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

Exploratory Search

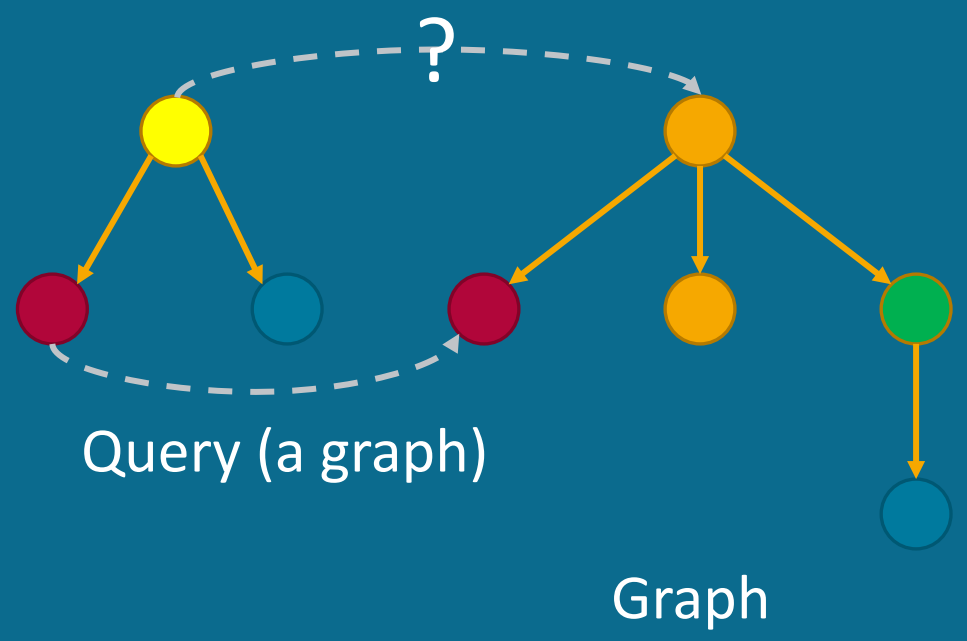
Approximate Graph Search

- 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

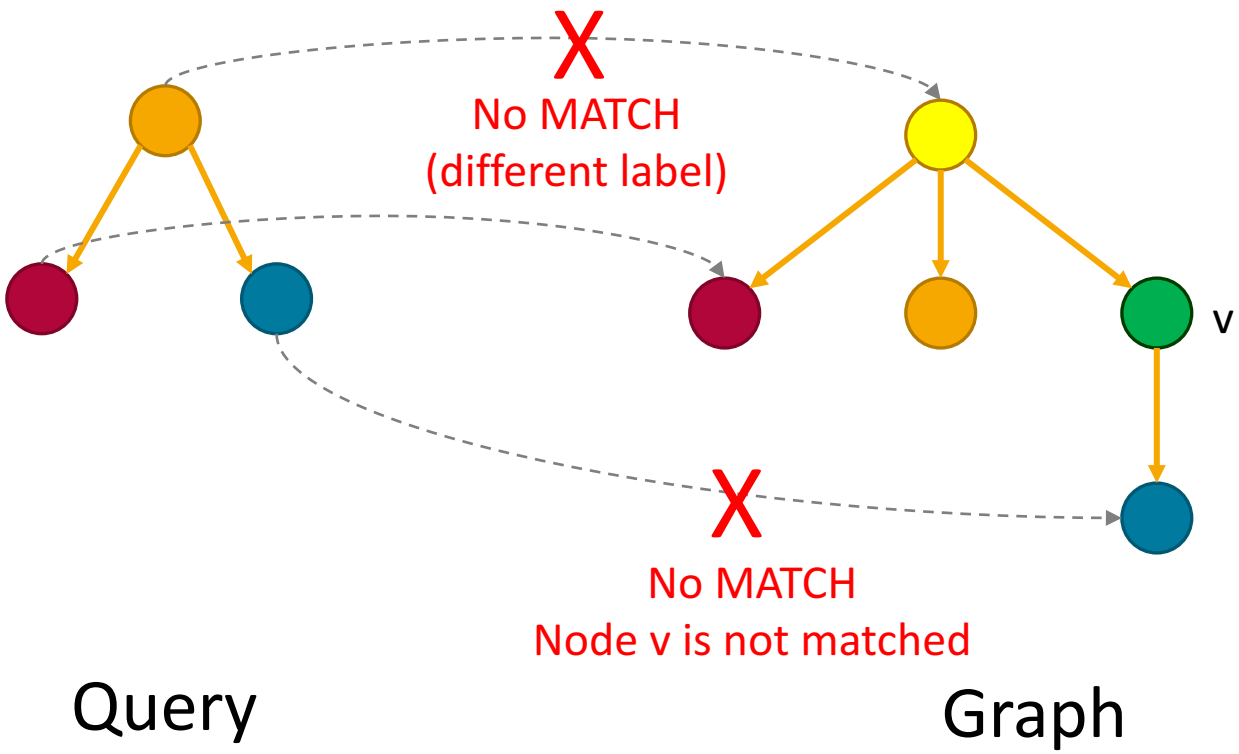
Solution

- 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

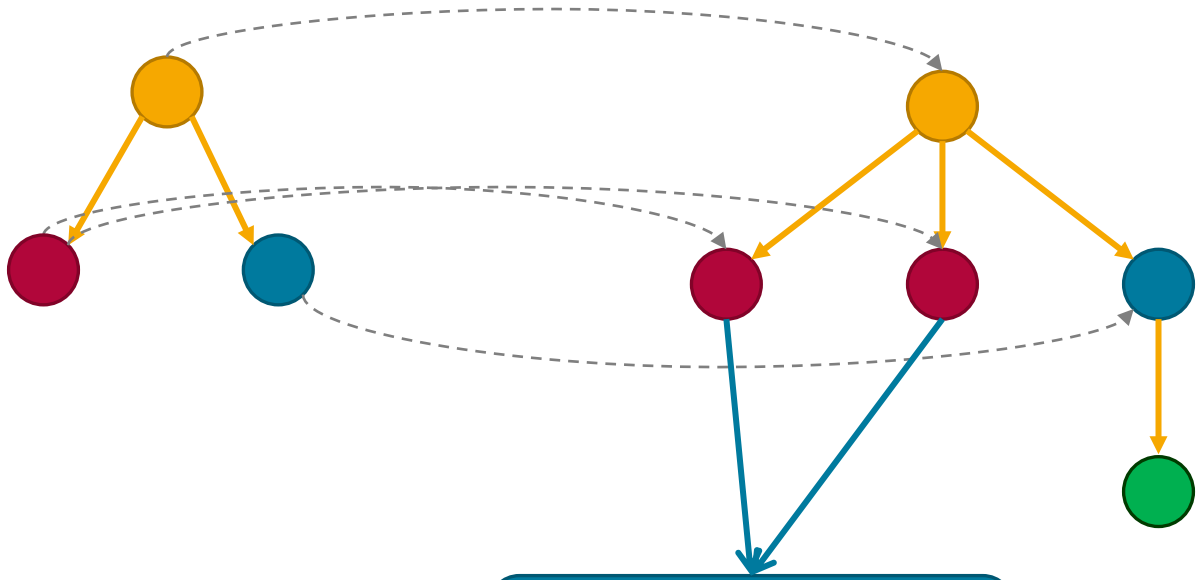
(Sub)Graph Isomorphism might be too restrictive



