Making Sense of Graph Query Results: Interactive Summarization and Exploration

Robert Pienta* Georgia Institute of Technology

Alex Endert† Georgia Institute of Technology

Shamkant Navathe‡ Georgia Institute of Technology

Duen Horng Chau§ Georgia Institute of Technology

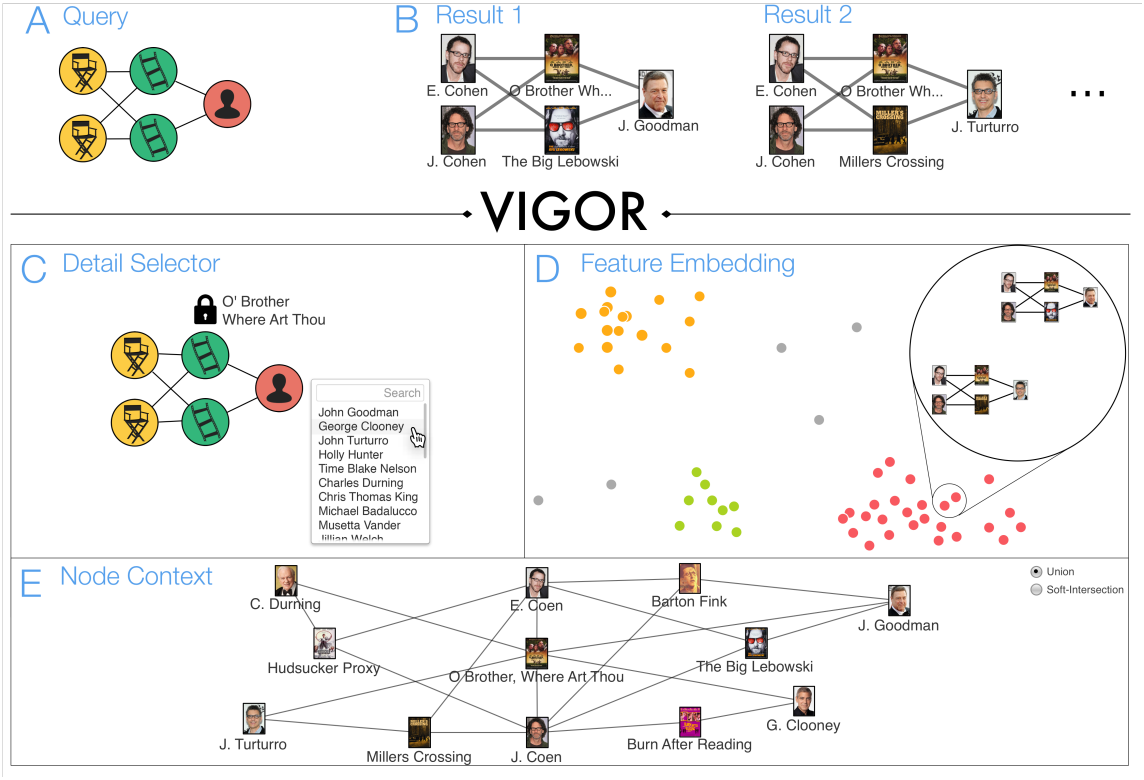

Figure 1: An analyst making sense of a movie graph dataset, after issuing a query (shown in A), which returns hundreds or more of results (as in B). Our VIGOR system helps analysts summarize and explore graph querying results through three views which improve the process of understanding results. **C**: the Narrowing Filter lets the analyst delve into the results by selecting specific node values (*O Brother, Where Art Thou?* selected as a value on the top film node in C). **D**: our Feature Embedding shows a high-level clustering of all the results based on result feature similarity. **E**: the Node Context summary shows the nodes shared across the current selection from C and D.

Abstract

Large data-rich network datasets are ubiquitous. Increasing research and development in graph databases offers analysts the ability to query large networks for interesting patterns. Graph querying allows analysts to quickly find matches for desired subgraphs from a large network (e.g., a clique of suspiciously communicating machines on a company intranet). Many of these systems do not support the exploration of results and leave the analyst with ineffective visualizations like tables and lists. We present our work in progress, VIGOR, an interactive visual system for summarizing and investigating large numbers of graph querying results.

Index Terms: K.6.1 [Management of Computing and Information Systems]: Project and People Management—Life Cycle; K.7.m [The Computing Profession]: Miscellaneous—Ethics

*pientars@gatech.edu

†endert@gatech.edu

‡sham@gatech.edu

§polo@gatech.edu

1 Introduction

Given the recent developments in graph databases and graph querying technologies [1, 3], new approaches are needed to facilitate discovery and exploration of results. In the data mining paradigm, a graph querying task is often considered finished when the results have been returned; but this is usually not the case for an analyst making sense of their data. Consider an analyst investigating a Rotten-Tomatoes[1] (RT) movie-actor-director graph. Two movies are connected if they are similar. Movie-to-director or movie-to-actor edges represent directing in and starring in that movie respectively. There can be a wide range of questions an analyst may want to ask when making sense of the results, such as: (a) which nodes come up most often in their results or nodes that appear commonly together (e.g., actors that often star in films together)? (b) are there common feature values shared across the results? (c) how can we group the results and identify representative instances for each group?

