

Assignment 2 : Graph algorithms and Mining

Instructions :

- You must describe your approach in sufficient detail. This would be helpful for partial credit in case you are not able to complete your implementation in time.
- You may use software libraries that implement the required functions. A few examples software libraries for graphs are SNAP (<http://snap.stanford.edu/> written in C++, python interface also provided), igraph (<https://igraph.org/>, available with R, python and C), NetworkX and Neo4j.
- While you may discuss with others, the submission must be your own work.
- There may be no single right answer. Hence, do not just report the final answers, conduct some analysis of the results and clearly explain any insights obtained from these analyses.

Problem description

Use the email-Eu-core graph dataset <https://snap.stanford.edu/data/email-Eu-core.html> for this assignment.

1. Identify the largest weakly connected component and on the largest connected component and compute communities according to any community detection approach you select. You may choose an approach which produces disjoint communities or overlapping communities. Compare the set of communities with the set of ground truth communities using a suitable method. Clearly explain how you compared the two

sets of communities, this may either be a method already proposed in the literature or your own (provide relevant citation clearly). Comment on how similar or different, the two sets of communities are, based on your score and comparison method.

2. Many community detection methods are computationally intensive, making it difficult to scale to large graphs. Describe briefly with suitable references/citations one or two methods proposed in the literature for performing community detection in large networks. Alternatively, you can propose your own strategy. Describe the method and show a partial implementation on the above or some other graph.