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

Duality

Locality

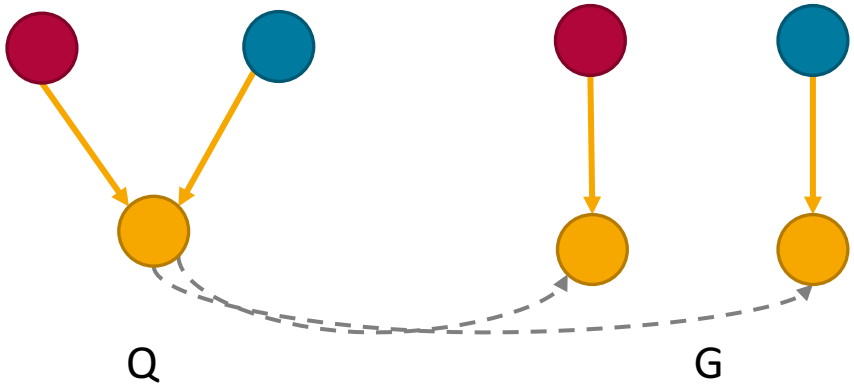
Graph Simulation [Milner 1989]

Parent-child relationship

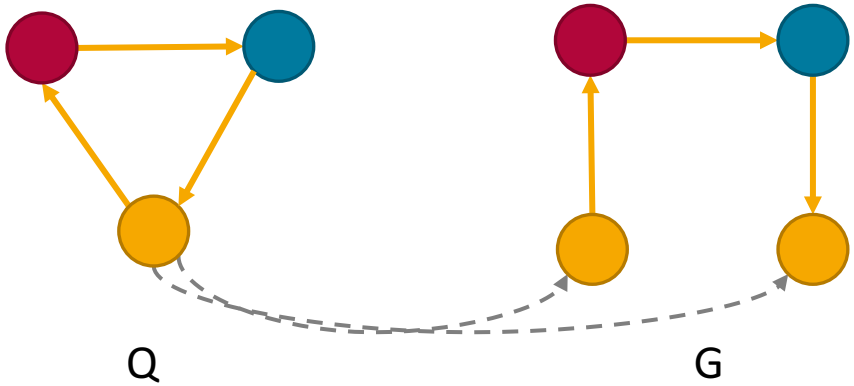
Child-parent relationship

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

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

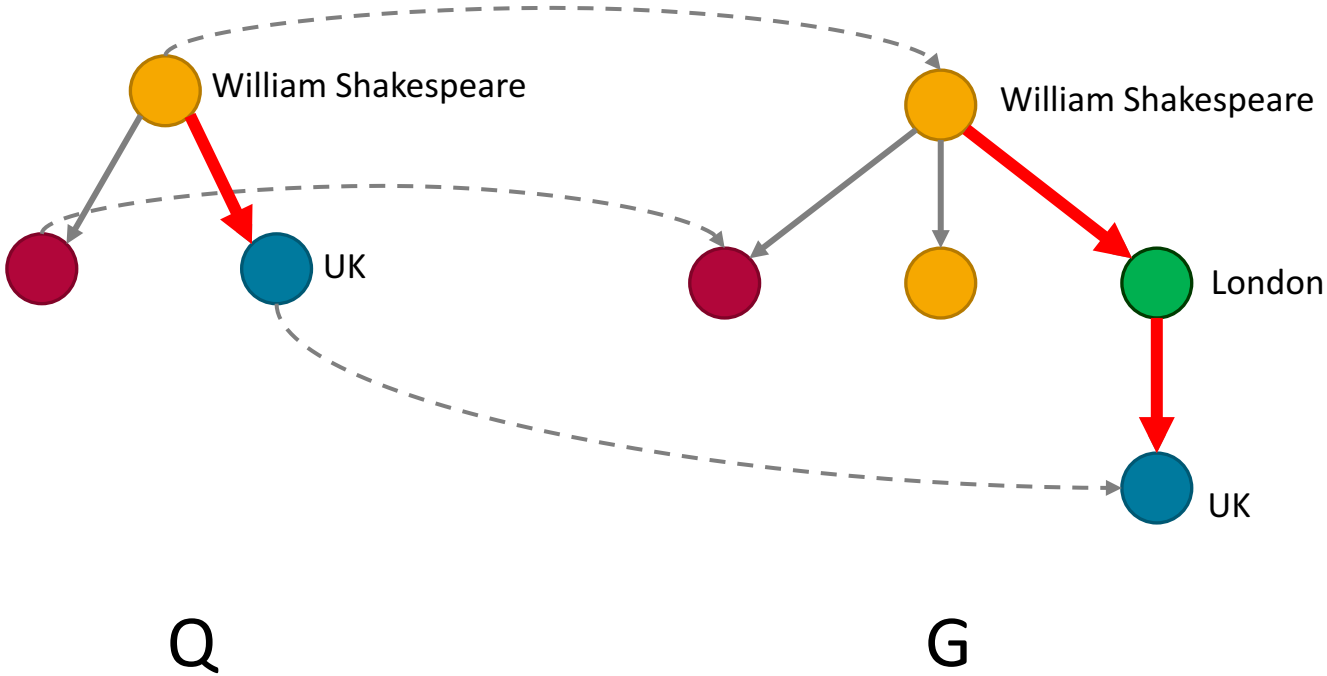
If Q matches G, via dual simulation,
then Q matches G, via graph simulation

