Graph Exploration: From the User to Large Graphs

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

May 14, 2017
SIGMOD 2017, Chicago, US
Who we are

Davide Mottin
- graph mining, novel query paradigms, interactive methods

Emmanuel Müller
- graph mining, stream mining, clustering and outlier mining on graphs, streams, and traditional databases
Big data and novice users
Data exploration

Efficiently extracting knowledge from data even if we do not know exactly what we are looking for

Idreos et al., Overview of Data Exploration Methods, SIGMOD 2015
The importance of graphs

Social Networks

Recommendation Graphs

Knowledge Graphs

Complex
Ubiquitous
Large
Valuable
Lost in the graph?
Current: Visualization tools

Several visualization tools:
• General: Gephi, GraphViz, ...
• Biological: Cytoscape, Network Workbench
• Social: EgoNet, NodeXL, ...
• Relational: Tulip

but

• No Scalability to large networks!
• No for novice users
• Limited expressivity
Current: Query languages

### SELECT

```sql
SELECT ?name ?email
WHERE
{
  ?person a foaf:Person .
  ?person foaf:name ?name .
}
```

### Query languages ARE:
- Expressive
- Powerful
- Scalable
- Compact

### not

- Not user friendly
- No guided search
- Not interactive
- Not scalable

---

**SPARQL**
```sql
g.V().hasLabel('movie').as('a','b').
where(inE('rated').count().is(gt(10))).
select('a','b').
by('name').
by(inE('rated').values('stars').mean()).
order().
by(select('b'),decr). limit(10)
```

**GREMLIN**
```sql
MATCH (node1:Label1)--> (node2:Label2)
WHERE node1.propertyA = {value}
RETURN node2.propertyA, node2.propertyB
```

**CYPERHER**
This tutorial is about ...

- Algorithms for helping the user finding the wanted information
- Approximate search on graphs to assist the user in finding the information
- Interactive methods on graphs based on user feedback
- Automatically discovery of portions of graphs using examples

NOT about

- Visualization methods for graphs
- Query languages and semantics
- Efficient indexing methods
- Pure machine learning on graphs
Our graph exploration taxonomy

- Exploratory Graph Analysis
- Focused Graph Mining
- Refinement of Query Results
Graph exploration taxonomy

Exploratory Graph Analysis

Other politicians like Angela Merkel?

Two exploratory options:
1. An imprecise query
   - Merkel
     President of
     ?

2. A by-example query
   - Merkel
     Chancellor
     Germany

Query is an example

- Schröder
- chancellor
- Germany
- Schröder
- chancellor
- Germany
Graph exploration taxonomy

Focused Graph Mining

How can I see only the part of the graph I’m interested in?

Targeted analysis on large graphs
1. Focused graph clustering
2. Space restriction methods
3. Graph Reweighting

They all like the Chicago Bulls

Ego-net analysis
Graph exploration taxonomy

Refinement of Query Results

Where is this molecule contained?

Dealing with generic queries:
1. Reformulation and refinement
2. Top-k results
3. Skyline queries

Dominance relation

Query:

\[ \text{OH} - \text{S} - \text{O} \]

Results:

- 270 results
- 220 results
The graph exploration ... graph

Exploratory graph analytics

- [Mottin14]
- [Jayaram15]
- [Khan13]
- [Yang14]
- [Fan10]
- [Ma14]

Refinement of query results

- [Ranu14]
- [Wu13]
- [Fan13]
- [Vasilyeva16]
- [Gupta14]
- [Zou10]
- [Mottin15]

Focused graph mining

- [Tong06]
- [Epasto15]
- [Staudt14]
- [Perozzi14]
- [Iglesias14]
- [Iglesias13]
Tutorial outline

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

Exploratory Graph Analysis (20 min)

Focused Graph Mining (20 min)

Refinement of Query Results (20 min)

Challenges and discussion
We are here

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

Exploratory Graph Analysis (20 min)

Focused Graph Mining (20 min)

Refinement of Query Results (20 min)

Challenges and discussion
Graphs

\[ G = (V, E, \phi) \]

Vertices \quad Edges \quad Probability function

\[ \lambda: V \cup E \rightarrow \Sigma \]

- Undirected Graphs
  - Co-authorship, Roads, Biological
- Directed graphs
  - Follows, ...
- Labeled Graphs
  - Knowledge graphs, ...
- Probabilistic graphs
  - Causal graphs
Graph databases (set of graphs)

\[ D = \{G_1, G_2, \ldots, G_n\}, G_i = (V_i, E_i, l_i), l_i: E_i \cup V_i \rightarrow \Sigma \]

