Graph Exploration: Taking the User into the Loop

Davide Mottin, Anja Jentzsch, Emmanuel Müller
Hasso Plattner Institute, Potsdam, Germany

2016/10/24
CIKM2016, Indianapolis, US
Tutorial outline

Background (5 min)
Graph models, subgraph isomorphism, subgraph mining, graph clustering

Exploratory Graph Analysis (35 min)

Focused Graph Mining (35 min)

Refinement of Query Results (35 min)

Real World-Use Case (15 min)
Linked Data graphs

Challenges and discussion
Refinement of Graph Query Results

Reformulation and Refinement

- Generate reformulations (explanations) for query with too-many too few results
- Explain results by providing summaries
- **User perspective**: even if the query is imprecise the system provides assistance

Top-k results

- Use user feedback to find the k results with the highest score
- **User perspective**: the results are potentially the most preferred items

Skyline queries
Reformulation and Refinement

Query (a graph)

The user query is too restrictive (few results) or too generic (many results)

Solution

- Change the query to include more/less results
  OR
- Summarize the results

Query Reformulation approaches: in Graph Databases (Mottin et al.), in connected networks (Vasilyeva et al.)

Result summarization approaches: top-k representative (Ranu et al.), keyword induced result summarization (Wu et al.)
Graph Query Reformulation

Results

Reformulations: query supergraphs

Exponential number of reformulations

Mottin, D., Bonchi, F. and Gullo, F. Graph Query Reformulation with Diversity. KDD, 2015
Graph Query Reformulation

Find $k$ meaningful reformulations:
1. Span all the results
2. Present different aspects of the results

Given a graph database $D$, query $Q$, and an integer $k$, find a set $Q^\star$ of $Q$ such that:

$$Q^\star = \text{arg max}_{Q} \left( \frac{1}{2} \sum_{Q' \in Q} |D_{Q'}| \right)$$

subject to $|Q| = k$.

Coverage:

$$cov(Q) = \left| \bigcup_{Q' \in Q} D_{Q'} \right|$$

Diversity:

$$\text{div}(Q', Q'') = |D_{Q'} \cup D_{Q''}| - |D_{Q'} \cap D_{Q''}|$$

Mottin, D., Bonchi, F. and Gullo, F. Graph Query Reformulation with Diversity. KDD, 2015
Graph Query Reformulation

Problem

Find a set $Q$ of $k$ reformulations that maximize a combination of coverage and diversity

$$f(Q) = \text{cov}(Q) + \lambda \sum_{Q', Q'' \in Q} \text{div}(Q', Q'')$$

$$Q^* = \arg \max_{Q \subseteq S_Q} \quad f(Q)$$

subject to $|Q| = k$.

Theorem (NP-hardness)

The problem reduces to **MAX-SUM Diversification** Problem, so it is NP-hard

Mottin, D., Bonchi, F. and Gullo, F. Graph Query Reformulation with Diversity. KDD, 2015
The Fast_MMPG Algorithm

\[ \Delta_f(Q, Q') = \text{marginal gain} \]
\[ \bar{\Delta}_f(Q, Q') = \text{upper bound} \]

Until the reformulation with the maximum upper bound and marginal gain is not found:

1. Expand the reformulation with the max upper bound
2. Prune Reformulations with marginal gain smaller than the upper bound so far

Mottin, D., Bonchi, F. and Gullo, F. Graph Query Reformulation with Diversity. KDD, 2015
**Why empty, Why so-many answers in graphs**

**Problem**
Given a query $Q$ and a graph $G$, restrict/enlarge the result set with minimal changes in the query.

---

Why empty, Why so-many answers in graphs

Why? Empty/Too Many

Change the query

Exponential variations!

Maximum Common Subgraph

Differential graph

Explinations

Modifications

Graphs and unexpected subgraphs

Answers to the new queries

Why empty, Why so-many answers in graphs

Cardinality estimation:
- Frequency of single edges
- Entropy

Generate candidates based on minimal modifications

Top-k representative queries

Select $k=2$ relevant objects

Top-2 answer: $g_1, g_2$

Two objects are close if they are similar

Object is relevant
Object is non-relevant

Ranu, S., Hoang, M. and Singh, A. Answering top-k representative queries on graph databases. SIGMOD, 2014
Top-k representative queries

Result of a query

Vector graph $\tilde{g}_i$: vectorial representation of $G_i$

**Example**: Binding compatibility with $m$ proteins, frequent subgraphs, belonged communities

Query: function from $\tilde{g}$ to $[-1,1]$, $q: \tilde{g} \rightarrow [-1,1]$

**Example**: Molecules with some properties, graphs with some structure, some community

Top-k Representative queries:

$$A = \arg \max_S \{ \pi_\theta(S) | S \subseteq R(q), |S| = k \}$$

where $R(q)$ = results of $q$, $\pi_\theta(S)$ = **representative power** of $S$, given threshold $\theta$