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*

Graph homomorphism

Revise graph homomorphism: match paths

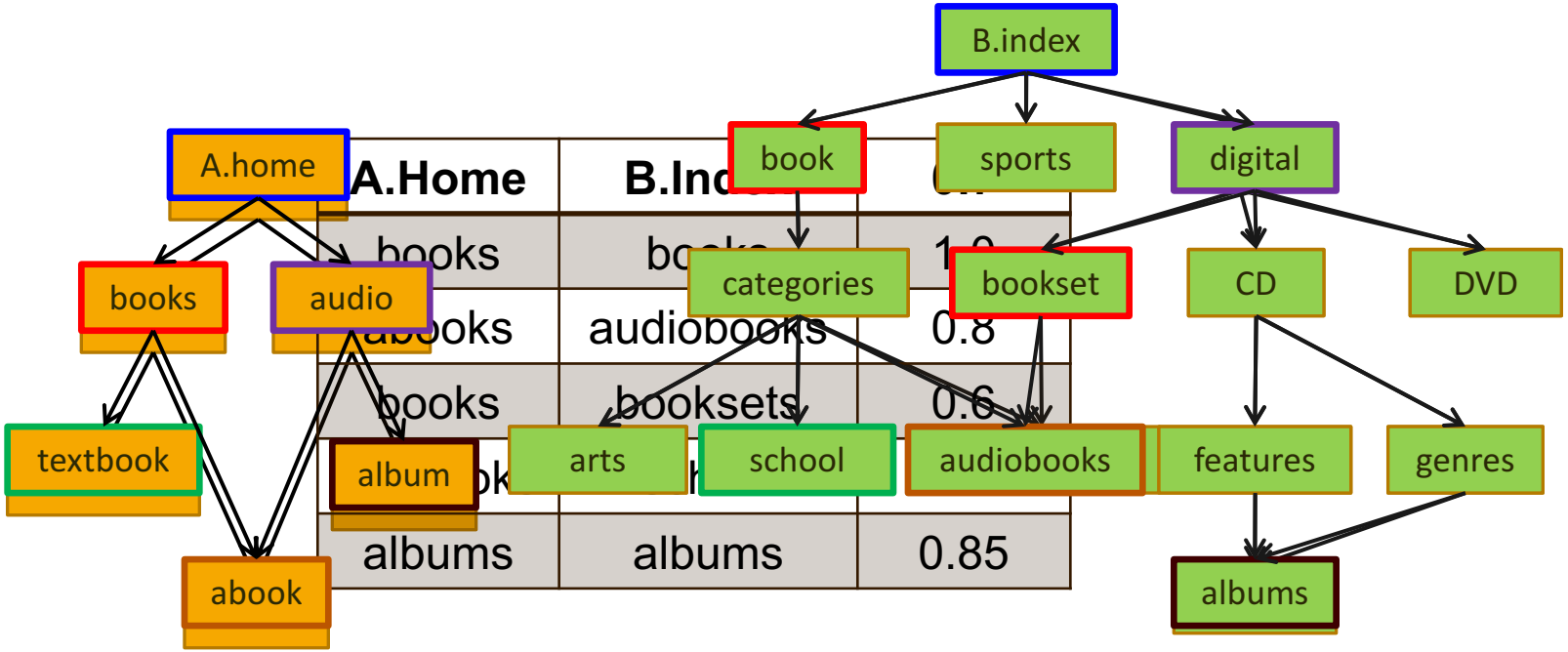


Fan, W., Li, J., Ma, S., Wang, H. and Wu, Y.. Graph homomorphism revisited for graph matching. PVLDB, 2010

P-Homomorphism

NP-complete

- 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 ξ

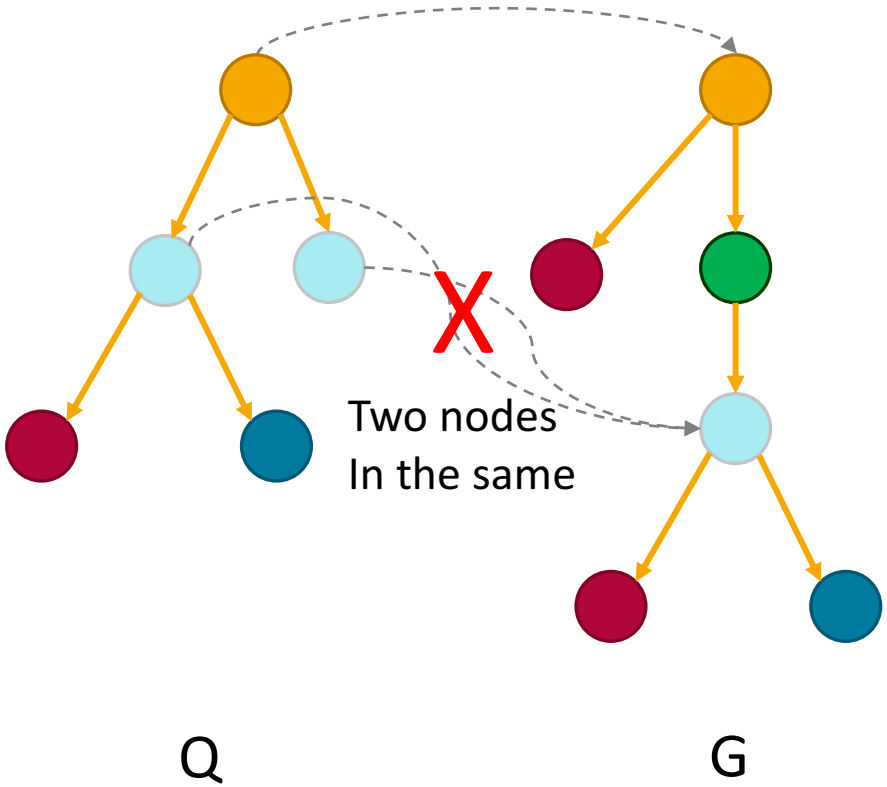


Fan, W., Li, J., Ma, S., Wang, H. and Wu, Y.. Graph homomorphism revisited for graph matching. PVLDB, 2010

1-1 P-Homomorphism

NP-complete

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 ρ with maximum Card(ρ).
- The cardinality of p-hom mapping from a subgraph G' = (V', E',L') of Q to G:
  - $Card(\rho) = |V'|/|V_Q|$

Maximum Overall similarity (SPH)

- Return the (1-1)P-hom mapping ρ with maximum Sim(ρ) .
- The overall similarity of p-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(G_1,G_2,M, ξ)

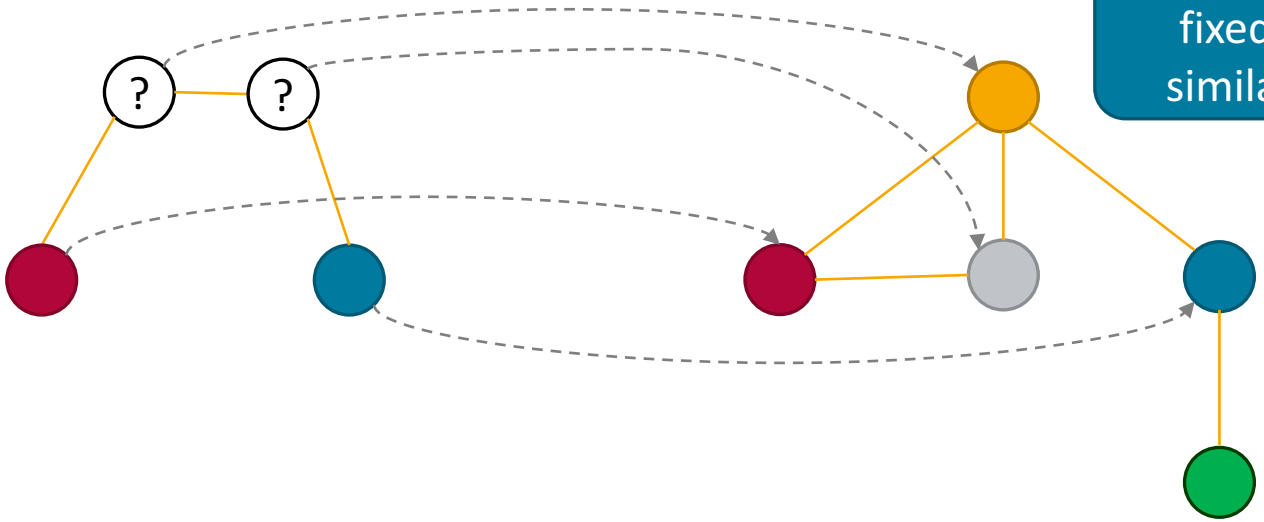
- Input: $G_1 = (V_1, E_1, L_1)$, $G_2 = (V_2, E_2, L_2)$, similarity matrix M, similarity threshold ξ
- Output: a P-hom mapping from subgraph of G_1 to G_2
- Procedure
  - initialize matching list for each node in G_1
  - compute the transitive closure of G_2 (connect two nodes if a path exists)
  - 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 **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

