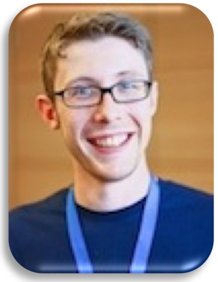
HPI Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

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



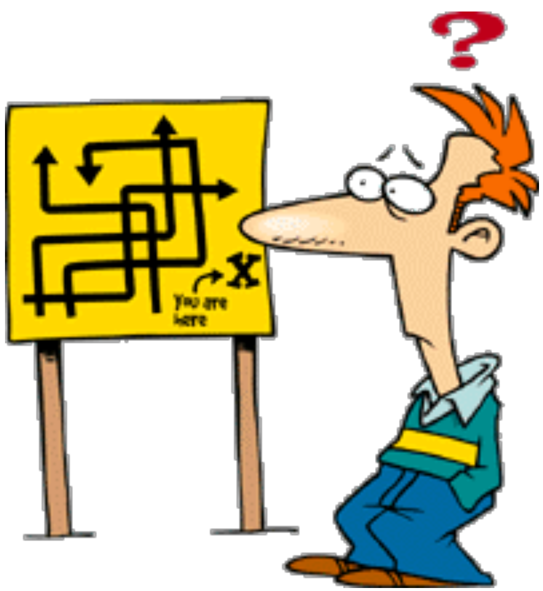
Davide Mottin

- graph mining, novel query paradigms, interactive methods
- https://hpi.de/en/mueller/team/davide-mottin.html

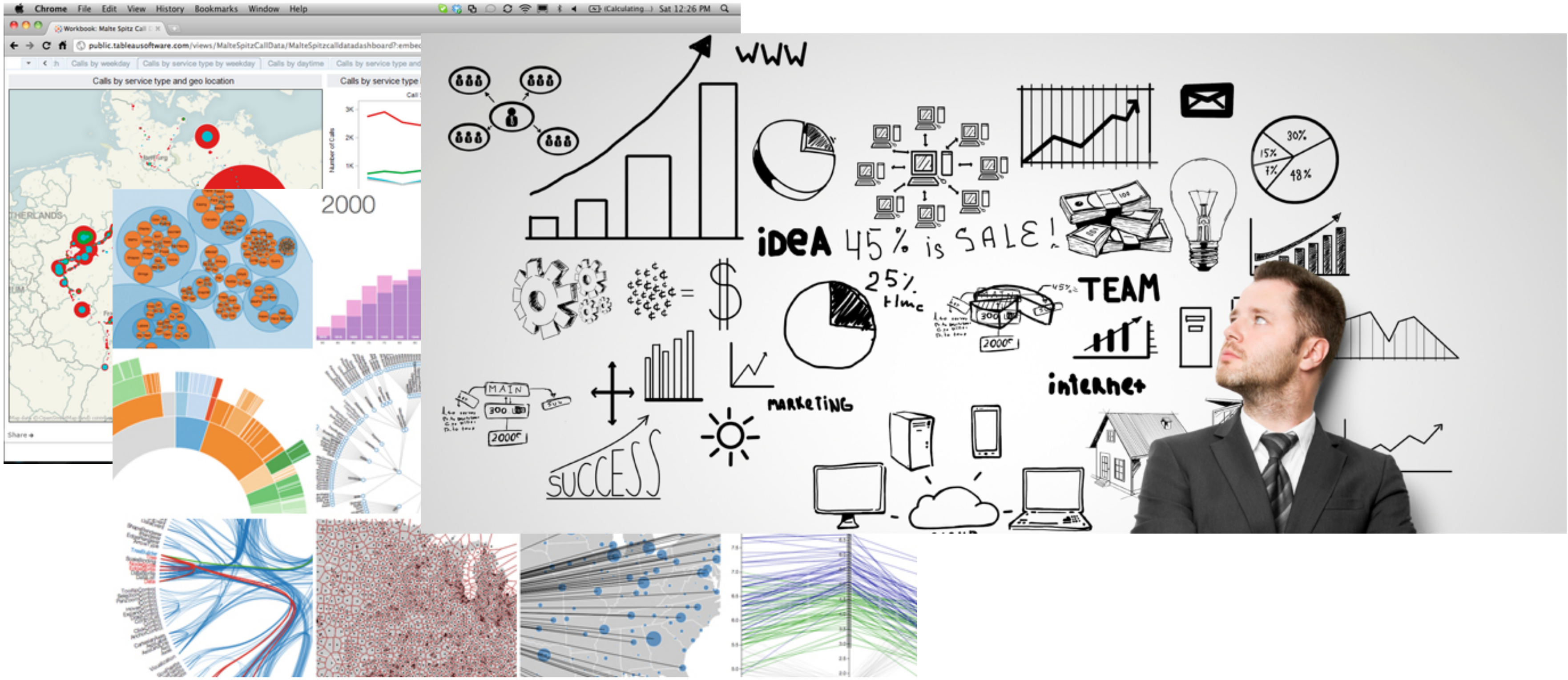
Emmanuel Müller

- graph mining, stream mining, clustering and outlier mining on graphs, streams, and traditional databases
- http://hpi.de/en/mueller/prof-dr-emmanuel-mueller.html

