CSE 5245 Introduction to Network Science – Lab 2

This assignment will allow you to explore various community detection algorithms for networks along the axes of effectiveness and efficiency. There are three components to this assignment:

1. Execution of various community discovery algorithms (source code will be provided)
2. For the datasets with ground truth analysis should be done using entropy.
3. For the datasets without ground truth, analysis should be done with various measures such as conductance, modularity, ncut etc.

Preliminaries:

You will focus on the following datasets available at:

https://snap.stanford.edu/data/wiki-Vote.html
https://snap.stanford.edu/data/ca-GrQc.html
https://snap.stanford.edu/data/p2p-Gnutella08.html
https://snap.stanford.edu/data/egonets-Facebook.html
https://snap.stanford.edu/data/com-Youtube.html

Note that different executables accept different formats. This preprocessing step may change the data statistics as compared to the webpage.

Also note that, for the youtube dataset ground truth community information is available for a subset of its nodes. You should do your analysis (entropy computation) based on this subset only.

Assignment Details:

1. You will use the following node community discovery algorithms:
   a. MLR-MCL
   b. METIS
   c. Clauset-Newman-Moore
   d. Documentation/source codes can be found at
   https://github.com/KarypisLab/METIS/blob/master/manual/manual.pdf
   https://sites.google.com/site/stochasticflowclustering/
   http://snap.stanford.edu/snap/index.html

2. All datasets and binaries can be found on Carmen "Files-Lab 2 files-CSE5245ForStudents.zip". After extracting the zip file, the datasets can be found at "CSE5245ForStudents/Lab2/data/"
    a. edgelist file ends with .txt
    b. metis file ends with .metis
    c. ground truth file ends with .GT

3. The binaries can be found at "CSE5245ForStudents/Lab2/bin/"
    a. "mlrmcl" for MLR-MCL
    b. "gpmetis" for METIS
    c. "community" for CNM

4. (Optional) You may choose to either run Lab2 locally or in stdlinux. If you choose to run in stdlinux, here are some steps you may begin with
    a. After logging in, you can either copy the zip file from your local machine or download it by:
       " wget http://web.cse.ohio-state.edu/~li.7533/CSE5245ForStudents.zip "
    b. Extract the file by:
       "unzip CSE5245ForStudents.zip"
    c. Go to the folder by:
       "cd CSE5245ForStudents/Lab2/ "
    d. One example:
       "`./bin/gpmetis ./data/facebook_combined.metis 100`" will create the output inside the ./data/ directory.

5. Perform a meta-level analysis of correlation of various measures of community detection quality such as modularity, conductance and ncut. For the dataset with ground truth (Youtube), you additionally need to compute entropy. *Please spend some time understanding and tuning some of the parameters in each method and comment on ease-of-use (tuning).*

The submitted set of files should include all code; a readme file describing how to execute/run code and your report. Your report, in pdf format, should be precise and defend both your basic design choices and your meta-level analysis. We will discuss possible conclusions/analysis choices in class.

This is optionally a team project – teams of two. Graduate students may do this assignment on their own. I recommend teams of two at least initially for undergraduates. Please ensure that both team members contribute equally to all aspects – implementation, documentation and report writing. Each report must have a small section describing the contribution of each team member.

Due Date: Feb 18 by midnight

Additional Clarifications on Lab 2

1. You have two sets of datasets:
    - With ground truth
        - Youtube
    - Without ground truth
        - p2p
        - wiki
        - Facebook
        - ca
2. For both type of datasets you have two representation format:
    - .txt representation (fromNodeID ToNodeID)
        - Use as input to the "community"
    - .metis representation
        - Use as input to "mlrmcl" and "gpmetis"
    - **Note: Snap2Metis.py script is used to convert snap format (.txt) to make .metis format**
    - **Note: Snap2Metis.py should run with python2**
3. Algorithms output format
    - community: NodeID to communityID
    - mlrmcl and gpmetis: Line number is nodeID and the corresponding number is community id

4. Ground truth format
    - Youtube: Each line represent a community which contains node ids in that community
5. Entropy analysis
    - **Note: NodeID in .txt format are different with NodeID in .metis format**
        - Youtube: node IDs in ground truth are consistent with node ID in .txt format
        - For computing the entropy for **MLR-MCL and GPMETIS** algorithms output (which are based on .metis input) you have to know the mapping of .txt nodeID to .metis NodeID
            - You can get that by hacking into **Snap2Metis.py** code, line 48-52 to get the mapping
            - You can store the dictionary idMap[ele] (e.g pickle format) to use it later for your analysis.
    - There is a possibility of overlap in communities in the ground truth. Part of this assignment is to figure out how you can compute entropy regarding this issue. Look at appendix B of the attachment to get some idea. You are free to find some other ways as well.

Out of Memory Issues with Youtube Dataset

This lab is designed to test your ability to handle an out-of-memory issue you may face on the Youtube dataset. Some hints on how to handle this.

1. mlrmcl and gpmetis have no problem with youtube so make sure to include them in your analysis
2. For community you have the following options:
    - Pruning graph by degree:
        - Read graph using NetworkX library:
            - G=nx.read_edgelist("data/com-youtube.ungraph.txt")
        - extract a list of nodes with degree lower than a threshold (2,3)

            - remove = [node for node,degree in dict(G.degree()).items() if degree < Threshold]
        - remove the nodes from the graph
            - G.remove_nodes_from(remove)
        - store the pruned graph:
            - nx.write_edgelist(G,'youtube_sparse.txt',data=False)
        - You need also to keep the list of removed nodes in order to not consider them in your entropy computation for ground truth.
- Using sparsifier in NetworkX called spanner:
    - Load the network like above.
    - apply spanner to the network
        - A spanner of a graph G = (V, E) with stretch t is a subgraph H = (V, E_S) such that E_S is a subset of E and the distance between any pair of nodes in H is at most t times the distance between the nodes in G.
        - https://networkx.github.io/documentation/stable/reference/algorithms/generated/networkx.algorithms.sparsifiers.spanner.html (Links to an external site.)
        - Set the stretch factor to more than 100 to see some differences
    - Note that spanner can also take a long time
- Prune the network by random sampling
    - you have to do it multiple times and report the average among sampled subgraphs.
- Prune the network by specific communities
    - you can optionally chose to use this method on largish communities identified by MLR-MCL or Metis

Overlapping NMI

Some helpful links. The correctness of the implementations is not guaranteed.

https://github.com/aaronmcdaid/Overlapping-NMI
https://github.com/eXascaleInfolab/OvpNMI
https://github.com/eXascaleInfolab/GenConvNMI

https://scikit-learn.org/stable/modules/generated/sklearn.metrics.normalized_mutual_info_score.html

https://course.ccs.neu.edu/cs6140sp15/7_locality_cluster/Assignment-6/NMI.pdf