Ranu, S., Hoang, M. and Singh, A. Answering top-k representative queries on graph databases. SIGMOD, 2014
Representative power

\[ R(q) = \text{answers to the query} \]

- \( q \): query

\( \theta \)-neighborhood

- \( N_\theta (G) = \{ G' \in R(q) | d(G, G') \leq \theta \} \)
- \( \theta \): distance threshold
- \( d(G, G') \): graph edit distance

Given a set of graphs \( S \)

- Representative power of \( S \)
- \( \pi_\theta (S) = \frac{|U_{G \in S} N_\theta (G)|}{R(q)} \)

Represent the coverage of a graph neighborhood

\[ \pi(\{G_1, G_3\}) = \frac{7}{8} \]
\[ \pi(\{G_1, G_2\}) = \frac{4}{8} \]

Ranu, S., Hoang, M. and Singh, A. Answering top-k representative queries on graph databases. SIGMOD, 2014
Greedy algorithm

\[ G^* = \arg \max_G \{ \text{Score}(A \cup \{G\}) - \text{Score}(A) \} \]

\[ A = A \cup \{G^*\} \]

Repeat until \(|A| = k\)

**NP-hard** (graph edit dist)

**PTIME**

Update representative powers of ALL graphs

\[ N_\theta(G') = N_\theta(G') \setminus N_\theta(G^*) \]

Indexed with vantage points and clustering

1-1/e-Approximation

**Example**: If \(d_v(g_1, g_2) > \theta\), \(g_2 \notin N_\theta(g_1)\)

\[ |b - c| \leq a \Rightarrow \]

\[ d_v(g_1, g_2) = |d(v, g_1) - d(v, g_2)| \]

Ranu, S., Hoang, M. and Singh, A. Answering top-k representative queries on graph databases. SIGMOD, 2014
Summarizing graph results

**Query:** keyword query on graph
e.g., Jaguar, America, History

---

Wu, Y., Yang, S., Srivatsa, M., Iyengar, A. and Yan, X. Summarizing answer graphs induced by keyword queries. *PVLDB, 2013*
Summarizing graph results

Q = \{a,b,c\}

Answer graph

Summary graph

**Answer graph**: keyword nodes and intermediate nodes

**Summary graph Gs**:
- Preserve connections between keyword nodes
- Each node is a hypernode
- For any path in Gs there is a path in the union of answer graphs with the same label

Quality of a summary (coverage)
\[ \alpha = 2 \times \frac{M}{|Q|(|Q| - 1)} \]
\[ M = \text{number of covered keyword pairs} \]

Two problems
1. Minimum \( \alpha \)-summarization: find the **minimum size** summary which covers at least \( \alpha \)
2. \( K \)-summarization: find \( K \) 1-summaries with minimum total size that form a \( K \)-partition on the answer graph sets (no repeated answers)

Wu, Y., Yang, S., Srivatsa, M., Iyengar, A. and Yan, X. Summarizing answer graphs induced by keyword queries. *PVLDB, 2013*
Summarizing graph results

\[ Q = \{a, c, e, f, g\} \]

\[(a, c), \{G1, G2\}\]

\[(a, e, g), \{G1, G2\}\]

\[(a, e, g), \{G3\}\]

0.1-summary Gs1

0.3-summary Gs2

1-summary Gs3
Summarizing graph results algorithms

1-summarization
1. Based on dominance relation: a node n1 dominates n2 if they have the same label and each path from a keyword pair that contains n2 also contains n1
2. Discover dominance relation and remove dominated nodes until no change

α-summarization
1. Greedy heuristic: compute 1-summaries for all keyword paths
2. Merge summaries with the minimum merge cost (extra edges added)
3. Repeat until the desired α is reached

K-summarization
1. Select K answer graphs as centers
2. Refine the clusters merging answer graphs with minimum merge cost until convergence
3. Compute 1-summary graphs for each cluster

Wu, Y., Yang, S., Srivatsa, M., Iyengar, A. and Yan, X. Summarizing answer graphs induced by keyword queries. PVLDB, 2013
Top-k Results

- Large query results
- Find interesting exact and similar matches

Query

(Singer) (Song)

Solution

- Ranking the results
- Optionally diversifying the matching

- Diversified top-k graph pattern matching (Fan et al.)
- Exploiting relevance feedback in knowledge graph search (Su et al.)
- Top-k interesting subgraph discovery in information networks (Gupta et al.)
- Querying web-scale information networks through bounding matching scores (Jin et al.)
Diversified top-k graph pattern matching

Query:
Find good PM (project manager) candidates collaborated with PRG (programmer), DB (database developer) and ST (software tester).

Fan, W., Wang, X. and Wu, Y. Diversified top-k graph pattern matching. VLDB, 2013
Diversified top-k graph pattern matching

- Graph pattern matching revised
  - extend a pattern with a designated output node $u_0$
  - matches $Q(G)$: the matches of $u_0$
  - readily extends to multiple output nodes

- Problem:
  - Find (diversified) top-K matches for graph pattern matching with a designated output node.

