Graph-based Exploration of Non-graph Datasets

Udayan Khurana
IBM Research
ukhurana@us.ibm.com

work with Deepak Turaga, Srinivasan Parthasarathy
Graph Analytics

• Graphs capture interconnections or interactions.
• Graphs are omnipresent - social media, emails, news stories, financial records, system logs, biological structures, …
• Network analysis provide useful insights
  • Node centrality metrics tell relative importance of entities
  • Global network metrics reflect collective behavior.
• Last few years have seen a rapid growth graph database management systems and graph analysis software.
Non-graph Datasets

- Majority of world’s data is present in non-graph formats
  - Relational Databases, JSON, XML, CSV, Plain Text, …
- Extracting graphs from a dataset is hard:
  - Identifying the appropriate graph
  - Writing code to perform the extraction
  - Time, skill and effort from an analyst or data scientist
The process of manual graph extraction is based on visual inspection and understanding of the data/domain. However, it is (a) time consuming; (b) Requires coding/query writing; (c) Possible to miss out on useful graphs. E.g., In DBLP database:

Example of an extraction query (Co-authorship graph):

Nodes: Select Author.id from Author;
Graph Analytics Methodology

- Graph Analytics is performed in an iterative manner:
  - (1) graph extraction from the dataset
  - (2) running an analysis algorithm on the graph
  - (3) interpretation of results
  - (4) running a different algorithm on the same graph
  - (5) finding another graph from the dataset.
GraphViewer: Overview

GraphMapper

Graph Extraction

Graph Specification

Graph Enumeration

Data source

MCS

Compact Graph Array

Graph-Algo Matching

Execution and Summarization

Interpretation
Alternative Graph Extraction Approach: Enumeration

- We propose an automated graph enumeration approach.
- Given a schema,
  - We construct a *Schema Graph*.
  - Any path starting and ending at a terminal node describes a graph definition.
    - E.g., 9 -> 10 -> 11 -> 10 -> 9 corresponds to the co-author graph described earlier.
Extracting Graphs from Non-structured Data

- Unstructured data such as text doesn't provide a schema
- Approach: Use Natural Language Processing to annotate:
  - Entities: Persons, Locations, Organizations, etc.
  - Relationships: Subject-Object-Predicate, Co-occurrence
- For semi-structured data such as JSON, XML, CSV, we
  - Adopt dual approaches - structured and unstructured
  - Try to introduce structure as well as treat it as text
Metric Computation and Summarization

- Graph enumeration produces multiple graphs,
  - $S_G = \{G_1, G_2, \ldots \}$
- These are matched with different algorithms,
  - $S_A = \{A_1, A_2, \ldots \}$
- We run $O(|S_A| \times |S_G|)$ graph metric algorithms
  - Produces $O(|S_A| \times |S_G| \times |E|)$ results where $E$ is the set of all entities
- The user can arrange the results in different ways
  - Aggregate by entity, entity-type, graph or algo.
  - Filter, Sort and Join to perform data exploration.
  - Potentially use data cubes.
Demo

- View this demo at VLDB 2016 at the following times (Graph and Semistructured Data):
  - Tuesday 4:00-5:30 PM (Maple - 3a)
  - Wednesday 11:15-12:45 (Maple - 3b)
- [http://localhost:8080/GraphViewer](http://localhost:8080/GraphViewer)
Demo: Data Source

GRAPH VIEWER

Specify data source type and location

Source Type:
- Database
- JSON
- XML
- CSV
- Text

DBName: dblp1
Server: localhost
Port: 5432

Username: postgres
Password: ****

Display Schema
Generate Graph Spec

Database Name: dblp1
conference
- year
- name
- location
- id
- title
- cid

authorpublication
- pid
- aid

author
- name
- id

publication
- id
- title
- cid

ForeignKey [authorpublication.aid -> author.id]
ForeignKey [authorpublication.pid -> publication.id]
Demo: Graph Specification

**GRAPH VIEWER**

**Node Specification:**

<table>
<thead>
<tr>
<th>NS-ID</th>
<th>Node Set</th>
<th>Node Count</th>
<th>Node Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1</td>
<td>publication.id</td>
<td>1677</td>
<td>publication.title, publication.cid,</td>
</tr>
<tr>
<td>N2</td>
<td>conference.id</td>
<td>27</td>
<td>conference.year, conference.name, conference.location,</td>
</tr>
<tr>
<td>N3</td>
<td>author.id</td>
<td>2576</td>
<td>author.name,</td>
</tr>
</tbody>
</table>

