Mining on Large Graphs

A short course at L'Università degli Studi Roma Tre
-- Some Suggested Projects and Exercises

Laks V.S. Lakshmanan

University of British Columbia

- The exercises are grouped into two units:
  - Communities k-cores, k-trusses, etc.
  - Densest Subgraphs and Influence Propagation.
- Solve one question from each unit.
- You are welcome to work in teams.
- If the question involves an implementation project, the team can have up to 3 members.
- Otherwise, the team can have up to 2 members.
- For non-implementation questions, you may present your solution as a power point presentation or as an overleaf document.

Unit I: Communities k-cores, k-trusses, etc.

(Q1) This exercise has two parts.

- Construct a small but detailed “run through” example illustrating how the efficient truss decomposition algorithm we discussed works. Your example should illustrate the algorithm’s working step by step. Make a (powerpoint or equivalent) presentation of your answer.

- Consider the k-core based definition of community search on p22. The definition is designed to ensure that the community found does not suffer from the so-called free rider effect. Suggest an alternative formulation of community search based on cores. Can you show that your formulation does not suffer from the free rider effect?

(Q2) Write a survey of research work on how multilayer graphs have been utilized in modeling biological data and how community detection / search over multilayer networks has been used to solve real-life problems over such data. Present your answer as a powerpoint presentation or an overleaf report: your choice.

(Q3) Propose and implement any project leveraging community search over undirected or directed graphs, single layer or multilayer and develop a prototype for an application of your choice. I'd be happy to discuss your proposal with you to help you flesh it out.

Unit II: Densest Subgraphs and Influence Propagation

(Q4) One of the techniques for speeding up the discovery of the densest subgraph in a single layer graph is to obtain a core decomposition and then search for the DSG in certain cores. In particular, suppose the maximum density subgraph observed during the core-decomposition is ρ. Then we know that the DSG must be contained in the $\lceil \rho \rceil - core$. Can you extend this technique to multilayer graphs using FirmCore? Discuss the details and explain your reasoning.

(Q5) For the same application as the one considered in Q3, implement any efficient algorithm for finding the densest subgraph. Using an application case study, contrast between the findings you obtained using community search versus those obtained using DSG.

(Q6) In the lecture, we discussed some efficient algorithms for ***influence maximization*** over single layer graphs. Recall that each edge (u, v) has an associated probability $p_{u,v}$ in the classic IC model. Consider a multilayer graph $G = (V, E, L)$, with vertices V, edges E, and layers L. Specifically, $E \subseteq V \times V \times L$. Each edge $(u, v, \ell) \in E$ has an associated probability $p_{u,v,\ell}$. In the context of social networks, the layers may correspond to interests or topics, depending on which users may influence each other with different probabilities. Consider an influence propagation starting from seed vertices corresponding to different layers. Assume that propagations in different layers are independent. Your goal is to find up to k seed vertices in each of λ layers such that the expected number of influenced users over all layers is maximized. To be clear, each influenced user in one layer counts toward a credit of 1 unit. If the same user in influenced in 2 layers, the campaign gets a credit of 2 units and so on. Can you extend any of the classic IM algorithms for solving this problem? Explain your reasoning.