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 Ne

s

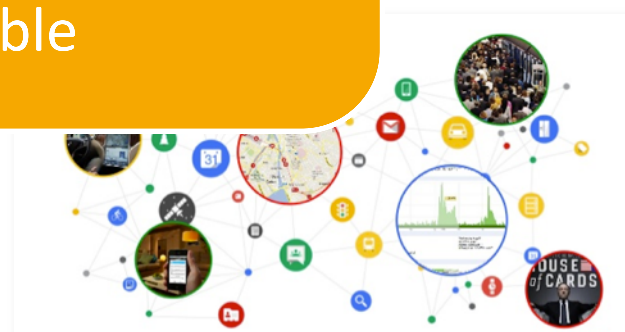
Complex

Ubiquitous

Large

Valuable

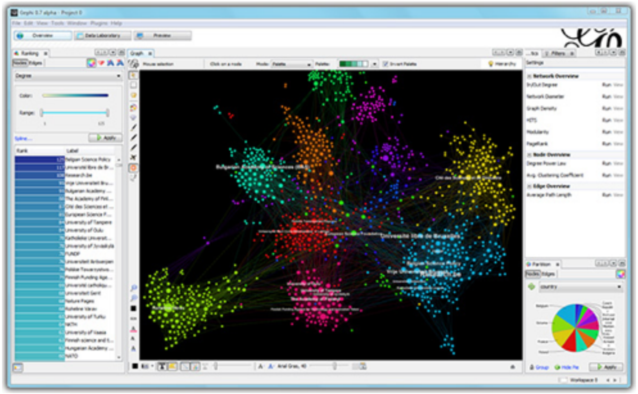
Recommendation Graphs

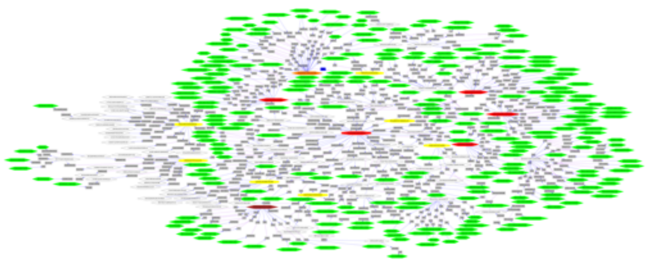


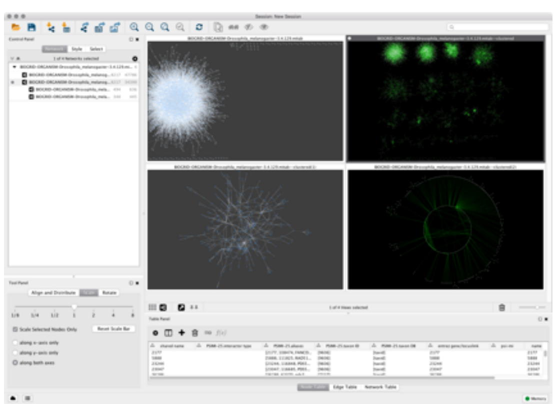
Knowledge Graphs

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

SPARQL

```
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

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

CYPHER

Query languages **ARE**:

- Expressive
- Powerful
- Scalable
- Compact

but

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

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

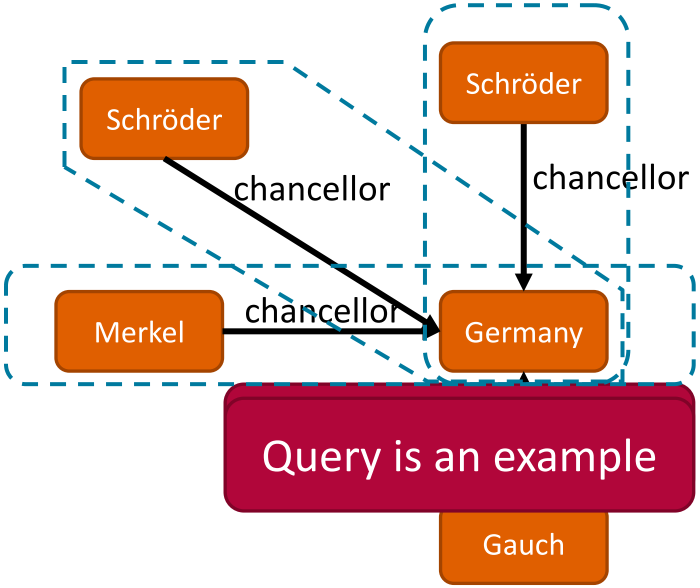
Exploratory Graph Analysis

Focused Graph Mining

Refinement of Query Results

Graph exploration taxonomy

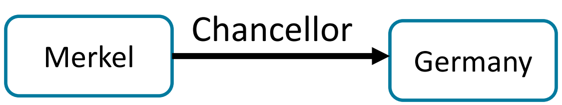
Exploratory Graph Analysis



Two exploratory options:

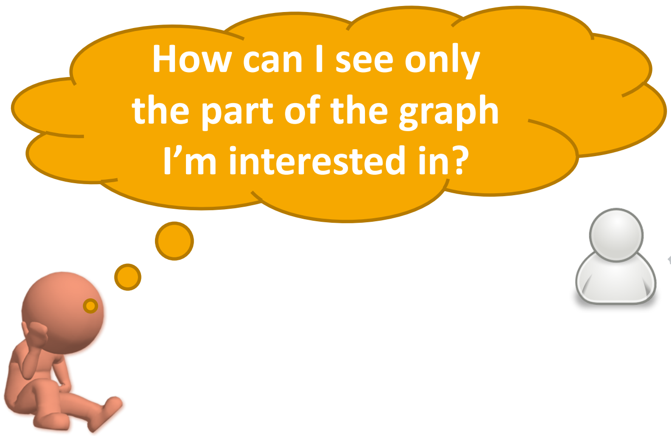
1. An imprecise query

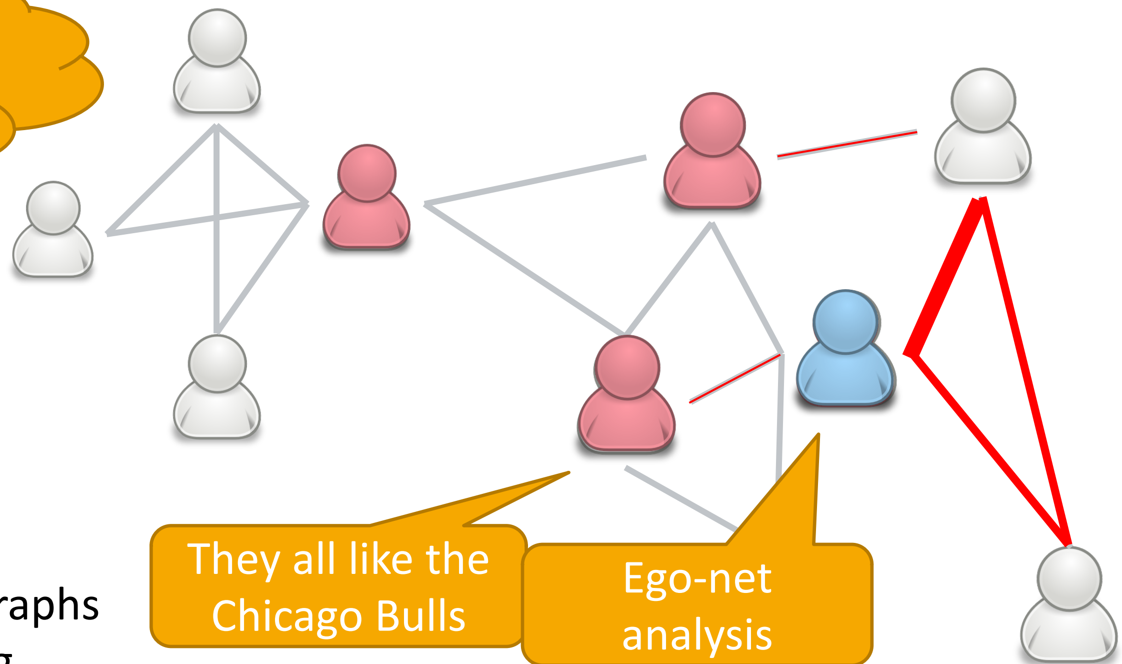
2. A by-example query



Graph exploration taxonomy

Focused Graph Mining





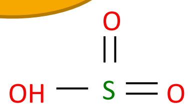
Targeted analysis on large graphs

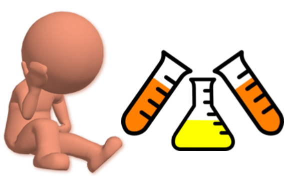
1. Focused graph clustering
2. Space restriction methods
3. Graph Reweighting

Graph exploration taxonomy

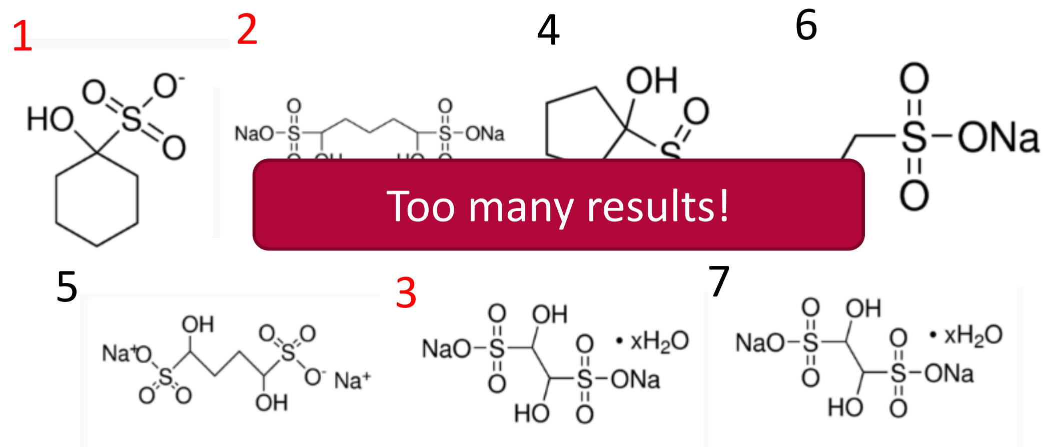
Refinement of Query Results

Where is this molecule contained?





1 2 4 6

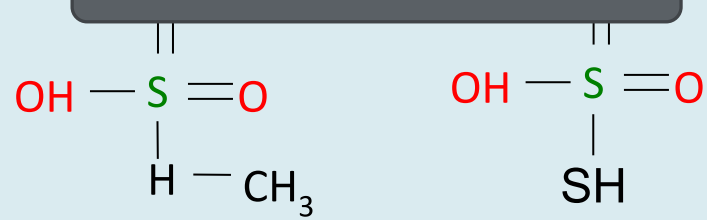


Too many results!

… + 500 results

Que…

Dominance relation



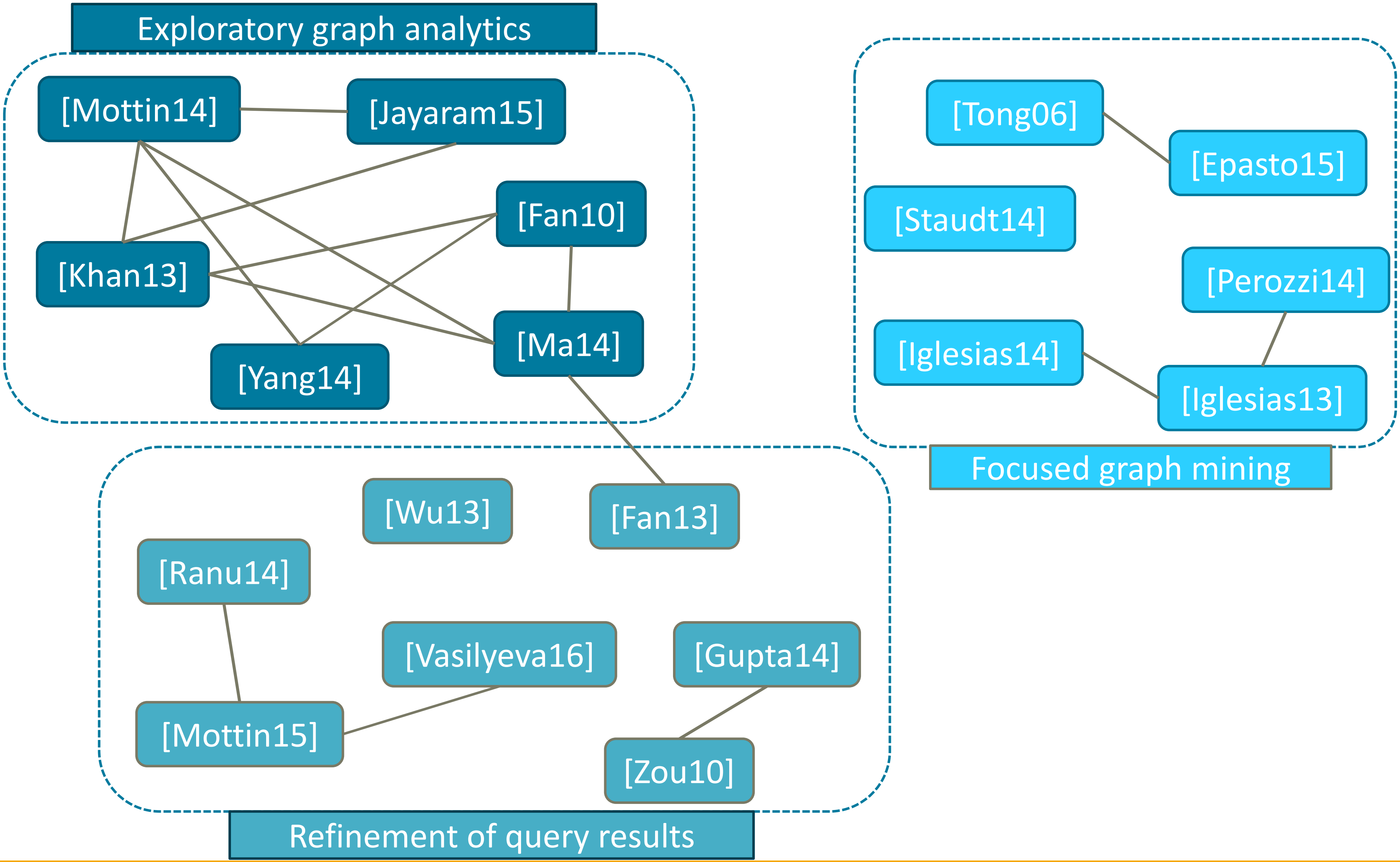
270 results

220 results

Dealing with generic queries:

1. Reformulation and refinement
2. Top-k results
3. Skyline queries

The graph exploration … graph



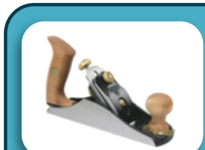
Tutorial outline

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

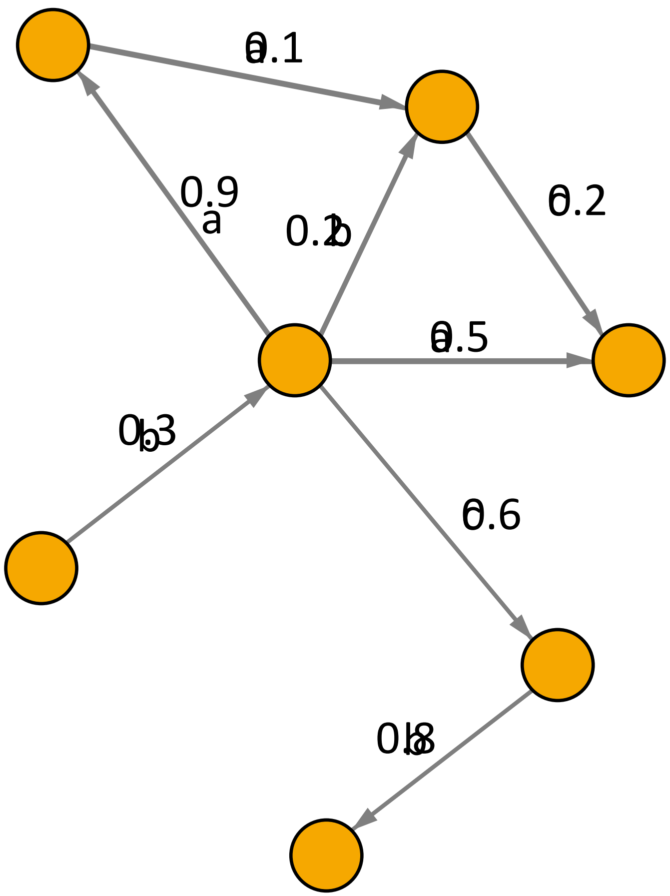
Exploratory Graph Analysis (20 min)

Focused Graph Mining (20 min)

Refinement of Query Results (20 min)

Challenges and discussion

Graphs



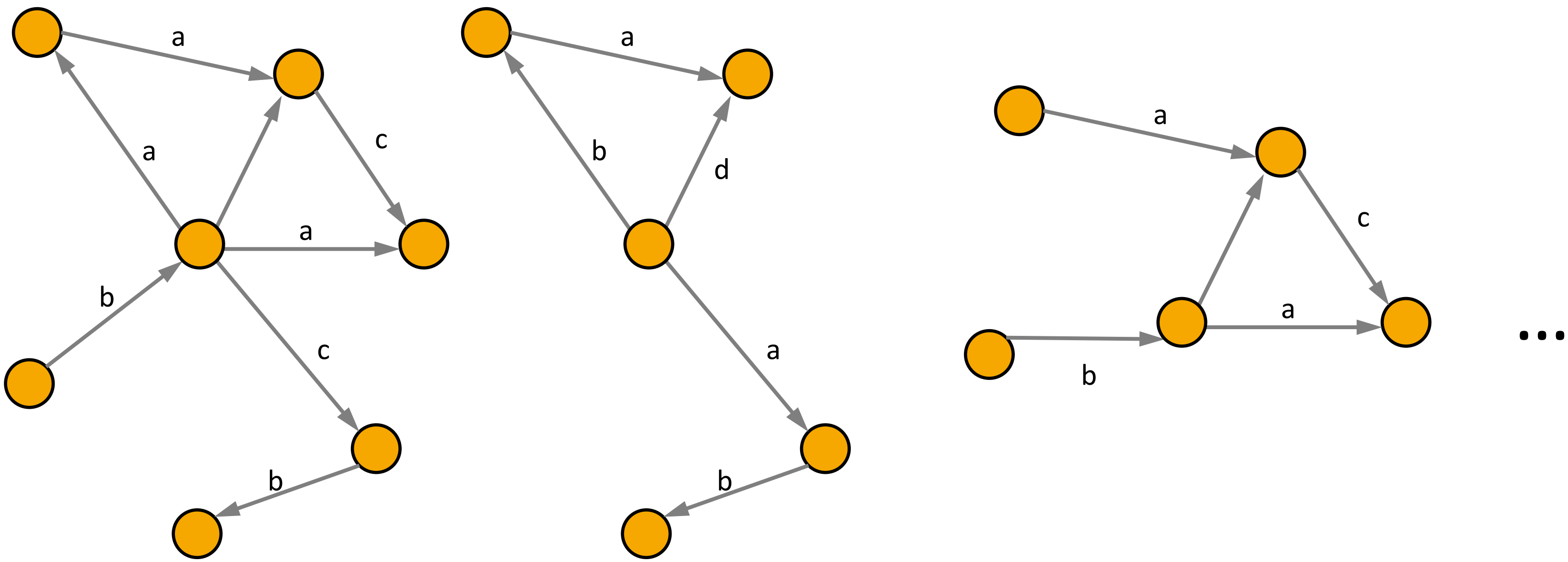
G = (V, E, l) / G = (V, E, p)

Vertices Edges Labeling function / Probability function

$l: V \cup E \to \Sigma$

- Undirected Graphs
  - Co-authorship, Roads, Biological
- Directed graphs
  - Follows, …
- Labeled Graphs
  - Knowledge graphs, …
- Probabilistic graphs
  - Causal graphs

Graph databases (set of graphs)

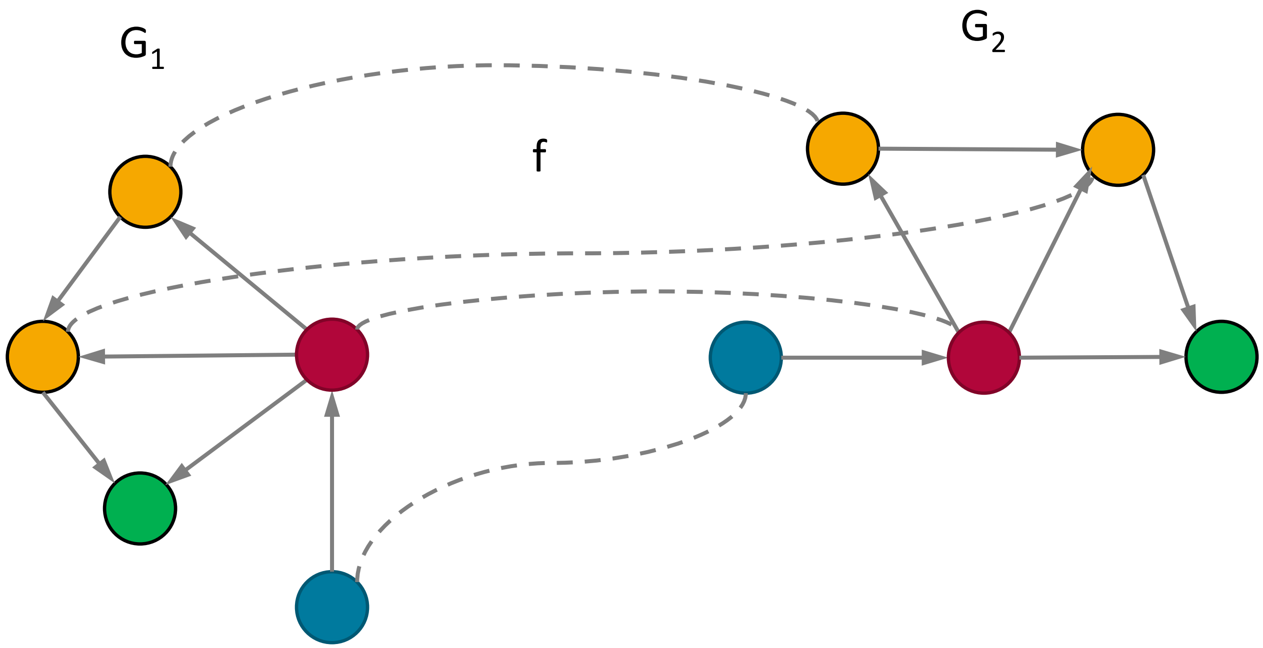


G_1 G_2 G_3

$$D = \{G_1, G_2, \dots, G_n\}, G_i = \langle V_i, E_i, l_i \rangle, l_i \colon E_i \cup V_i \rightarrow \Sigma$$

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

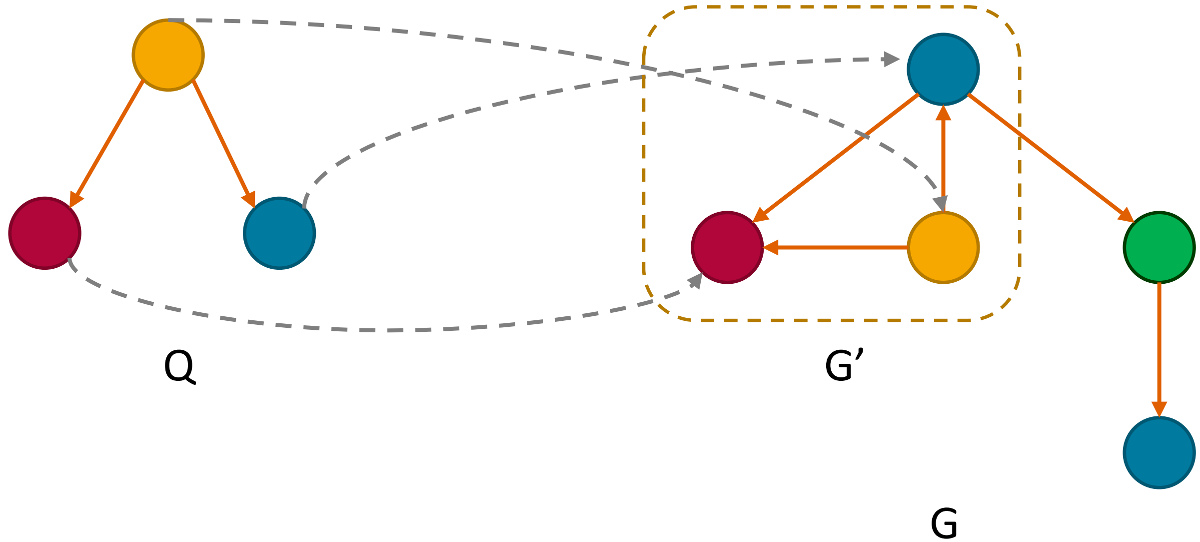
Graph Isomorphism



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 G_2 iff exists a **bijective** function $f: V_1 \rightarrow V_2$ s.t.:

