Local Graph Clusters via Capacity Releasing Diffusion

Project Title: Local Graph Clusters via Capacity Releasing Diffusion

Project Webpage: https://www.andrew.cmu.edu/user/tianl1

Project Description

Local Graph Clustering describes the problem of finding a cluster of closely connected vertices around a vertex in a large input graph; furthermore, the cluster must sparsely connected to the rest of the graph. The main point of interest of the problem is to find such a cluster without traversing the entire graph. Usually this implies traversing or touching only the output cluster and perhaps a few vertices outside of the output cluster. This is important because given graphs with vertices with perhaps a billion or more nodes, the algorithm should still be able to find the cluster quickly and should not depend on the size of the input graph; the algorithm only cares about the cluster and thus should be able to solve this problem locally.

This problem is widely used in machine learning, data analysis, and mathematics; more applications are highlighted in the paper published by Wang et al. [4]. The exact uses are out of the scope of this project proposal; however, we give a contrived example on how it might be useful. Consider a graph that connects to items (vertices) with an edge if they are similar. Then given an input vertex, finding a cluster around it would be equivalent to finding the group of items that is similar to the input item. Surely, this leads to the natural idea of classifying objects.

Historically, this problem has been solved with random walks with resets. The algorithm is described as such: you start on the input vertex and traverse the graph randomly but at each step, there is some probability of resetting back to the input vertex. After some number of traversals, you take all vertices that has been visited more than a threshold amount. Intuitively, vertices that are highly connected to the start vertex will be visited more; however, in practice, there are graphs in which the random walk method is too aggressive as in the algorithm tends to add in vertices that are not highly connected with the rest of the cluster into the cluster.

Wang et al. [4] published a result in 2017 that attempted to fix the overaggressiveness of the random walk methods. The paper presented the Capacity Releasing Diffusion (CRD) method which was derived from the classic Max-Flow Min-Cut algorithm push relabel. In short, the algorithm runs as such:

We begin initially with some amount of weight on the input vertex, then the algorithm spread the weights around vertices close to the initial vertex. Then, it double the weights and attempted to spread the weights. This is repeated until it is too ”hard” to spread the weights, in which case we terminate and take a subset of the vertices that have high amount of weights.

Another point to note is that any variation of the random walk method is subject to the Cheeger barrier, a bound that limits how well a random walk method can estimate how closely connected
the cluster is; however, since Wang et al. [4]’s result was not a random walk algorithm, it broke the Cheeger barrier for the first time.

The paper showed promising experimental results and theoretical results; however, this project will try to extend on both the experimental theoretical results shown in the paper. For the experiments, the paper considered two classes of graphs: synthetic and real-world graphs. For synthetic graphs, they generated a 60 by 60 grid and connected corners (vertices) of the grids uniformly randomly and the real-world graphs came from various Facebook datasets on friend groups. However, even though these experiments show that the CRD method was rather consistently edging out the standard random walk algorithm, there might be contrived cases where the CRD method does not perform well. For example, analogously, we know that the random walk methods don’t work well for Cockroach graphs. The alternate focus of this project, which is just as important, is to prove various theoretical results that the paper mentions and perhaps fine-tune and improve the CRD algorithm.

This project will be done under the mentorship and guidance of Professor Gary Miller.

Project Goals

This project will aim to reproduce various experimental results of the paper and explore the theoretical implications of CRD algorithm.

75%: Reproduce experimental results shown in the paper. Run various experiments with the CRD graph on contrived examples to test performance outside of the graphs used in the paper.

100%: Improve and fine tune the algorithm in various ways such as improving runtime or more importantly improving the accuracy of the algorithm in finding good clusters. OR use the idea of push relabel to apply to another similar problem and prove some theoretical result.

125%: Write and submit a paper with novel results from the previous part. OR use the idea of push relabel to apply to another similar problem and prove some theoretical result.

Project Milestones

1st Technical Milestone for 15-300:

• Be well read in the the background information of CRD and other related algorithms
• Reading about related concepts such as the random walk local cluster algorithm, isoperimetric cuts and the Cheeger inequality
• Build intuition on the Push-Relabel algorithm
• Contact Di Wang, the author of the paper that introduced the CRD process, for his code and implementation of the CRD algorithm.

January 27th:

• Be well aquainted with the CRD process question and have a direction moving forward on the specific subproblems we want to experiment on/prove.

February 10th:

• Not enough information. TBD

February 24th:
• Not enough information. TBD

March 16th:
• Decide whether to aim for 75% or 100% or 125% goal

Match 30th:
• Not enough information. TBD

April 13th:
• Not enough information. TBD

April 27th:
• Reach the goal that was set on March 16th

Literature Search
Currently the following papers/topics are on the reading list. Further papers will be added based on need (for example if another paper is needed to understand a paper already on the reading list) and on recommendation from Prof. Miller.

Papers
• Wang et al., Capacity Releasing Diffusion for Speed and Locality[4]
• Goldberg and Tarjan, A new approach to the maximum-flow problem [3]
• Andersen and Lang, An algorithm for improving graph partitions [1]

Topics
• Random walks (Andersen and Lang [1])
• Push relabel (Goldberg and Tarjan [3])
• Cheeger Inequality, (Cheeger [2])

Resources Needed
Nothing as of now.
References