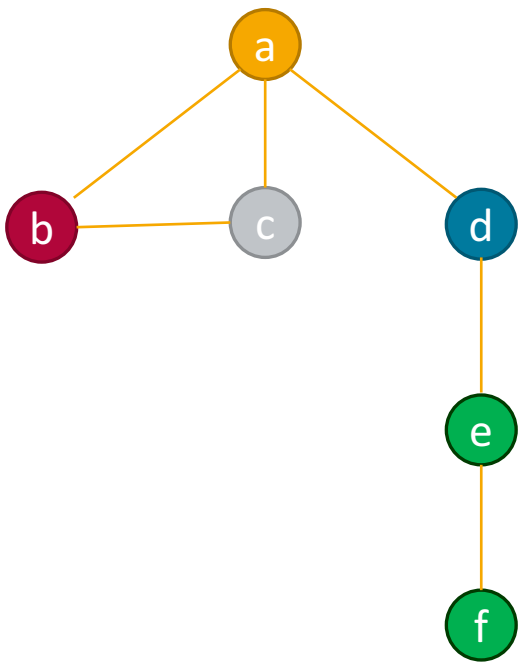
NeMa: compute node vectors

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

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)



$h = 2, \alpha = 0.5$

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

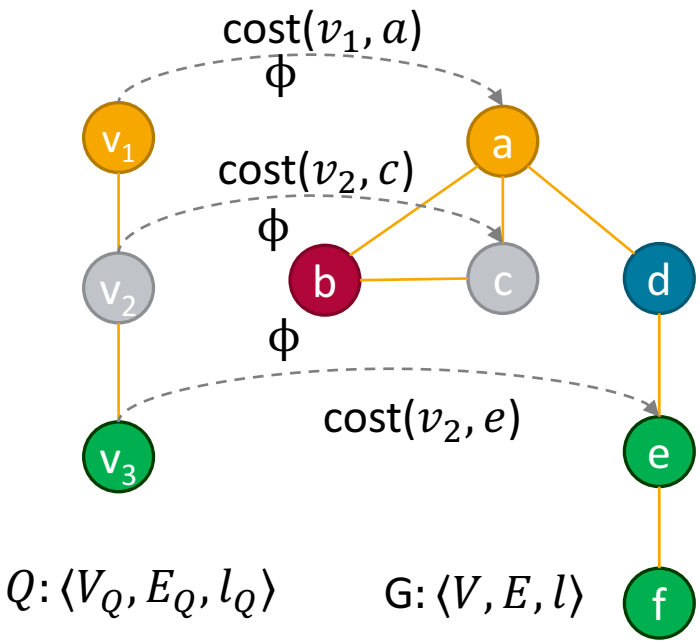
Vector of nodes at distance <= h from a

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

NeMa

NP-hard

APX-hard



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

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 $C(\phi)$

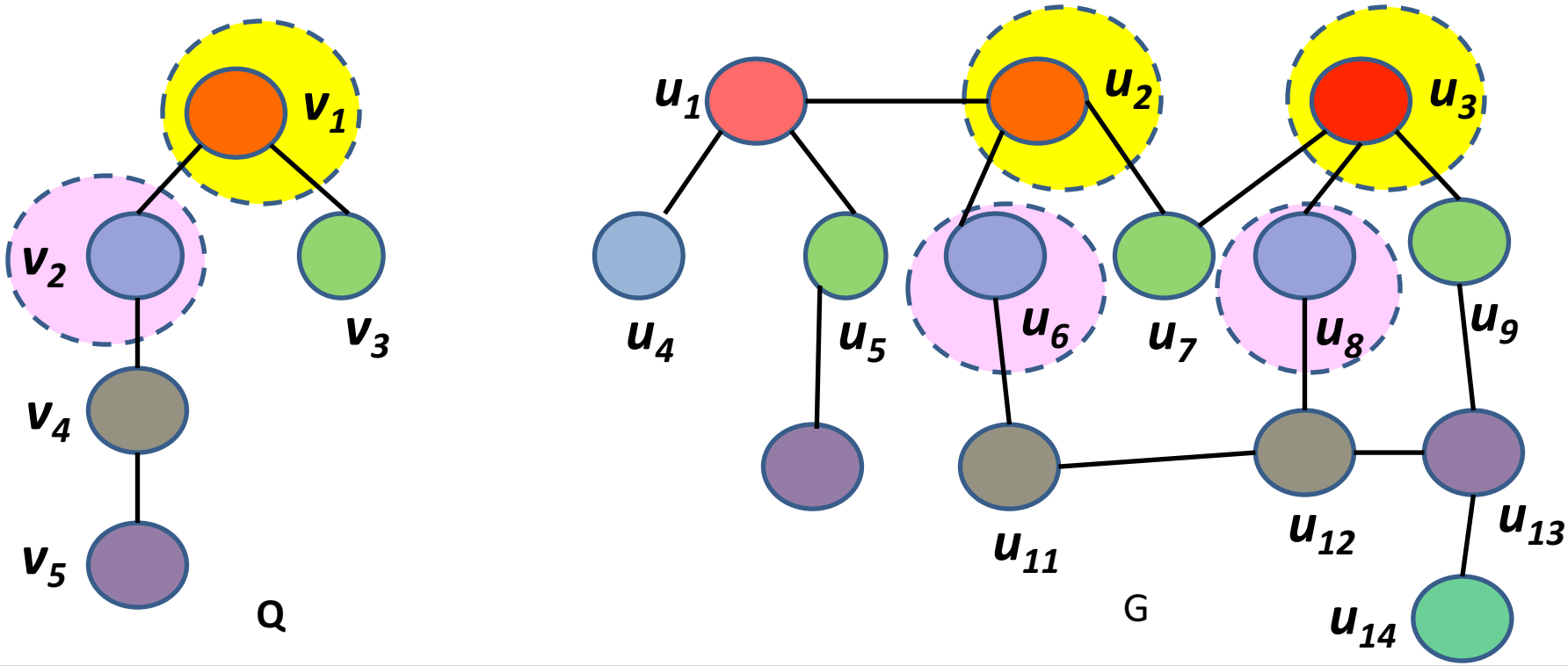
Solved with a belief propagation approach

