NSF ITRG Research Proposal

Visualizing and Interacting with Time-Evolving Graphs

Lisa Jin

1 Project Summary

Large, time-evolving graphs can be directly and indirectly observed in a variety of phenomena across domains. Specifically, for following the evolution of communities over time, applications can range from tracking modules in protein-protein interaction networks [10] to groups in scientific co-authorship networks [3]. Other tasks may include anomaly detection and graph matching across graph snapshots in time. Whether these graphs are naturally present, as in friendships within a social network, or inferred, as in functional brain networks from fMRI data, pattern discovery in data of this scale is non-trivial. Furthermore, as temporal graph analysis is of interest to a diverse set of disciplines, identifying an approach that is both relatively unbiased and interpretable is critical.

To this end, the primary task is to develop a pipeline that includes data preprocessing, summarization, and visualization. The primary objectives of the system are to (a) provide a basis for comparing network inference methods, (b) leverage user interaction to tailor summary results to a domain, and (c) intelligently display temporal graph evolution. In network data that are inferred, a clear depiction of its ‘important’ trends can provide insight on the quality of its generation. Adjusting preprocessing parameters, such as number of temporal graph snapshots, results in varying summaries for the same raw data. One may hypothesize that certain attributes (e.g., simplicity or expressiveness) of a summary indicate success of the generated graph in capturing underlying trends. The pipeline can then incorporate this user knowledge to iteratively improve upon its summaries. Real-time user interaction with the system to produce succinct, task-specific visualizations also remains a key focus of the system.

Visualizing temporal graphs effectively not only improves the user experience, but also aids in evaluation of unsupervised analysis. Existing static graph visualization frameworks for anomaly detection include OPAvion [1] and NetRay [8], which target node-level deviations. Domain specific data mining, such as that of connectomics, depends heavily on visualization for revealing larger-scale patterns [11]. In addition to the challenges of static graph layout, the best method to denote transitions between graph snapshots (e.g., small-multiples vs. animation) remains under debate [4]. Therefore, it is a priority to determine which transition mode is best suited for each user-defined task through empirical experiments. According to results from previous experiments, the most effective layouts and transitions depend on a combination of data attributes and the task at hand.

With the prevalence of temporal graphs in numerous domains, there is great utility in creating an end-to-end analysis system. Aggregating knowledge about various user goals and datasets into a single system can make mining massive networks more accessible.
2 Project Description

In this section, we explore objectives for reaching the outcomes of our system, outline an approach to achieving the objectives, and discuss the staffing and resources needed for timely execution.

2.1 Background Overview

Temporal graph summarization encompasses the areas of graph compression and mining. An existing approach, TIMECRUNCH [12], is a principled and parameter-free algorithm that uses MDL (Minimum Description Length) to select representative temporal subgraphs. Although it requires no user intervention, it is limited by its predefined static (e.g., star, clique, bipartite core) and temporal (e.g., oneshot, flickering, periodic) vocabularies to express patterns. We plan to explore system grounded in a similar theoretical basis, but that extends the fixed vocabulary to one specific to the dataset input. To do so automatically, fast frequent-subgraph mining algorithms [7] may be a promising initial direction if such subgraphs can be discovered efficiently.

In terms of visualizing temporal graphs, a major issue is how to represent changes in network structure over time. Early approaches such as simulated annealing [9] and supergraph creation to encode nodes at all time steps [5] would not provide the required speed for fast graph drawing. More recent visualization methods use task-specific qualities, such as the animated radial layout proposed by Yee et. al [13] targeted for egocentric, tree layout networks. As there is likely no single optimal layout for all networks, it would be sufficient to select from a combination of different graph layout attributes that depend on the particular graph task.

2.2 Objectives

The overarching goal of this work is to develop a system with unsupervised analysis that uses user feedback to improve specificity of the network summarization. In effect, domain knowledge can be utilized prior to summarizing the network and/or during visualization of the summary results. While the former can be evaluated in the context of graph processing technique, the latter requires finding the optimal pairings between graph mining task and visualization method. For example, in following the dynamics of communities over time, does preserving the mental model of a graph [2, 6] improve user performance? If so, what aspects of a visualization or interaction method is effective in aiding the user? Given the present lack of consensus in experimental results, we are interested in the behavioral component of how users uncover salient temporal patterns. Experiments measuring user performance on standard graph analysis tasks will be the primary metric.

Although it is possible to ask the user for domain specific knowledge (e.g., node community labels) before graph summarization, there are instances where such expertise is incomplete or unavailable. One solution is to investigate ways to implicitly capture information from user interaction with an initial visualization of the network summary, and then iteratively improve upon the summarization using this data. Though this approach requires more than one cycle through the pipeline, this active learning approach could greatly improve the visualization results, starting anew from the summarization step.

As a semi-supervised system would benefit from large quantities of data, this quality would incentivize deploying the system online. In addition to improving accessibility of a temporal graph analysis pipeline, the system could benefit from generalizing rules between datasets of diverse disciplines. Thus, a final objective of online deployment would be central to improving the usefulness of our proposed system.
2.3 Research Approach

In order to automatically select the best graph visualization according to the context, we plan to conduct a series of usability studies on the supported graph analysis tasks. These tasks will include detecting intra- and inter-community patterns, as well as anomaly detection. Additionally, as the graph structures of different datasets can greatly differ, we will sample attributes of graph topology (e.g., power-law distribution vs. small world architecture) as possible variables. The rules underlying the relationship between visualization method and task and/or graph structure may even be learned through a simple classification model so that the findings can easily be included in a deployed system.

To derive domain knowledge from user interactions, we must first develop a graph exploration framework that is conducive to meaningful interaction. More specifically, the user interface must be designed in such a way that encourages the user to navigate through the graph components in a non-random manner. This will likely require more user behavior studies in which we ask the user qualitative questions about their interactions with the system on simplified datasets. Mining the user data may be done through a commercial JavaScript framework. After data is scraped, we may collect ‘ground truth’ information from the user with respect to areas of interest in the graph, such as known membership in communities or anomalous groups. This can then be used in model selection that can determine patterns of how a user explores graph summary subgraphs.

Building a system that may be deployed from a centralized server will involve heavy amounts of programming and planning from an engineering viewpoint. The database storing users’ graph data will need to be robust and very high capacity, which will require significant planning.

2.4 Staffing and Resources

The research team would include one research investigator and three research assistants. Each of the research assistants would be responsible for a separate objective, though the initial development of a prototype would require a combined effort. A significant portion of the $100,000 funds would be dedicated to securing a cloud-based computing cluster and database for deployment.
3 References


