Where we are

Background (15 min)
- Graph models
- Subgraph isomorphism
- Subgraph mining
- Graph clustering

Exploratory Graph Analysis (40 min)

Focused Graph Mining (40 min)

Refinement of Query Results (40 min)

Machine Learning and Visualization (40 min)

Challenges and discussion
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

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

Graph Simulation [Milner 1989]

Poly-time (cubic)
Strong simulation

Without Locality: unconnected and unbounded graphs

Without Duality: Trees match cycles

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*
Graph homomorphism

Revise graph homomorphism: match paths

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

- Matches paths instead of single edges
- Similarity matrix between nodes $M$ over $Q$ and $G$, $M(u,v)$ similarity score of node $u$ in $Q$ and $v$ in $G$.
- Similarity threshold $\xi$

Fan, W., Li, J., Ma, S., Wang, H. and Wu, Y. Graph homomorphism revisited for graph matching. PVLDB, 2010
1-1 P-Homomorphism

**Injective** P-Homomorphism mapping from Q and G

Fan, W., Li, J., Ma, S., Wang, H. and Wu, Y. Graph homomorphism revisited for graph matching. PVLDB, 2010
Queries with (1-1)P-Homomorphism

Maximum cardinality problem (CPH)
- Return the (1-1)P-hom mapping $\rho$ with maximum $\text{Card}(\rho)$.
- The cardinality of $\rho$-hom mapping from a subgraph $G' = (V', E', L')$ of $Q$ to $G$:
  - $\text{Card}(\rho) = \frac{|V'|}{|V_Q|}$

Maximum Overall similarity (SPH)
- Return the (1-1)P-hom mapping $\rho$ with maximum $\text{Sim}(\rho)$.
- The overall similarity of $\rho$-hom mapping from a subgraph $G'$ of $Q$ to $G$:

Decision problems
NP-hard for DAGs

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

Algorithm compMaxCard(G₁, G₂, M, ξ)

- **Input**: G₁ = (V₁, E₁, L₁), G₂ = (V₂, E₂, L₂), similarity matrix M, similarity threshold ξ
- **Output**: a P-hom mapping from subgraph of G₁ to G₂
- **Procedure**
  - initialize matching list for each node in G₁
  - compute the transitive closure of G₂ (connect two nodes if a path exists)
  - starting from a match pair, recursively choose and include new matches to the match set until it can no longer be extended, via a greedy strategy.
- **Complexity**: $O(|V_1|^3 |V_2|^2 + |V_1||E_1||V_2|^3)$

P-Hom problems can be solved with a provable performance guarantee
NeMa

Relax \textit{p-homomorphism}:
- Structure and some labels are unknown
- Node closed in the query must be closed in the graph

NeMa: compute node vectors

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

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

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

\[ h = 2, \alpha = 0.5 \]
\[ R_G(a) = \{ (b, 0.5), (c, 0.5), (d, 0.5), (e, 0.25) \} \]

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

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

---

NeMa

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

$Q: \langle V_Q, E_Q, l_Q \rangle \quad G: \langle V, E, l \rangle$

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

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

$\text{cost}(v_1, a) = \phi$

$\text{cost}(v_2, c) = \phi$

$\text{cost}(v_2, e) = \phi$

$\text{cost}(v_1, a)$

NP-hard

APX-hard

Label comparison cost

Node vectors difference

Overall cost of mapping $\phi$

Solved with a belief propagation approach

NemaInfer algorithm

If a node has “good” neighbors, more likely it is a “good” match.

\[ U_i(v, u) = \min_{\{\phi: \phi(v) = u\}} \left[ F_\phi(v, u) + \sum_{v' \in N(v)} U_{i-1}(v', u') \right] \]
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.*
## Transformations

<table>
<thead>
<tr>
<th>Transformation</th>
<th>Category</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>First/last token</td>
<td>String</td>
<td>Barack Obama -&gt; Obama</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>String</td>
<td>Jeffrey Jacob Abrams -&gt; J. J. Abrams</td>
</tr>
<tr>
<td>Prefix</td>
<td>String</td>
<td>Engineer-&gt; Eng.</td>
</tr>
<tr>
<td>Acronym</td>
<td>String</td>
<td>Microsoft -&gt; MS</td>
</tr>
<tr>
<td>Synonym</td>
<td>Semantic</td>
<td>Country -&gt; Nation</td>
</tr>
<tr>
<td>Ontology</td>
<td>Semantic</td>
<td>Table -&gt; Furniture</td>
</tr>
<tr>
<td>Range</td>
<td>Numeric</td>
<td>~30 -&gt; 33</td>
</tr>
<tr>
<td>Distance</td>
<td>Topology</td>
<td>Dallas – USA -&gt; Dallas – Texas – USA</td>
</tr>
</tbody>
</table>

They can be expanded arbitrarily

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?