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

NemaInfer algorithm

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

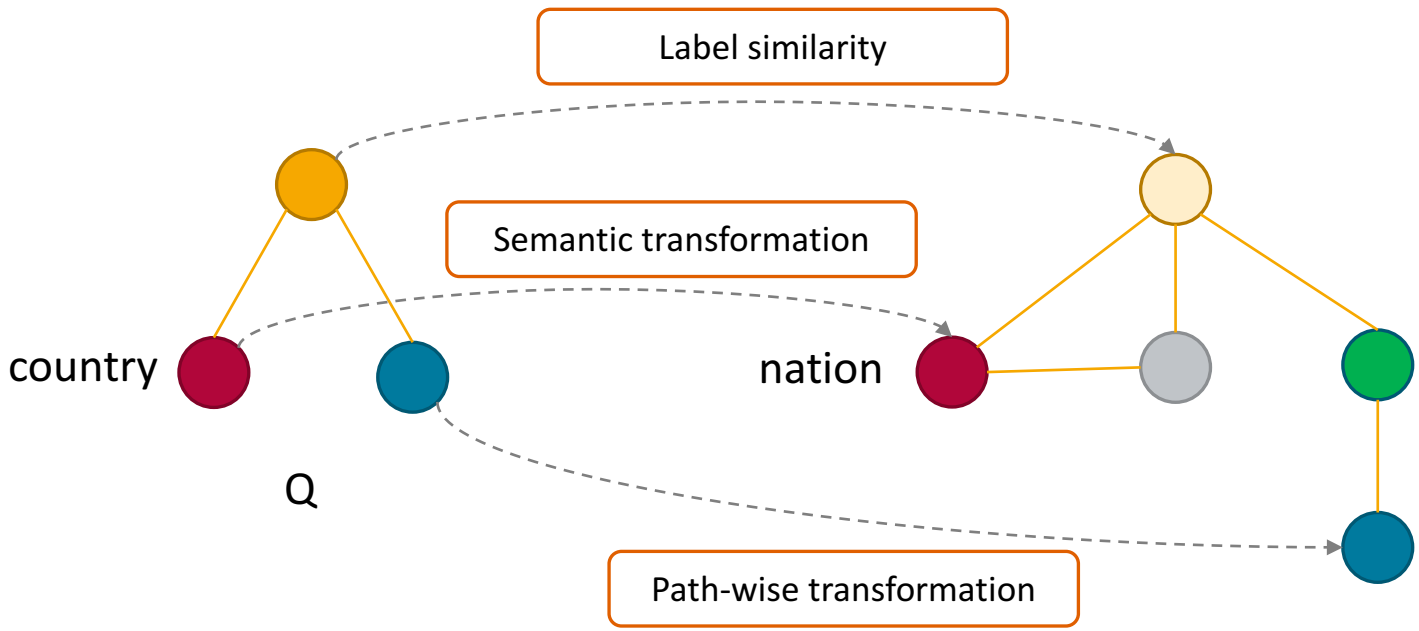
$$U_i(v,u) = \min_{\{\phi:\phi(v)=u\}} \left[F_\phi(v,u) + \sum_{v' \in \mathbb{N}(v)} U_{i-1}\left(v',u'\right)\right]$$



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*.

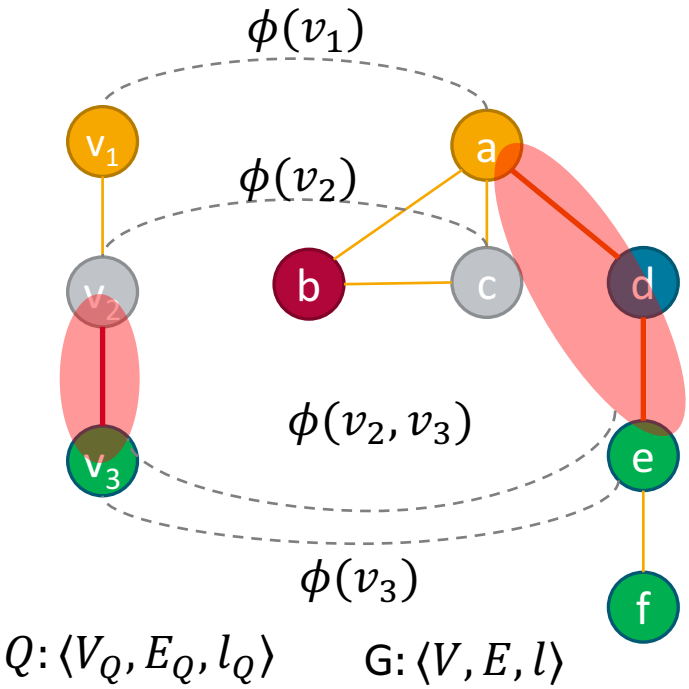
Transformations

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

They can be expanded arbitrarily

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

Model on transformations



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

Node matching score

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

Edge matching score

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

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 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

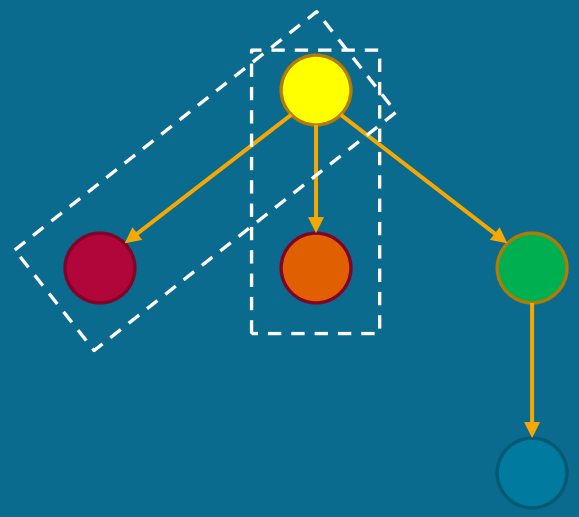
Query phase

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

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

Querying by Example

Query
(an example)



