VIGOR: Interactive Visual Exploration of Graph Query Results (VAST 2017)

ROBERT PIENTA, FRED HOHMAN, ALEX ENDERT, ACAR TAMERSOY, KEVIN ROUNDY, CHRIS GATES, SHAMKANT NAVATHE, DUEN HORNG CHAU

PRESENTED BY: RITESH SARKHEL, OMID ASUDEH, MONIBA KEYMANESH
Background

- What is graph querying?
- Why do we query graphs?
  - Detecting money laundering rings
  - Discovering near bipartite cores in auction fraud
  - Uncovering fraudulent near-clique reviews
Background

Results: subgraph matches
Motivation

- Different node features
- Arbitrary user queries
- Large number of results
- Shared nodes and edges among the results

Cannot understand underlying patterns
Cannot explore the result set

Solution: **VIGOR** Graph Query Visualization and Exploration System
Overview of VIGOR

Exemplar View

Fusion Graph (12 nodes shown, 2569 Results)

Embedded Results (68 Results Selected)

Minning evolutionary multi-branch trees from text streams

Paper 2013

Xirong Wang, Shixia Liu, Yangqiu Song, Baining Guo
At KDD-2013 on pages: 722 to 730
ACM ID: 2487603
Interface Overview

- Exemplar view
- Fusion graph
- Subgraph Embedding View
- Feature Explorer View
Alice enters a graph query

Cypher QL transforms query to a graph

“Find authors who have papers both in VAST and KDD”
Fusion graph

- VIGOR provides a hybrid view of how the query results match up with the input query.
- Selecting a result node highlights the relevant attributes related to that tuple.
Subgraph Embedding View

- Each square point represents a node in result-subgraph
- Similar results are spatially close in embedding space
- Points in same cluster are bounded by a concave hull
Subgraph Embedding View

Alice’s graph query on DBLP

~ 2500 results of (paper, author-list)

Result subgraphs clustered in a feature aware space

Considers all possible combinations of features and PCA to a two-dimensional space
Feature Explorer View

- VIGOR provides the feature-level flexibility
- Subgraph Embedding changes based on modified features
- Supports both continuous and discrete feature values
Design Rationale

- Exemplar View provides:
  - Fast error-checking for input query
  - Bottom-up exploration of result-set
  - Ability to start from a familiar result and on the fly query expansion or filtering

- Subgraph Embedding provides:
  - Top-down exploration of the result-set
  - Global view of the result set
  - Clustering similar results together help analysts get a macro-view
Design Rationale

- Feature-centric sense-making by Feature Explorer View
- VIGOR provides coordination among multiple views
  - Fusion graph combines top-down and bottom-up views of result set
  - Clicking on the node explains why that result was included
  - Hovering over the squares on Subgraph Embedding View shows detailed values
Methodology
Methodology contd.

1. Extract - Structural/Node features for each node
   - **Structural features:**
     - Node degree
     - Egonet edges
     - Egonet neighboring nodes
     - Clustering coefficients
   - **Node features:**
     - For author: number of co-authors, number of conferences

2. Vectorize - Merge the common features into per-result vectors

3. Aggregate & Normalize into Signature- by computing mean, variance, skewness and kurtosis for each feature for each node.
Merge common feature
4. Reduction and Clustering

- Reduce dimensionality to 2 using PCA / kernel-PCA / t-SNE
- Cluster results based on feature combinations using OPTICS
  - Canberra distance (weighted version of manhattan distance)

\[ d(p, q) = \sum_{i=1}^{n} \frac{|p_i - q_i|}{|p_i| + |q_i|} \]

where

\[ p = (p_1, p_2, \ldots, p_n) \text{ and } q = (q_1, q_2, \ldots, q_n) \]

- Canberra distance is sensitive to small changes near zero, which helps preserves small distances in the final reduction.
Architecture

- Client-server architecture using D3 and jQuery
- Backend in python
Experimental results

- User study
- Think-aloud explorative study
User Study 1: DBLP

Within-subject User Study

Participants: 12 (7 female, 5 male; ages 21 to 31)
Dataset: DBLP Co-authorship Network
Tasks: 4 tasks related to co-authorship and conferences
Measured: Task completion times and error rates

VIGOR vs neo4j

Slides adapted from Robert Pienta et al. @GaTech with the authors’ permission
Experimental results (contd.)

Slides adapted from Robert Pienta et al. @GaTech with the authors’ permission
Experimental results (contd.)

Slides adapted from Robert Pienta et al. @GaTech with the authors’ permission
Experimental results (contd.)

**User Study Tasks**

**Task 1**
- **Query**: Author → Paper → Conference
- **Objective**: Find the number of papers Daniel Keim has published at ICDM.

**Task 2**
- **Query**: Conference → Paper → Author
- **Objective**: Find all authors from 2015 or later at KDD, that have published at least two papers with 'entry' in the name.

**Task 3**
- **Query**: Conference → Paper → Author
- **Objective**: List the distinct number of INFOVIS publication groups that Tobias Shreck is in.

**Task 4**
- **Query**: Author → Paper → Conference
- **Objective**: For co-authors with a paper at INFOVIS and another at KDD, find the author with the most distinct papers.

Slides adapted from Robert Pienta et al. @GaTech with the authors’ permission
Experimental results (contd.)

- Confounding factors:
  - ✓ Software (VIGOR or Neo4j)
  - ✓ Complexity of the task
  - ✓ Software Order (VIGOR or Neo4j going first)

- ANOVA results show only varying the software produced significant improvement.
Experimental results (contd.)

User Study Results for VIGOR & Neo4j

- Task 1
- Task 2
- Task 3
- Task 4

<table>
<thead>
<tr>
<th>Average Task Time (s) (Shorter is better)</th>
<th>Average # of Errors (Shorter is better)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>150</td>
</tr>
<tr>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>1.5</td>
<td>2.0</td>
</tr>
</tbody>
</table>

Slides adapted from Robert Pienta et al. @GaTech with the authors’ permission
Experimental results (contd.)

Which Software Seemed...

- VIGOR
- Neo4j
- The Same

Number of Participants

- Easier to learn
- Easier to use
- More accurate
- Faster
- More enjoyable
- More liked
- More likely to use in future

Slides adapted from Robert Pienta et al. @GaTech with the authors’ permission
User Study 2: Cyber Security

- Network: Cyber Security network with 7,651 nodes and 384,182 edges
- Nodes connect clients of Symantec with Security incidents
  - Active signature
  - Passive signature
- Dataset contains > 11,000 incidents
- Participants: 3 Cyber Security experts from Symantec
- Number of queries: 2
Experimental results

**Query 1:** Compare two companies with at least one active and one passive signature where one took actions to every threat detected and one did not.

**Query 2:** Find companies that are inconsistent in taking actions when a threat is detected.
Experimental results (contd.)

Visualization of Query 1 results:

Security blindspots for Company “7” is detected using Fusion Graph.
Experimental results (contd.)

Visualization of Query 2 results:
A company with inconsistent behavior when an incident is reported
Pros/Cons

Pros:
- Exemplar-based interaction technique
- Feature-aware subgraph result summarization

Cons:
- User study only with few professional participants
- Users should be familiar with Cypher QL
- No arguments are provided for used structural features
- Insufficient analysis on clustering quality
- No evaluation on query creation and refinement
- Some information loss in the embedding process
- No information about the scope of the graph results that can be visualized (directed?, weighted?)
- User cannot filter based on the features in the feature explorer view
Thank You!