**Graph Specification:**

<table>
<thead>
<tr>
<th>GID</th>
<th>Nodesets</th>
<th>Relationships</th>
<th>Est. Edges</th>
<th>Graph Summary</th>
<th>Graph Layout</th>
</tr>
</thead>
<tbody>
<tr>
<td>G1</td>
<td>N1</td>
<td>Path=publication.id, publication.title,</td>
<td>1677</td>
<td>Graph Summary</td>
<td>View Graph</td>
</tr>
<tr>
<td>G2</td>
<td>N1</td>
<td>Path=publication.id, publication.cid,</td>
<td>195007</td>
<td>Graph Summary</td>
<td>View Graph</td>
</tr>
<tr>
<td>G3</td>
<td>N2</td>
<td>Path=conference.id, conference.year,</td>
<td>75</td>
<td>Graph Summary</td>
<td>View Graph</td>
</tr>
<tr>
<td>G4</td>
<td>N2</td>
<td>Path=conference.id, conference.name,</td>
<td>249</td>
<td>Graph Summary</td>
<td>View Graph</td>
</tr>
<tr>
<td>G5</td>
<td>N2</td>
<td>Path=conference.id, conference.location,</td>
<td>147</td>
<td>Graph Summary</td>
<td>View Graph</td>
</tr>
<tr>
<td>G6</td>
<td>N3</td>
<td>Path=author.id, author.name,</td>
<td>2578</td>
<td>Graph Summary</td>
<td>View Graph</td>
</tr>
</tbody>
</table>

**Buttons:**

- Add Graph Definition
- Edit Graph Criteria
- Create Run Spec
- Clear Graph Spec
### Run Specification:

<table>
<thead>
<tr>
<th>Graph</th>
<th>Algo</th>
<th>Max Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>G1</td>
<td>LocalClusteringCoefficient</td>
<td>10</td>
</tr>
<tr>
<td>G1</td>
<td>BetweennessCentrality</td>
<td>10</td>
</tr>
<tr>
<td>G1</td>
<td>PageRank</td>
<td>10</td>
</tr>
<tr>
<td>G1</td>
<td>ClosenessCentrality</td>
<td>10</td>
</tr>
<tr>
<td>G2</td>
<td>LocalClusteringCoefficient</td>
<td>10</td>
</tr>
<tr>
<td>G2</td>
<td>BetweennessCentrality</td>
<td>10</td>
</tr>
<tr>
<td>G2</td>
<td>PageRank</td>
<td>10</td>
</tr>
<tr>
<td>G2</td>
<td>ClosenessCentrality</td>
<td>10</td>
</tr>
<tr>
<td>G3</td>
<td>LocalClusteringCoefficient</td>
<td>10</td>
</tr>
<tr>
<td>G3</td>
<td>BetweennessCentrality</td>
<td>10</td>
</tr>
<tr>
<td>G3</td>
<td>PageRank</td>
<td>10</td>
</tr>
<tr>
<td>G3</td>
<td>ClosenessCentrality</td>
<td>10</td>
</tr>
<tr>
<td>G4</td>
<td>LocalClusteringCoefficient</td>
<td>10</td>
</tr>
<tr>
<td>G4</td>
<td>BetweennessCentrality</td>
<td>10</td>
</tr>
<tr>
<td>G4</td>
<td>PageRank</td>
<td>10</td>
</tr>
</tbody>
</table>
Demo: Exploring Results

**GRAPH VIEWER**

**Browse Results**

*Run Complete*

Results are stored in the following schema. Please write a SQL query or populate one by clicking an option below.

```sql
Results(entityID, entityType, graph, algo, score)

SELECT entityid, AVG(score) FROM vals GROUP BY entityid order by avg(score) desc;
```

<table>
<thead>
<tr>
<th>entityid</th>
<th>avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>N2.91</td>
<td>1.00000000000000000000000000000000</td>
</tr>
<tr>
<td>N2.33</td>
<td>1.00000000000000000000000000000000</td>
</tr>
<tr>
<td>N3.1744</td>
<td>1.00000000000000000000000000000000</td>
</tr>
<tr>
<td>N3.2447</td>
<td>1.00000000000000000000000000000000</td>
</tr>
<tr>
<td>N2.86</td>
<td>1.00000000000000000000000000000000</td>
</tr>
<tr>
<td>N2.89</td>
<td>1.00000000000000000000000000000000</td>
</tr>
<tr>
<td>N2.85</td>
<td>1.00000000000000000000000000000000</td>
</tr>
<tr>
<td>N2.35</td>
<td>1.00000000000000000000000000000000</td>
</tr>
</tbody>
</table>
System Aspects

• Extensibility:
  • New data types; new analytics; different aggregators
• Storage efficiency:
  • Shared nodes and edges in multiple fetched graphs present redundancy.
  • We store the graphs efficiently in an overlaid manner
• Runtime computational sharing:
  • Multi-query optimization in relational databases eliminate computational redundancy
  • Sharing text parsing across multiple graphs
Contributions

• We propose a new methodology for performing graph-based analysis on non-graph datasets:
  • Through automated graph enumeration
  • Through automated code assembly and deployment
  • Combines an analyst’s input with automated specs
• It assists an analyst in graph-based analysis by:
  • Reducing time, skill and effort
• It is efficient in exploring different kinds of datasets through graph-based analytics
Research Problems

• Efficiently find graphs, subgraphs or nodes that satisfy a certain criteria:
  • For instance, “list graphs with density > d”.
  • Can we do it without extracting all possible graphs?
• Predict the more useful graph/analytics for a given task:
  • Use training examples to learn correlations.
  • Avoid generating all graphs.
• System efficiency in extraction and execution of:
  • Multiple graphs and multiple algorithms.