Set of small labeled graphs
Chemical compounds, Business models, 3D objects
Graph Isomorphism

Given two graphs, $G_1: \langle V_1, E_1, l_1 \rangle$, $G_2: \langle V_2, E_2, l_2 \rangle$ $G_1$ is isomorphic to $G_2$ iff exists a bijective function $f: V_1 \rightarrow V_2$ s.t.:

1. For each $v_1 \in V_1$, $l(v_1) = l(f(v_1))$
2. $(v_1, u_1) \in E_1$ iff $(f(v_1), f(u_1)) \in E_2$
A graph $Q: \langle V_Q, E_Q, l_Q \rangle$ is subgraph isomorphic to a graph $G: \langle V, E, l \rangle$ if exists a subgraph $G' \subseteq G$, isomorphic to $Q$. 

\[ Q \quad G' \quad G \]
Graph Clustering and Community Detection

**Given:** graph with nodes, edges, labels

\[ G = (V, E, l) \]

- Vertices
- Edges
- Labeling function \( l: V \cup E \rightarrow \Sigma \)

**Discover:** a partitioning of communities

\[ C = \{C_1, C_2, C_3, \ldots, C_k\} \]

- Optimize a given quality criterion \( Q(C) \), e.g. **Modularity** or other measures
- Is an **NP-hard problem** to find the optimal partitioning
We are here

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

Exploratory Graph Analysis (20 min)

Focused Graph Mining (20 min)

Refinement of Query Results (20 min)

Challenges and discussion
Exploratory Search

Approximate Graph Search

- Given an imprecise query find the closest answers to that query
- **User perspective:** no need to know about the entire details of the data

Searching by Example

- Given an example from the results, find the other results of an unspecified query
- **User perspective:** it is not necessary to know how to describe the results
Approximate Graph Search

Query (a graph)

Graph

Solution

• The user might be imprecise in the search terms

• Find (partial) correspondence from the query to the graph

Structural mapping: Strong-simulation (Ma et al.)
Node similarity approaches: P-homomorphism (Fan et al.), Nema (Khan et al.)
Probabilistic approaches: SLQ (Yang et al.)
Subgraph isomorphism issues

(Sub)Graph Isomorphism might be too restrictive

No MATCH
(different label)

No MATCH
Node v is not matched

Fan, W., Li, J., Ma, S., Wang, H. and Wu, Y.. Graph homomorphism revisited for graph matching. PVLDB, 2010
Strong simulation

Revise subgraph isomorphism:
Instead of bijection, compute a binary relation between nodes

Nodes with the same structural "role" are matched

Ma, S., Cao, Y., Fan, W., Huai, J. and Wo, T. Strong simulation: Capturing topology in graph pattern matching. *TODS, 2014*
Strong simulation

Given $Q: \langle V_q, E_q, l_q \rangle$ and data graph $G: \langle V, E, l \rangle$, a binary relation $S \subseteq V_q \times V$ is said to be a dual simulation if

- for each $(u, v) \in S$, $l(u) = l(v)$
- for each $v \in V_q$ exists a node $u \in V$ s.t. $(v, u) \in S$
  - for each edge $(v, v') \in E_q$, there exists an edge $(u, u') \in E$ such that $(v', u') \in S$
  - for each edge $(v'', v) \in E_q$, there exists an edge $(u'', u) \in E$ such that $(v'', u'') \in S$

- The matching subgraph is:
  - connected graph
  - the diameter is not larger than twice the diameter of the query

Ma, S., Cao, Y., Fan, W., Huai, J. and Wo, T. Strong simulation: Capturing topology in graph pattern matching. *TODS, 2014*
Properties of Strong Simulation

If Q matches G, via subgraph isomorphism, then Q matches G, via strong simulation.

If Q matches G, via strong simulation, then Q matches G, via dual simulation.

If Q matches G, via dual simulation, then Q matches G, via graph simulation.