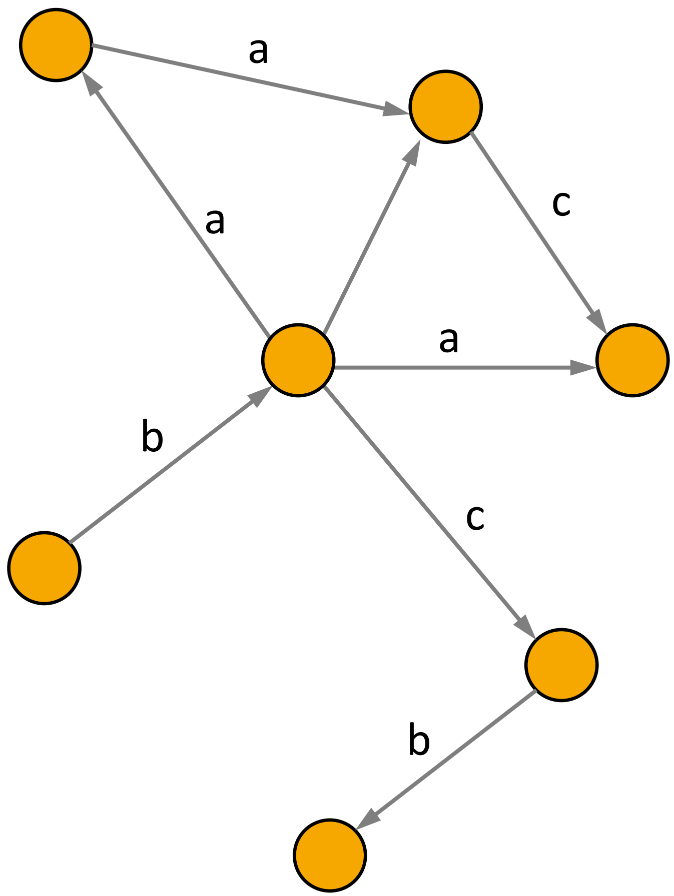
1. For each $v_1 \in V_1$, $l(v_1) = l(f(v_1))$
2. $(v_1, u_1) \in E_1$ iff $\big(f(v_1), f(u_1)\big) \in E_2$

Subgraph Isomorphism



A graph ,$Q\colon \langle V_Q, E_Q, l_Q \rangle$ is subgraph isomorphic to a graph $G\colon \langle V, E, l \rangle$ if exists a subgraph $G' \sqsubseteq G$, isomorphic to Q

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

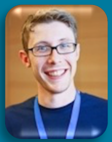
$$C = \{C_1, C_2, C_3, \ldots, C_k\}$$

- **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

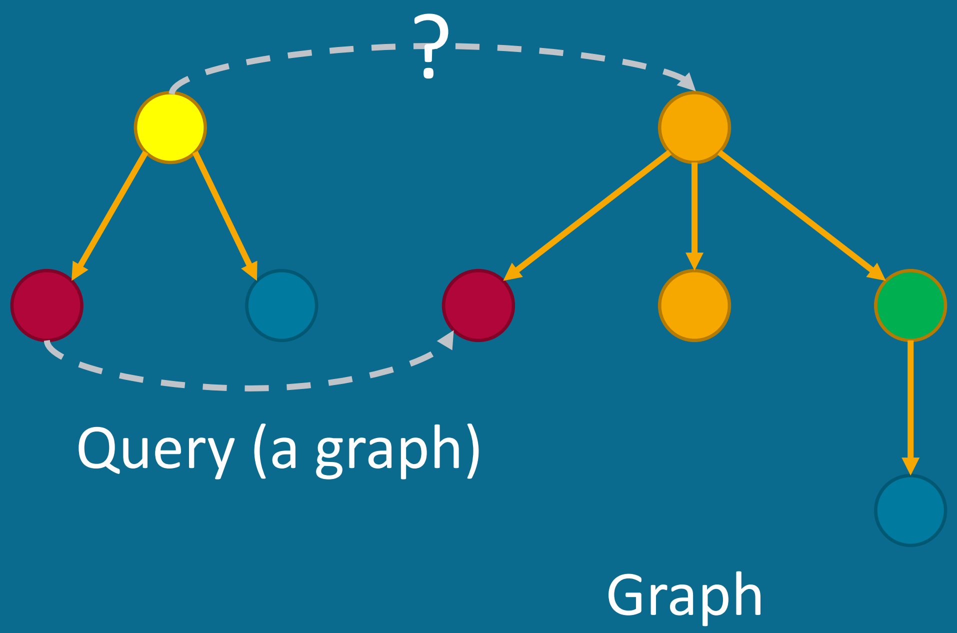
Approximate Graph Search

- 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



- The user might be imprecise in the search terms

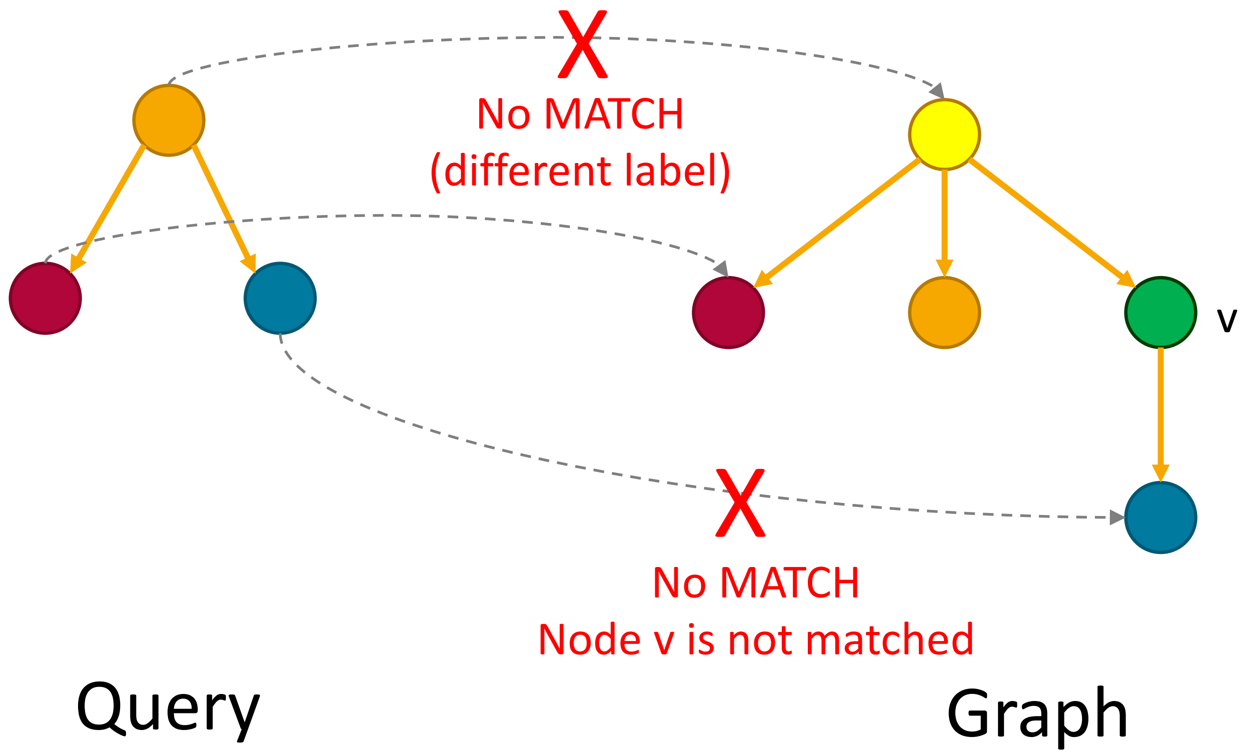
Solution

- 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

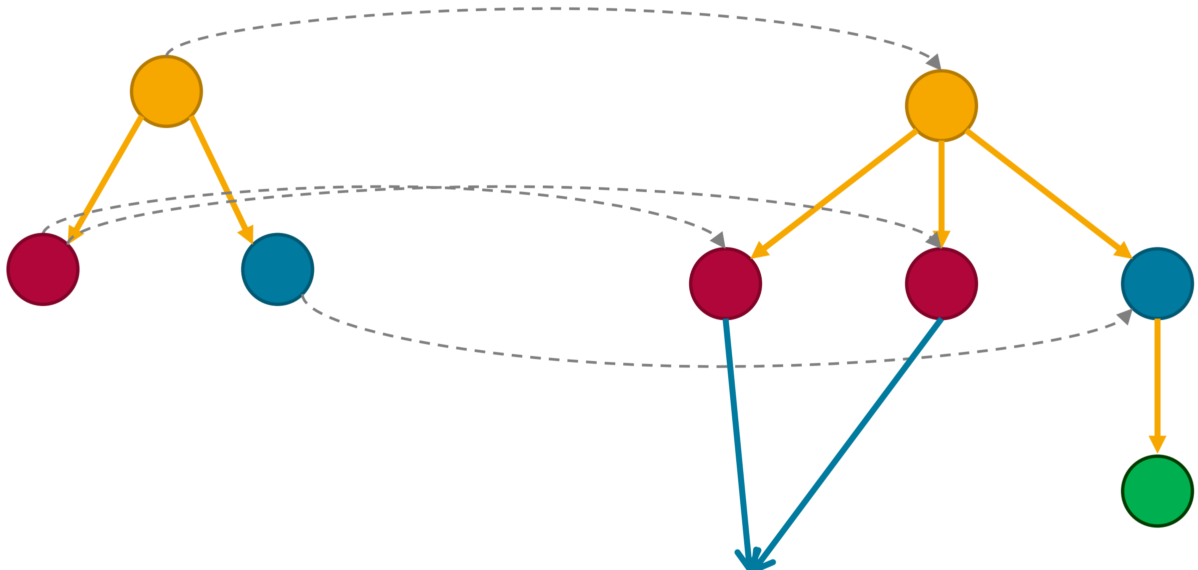
(Sub)Graph Isomorphism might be too restrictive



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

Poly-time (cubic)

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]

Parent-child relationship

Child-parent relationship

Duality

Locality

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

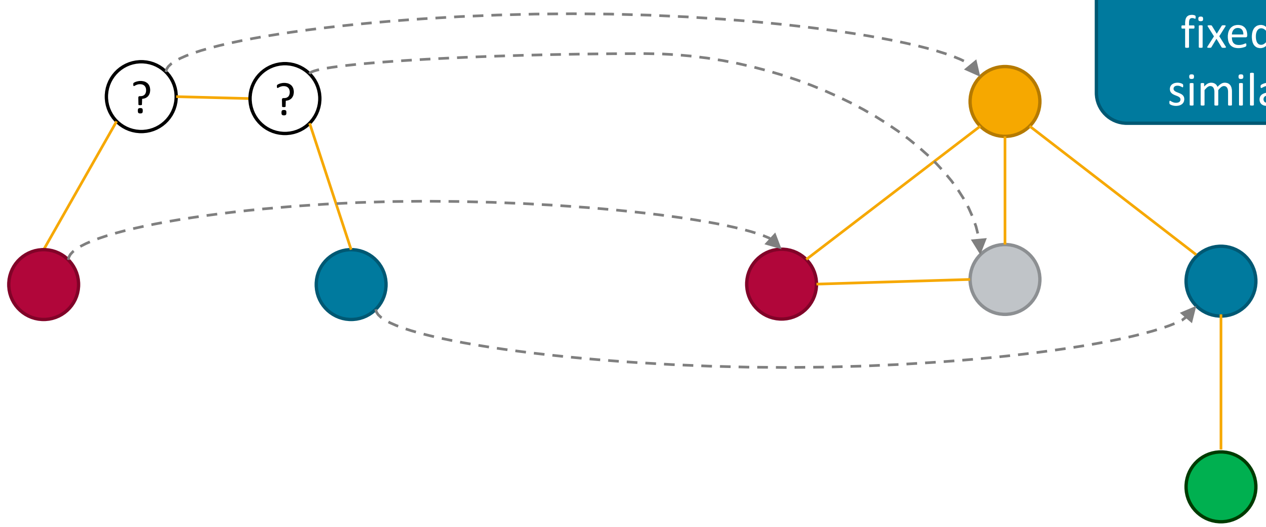
Subgraph Isomorphism ❯ Strong Simulation ❯ Dual Simulation ❯ Graph Simulation

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

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

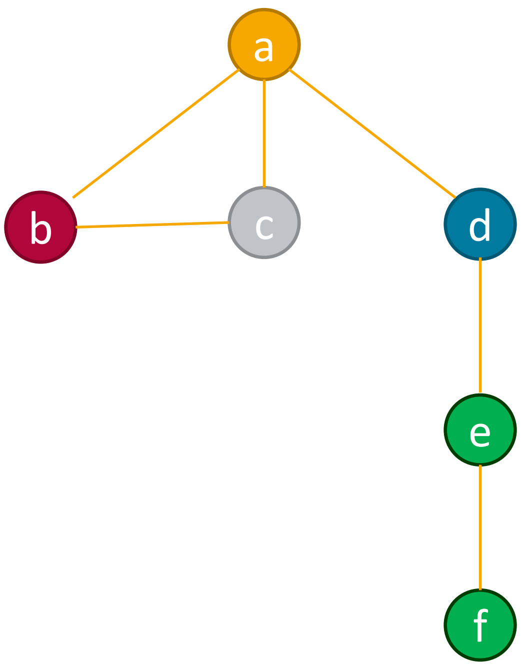
NeMa: compute node vectors

$$R_G(u) = \{\langle u', w_u(u') \rangle\}$$

Convert node u into a vector of neighbors

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

Distance less than h (h-hop neighbor)



$$h = 2, \alpha = 0.5$$

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

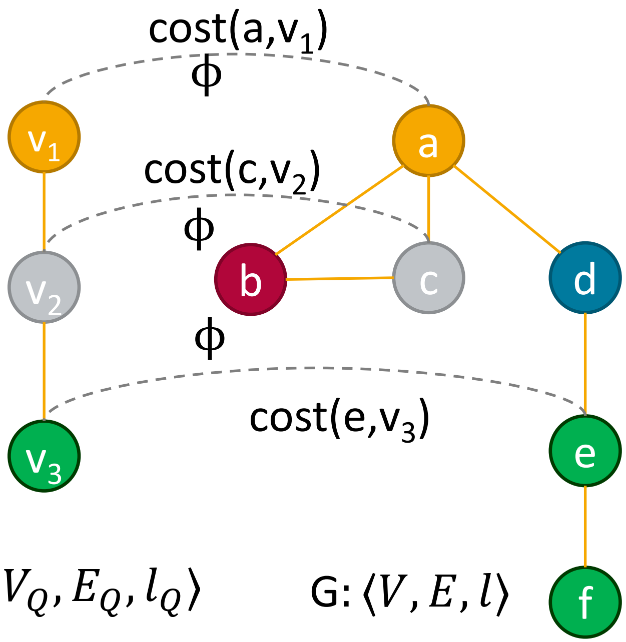
Vector of nodes at distance <= h from a

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

NeMa

NP-hard

APX-hard



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

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

Label comparison cost

Node vectors difference

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

Overall cost of mapping ϕ

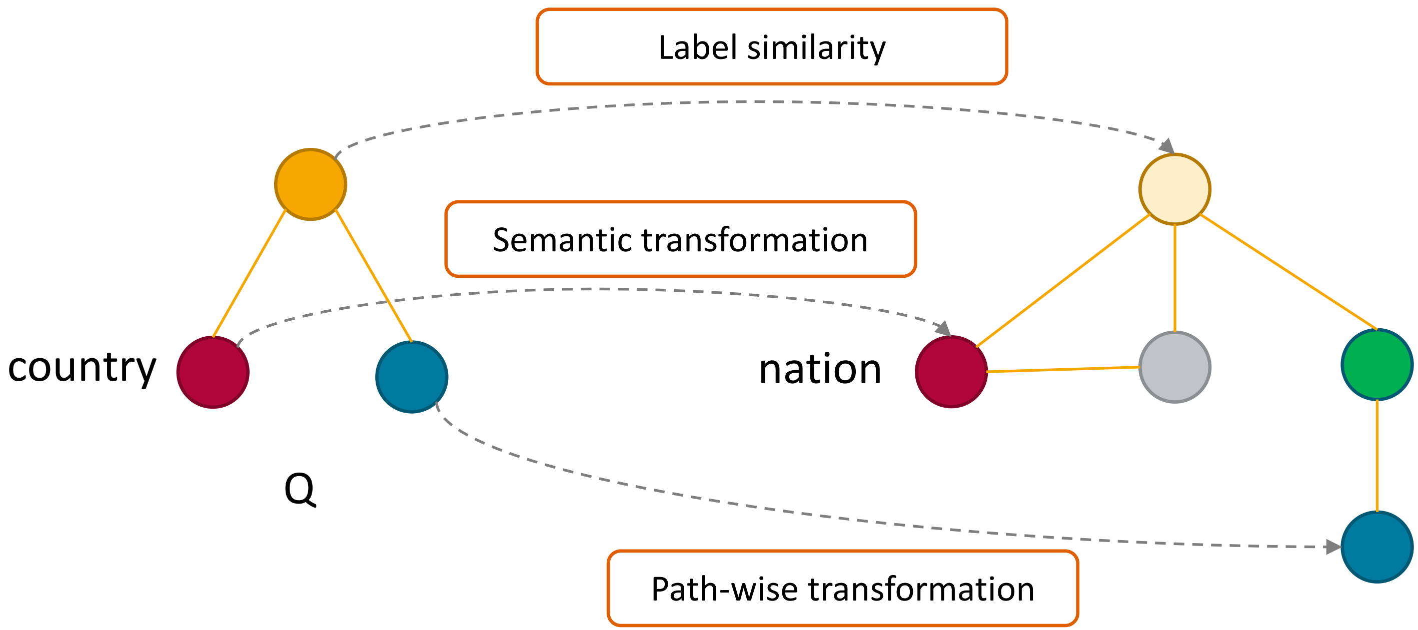
Problem
Given Q and G, find the mapping ϕ with the minimum cost $C(\phi)$

Solved with a belief propagation approach

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

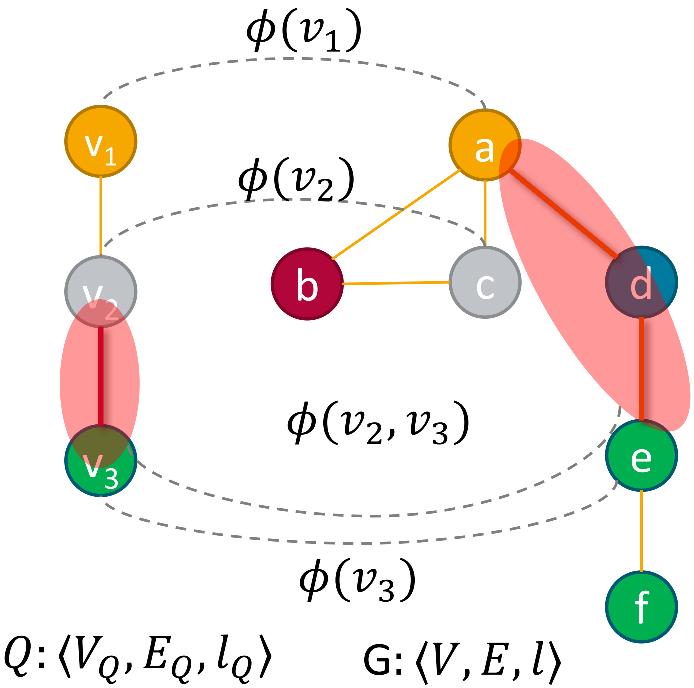
SLQ

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



$$F_V(v, \phi(v)) = \sum_i \alpha_i f_i(v, \phi(v))$$

Node matching score

$$F_E(e, \phi(e)) = \sum_i \beta_i f_i(e, \phi(e))$$

Edge matching score

$$P(\phi|Q) \propto \exp\Big(\sum_{v \in V_Q} F_V(v, \phi(v)) + \sum_{e \in E_Q} F_E(e, \phi(e))\Big)$$