Graph

- 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

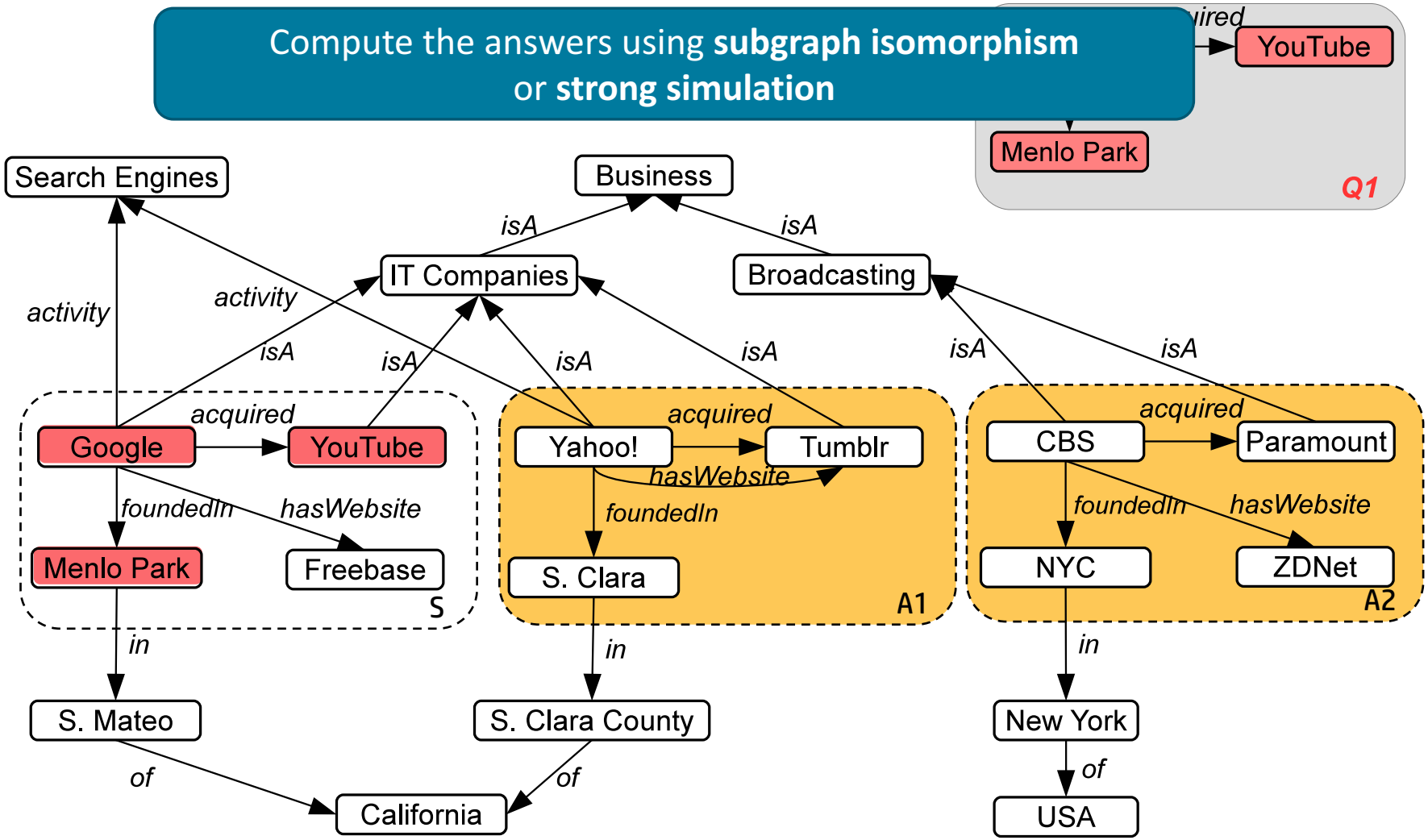
Exemplar Query Evaluation

- 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

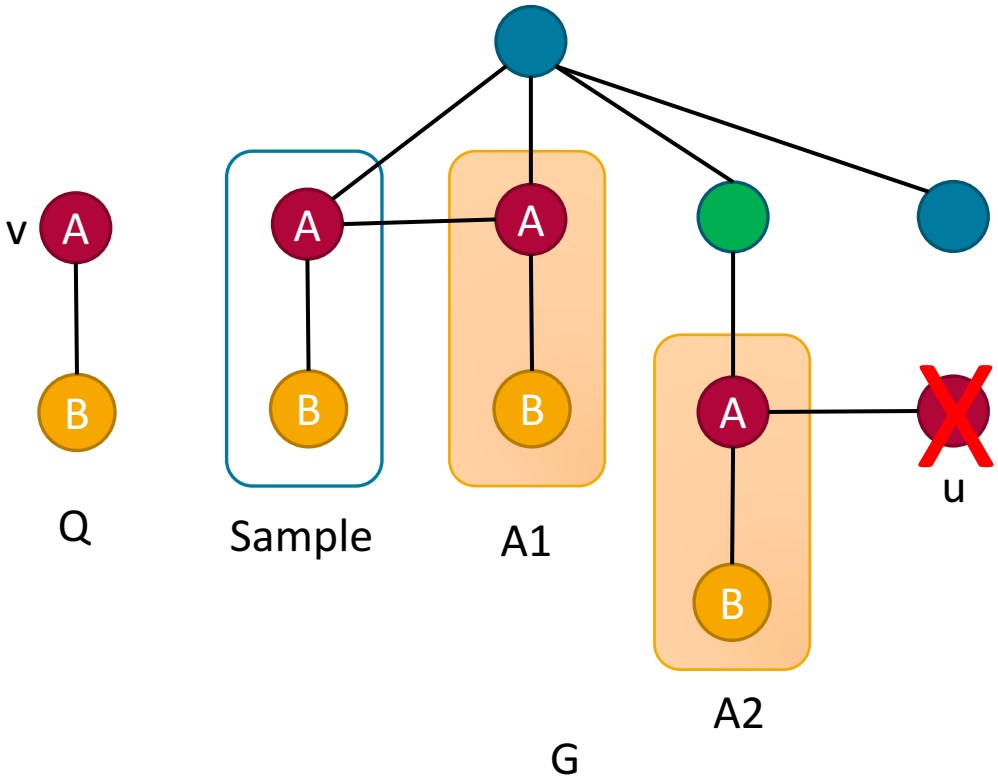
Compute the answers using **subgraph isomorphism** or **strong simulation**



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)}

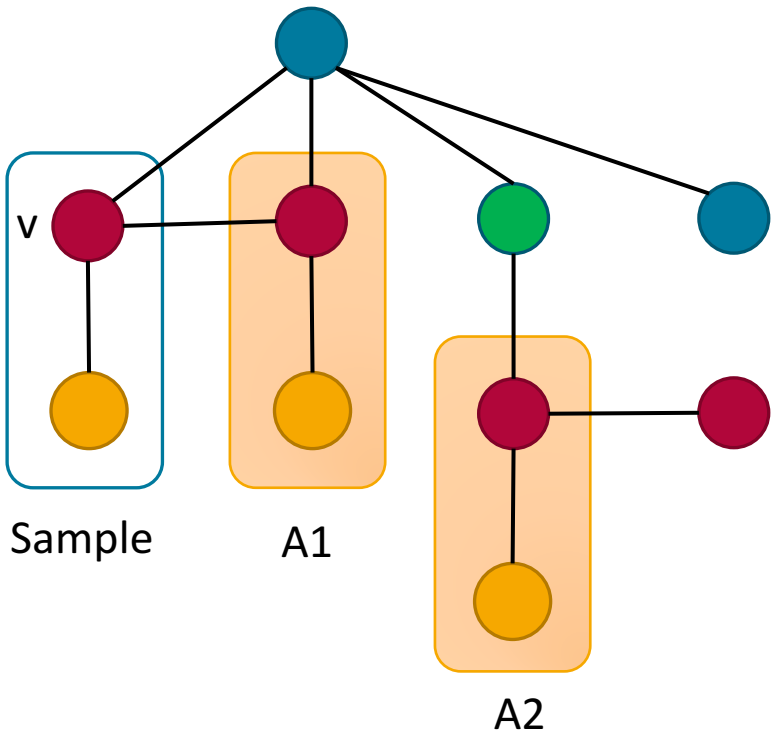
No Match

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

Computing exemplar queries

NP-complete (subgraph isomorphism)