Ma, S., Cao, Y., Fan, W., Huai, J. and Wo, T. Strong simulation: Capturing topology in graph pattern matching. *TODS, 2014*
NeMa

Relax p-homomorphism:
• Structure and some labels are unknown
• Node closed in the query must be closed in the graph

The structure is not fixed anymore but similar to the query

**NeMa: compute node vectors**

\[ R_G(u) = \{ (u', w_u(u')) \} \]

where \( w_u(u') = \begin{cases} 
\alpha^d(u, u') & d(u, u') \leq h \\
0 & \text{otherwise}
\end{cases} \)

**Convert node \( u \) into a vector of neighbors**

**Distance less than \( h \) (h-hop neighbor)**

**Vector of nodes at distance \( \leq h \) from \( a \)**

\( h = 2, \alpha = 0.5 \)

\[ R_G(a) = \{ (b, 0.5), (c, 0.5), (d, 0.5), (e, 0.25) \} \]

**Khan, A., Wu, Y., Aggarwal, C.C. and Yan, X. Nema: Fast graph search with label similarity. PVLDB, 2013**
**NeMa**

**Problem**
Given Q and G, find the mapping $\phi$ with the minimum cost $C(\phi)$

\[ C(\phi) = \sum_{v \in V_Q} \text{cost}(v, \phi(v)) \]

\[ \text{cost}(v, u) = \Delta_L(l(v), l(u)) + \sum_{v' \in N(v)} \Delta_+(w_v(v'), w_u(u')) \]

**Solution**
Solved with a belief propagation approach

Similar to NEMA
Assume that a match is obtained by a sequence of transformations of the query nodes into the graph

Yang, S., Wu, Y., Sun, H. and Yan, X. Schemaless and structureless graph querying. PVLDB, 2014.
Model on transformations

\[ F_V(v, \phi(v)) = \sum_i \alpha_i f_i(v, \phi(v)) \]
\[ F_E(e, \phi(e)) = \sum_i \beta_i f_i(e, \phi(e)) \]

\[ P(\phi|Q) \propto \exp\left( \sum_{v \in V_Q} F_V(v, \phi(v)) + \sum_{e \in E_Q} F_E(e, \phi(e)) \right) \]

Problem
- How to learn the parameters \( \alpha_i, \beta_i \)?
- How to find the matching with the highest score?

Yang, S., Wu, Y., Sun, H. and Yan, X. Schemaless and structureless graph querying. *PVLDB, 2014.*
Querying by Example

**Query (an example)**

**Solution**
- The user query is an example result
- Find results that are similar to the one in input

**Graph**

**Exemplar Queries** (Mottin et al.), GQBE (Jayaram et al.)

**NOT approximate queries:**
A result to an approximate query is the closest possible to the query itself
Exemplar Queries

**Input:** $Q_e$, an example element of interest

**Output:** set of elements in the desired result set

**Exemplar Query Evaluation**

- evaluate $Q_e$ in a database D, finding a sample $s$
- find the set of elements $a$ similar to $s$ given a *similarity relation*
Compute the answers using **subgraph isomorphism** or **strong simulation**

Exemplar Queries

- **Q1**: Compute the answers using **subgraph isomorphism** or **strong simulation**

**Google** → **YouTube** → **Menlo Park**

**IT Companies**

- **Yahoo!** → **Tumblr**
- **CBS** → **Paramount**

**Search Engines**

- **Freebase** → **S. Clara County**
- **S. Mateo** → **California**
- **NYC** → **USA**

**Business**

- **S. Clara County** → **New York**
- **Menlo Park** → **Q1**

**Broadcasting**

- **hasWebsite** from **Yahoo!** to **S. Clara County**
- **acquired** from **CBS** to **S. Clara County**
- **acquired** from **YouTube** to **Menlo Park**

**D. MOTTIN, E. MÜLLER**
Computing exemplar queries

Pruning technique:
- Compute the neighbor labels of each node
  \[ W_{n,a,i} = \{ n_1 | l(n_1, n_2) = a \ \forall \ n \in N_{i-1}(n) \} \]
- Prune nodes not matching query nodes neighborhood labels
- Apply the technique iteratively on the query nodes

Labels at distance 1

\[ v \text{ neighborhood} = \{(B,1)\} \]
\[ u \text{ neighborhood} = \{(A,1)\} \]

No Match

Mottin, D., Lissandrini, M., Velegrakis, Y. and Palpanas, T. Exemplar queries: Give me an example of what you need. PVLDB 2014
Graph query by example (GQBE)

In GQBE Input is a set of (disconnected) entity mention tuples

\[ Q = (Google, S. Mateo) \]

Results = (Yahoo, S. Clara) (CBS, New York)

Jayaram, N., Khan, A., Li, C., Yan, X. and Elmasri, R. Querying knowledge graphs by example entity tuples. TKDE, 2015
**GQBE**