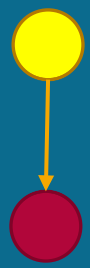
Overall score for matching ϕ

Problem

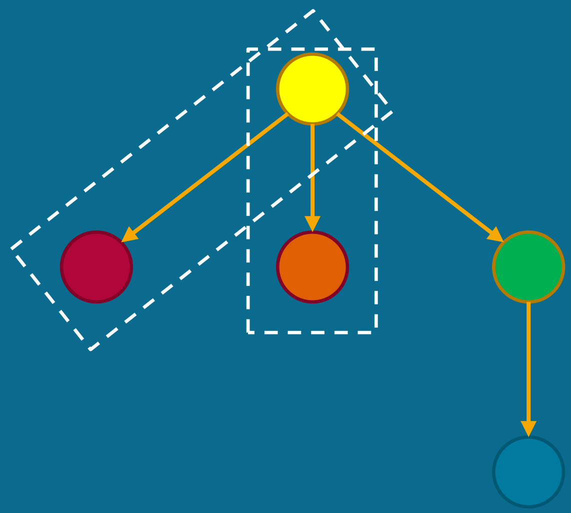
- How to learn the parameters α_i, β_i ?
- How to find the matching with the highest score?

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

Querying by Example



Query
(an example)



Graph

- The user query is an example result

Solution

- 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

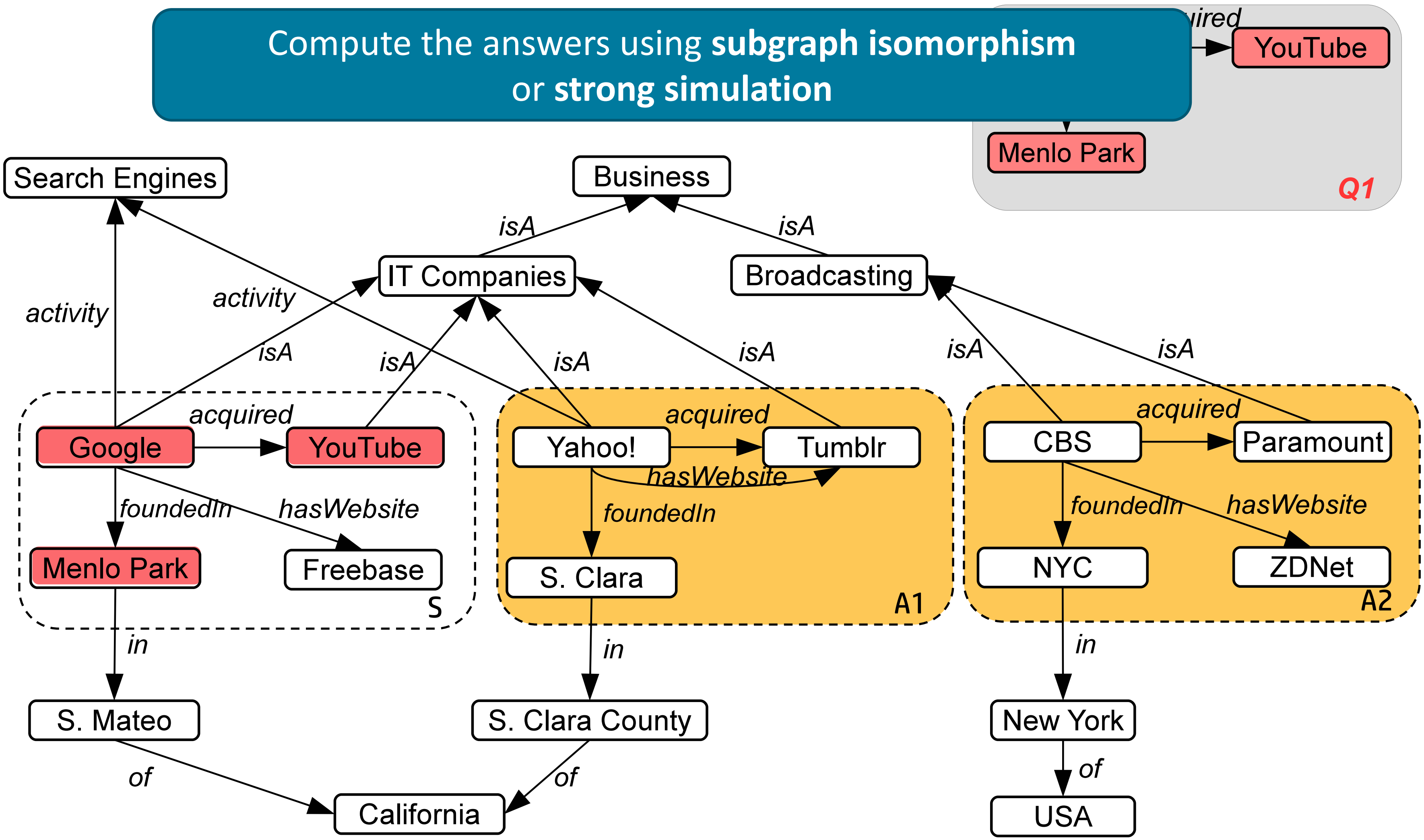
Exemplar Query Evaluation

- evaluate Q_e in a database D, finding a sample *s*
- find the set of elements *a* similar to *s* given a *similarity relation*

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

Exemplar Queries

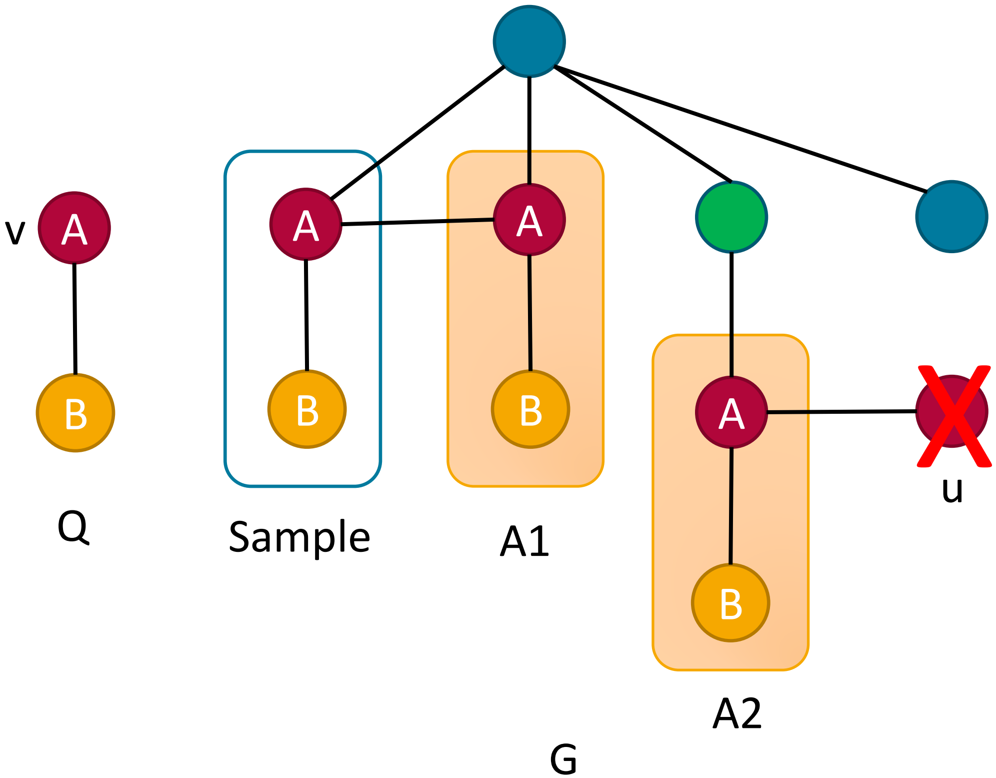
Compute the answers using **subgraph isomorphism** or **strong simulation**



Computing exemplar queries

NP-complete (subgraph isomorphism)

$O(|V|^4)$ (simulation)



Pruning technique:

- Compute the neighbor labels of each node

$$W_{n,a,i} = \{n_1 | l(n_1, n_2) = a \vee \in N_{i-1}(n)\}$$

- Prune nodes not matching query nodes neighborhood labels
- Apply the technique iteratively on the query nodes

Labels at distance 1

v neighborhood = {(B,1)}

$\not\subseteq$

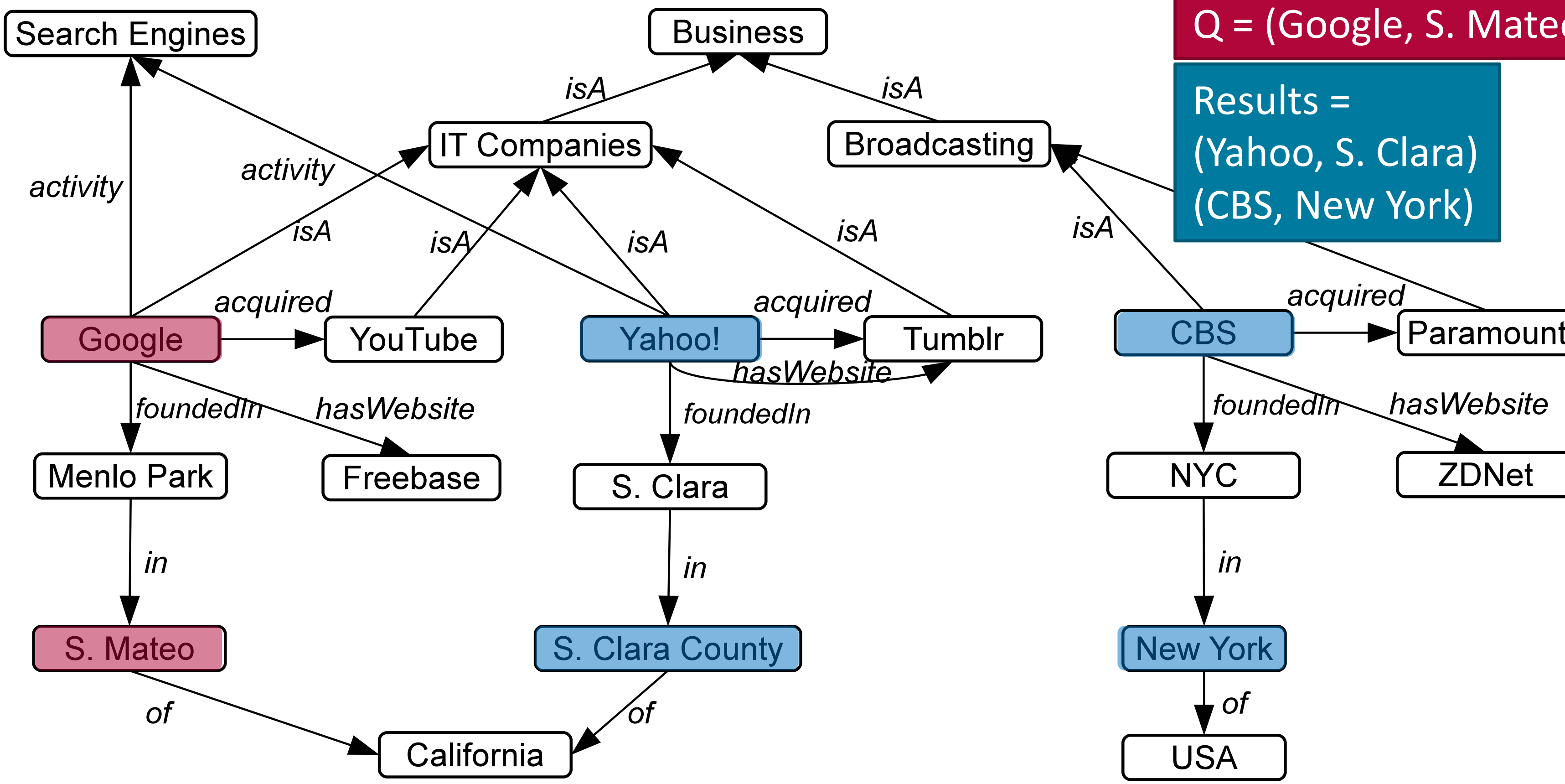
u neighborhood = {(A,1)}

No Match

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

Graph query by example (GQBE)

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



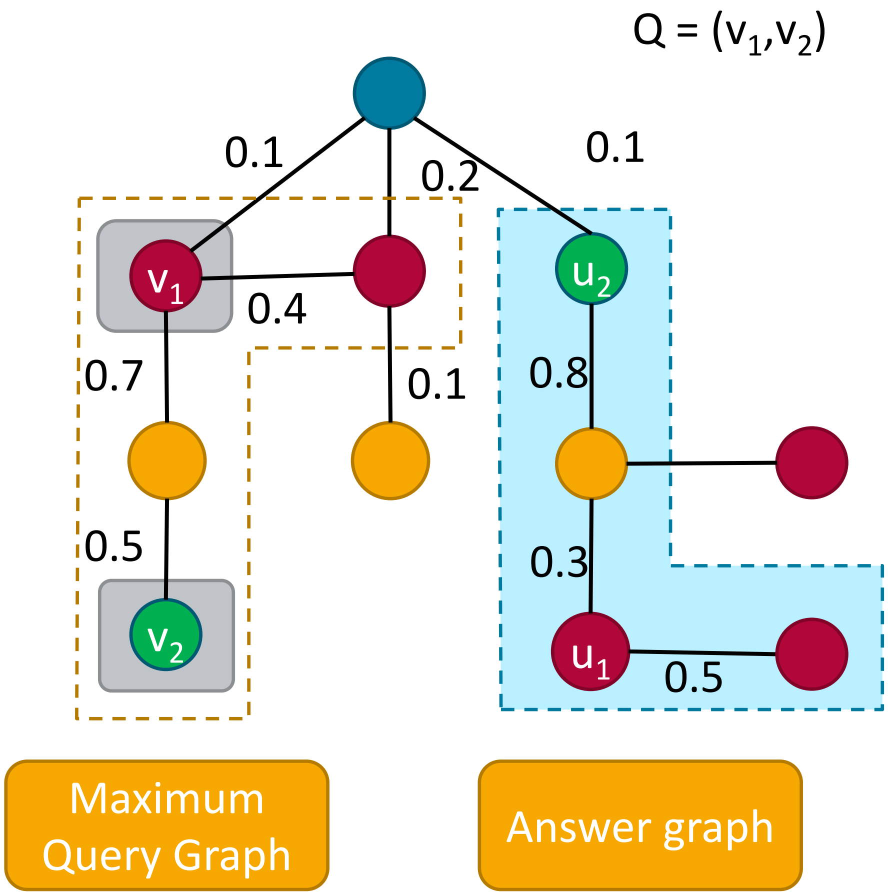
Q = (Google, S. Mateo)

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

NP-hard



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

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

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 …

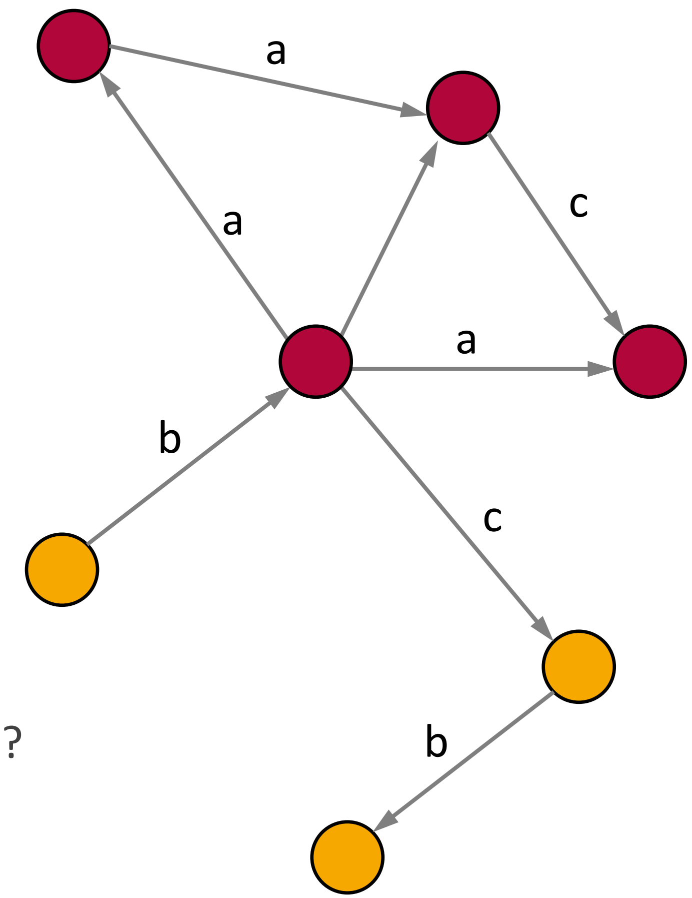
Graph Mining Focused on User Interest

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

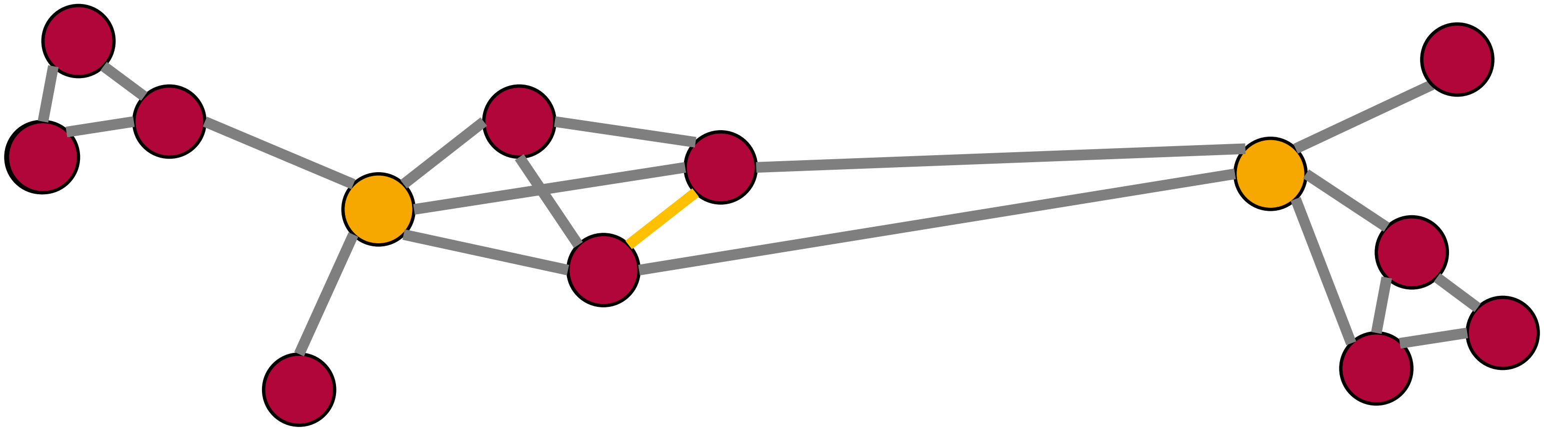
C. Staudt, Y. Marrakchi, H. Meyerhenke: Detecting Communities Around Seed Nodes in Complex Networks (BigData 2014)

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).

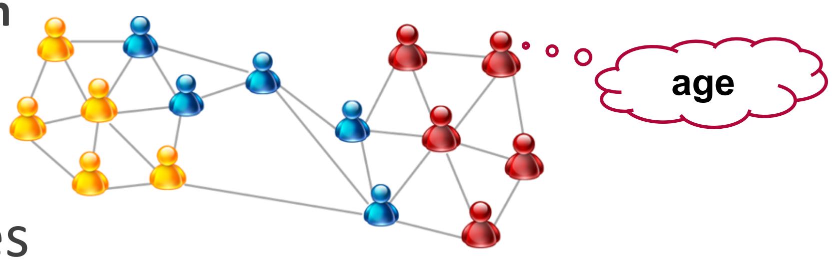


Epasto et al. Ego-Net Community Mining Applied to Fried Suggestion. (VLDB 2015)

Mining Attributed Graphs

Different graph mining techniques

- Clustering / graph partitioning / …
- **Community detection and anomaly detection**



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

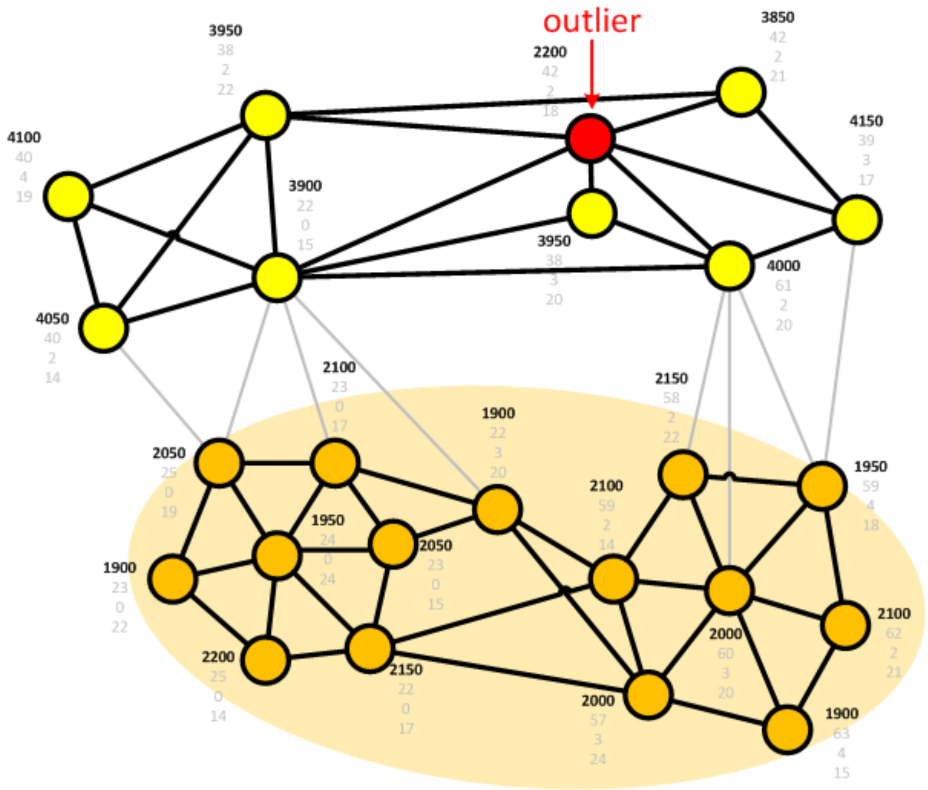
Problem: **disassortative mixing** [Newman 2003]
hinders the detection of communities
(i.e. similarity assessment of nodes)

Solution: Selection of relevant views ensuring homophily

Newman. Mixing patterns in networks. Physical Review, 2003

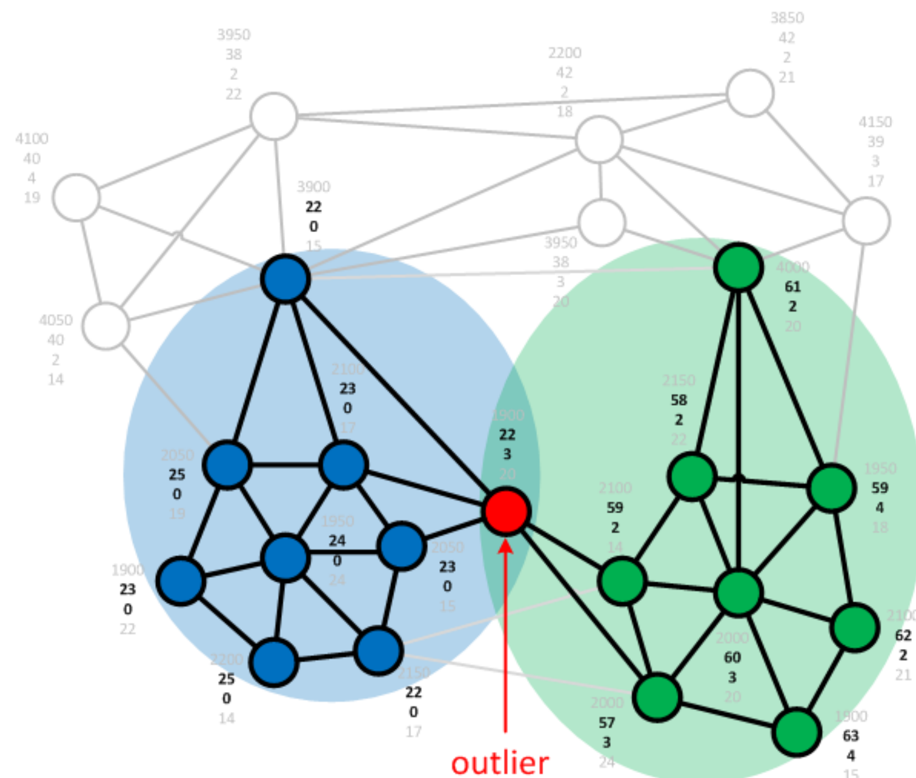
Multiple Views in Attributed Graphs

