two partitions shown in Fig. 1 (in the first, there are two clusters, and in the second, the whole network is taken as a single cluster).

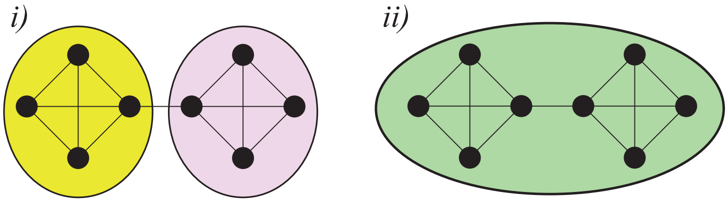

Figure 1: Two module configurations for a)

In the rest of this exercise, we will show that the modularity Q defined in the above Eq. (3) contains an intrinsic scale. This means that the size of communities that are favored by modularity optimization depends on the number of links L in the network. Therefore, sometimes subgraphs which should by common sense be labeled as separate modules, such as full cliques attached to the rest of the network with a single link, get merged when Q is maximized.

Consider now a very general case – a network where there are altogether L links. Suppose that we have somehow *a priori* identified two groups of nodes a and b which we consider as modules (say, because they are cliques). Let us also assume that there is a single link connecting these two modules[2]. Instead, we make no assumptions about the number of links that connect a and b to the rest of the network. For a schematic illustration, see Figure 2.

Let's now consider two alternative ways of parcellating this network into communities: either 1) consider the two real modules as two modules, or 2) merge them into a single module ab. In community detection based on modularity optimization, we select the partition that yields

[2] Similar, although slightly more general, calculations could be made by assuming that there are $l_{a \leftrightarrow b}$ links connecting the two communities without much difference in the results. Here we, however, concentrate on the limiting case of only a single connecting link.

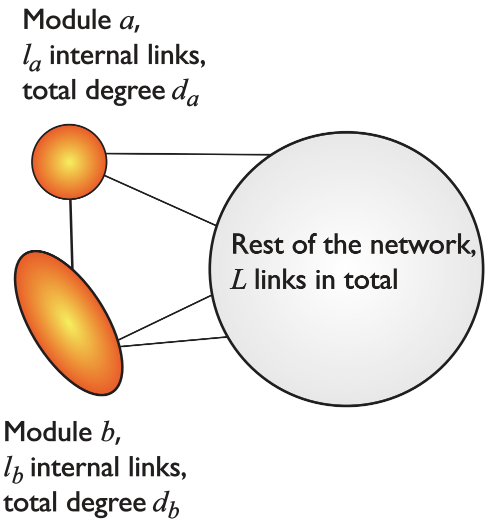

Figure 2: Schematic figure of two modules a and b, connected with a single link. Note that in b) and c) we do not make any assumptions of how many links are connecting the two modules to the rest of the network.

higher modularity. So, the idea now is to calculate the difference in modularity $\Delta Q = Q_2 - Q_1$ between these two partitions using the formulation of Eq. 3 for modularity. When this difference is positive modularity optimization will merge the two "physical" modules.

b) (2 pts) **Write** ΔQ as a function of L, d_a, d_b, d_{ab}, l_a, l_b, and l_{ab}.

   Hint: Note that the part of the modularity that comes from any other community than a, b or their merger ab is cancelled out when Q_1 is substracted from Q_2. This can be seen easily by writing in both formulas Q_1 and Q_2 the part of the sum that deals with communities in the rest of the network as $Q_R = \sum_{c \in \mathcal{P}'} \left[\frac{l_c}{L} - \left(\frac{d_c}{2L} \right)^2 \right]$, where $\mathcal{P}'$ is the set of communties in the rest of the network.

c) (2 pts) **Write** d_{ab} as the function of d_a and d_b, and l_{ab} as the function of l_a and l_b. Using these substitutions **show** that the option of merging two clusters becomes favored by modularity optimization when $L > \frac{d_a d_b}{2}$. **Explain** why this result indicates that modularity optimization must have a *resolution limit*, or a minimum size of community that can be found that depends on the size of the network (where size is the number of links). You may assume that the network is sparse so d_a, d_b, and d_{ab} are constants when L increases.

d) (2 pts) Next, **provide an argument** that the resolution limit of modularity has a form such that in a network with L links it is not possible to find communities of size smaller than $n \propto \sqrt{L}$. **Use** the following steps in your argument:

   i) Assume that community a is a clique of n nodes and that there is one single link connecting a to the rest of the network. Write the formula for d_a as function of n. To make things slightly simpler, you can approximate $d_a \approx n^2$ for the rest of this exercise.

   ii) Using the results of c) show that the community a is always merged to some other community when $n < C\sqrt{L}$, where C is some constant that doesn't depend on L.

Aalto University, School of Science
CS-E5740 Complex Networks, Fall 2022
Saramäki, Hiraoka
Exercise set #8

Feedback (1 pt)

To earn one bonus point, give feedback on this exercise set and the corresponding lecture latest two days after the report's submission deadline. You can find the feedback form at the Assignments tab in MyCourses.

References