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



Approximation:

- Nodes closed to the sample are more important
- Use Personalized PageRank with a weighted matrix

$$\boldsymbol{v} = (1-c)A\boldsymbol{v} + c\boldsymbol{p}$$

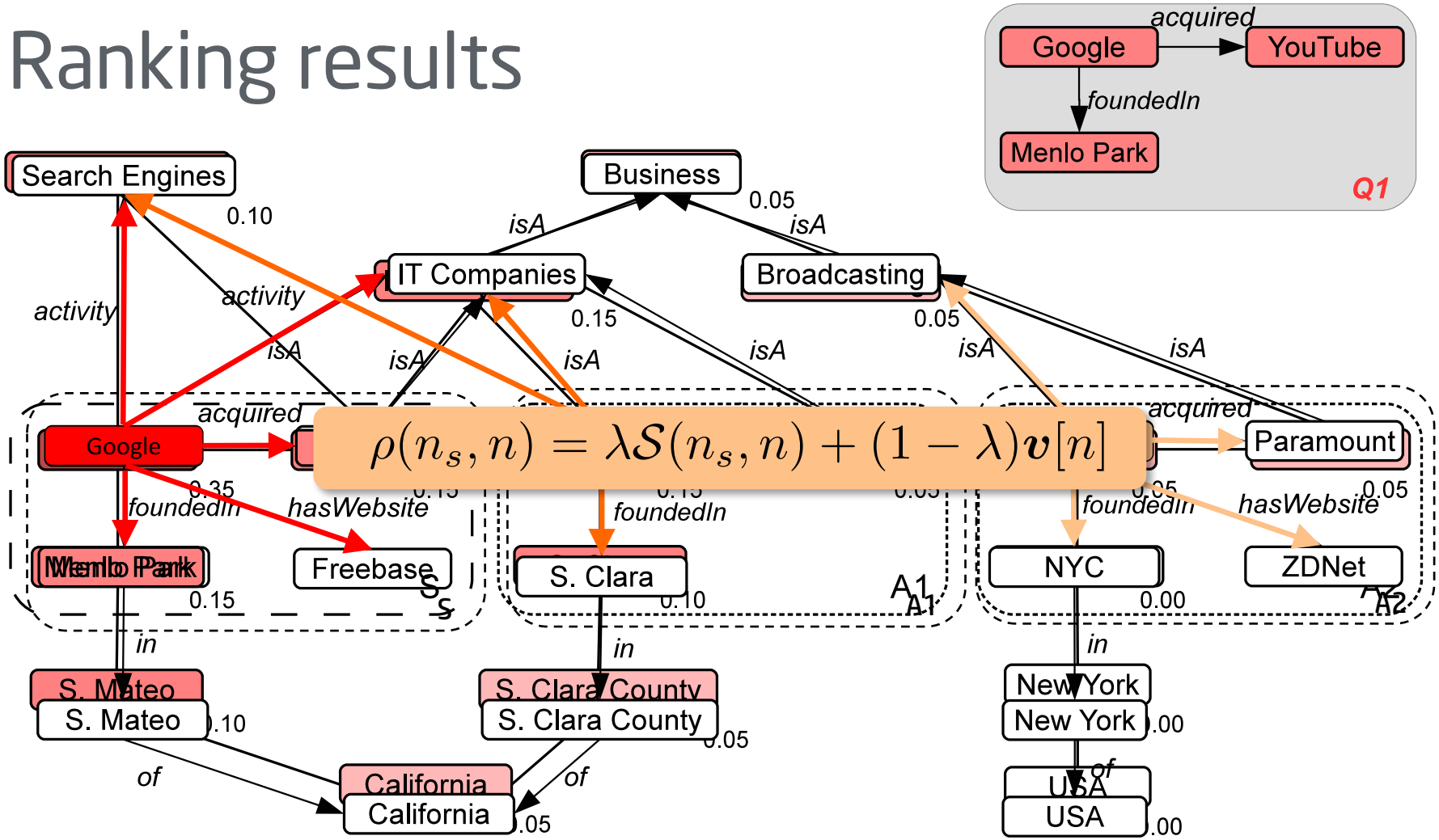
- Weight edges using the frequency of the edge-label