Different structures depending on the subset of attributes



Multiple Views in Attributed Graphs

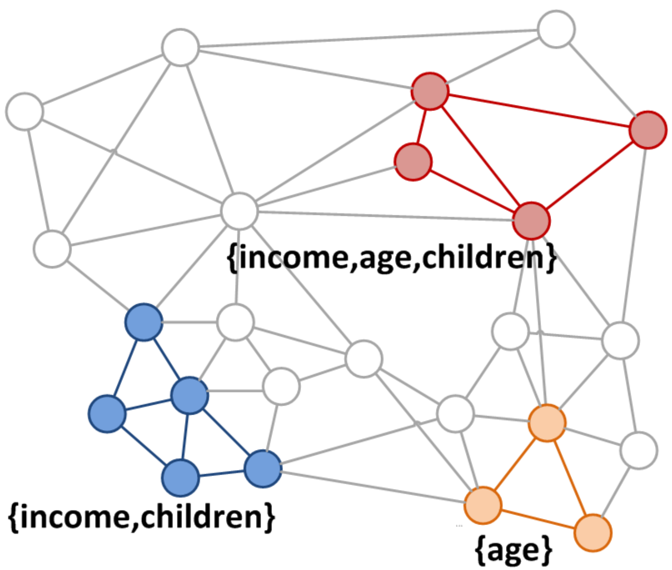
Different structures depending on the subset of attributes

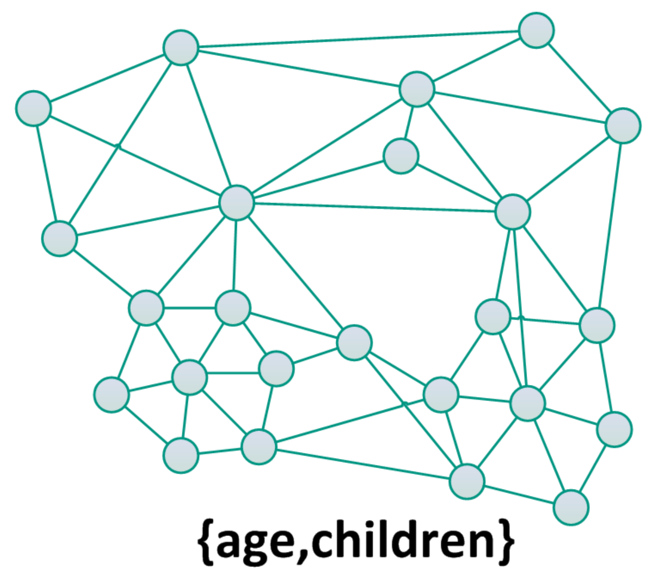


Specialized Approaches

Frequent subgraph mining, subspace clustering ...

- Local selection of the attributes
- Individual subgraphs





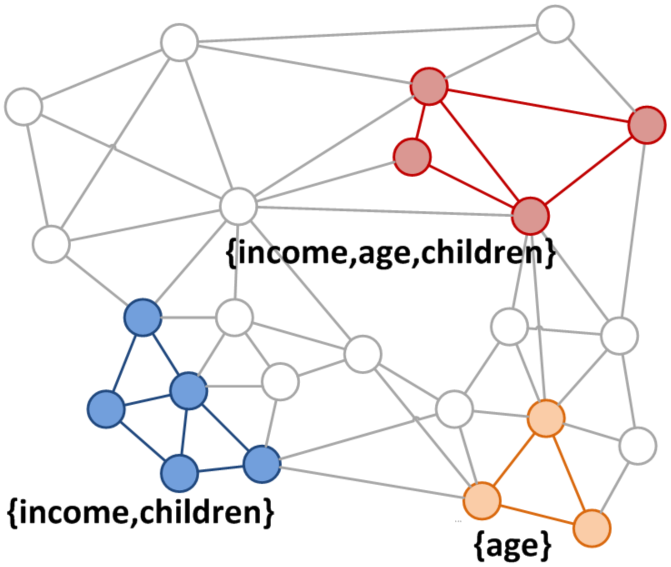
First Idea: Local Context Selection

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

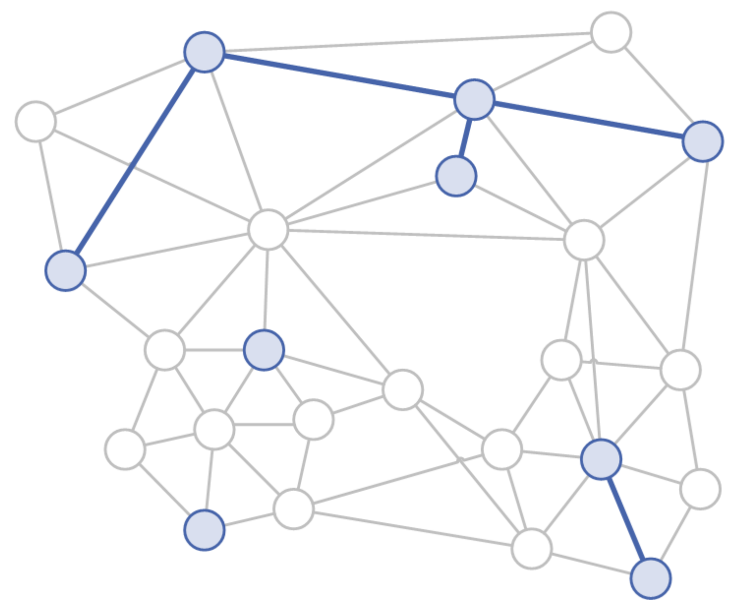
- **Mutual similarity** between attribute values in subspace *S*
- **Significantly more edges** than expected by a random distribution

Constraint Subgraph $G_{C,S}$

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

S = {shoe size}
nodes with **8 ≤ *shoe size* ≤ 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

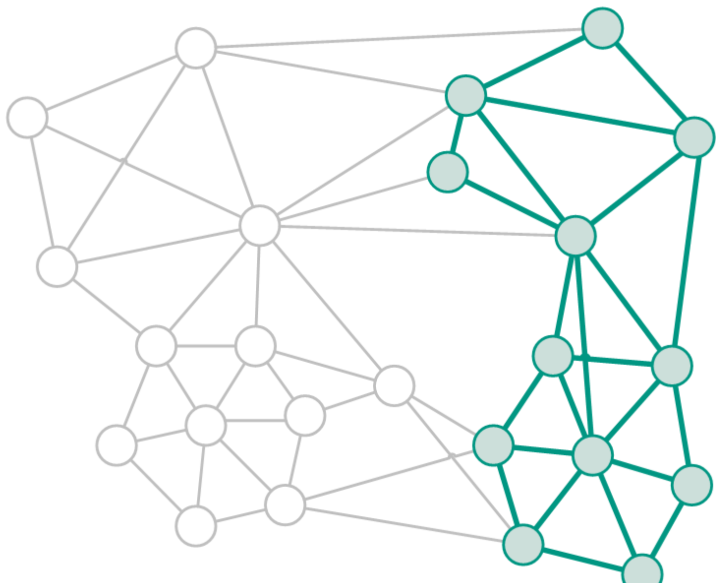
- **Mutual similarity** between attribute values in subspace *S*
- **Significantly more edges** than expected by a random distribution

Constraint Subgraph $G_{C,S}$

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

S ={age,income}
nodes with **45 ≤ *age* ≤ 60** and
1900 ≤ *income* ≤ 4500

high number of edges

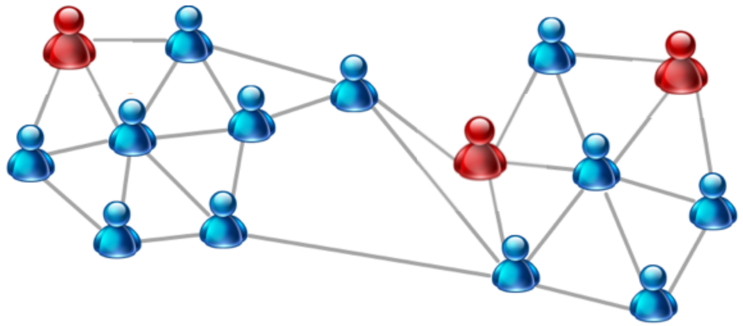


Iglesias et al. Statistical Selection of Congruent Subspaces for Mining Attributed Graphs (ICDM 2013)

Focus on User Preference

Examples for user preference:

- attribute weighting
- examples of similar nodes
- some notion of similarity



examples of similar nodes

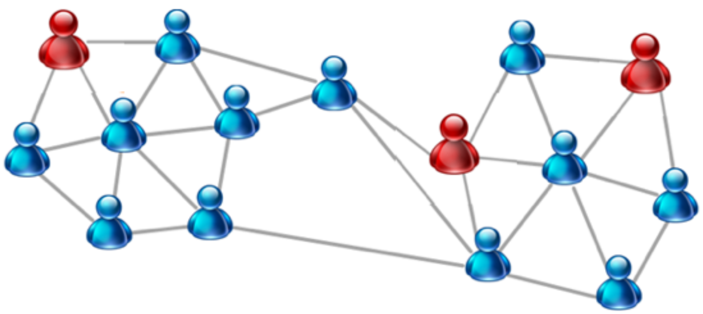
attribute weighting

Focused Selection of Subsaces (FocusCO)

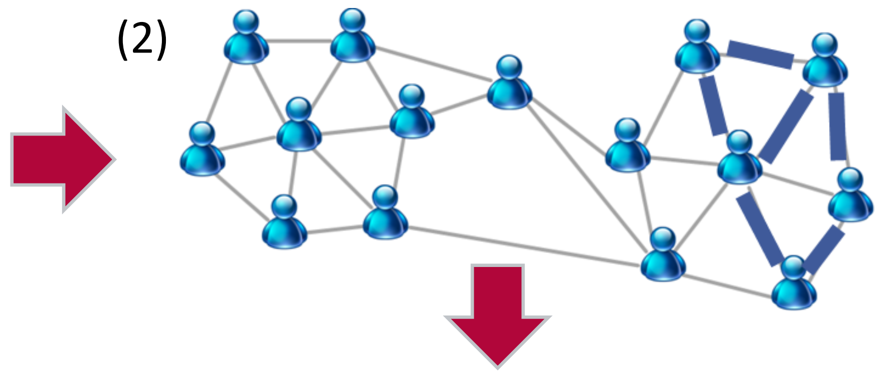
Decoupled mining for given user preference

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

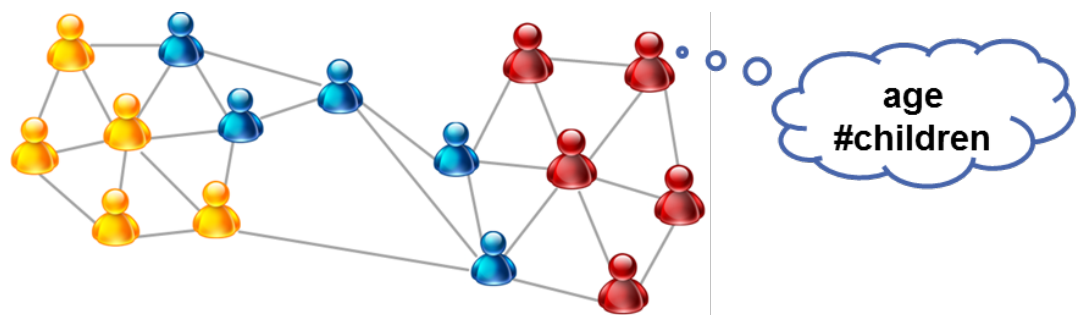
(1)



(2)



(3) applicable for various community detection models

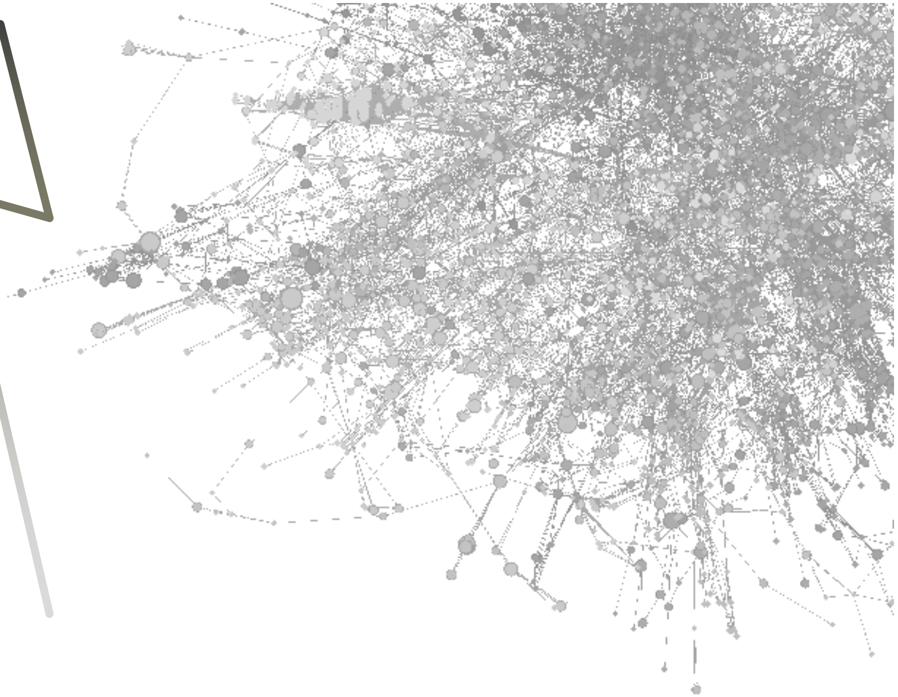


Perozzi et al. Focused Clustering and Outlier Detection in Large Attributed Graphs (KDD 2014)

Knowledge Discovery by Focused Graph Mining

Example Sociology:

hypothesis testing vs. hypothesis generation



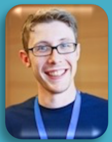
We are here

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

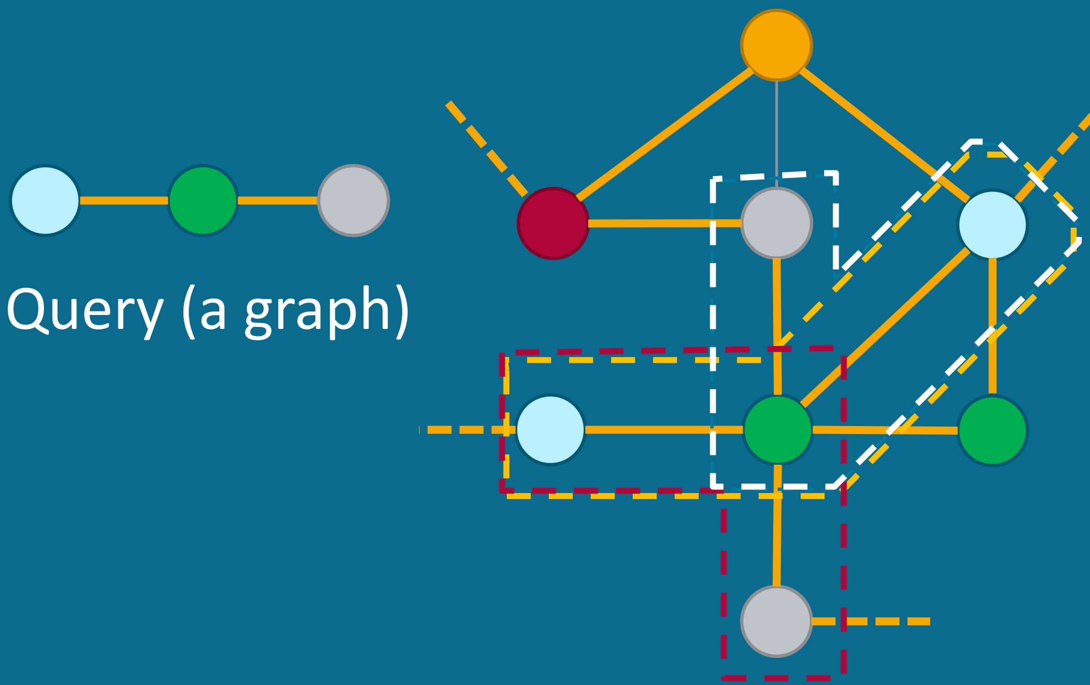
- Use user feedback to find the k results with the highest score
- User perspective: the results are potentially the most preferred items

Skyline queries

Not in this tutorial ☹

- Optimize one single [illegible] when finding results of a query
- User perspective: sh[illegible] those nodes/graphs that are no worse than others

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

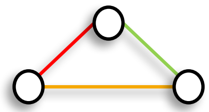
OR

- Summarize the results

