Oliver, Xander, Holden, and Jacob
COSC 254
3/24/2021

Group 3 Project Plan: Summarizing Large Graphs with VoG

Introduction:

Our project examines the process of succinctly summarizing large graphs. The motivating question is as follows: when presented with a large, million-node graph, how can we discern meaningful information from it? It is a difficult task to discover interesting patterns in such a graph, as visual representations end up being extremely busy and incomprehensible. So how can we then visualize large graphs and find/measure their important substructures? To combat these issues, the research paper "Summarizing and Understanding Large Graphs" (Koutra et al.) proposes VoG (Vocabulary-based summarization of Graphs), a method of summarizing large graphs by approximating the optimal Minimum Description Length (MDL) in terms of local graph structures. This allows for both the discovery of interesting patterns and a better understanding of the underlying characteristics of a large graph.

Consider, for example, a large graph containing all of the edits made on a popular Wikipedia article where nodes represent individual editors and a shared edge represents edits to the same part of the article. A basic visualization of the resulting graph gives us almost no insight into the graph itself, but a summarization of the graph (using VoG) can reveal interesting information about the most important substructures of the graph. For example, we might find a list of the most important “stars” (one of six prevalent graph substructure types that the research paper outlines) which might indicate either Wikipedia super-users or admins who edited many different parts of the article. We might also get (as an output from VoG) a list of the most important “bipartite cores” which could indicate two groups of editors constantly overwriting one another’s changes. This sort of pattern discovery from the summarization of large graphs is extremely useful not only in the context of Wikipedia but across large graphs of all types.

In terms of computational challenges, graphs can be particularly computationally expensive and CPU intensive depending on the number of edges and vertices. In the case of the large graphs that VoG works on—usually on the scale of over a million nodes and well over a million edges—any sort of summarization task is very computationally expensive. Since there is no structure like anti-monotonicity to efficiently search through all possible structures from the given vocabulary, heuristics are required to get usable results. With heuristics, VoG, in the best-case scenario, has a run-time that is near-linear on the number of edges on the graph. We will have to work to code VoG as efficiently as possible.

Algorithm Descriptions:

The main algorithm we will be coding is the Vocabulary-based summarization of Graphs (VoG), which builds off of compression through minimum description length (MDL). MDL uses the database in order to compress a dataset into a lossless subset that can completely represent the entire original dataset. VoG builds off of the MDL concept and applies it to graphs, using subsets of possibly overlapping subgraphs to succinctly describe a larger graph. MDL essentially minimizes the length of the description of a model plus the length (in bits) of the description of that model when it is encoded with the data. VoG uses a set vocabulary and identifies subgraphs, labeling them as either perfect structures or approximate structures. Then using specific heuristics, VoG chooses the best graph summarization. From this paper, the first heuristic used was Plain, which acts as a baseline that uses all of the candidate structures. The second is Top-K which selects the top k candidate structures after they have been sorted by quality. The third heuristic is Greedy'nForget. This heuristic looks at a sorted list of candidate structures by quality in sequential order and will add each candidate as long as the total encoded cost of the graph does not increase. Candidate structures refer to subgraphs defined by the vocabulary being used which includes cliques, stars, bipartite cores, and chains, with this specific application.

Experiments/Evaluation:

We will be applying VoG to a variety of datasets, some of which we will be the same as those from the study—e.g. WWW-Barabasi and AS-Oregon—in order to verify our implementation of VoG, and some of which will be novel. More specifically, we can verify (based on the different heuristics used) the numbers presented in the paper in Table 4—such as the number of stars, near-bipartite cores, and full cliques—with the numbers we get from our implementation of VoG. Once we have successfully implemented VoG—and verified it on the same datasets from the paper—we are hoping to run VoG on a dataset that examines the interactions between different subreddits. By implementing our algorithms on this dataset, we will have the opportunity to find interesting insights into how communities form online and how interactions manifest around different topics in an online setting.

Work to be Done:

1) Set up framework for VoG
    a) Figure out how to use Python graph analysis libraries for graph decomposition and identifying relevant subgraphs.
    b) Implement heuristics described in the Koutra paper.
2) Figure out how to implement VoG.
    a) Determine how to identify the important subgraph structures.
        i) Consider implementing heuristics?
    b) Determine how to encode these subgraphs for MDL's use.
    c) Worst case scenario, ask Matteo for help with contacting the paper author for VoG help.
3) Test implementations on datasets from the Koutra paper

    a) Record/analyze results and compare algorithms' performance to the experiments in the Koutra paper
        i) How effective is the model? Can we find the same patterns found in the original experiment?

4) Once we have a working implementation, we will run our own experiments using our own dataset (described above)
    a) Analyze the performance of the algorithm on our data, compare it to the original experiment.
    b) Use the results of our experiment to find any interesting information about the Reddit community, in the same vein as the Wikipedia experiment mentioned in the paper and in the introduction of this project plan.

Phase 3 Deliverables:

Our goal is to set up the entire framework for tackling the VoG algorithm. That involves learning different Python graph analysis libraries such as igraph, NetworkX, and NetworKit (and selecting the best one); writing functions/implementing algorithms to determine the structure of a given subgraph (such as star, bipartite core, or clique); and implementing (or finding an implementation of) MDL.

Phase 4 Deliverables:

We plan to complete our implementation of VoG, and perform experiments using the Wikipedia dataset in the Koutra article and our own Reddit dataset. We'll use our findings to analyze both what MDL and VoG can teach us, as well as what we can determine about Reddit and the interactions between Subreddits. We will at this point have an analysis of our data versus the study's paper as well as a new analysis that we can apply towards social media groups (specifically Reddit). This phase also involves formally writing up a report of our findings and preparing our final presentation.

Note: If the VoG is too difficult to implement, Matteo can reach out to the author of the paper to ask for some code.

Logistics:

We'll be using a GitHub repository for sharing all of our code. The repository can be found here. We plan to meet over Zoom/in person (for those on campus) to develop our code synchronously since our programming tasks need to be completed in a linear fashion. We anticipate that the majority of our work will be done during these frequent group coding sessions and that we will find the chance to divvy up smaller tasks (such as writing smaller helper functions) as we progress into our coding. As we begin tackling this project, however, we will all meet as a group so that we’re all on the same page as we lay the groundwork/foundation for our algorithm implementations. Once our implementation of VoG is complete, we can then decide how to divide the labor with respect to running experiments on the two datasets, analyzing the results, and preparing the report/presentation.