$$I(e_{ij}^{\ell}) = I(\ell) = \log \frac{1}{P(\ell)} = -\log P(\ell)$$

$$P(\ell) = \frac{|E^{\ell}|}{|E|}$$

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

Ranking results



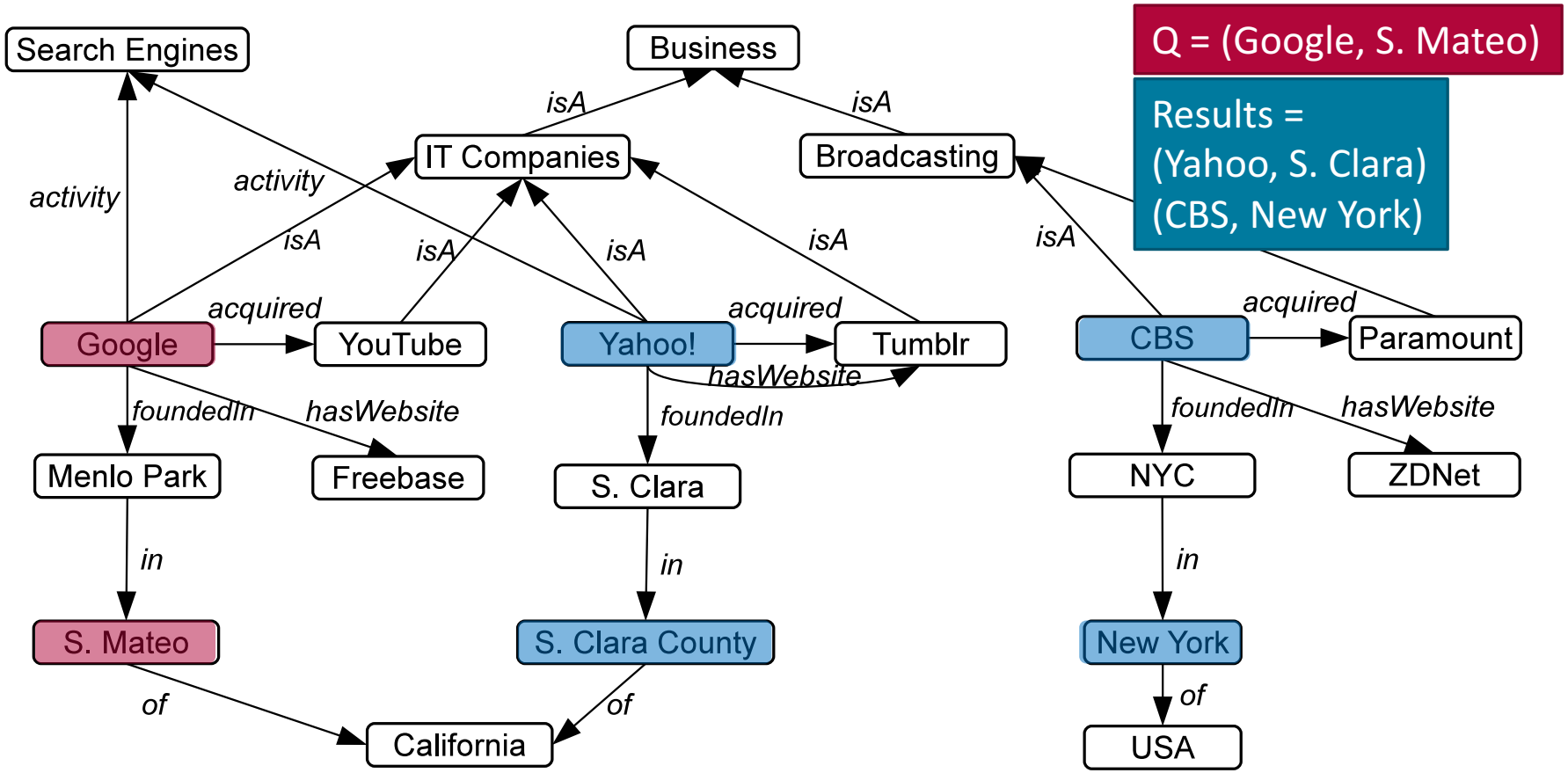
$$\rho(n_s, n) = \lambda \mathcal{S}(n_s, n) + (1 - \lambda)\boldsymbol{v}[n]$$

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)

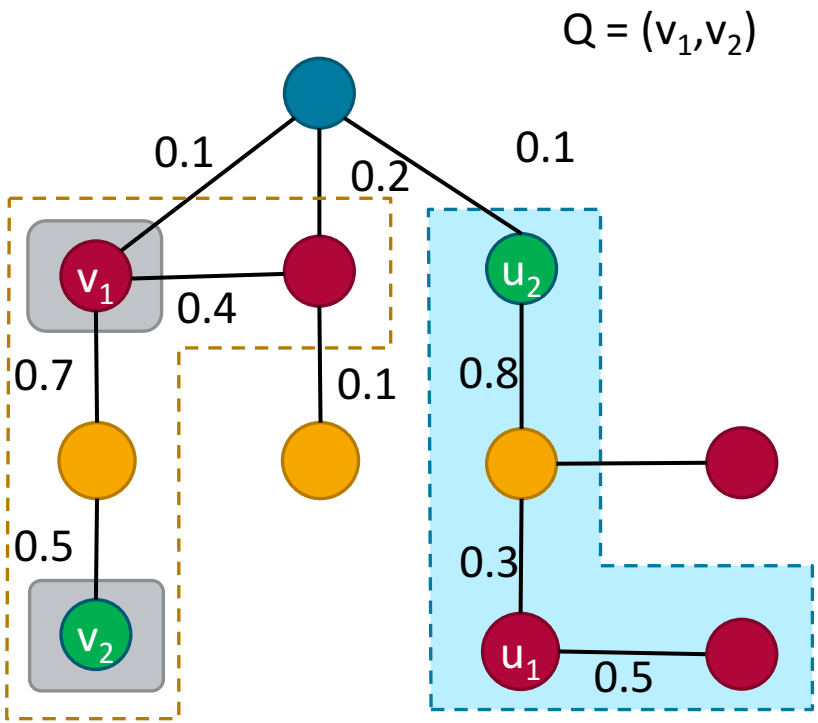
In GQBE Input is a set of (disconnected) entity mention tuples



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

GQBE

NP-hard



Maximum Query Graph

Answer graph

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

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

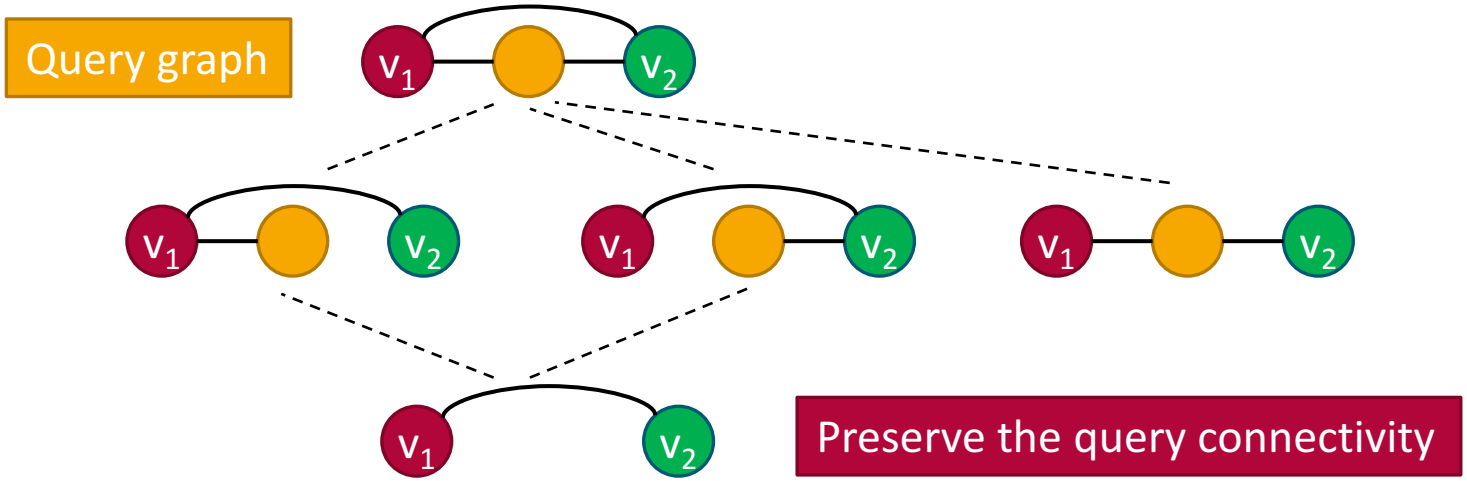
Jayaram, N., Khan, A., Li, C., Yan, X. and Elmasri, R. Querying knowledge graphs by example entity tuples. *TKDE, 2015*

Multiple query tuples

NP-hard

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*