1. Find the maximum query graph
   - Neighborhood Graph with $m$ edges having the maximum weight
2. Find all the answers subgraph isomorphic to the query graph
3. Rank the answers and return the top-$k$ tuples

**Answer score:**
- Sum of query graph weights
- Similarity match between edges in the answer and the query

$$\text{match}(e, e')= \begin{cases} 
\frac{w(e)}{|E(v)|} & \text{if } u = f(u) \\
\frac{w(e)}{|E(v)|} & \text{if } v = f(v) \\
\frac{w(e)}{\min(|E(u)|, |E(v)|)} & \text{if } u = f(u), v = f(v) \\
0 & \text{otherwise}
\end{cases}$$

Jayaram, N., Khan, A., Li, C., Yan, X. and Elmasri, R. Querying knowledge graphs by example entity tuples. *TKDE, 2015*
We are here

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

Exploratory Graph Analysis (20 min)

Focused Graph Mining (20 min)

Refinement of Query Results (20 min)

Challenges and discussion
Graph Mining – a very broad topic

- Link Prediction
- Community Detection
- Anomaly Detection
- Frequent Subgraph Mining
- Graph Partitioning
- ... many more ...

D. MOTTIN, E. MÜLLER
Graph Mining Focused on User Interest

We consider “user interest” at a major tool for adaptive graph mining

- In contrast to raw analysis of graphs (i.e. with no or very little user interaction)
- Example (modularity based clustering):

Given a graph
discover best partitioning of the nodes

Optimize a given quality criterion \( Q(C) \),
e.g. Modularity or other measures

- Where is the user interest in such definitions?
- How to include the user into the loop?
- How do we need to change the algorithmic search?
Focus: Given a Set of Query Nodes

Given Q nodes (by the user)

How can we find the center-piece node that has direct or indirect connections to all or most of these nodes?

• Neither a clustering of nodes
• Nor the shortest path between pairs of nodes
• Nor any other graph mining method (with lack of user input)

H. Tong & C. Faloutsos: Center-Piece Subgraphs: Problem Definition and Fast Solutions. (KDD 2006)
Focused Communities: Given a Set of Seed Nodes

Traditional detection of communities as internally dense subgraphs (e.g. measured by modularity or conductance)

Given seed nodes (by the user)

Perform selective search for communities

local community detection
seed set expansion

- Global search is not appropriate for such local/selective models
- Communities may overlap or coincide

C. Staudt, Y. Marrakchi, H. Meyerhenke: Detecting Communities Around Seed Nodes in Complex Networks (BigData 2014)
Egoistic Focus on Yourself: Ego-Nets

For a given node consider their neighbors and the connections among these neighbors

Compute ego-nets for each given node that is of interest.

Useful for link prediction, community detection, anomaly detection, and many more, as pre-processing (feature extraction).

Epasto et al. Ego-Net Community Mining Applied to Fried Suggestion. (VLDB 2015)
Different graph mining techniques
- Clustering / graph partitioning / ...
- Community detection and anomaly detection

Used assumption: **Homophily** has to be fulfilled for **all** the attributes

Problem: **disassortative mixing** [Newman 2003] hinders the detection of communities (i.e. similarity assessment of nodes)

**Solution:** Selection of relevant views ensuring homophily

Multiple Views in Attributed Graphs

Different structures depending on the subset of attributes
Multiple Views in Attributed Graphs

Different structures depending on the subset of attributes

age
income
shoe size
#children

outlier
Specialized Approaches

Frequent subgraph mining, subspace clustering ...

- Local selection of the attributes
- Individual subgraphs
First Idea: Local Context Selection

**Local Context:**
- Subset of relevant attributes
- Selection w.r.t. a subgraph

**How to define a local context for each node?**

**How to efficiently select only the relevant attributes?**

**Model dependent solution for community outlier mining**
- Statistical test of attribute value distribution for each local context
- Measure deviation of each node w.r.t. its local context only

Iglesias et al. Local Context Selection for Outlier Ranking in Graphs with Multiple Numeric Node Attributes (SSDBM 2014)
Selection of Congruent Subspaces (ConSub)

**Definition: Congruent subspaces**
- **Mutual similarity** between attribute values in subspace $S$
- **Significantly more edges** than expected by a random distribution

**Constraint Subgraph $G_{C,S}$**
- Set of constraints formed by all the pairs $(I_j = [low_j, high_j], A_j \in S)$

$S = \{\text{shoe size}\}$
nodes with $8 \leq \text{shoe size} \leq 9$

small number of edges

Iglesias et al. Statistical Selection of Congruent Subspaces for Mining Attributed Graphs (ICDM 2013)
Selection of Congruent Subspaces (ConSub)

