Analysis of Large-Scale Social Networks

Exercise Session 4: Graph Clustering

Software

- Suggestion: iGraph in Python using Google Colab
- Any other network package can be used

Data Sets

The provided network consists of a set of publications connected to each other based on their similarity. There are two files:

- Network file in Pajek format. Nodes have an UT-code as identifier (15 character code). Edges are weighted similarities. The similarities between two documents are based on two parameters: the number of shared references and the textual similarity between the abstracts of the publications
- Flat text file, tab-separated, with the bibliographic information on the papers holding title, authors, publication year, …

Objectives

The objective of this exercise is two-fold:

- Demonstrate the difference between distinct cluster solutions
- Demonstrate the existence of the degeneracy problem in the Louvain cluster algorithm.

Pre-processing

Tasks

1. *Create a notebook to log all the commands and steps.*
2. *Read the network data*
3. *Check if the network is undirected/directed and weighted/unweighted*
4. *Calculate degree and plot degree distribution*
5. *Check if the network is connected*
6. *Identify the largest connected component*
7. *Retain only this largest component*
8. *Calculate degree and plot degree distribution again.*
9. *Save new network in pajek format*
10. *Save the label list (UT-codes)*

Clustering & Visualization

Tasks:

1. *Run three community detection algorithms from the igraph packages (eg cluster_leading_eigen; cluster_multilevel; cluster_leiden)*
2. *Calculate modularity scores for each cluster solution*
3. *Try to visualize (a sample of) at least one network with nodes in different colour according to their cluster membership*
4. *Compare the obtained cluster solutions using Normalized Mutual Information*

5. *Identify the topics of one cluster solution based on the title of the members of the clusters. You could use the nodes with the highest weighted degree in the different communities.*

Degeneracy in Clustering

Tasks:

1. *Run the Multilevel community detection algorithms on 5 permuted versions of the network. The permutation of the network can be obtained with this command:*

```
perm_net=net.permute_vertices(
   np.random.permutation(
         hb.vcount()
   ).tolist())
```

2. *Calculate modularity scores for each cluster solution*
3. *Compare the obtained cluster solutions. How stable is the community detection across different permutations?*