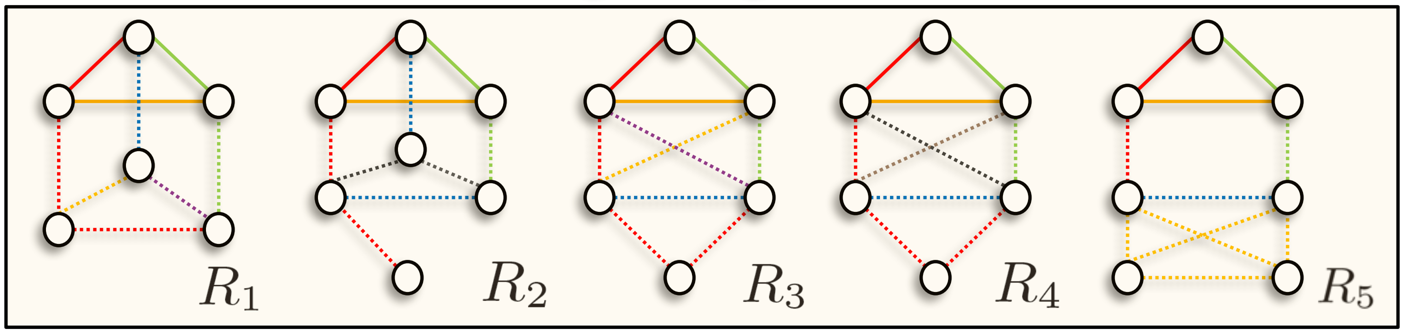
- Query Reformulation approaches: in Graph Databases (Mottin et al.), in connected networks (Vasilyeva et al.)
- Result summarization approaches: top-k representative (Ranu et al.), keyword induced result summarization (Wu et al.)

Graph Query Reformulation

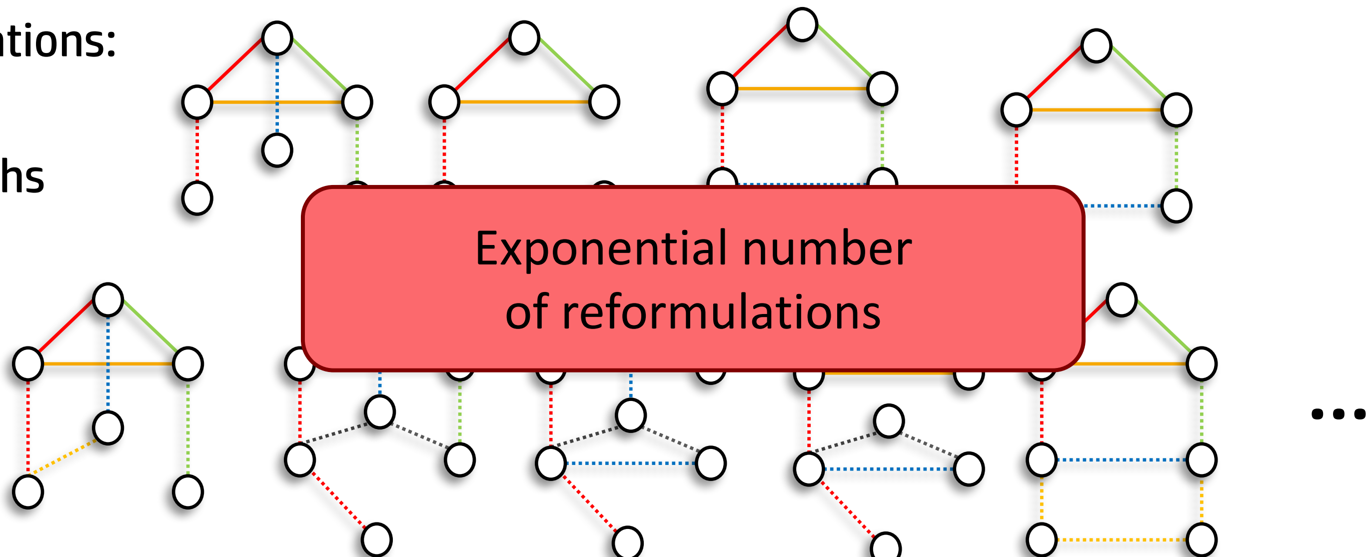
Query



Results

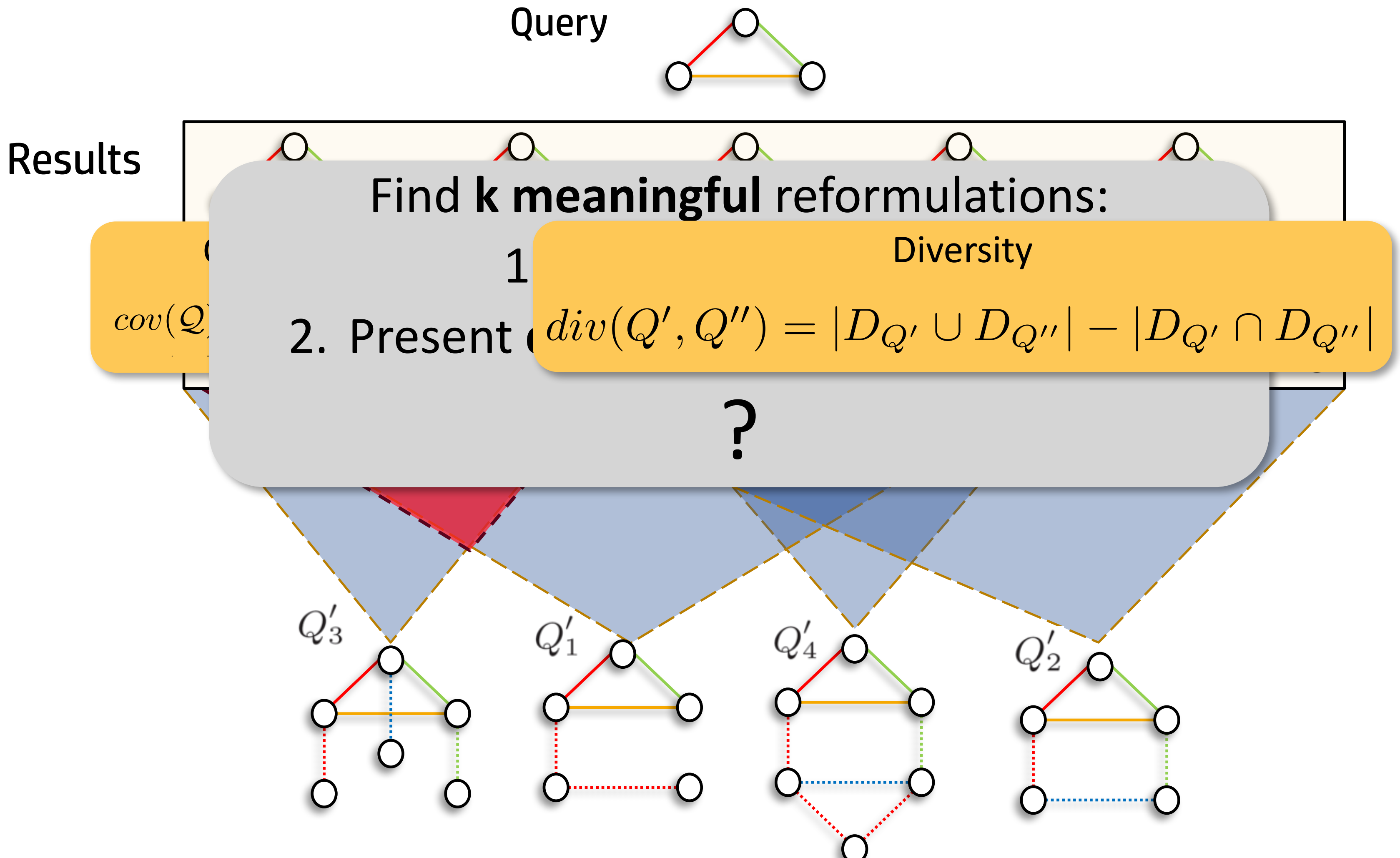


Reformulations:
query
supergraphs



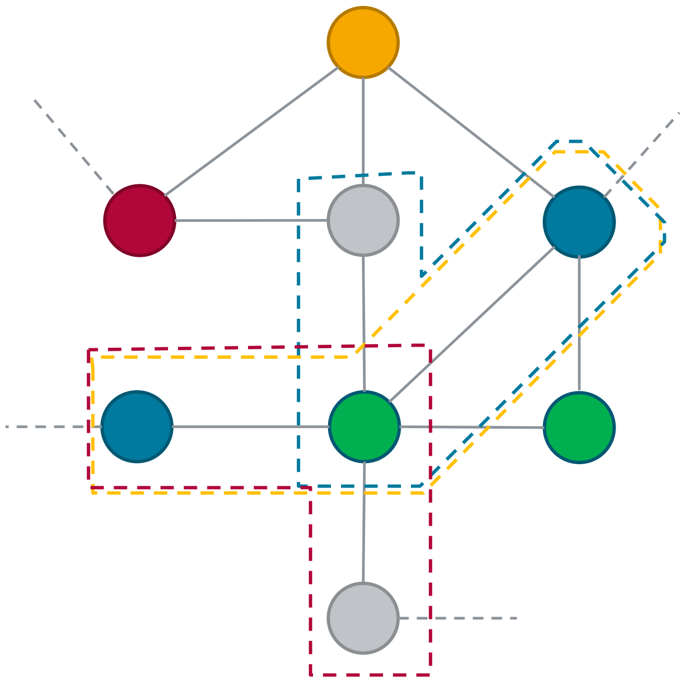
Mottin, D., Bonchi, F. and Gullo, F. Graph Query Reformulation with Diversity. KDD, 2015

Graph Query Reformulation



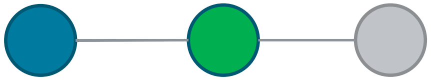
Mottin, D., Bonchi, F. and Gullo, F. Graph Query Reformulation with Diversity. KDD, 2015

Why empty, Why so-many answers in graphs



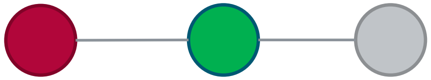
Large graph

Query



Too Many answers

Query



Empty-answer

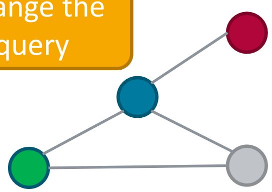
Problem
Given a query Q and a graph G, restrict/enlarge the result set with minimal changes in the query.

Vasilyeva, E., Thiele, M., Bornhövd, C. and Lehner, W.. Answering “Why Empty?” and “Why So Many?” queries in graph databases. *JCSS, 2016*

Why empty, Why so-many answers in graphs

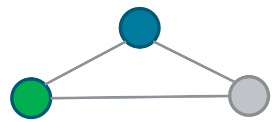
Why?
Empty/Too Many

Change the query

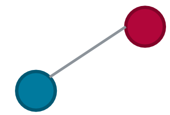


Exponential variations!

Explanations



+

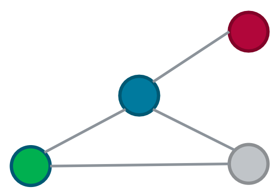


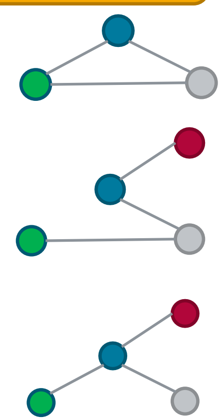
Maximum Common Subgraph

Differential graph

Graphs and unexpected subgraphs

Modifications



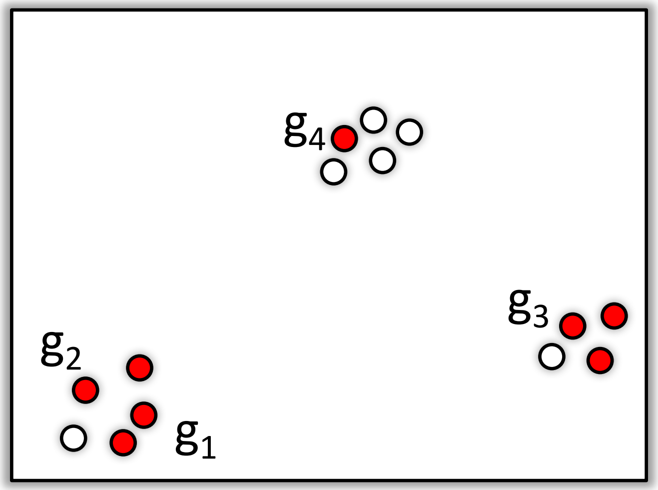


Answers to the new queries

Vasilyeva, E., Thiele, M., Bornhövd, C. and Lehner, W.. Answering "Why Empty?" and "Why So Many?" queries in graph databases. *JCSS, 2016*

Top-k representative queries

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

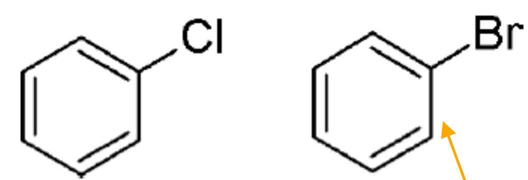


- Object is *relevant*
- Object is non-relevant

Two objects are close if they are similar

Select k=2 relevant objects

Top-2 answer: g_1, g_2

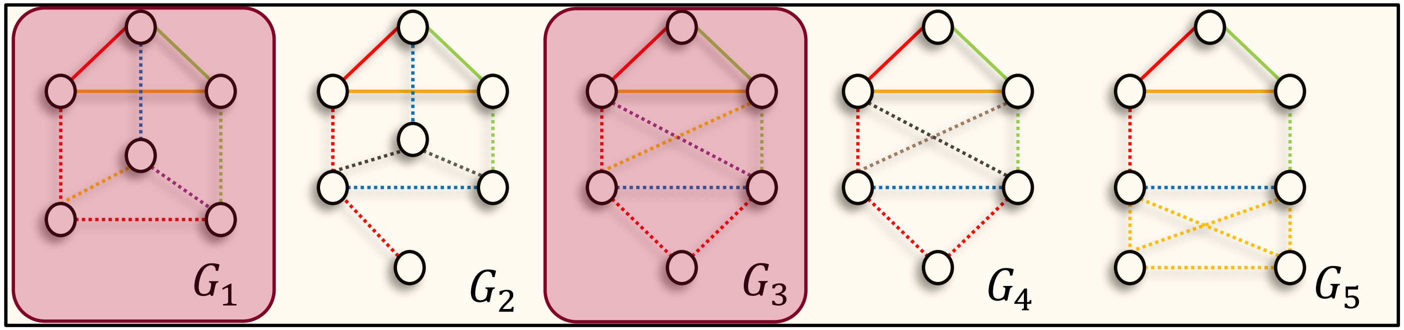


Redundant

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 $\vec{g}_i$: vectorial representation of G_i

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

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

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 θ

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

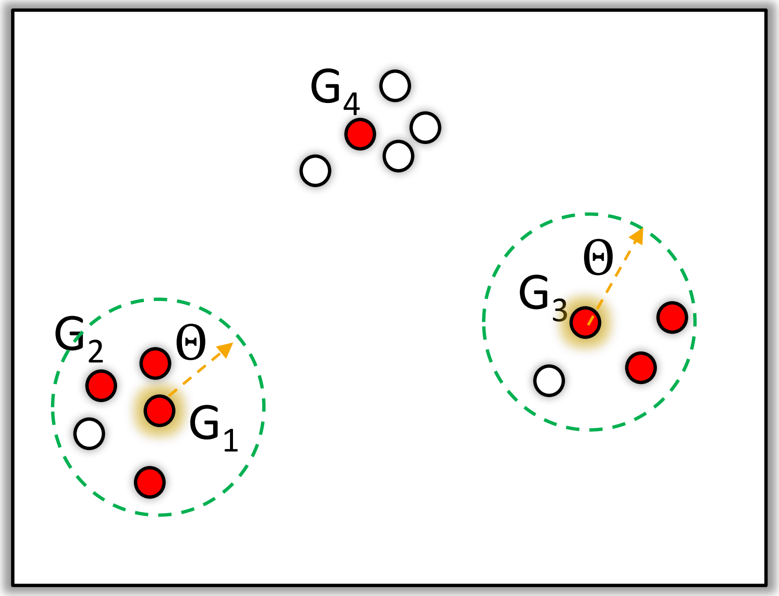
Representative power

R(q) = answers to the query

- q : query

θ-neighborhood

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



Given a set of graphs S

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

Represent the coverage of a graph neighborhood

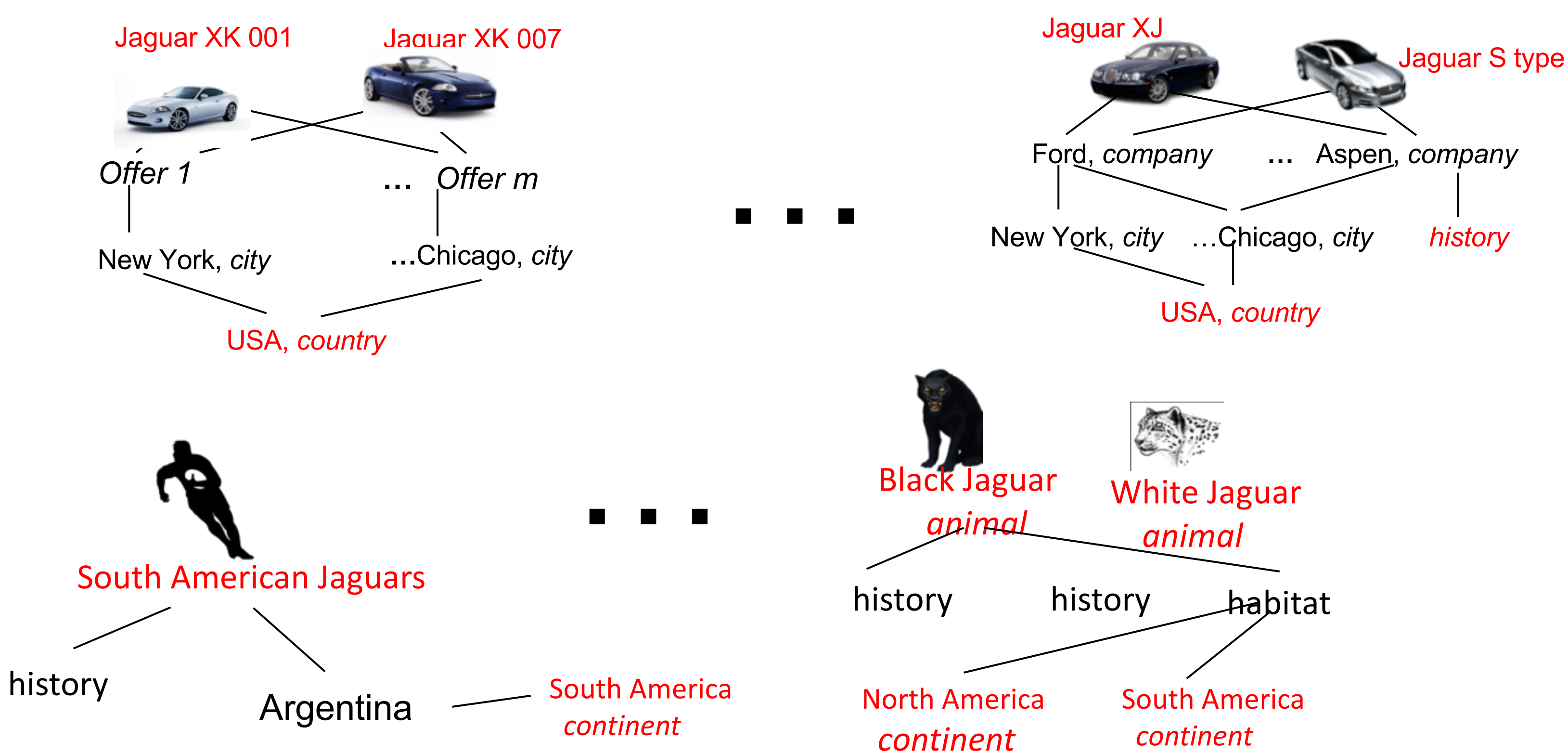