**Definition: Congruent subspaces**
- **Mutual similarity** between attribute values in subspace $S$
- **Significantly more edges** than expected by a random distribution

**Constraint Subgraph** $G_{C,S}$
- Set of constraints formed by all the pairs $(I_j = [low_j, high_j], A_j \in S)$

$S = \{\text{age}, \text{income}\}$
- Nodes with $45 \leq \text{age} \leq 60$ and $1900 \leq \text{income} \leq 4500$

*high number of edges*
Focus on User Preference

Examples for user preference:
- attribute weighting
- examples of similar nodes
- some notion of similarity

examples of similar nodes

attribute weighting

age
income
shoe size
#children
...

SIGMOD 2017 TUTORIAL
D. MOTTIN, E. MÜLLER
53
Focused Selection of Subsaces (FocusCO)

Decoupled mining for given user preference

1. Infer similarity measure
2. Re-weighting of graph edges
3. Community detection & community outlier mining

(3) applicable for various community detection models

Perozzi et al. Focused Clustering and Outlier Detection in Large Attributed Graphs (KDD 2014)
Knowledge Discovery by Focused Graph Mining

Example Sociology:

hypothesis testing vs. hypothesis generation
We are here

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

Exploratory Graph Analysis (20 min)

Focused Graph Mining (20 min)

Refinement of Query Results (20 min)

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

- Optimize one single objective at a time when finding results of a query
- User perspective: show only those nodes/graphs that are no worse than others

Not in this tutorial 😞
Reformulation and Refinement

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

Solution

- Change the query to include more/less results
- 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. \( \text{cov}(Q) = \sum_{Q_0 \in S} f(Q) \) subject to \( |Q| = k \).

2. Present different aspects of the results.

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

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

Explinations

Maximum Common Subgraph

+ Differential graph

Modifications

Graphs and unexpected subgraphs

Answers to the new queries

Top-k representative queries

Graphs are points in a metric space with d as a distance function

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 $\theta$
Representative power

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

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

Given a set of graphs \( S \)
- Representative power of \( S \)
- \( \pi_\theta(S) = \frac{|\bigcup_{G \in S} N_\theta(G)|}{R(q)} \)

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

Represent the coverage of a graph neighborhood

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**: keyword nodes and intermediate nodes

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

**Quality of a summary (coverage)**  
\[ \alpha = 2 \times \frac{M}{|Q|(|Q| - 1)} \]  
\( M \) = 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

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

\[
\begin{align*}
&Q = \{a, c, e, f, g\} \\
&('a, c'), \{G1, G2\} \\
&('a, e, g'), \{G1, G2\} \\
&('a, e, g'), \{G3\} \\
&0.1\text{-summary } Gs1 \\
&0.3\text{-summary } Gs2 \\
&1\text{-summary } Gs3
\end{align*}
\]
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

\[ \alpha \text{-summarization} \]
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 \( \alpha \) is reached

\[ K \text{-summarization} \]
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**

![Graph Query Example]

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

Find matches using graph simulation, which computes a binary relation on the pattern nodes in Q and their matches in G

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 \( u \)
- **Relevance function**
  - The more reachable matches, the better
    \[
    \delta_r(u, v) = |R(u, v)|
    \]

- **Top-k matching, k-match maximizing**

**Pattern Q**

**Relevance**

\[
\delta_r(S) = \arg \max_{S' \subseteq M(u(Q,G,u_o),|S'|=k} \sum_{v_i \in S'} \delta_r(u_o, v_i)
\]

**Diversity**

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

\[
F(S) = (1 - \lambda) \sum_{v_i \in S} \delta'_r(u_o, v_i) + \frac{2 \cdot \lambda}{k - 1} \sum_{v_i \in S, v_j \in S, i < j} \delta_d(v_i, v_j)
\]

Finding Top-k Matches (acyclic)

Starting propagation from DB2, after propagation, parts of the vectors are as below.