Diversified top-k graph pattern matching

- **Relevance**
  - Relevant set $R(u,v)$ for a match $v$ of a query node $u$: 
  - all descendants of $v$ as matches of descendants of $u$
  - a unique, maximum relevance set
  - Relevance function
    - The more reachable matches, the better
    
    $$\delta_r(u,v) = |R(u,v)|$$

- **Top-k matching:**
  - find top-k match set that maximizes total relevance

$$\delta_r(S') = \arg \max_{S' \subseteq M_u(Q,G,u_o), |S'|=k} \sum_{v_i \in S'} \delta_r(u_o, v_i)$$

Match Diversification

Match diversity
  ◦ Diversity function: set difference of the relevant set

$\delta_d(v_1, v_2) = 1 - \frac{|R(u,v_1) \cap R(u,v_2)|}{|R(u,v_1) \cup R(u,v_2)|}$

Diversification: a bi-criteria combination of both relevance and diversity

$F(S) = (1 - \lambda) \sum_{v_i \in S} \delta_r(u_o, v_i) + \frac{2 \cdot \lambda}{k - 1} \sum_{v_i \in S, v_j \in S, i < j} \delta_d(v_i, v_j)$

  ◦ relevance: common neighbors, Jaccard coefficient...
  ◦ diversity: neighborhood diversity, distance-based diversity

Diversified Top-k Matching: find a set $S$ of matches for output node

$F(S') = \arg \max_{S' \subseteq M_u(Q,G,u_o)} F(S')$
Finding Top-k Diversified Matches

<table>
<thead>
<tr>
<th>V</th>
<th>R(u₀, v)</th>
<th>δr ()</th>
</tr>
</thead>
<tbody>
<tr>
<td>PM₁</td>
<td>{PRG₁, DB₁, ST₁, ST₂}</td>
<td>4</td>
</tr>
<tr>
<td>PM₂</td>
<td>{PRG₄, PRG₃, PRG₂, DB₂, DB₃, ST₂, ST₃, ST₄}</td>
<td>8</td>
</tr>
<tr>
<td>PM₃</td>
<td>{PRG₃, PRG₂, DB₂, DB₃, ST₃, ST₄}</td>
<td>6</td>
</tr>
<tr>
<td>PM₄</td>
<td>{PRG₃, PRG₂, DB₂, DB₃, ST₃, ST₄}</td>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>δd ()</th>
<th>PM₁</th>
<th>PM₂</th>
<th>PM₃</th>
<th>PM₄</th>
</tr>
</thead>
<tbody>
<tr>
<td>PM₁</td>
<td>0</td>
<td>10/11</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>PM₂</td>
<td>10/11</td>
<td>0</td>
<td>1/4</td>
<td>1/4</td>
</tr>
<tr>
<td>PM₃</td>
<td>1</td>
<td>1/4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PM₄</td>
<td>1</td>
<td>1/4</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

PM₁ and PM₃ are picked by TopKDiv as top-2 diversified matches.

\[
F'(PM₁, PM₃) = 0.5 \times (\frac{4}{11} + \frac{6}{11}) + 1 = 1.45
\]

PM₁ and PM₃ have no descendant matches in common, and influence a large part of the matches.
Top-k interesting subgraph discovery in information networks

- **Given**
  - Typed unweighted query
  - A heterogeneous edge-weighted information network
  - Edge interestingness measure

- **Find**
  - Top-k interesting subgraphs

Gupta, M., Gao, J., Yan, X., Cam, H. and Han, J. Top-k interesting subgraph discovery in information networks. ICDE, 2014
Top-k interesting subgraph discovery in information networks

- 3 new graph indexes for building a top-k solution
  - Graph topology index
  - Sorted edge lists
  - Graph maximum metapath weight index

Gupta, M., Gao, J., Yan, X., Cam, H. and Han, J. Top-k interesting subgraph discovery in information networks. ICDE, 2014
Skyline Queries

- Prune a search space of large numbers of multi-dimensional data items to a small set of interesting items

Solution

- Eliminating items that are dominated by others

- Dynamic skyline queries in large graphs (Zou et al.)
- Efficient subgraph skyline search over large graphs (Zheng et al.)
Dynamic skyline queries

- Users can specify different sets of query points
  - Offer users more flexibility in specifying their search criteria
- Skylines are dynamically updated

Naïve approach:
- Computing all new vectors according to the query points and then searching the skylines over the generated vectors
Dynamic skyline queries in large graphs

Shared Shortest Path (SSP) pruning

- if there exists at least one joint (common) vertex $v'$ among all shortest paths between $v$ and $q_i$, $v$ can be pruned safely

Zou, L., Chen, L., Özsu, M.T. and Zhao, D. Dynamic skyline queries in large graphs. DASFAA, 2010