To answer these questions, often an analyst will create and iteratively refine a query based on the intermediate query results. Contemporary

[1] https://www.rottentomatoes.com/

Figure 2: (A) Table view from Neo4j, a commercial system; and (B) list view [3] used to display results. Even for modest sized queries, these conventional approaches require significant scrolling, and cannot easily reveal broader patterns and relationships among results.

systems provide basic methods for displaying results, such as tables or lists (as in Figure 2), which often cannot easily reveal the broader emergent patterns among the results, like the ones previously mentioned (e.g., whether results form topological groups). In addition, as the number of nodes per query and the amount of information per node increase, the amount of space required to show results in a table or list quickly overruns high-definition monitors.

Little investigation has been done in the space of graph query result visualization and exploration. We are working on VIGOR, an exploratory system to fill this research gap. This paper's remaining discussion will continue to use the RottenTomatoes dataset.

2 VIGOR Interface Design

The VIGOR system employs three key views:

- **Narrowing Filter** — As the analyst selects values of interest from the menu (as in Figure 1.A top film), the node-value lists for the remaining nodes are narrowed to fit the desired choices (in Figure 1 the list only shows actors who starred in O Brother, Where Art Thou and another co-directed film). An analyst may select any number of values for a node and explore the features of the current selection via a feature summary (see Figure 3).

- **Feature Embedding** — to help analysts group results and understand similar matches, VIGOR provides a high-level overview where each result is reduced to a single point by its features and then clustered with the other results. Classically, structural metrics like centrality, PageRank, degree, and connectivity have been used as features for embedding graphs [2]. We are experimenting with blending both structural and attribute features to improve the quality of our low-dimensional representation.

- **Node Context** — the results for a given query can induce complex structures like components of varying sizes, or sharing sets of common nodes. This view shows the union or soft-intersection of any selections made in the Feature Embedding or Narrowing Filter. By selecting union, VIGOR provides node-level context to the structure of the returned results, rather than keeping them completely separate. The soft-intersection (see Figure 4) scales the commonly shared nodes (and lightens uncommon values), such that commonly reoccurring nodes across the results are easier to see. In Figure 4-top, the query results for films starring Sean Connery or Nicolas Cage.

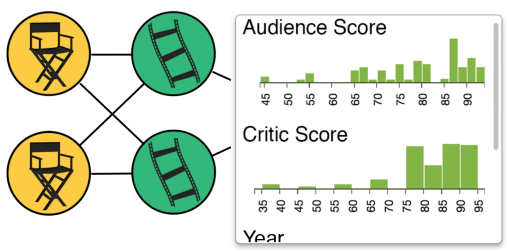

Figure 3: The features for each node in the Node Context view are summarized with small distributions. The displayed distributions are drawn from the matching results and updated as particular values are selected (narrowing the selected result set).

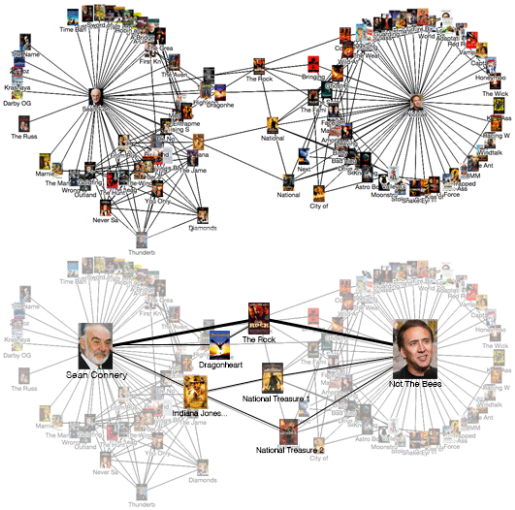

Figure 4: The Node Context view showing the results (top) of a query investigating films by Sean Connery or Nicholas Cage. Soft intersection (bottom) is used to highlight the nodes and edges which are shared across many results (which are the movies that star both actors).

An analyst can narrow down results via (1) lassoing groups of results in the Feature Embedding or (2) by choosing node-values in the Narrowing Filter. As the result-set is narrowed down the Node Context is updates to reflect the new content. Hovering over results in Feature Embedding highlights them in Node Context, while hovering over a node in Node Context highlights all results that contain that node.

3 Conclusions and Ongoing Work

We are in the process of creating and polishing series of techniques to improve the ease with which analysts can (1) explore their numerous graph query results by node values and (2) explore the patterns across results with our Feature Embedding view and feature distribution views. These techniques improve the ways in which an analyst may quickly explore a set of results.

We have completed many of the necessary features of VIGOR and are now designing and planning a user study, where we will investigate the usability, speed, and accuracy with which participants can perform common graph querying-related tasks. We will recruit analysts for our study, during which they will perform tasks on real network datasets from different domains like computer security and material science; wherein, not only the networks themselves vary greatly, but also the patterns and desired insight.

References

[1] R. Angles and C. Gutierrez. Survey of graph database models. *ACM Comput. Surv.*, 40(1):1:1–1:39, Feb. 2008.

[2] M. Berlingerio, D. Koutra, T. Eliassi-Rad, and C. Faloutsos. Network similarity via multiple social theories. In *Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on*, pages 1439–1440, Aug 2013.

[3] R. Pienta, A. Tamersoy, A. Endert, S. Navathe, H. Tong, and D. H. Chau. Visage: Interactive visual graph querying. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI '16, pages 272–279, New York, NY, USA, 2016. ACM.