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PM1</td>
<td>&lt;XPM1 = XPRG1 ∧ XDB1, Φ, 0, 2&gt;</td>
</tr>
<tr>
<td>PM2</td>
<td>&lt;XPM2 = ((XPRG3 = true) V (XPRG4=true)) ∧ XDB2=true, {DB2, PRG4, PRG3}, 3, 3&gt;</td>
</tr>
<tr>
<td>PM3</td>
<td>&lt;XPM3 = (XPRG3 = true) ∧ (XDB3=true), {DB2, PRG3}, 2, 2&gt;</td>
</tr>
<tr>
<td>PM4</td>
<td>&lt;XPM4 = XPRG4 ∧ XDB1, Φ, 0, 2&gt;</td>
</tr>
<tr>
<td>PRG1</td>
<td>&lt;XPRG1 = XPRG1 ∧ XDB1, Φ, 0, 2&gt;</td>
</tr>
<tr>
<td>PRGj (j ∈ [3,4])</td>
<td>&lt;XPRGj = XPRGj ∧ XDB1, Φ, 0, 2&gt;</td>
</tr>
<tr>
<td>DB2</td>
<td>&lt;XDB2 = XDB2, Φ, 0, 0&gt;</td>
</tr>
<tr>
<td>DBk (k ∈ [1,3])</td>
<td>&lt;XDBk = XDBk, Φ, 0, 0&gt;</td>
</tr>
</tbody>
</table>

PM2 is verified to be a valid match, and its relevant set includes \{DB2, PRG4, PRG3\}, which is the largest relevant set compared with other PMs.

Early termination condition is met.
We are here

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

Exploratory Graph Analysis (20 min)

Focused Graph Mining (20 min)

Refinement of Query Results (20 min)

Challenges and discussion
Approximate Queries

- User query is imprecise
- User query is an example result

By-Example methods

- Only need partial knowledge on the data
- No need for complicate query languages (use examples, partial descriptions)
- The query adapts to user need
- Enable exploratory search by using small queries on the data
Challenges for Exploratory Graph Analysis

**Database**
- Unsupported in most of the current graph databases
- No "universal" index to answer multiple type of queries
- Partitioning methods for approximate query answering

**Data mining**
- User interactivity in the exploration process
- No solutions for probabilistic graphs
- Respond to queries in dynamic graphs
- Find examples in streaming settings

**Information retrieval**
- Exploiting query logs for personalized query answering
- Retrieve results in form of documents converting the query structures
Summary of Focused Graph Mining

The focus on individual user interest
... as **Query** to the Graph Mining System
... as **Seed Node(s)** for Local Search
... as **Attributes** and **Weights**

- get or infer user interest → unexpected results
- interactive exploration → intuitive parametrization
- adaptive graph mining → individual local search
Challenges for Focused Graph Mining

User interactivity in the graph mining process
- unsupported in most of the current graph mining algorithms
- huge variety of user interactions possible
- feedback loop needs to be unified and become exchangeable

Revolution of formal models and search algorithms
- insufficient extensions of existing models and algorithms
- adaptive steering of algorithms vs. fixed parametrization
- evaluation of algorithms with user studies

Scalability of algorithms for real-time interaction
- NP-hard problems, heuristic algorithms, ..., still not scalable
- exploit the user interest for pruning the search space
Summary of Refinement of Query Results

Refinement

• The user query is too restrictive or too generic

Top-k Results

• Queries typically have inexact matches

Skyline Queries

• Find small set of interesting items with many dimensions and incremental updates

- The user might have a very generic idea of how to describe the structure of interest
- The system guides the user towards the answer with simple steps
- The results are explained with reformulations
- The queries can be inexact
Challenges for Refinement of Query Results

- Profiling of queries for optimized performance
- Provenance and explainability of queries
- Managing uncertainty in data

- Personalized reformulations and interactivity
- Facet search discovery in graphs
- Learning of user preferences while refining

- Real time performance not achieved
- Avoiding traverse the entire space using query workloads and query logs
The missing tiles in graph exploration

- Interactivity
- Adaptivity
- Personalization
- Scalability
Slides: https://hpi.de//mueller/tutorials/graph-exploration-sigmod.html
References


References


[Staudt14] C. Staudt, Y. Marrakchi, H. Meyerhenke: Detecting Communities Around Seed Nodes in Complex Networks (BigData 2014)


[Iglesias14] Iglesias et al. Local Context Selection for Outlier Ranking in Graphs with Multiple Numeric Node Attributes (SSDBM 2014)

[Iglesias13] Iglesias et al. Statistical Selection of Congruent Subspaces for Mining Attributed Graphs (ICDM 2013)

[Perozzi14] Perozzi et al. Focused Clustering and Outlier Detection in Large Attributed Graphs (KDD 2014)
References


[Ranu14] Ranu, S., Hoang, M. and Singh, A. Answering top-k representative queries on graph databases. SIGMOD, 2014