Querying with SLQ

**Learning the parameters (offline)**
1. Random sample a structure from the graph
2. Apply random transformations on the found structure
3. Search the generated queries on the graphs
4. Label the results as **positive** or **negative**
5. Train a Conditional Random Field on the examples

**Query phase**
1. Construct a CRF model on the query and matching candidates
2. Use Loopy Belief Propagation to find the most likely (top-1) assignment
   - \( m_{ji}^{(t)}(u_i) = \max_{u_j} F_V(v_j, u_j) F_E((v_j, v_i), (u_j, u_i)) \prod_{v_k \in N(v_j) \setminus v_i} m_{kj}^{(t-1)}(u_j) \)

---

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

Query (an example)

Graph

Solution

• The user query is an example result

• Find results that are similar to the one in input

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
Exemplar Queries

Compute the answers using **subgraph isomorphism** or **strong simulation**

- **Q1**
  - **Google**
  - **Menlo Park**
  - **Yahoo!**
  - **S. Clara County**
  - **S. Mateo**
  - **Freebase**
  - **S. Clara**
  - **NYC**
  - **CBS**
  - **ZDNet**
  - **YouTube**
  - **CBS**
  - **Paramount**
  - **New York**
  - **USA**

**Subgraph Isomorphism**

- **isA** relationships:
  - Google -> IT Companies
  - YouTube -> IT Companies
  - Yahoo! -> IT Companies
  - Tumblr -> IT Companies
  - CBS -> IT Companies
  - Paramount -> IT Companies
  - S. Clara County -> S. Clara
  - NYC -> S. Clara
  - New York -> USA

**Strong Simulation**

- **acquired** relationships:
  - Google acquired YouTube
  - Yahoo! acquired Tumblr
  - CBS acquired Paramount
  - S. Clara acquired S. Clara County
  - NYC acquired New York
  - USA acquired S. Clara

- **activity** relationships:
  - Google Search Engines
  - YouTube Business
  - Yahoo! Business
  - Tumblr Business
  - CBS Business
  - Paramount Business

- **foundedIn** relationships:
  - Google founded in Menlo Park
  - YouTube founded in Menlo Park
  - Yahoo! founded in Menlo Park
  - S. Clara County founded in S. Clara
  - NYC founded in New York
  - USA founded in New York

- **hasWebsite** relationships:
  - Google has Website
  - YouTube has Website
  - Yahoo! has Website
  - Tumblr has Website
  - CBS has Website
  - Paramount has Website

- **in** relationships:
  - Google in S. Mateo
  - YouTube in S. Mateo
  - Yahoo! in S. Clara
  - Tumblr in S. Clara
  - CBS in NYC
  - Paramount in NYC
  - S. Clara County in S. Clara
  - New York in USA
  - California of USA
Computing exemplar queries

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

**Labels at distance 1**
- \( v \) neighborhood = \{ (B,1) \}
- \( u \) 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*
Computing exemplar queries

Approximation:
- Nodes closed to the sample are more important
- Use Personalized PageRank with a weighted matrix
  \[ \mathbf{v} = (1 - c) \mathbf{A} \mathbf{v} + c \mathbf{p} \]
- Weight edges using the frequency of the edge-label
  \[ I(e_{ij}^\ell) = I(\ell) = \log \frac{1}{P(\ell)} = - \log P(\ell) \]
  \[ P(\ell) = \frac{|E|}{|E|} \]

Mottin, D., Lissandrini, M., Velegrakis, Y. and Palpanas, T. Exemplar queries: Give me an example of what you need. PVLDB 2014
Ranking results

Combination of two factors
1. **Structural**: similarity of two nodes in terms of neighbor relationships
2. **Distance-based**: the PageRank already computed
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
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

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

GQBE finds answers for multiple query tuples
1. Compute a re-weighted union graph of the individual query graphs
2. Find answers using a lattice obtained removing edges from the union graph

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