$$\pi(\{G_1, G_3\}) = \frac{7}{8}$$

$$\pi(\{G_1, G_2\}) = \frac{4}{8}$$

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

Summarizing graph results

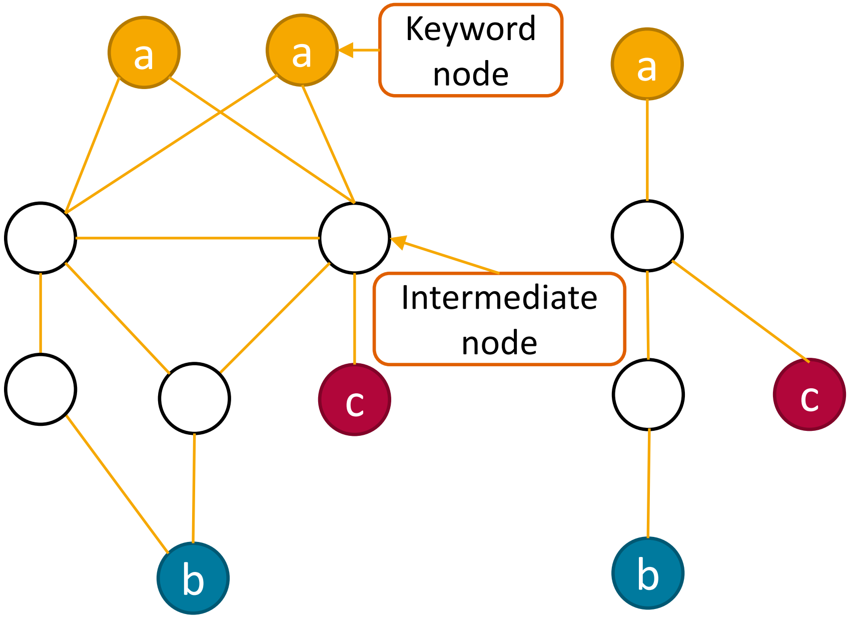
Query: keyword query on graph e.g., Jaguar, America, History



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)
$\alpha = 2 * M/(|Q|(|Q| - 1),$
M = number of covered keyword pairs

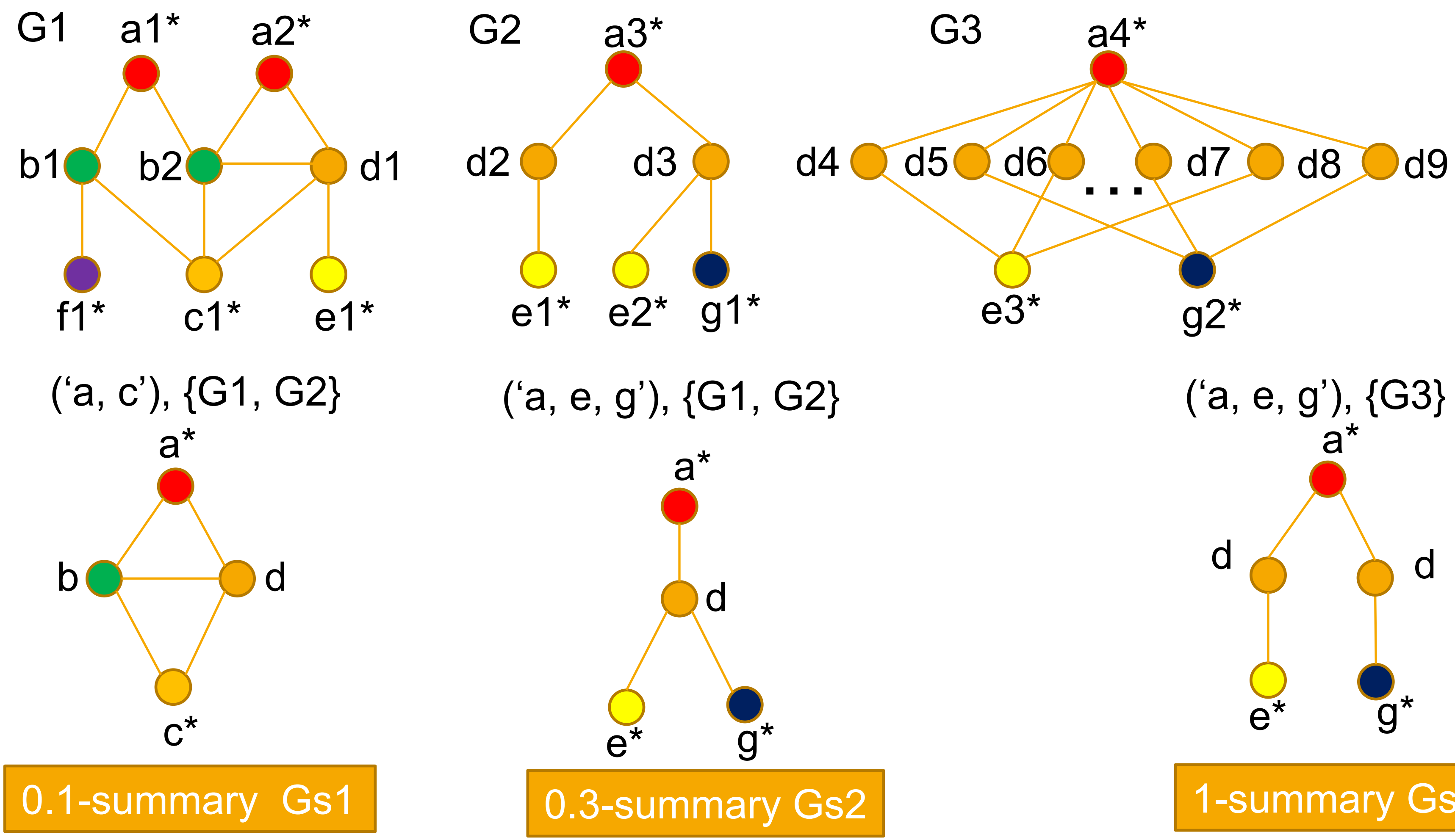
Two problems

1. Minimum α-summarization: find the **minimum size** summary which covers at least α
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}



Summarizing graph results algorithms

PTIME

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

NP-complete

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

NP-complete

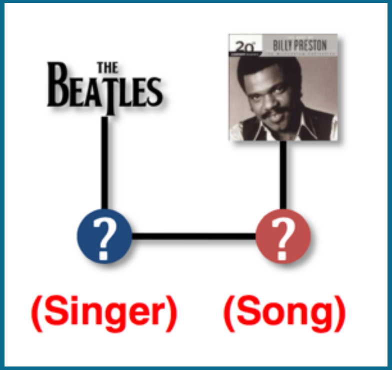
K-summarization

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

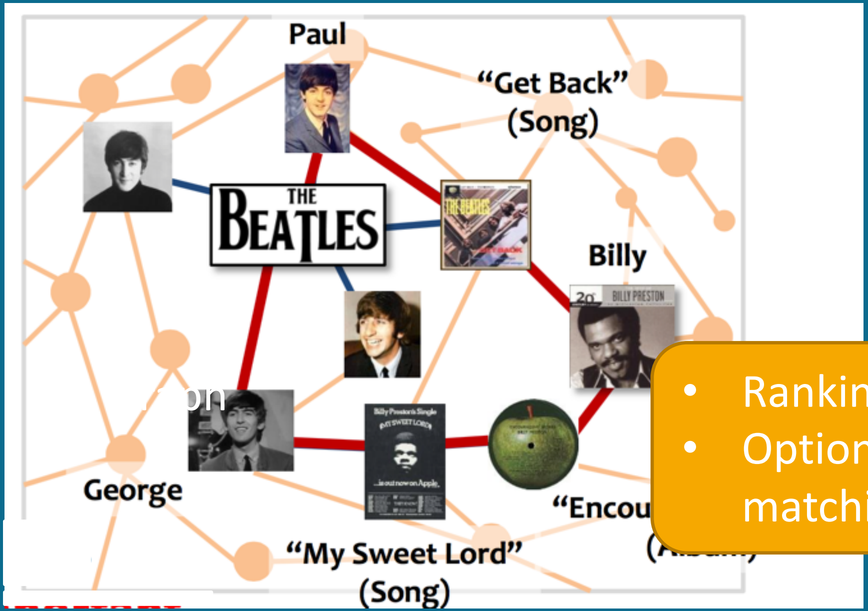
Wu, Y., Yang, S., Srivatsa, M., Iyengar, A. and Yan, X. Summarizing answer graphs induced by keyword queries. *PVLDB, 2013*

Top-k Results

- Large query results
- Find interesting exact and similar matches



Query



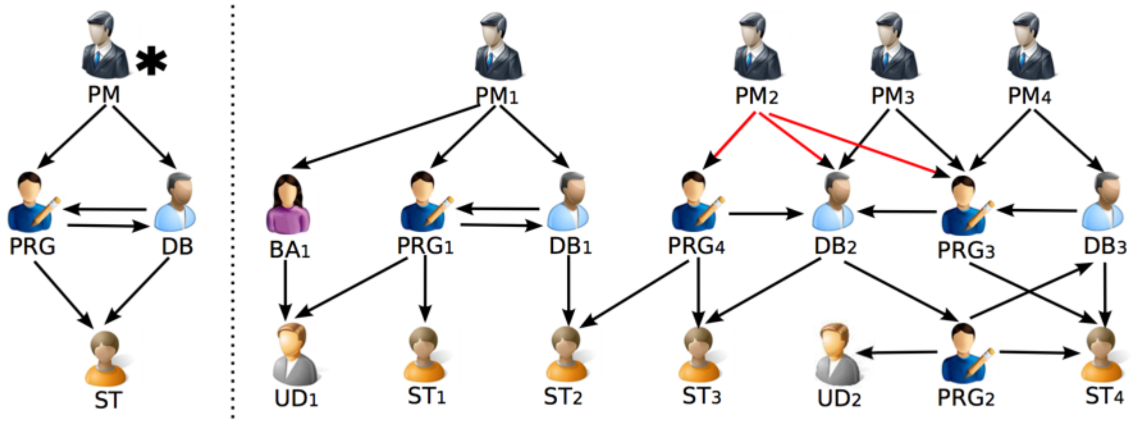
Solution

- 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).



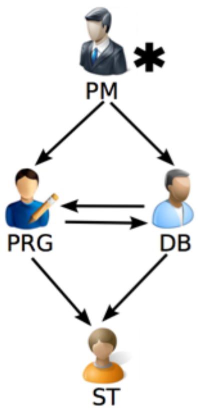
Pattern Q

Graph G

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

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

Diversified top-k graph pattern matching

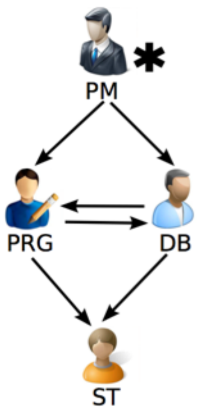


Pattern Q

- 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.

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

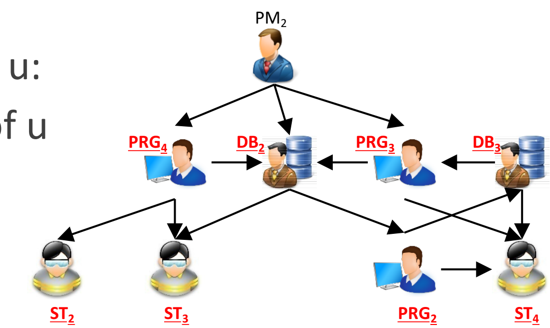
Diversified top-k graph pattern matching



Pattern Q

- 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

Relevance

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

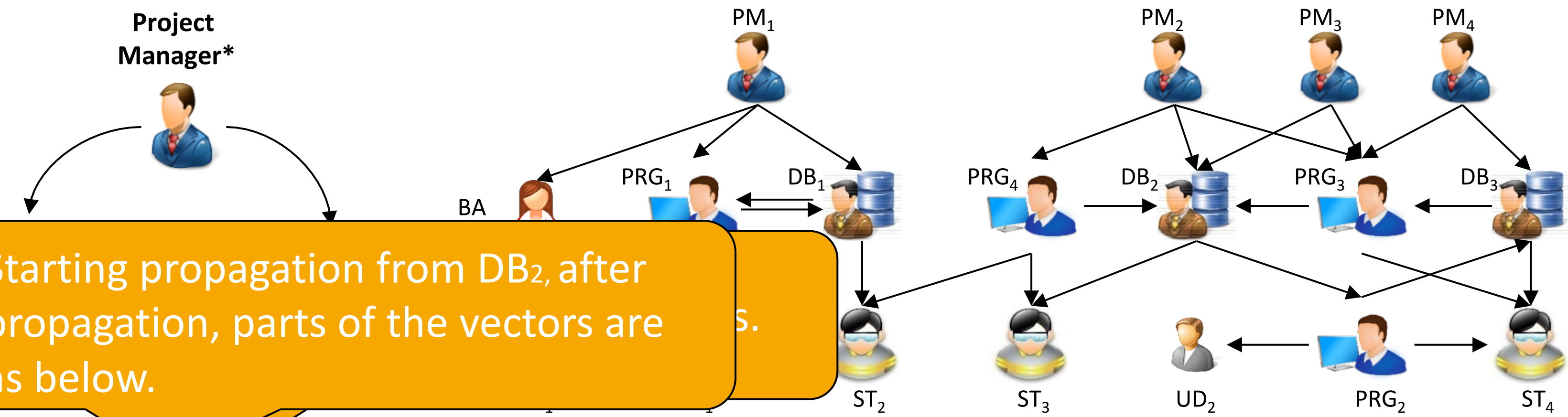
Diversity

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

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

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

Finding Top-k Matches (acyclic)



Starting propagation from DB_2, after propagation, parts of the vectors are as below.

v	v.T = <v.bf, v.R, v.l, v.h>
PM_1	$\langle X_{PM1} = X_{PRG1} \wedge X_{DB1}, \Phi, 0, 2\rangle$
PM_2	$\langle X_{PM2} = ((X_{PRG3} = true) \vee (X_{PRG4}=true)) \wedge X_{DB2}=true, \{DB_2, PRG_4, PRG_3\}, 3, 3\rangle$
PM_3	$\langle X_{PM3} = (X_{PRG3} = true) \wedge (X_{DB2}=true), \{DB_2, PRG_3\}, 2, 2\rangle$
PM_4	$\langle X_{PM4}$
PRG_1	$\langle X_{PRG1}$
PRG_j ($j \in [3,4]$)	$\langle X_{PRGj}$
DB_2	$\langle X_{DB2}$
DB_k ($k \in [1,3]$)	$\langle X_{DBk}$

PM2 is verified to be a valid match, and its relevant set includes $\{DB_2, PRG_4, PRG_3\}$, 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

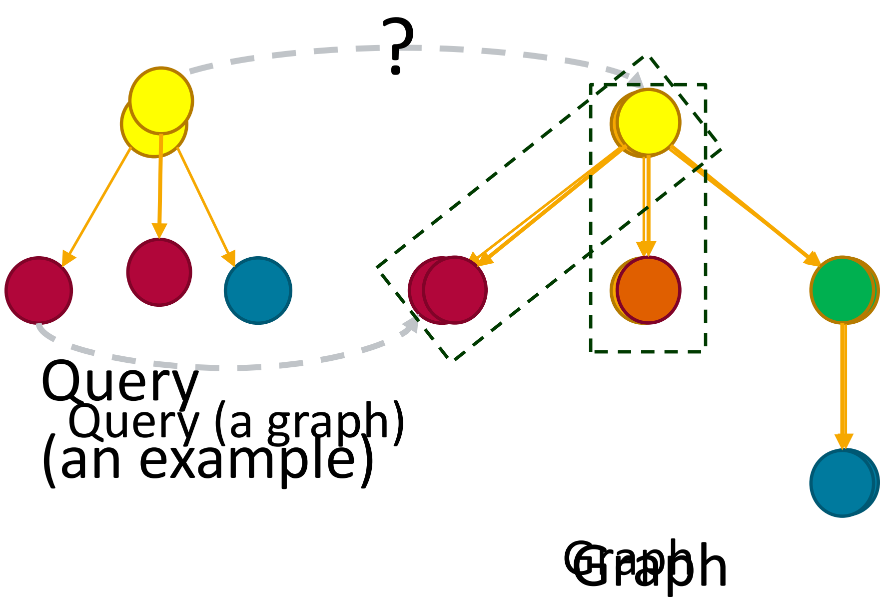
Summary of Exploratory Graph Analysis

Approximate Queries

- User query is imprecise

By-Example methods

- User query is an example result



- Only need a 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

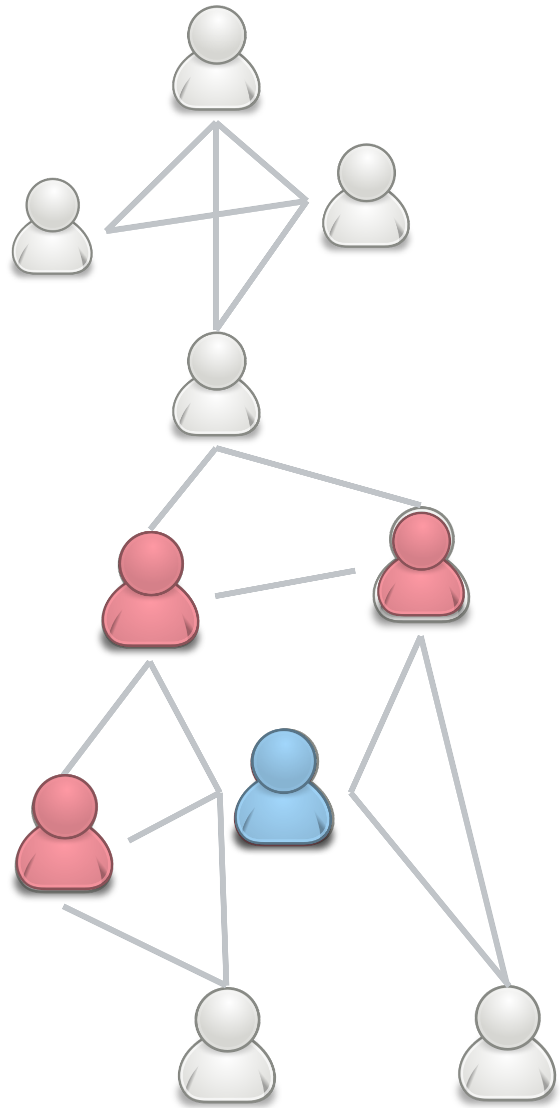
The focus on individual user interest

... as **Query** to the Graph Mining System

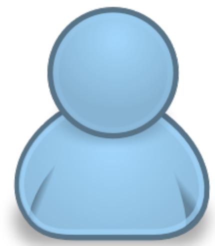
... as **Seed Node(s)** for Local Search

... 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

Data mining

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

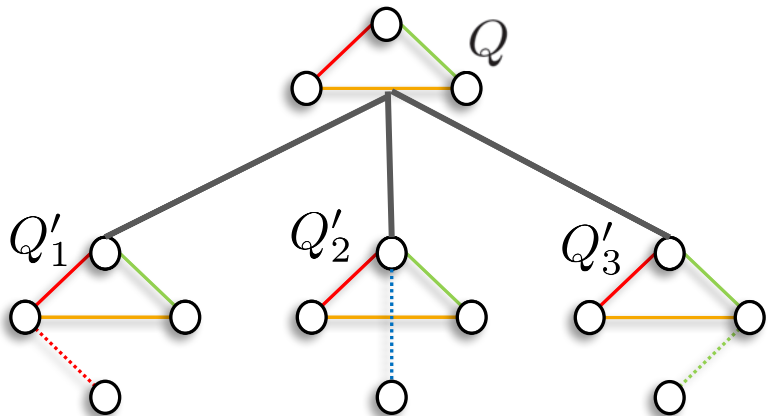
Top-k Results

- 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

Database

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

Data mining

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

scale

- 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

[Ma14] Ma, S., Cao, Y., Fan, W., Huai, J. and Wo, T. Strong simulation: Capturing topology in graph pattern matching. *TODS, 2014*

[Fan10] Fan, W., Li, J., Ma, S., Wang, H. and Wu, Y.. Graph homomorphism revisited for graph matching. PVLDB, 2010

[Khan13] Khan, A., Wu, Y., Aggarwal, C.C. and Yan, X. Nema: Fast graph search with label similarity. PVLDB, 2013

[Yang14] Yang, S., Wu, Y., Sun, H. and Yan, X. Schemaless and structureless graph querying. *PVLDB, 2014*.

[Mottin14] Mottin, D., Lissandrini, M., Velegrakis, Y. and Palpanas, T. Exemplar queries: Give me an example of what you need. *PVLDB 2014*

[Jayaram15] Jayaram, N., Khan, A., Li, C., Yan, X. and Elmasri, R. Querying knowledge graphs by example entity tuples. *TKDE, 2015*

References

[Tong06] H. Tong & C. Faloutsos: Center-Piece Subgraphs: Problem Definition and Fast Solutions. (KDD 2006)

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

[Epasto15] Epasto et al. Ego-Net Community Mining Applied to Fried Suggestion. (VLDB 2015)

[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

[Mottin15] Mottin, D., Bonchi, F. and Gullo, F. Graph Query Reformulation with Diversity. KDD, 2015

[Vasilyeva16] Vasilyeva, E., Thiele, M., Bornhövd, C. and Lehner, W.. Answering "Why Empty?" and "Why So Many?" queries in graph databases. *JCSS, 2016*

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

[Wu13] Wu, Y., Yang, S., Srivatsa, M., Iyengar, A. and Yan, X. Summarizing answer graphs induced by keyword queries. *PVLDB, 2013*

[Fan13] Fan, W., Wang, X. and Wu, Y. Diversified top-k graph pattern matching. *VLDB*, 2013

[Gupta14] Gupta, M., Gao, J., Yan, X., Cam, H. and Han, J. Top-k interesting subgraph discovery in information networks. ICDE, 2014

[Zou10] Zou, L., Chen, L., Özsu, M.T. and Zhao, D. Dynamic skyline queries in large graphs. DASFAA, 2010