ICDM 2016
IEEE International Conference on Data Mining
BARCELONA

Mining Summaries for Knowledge Graph Search

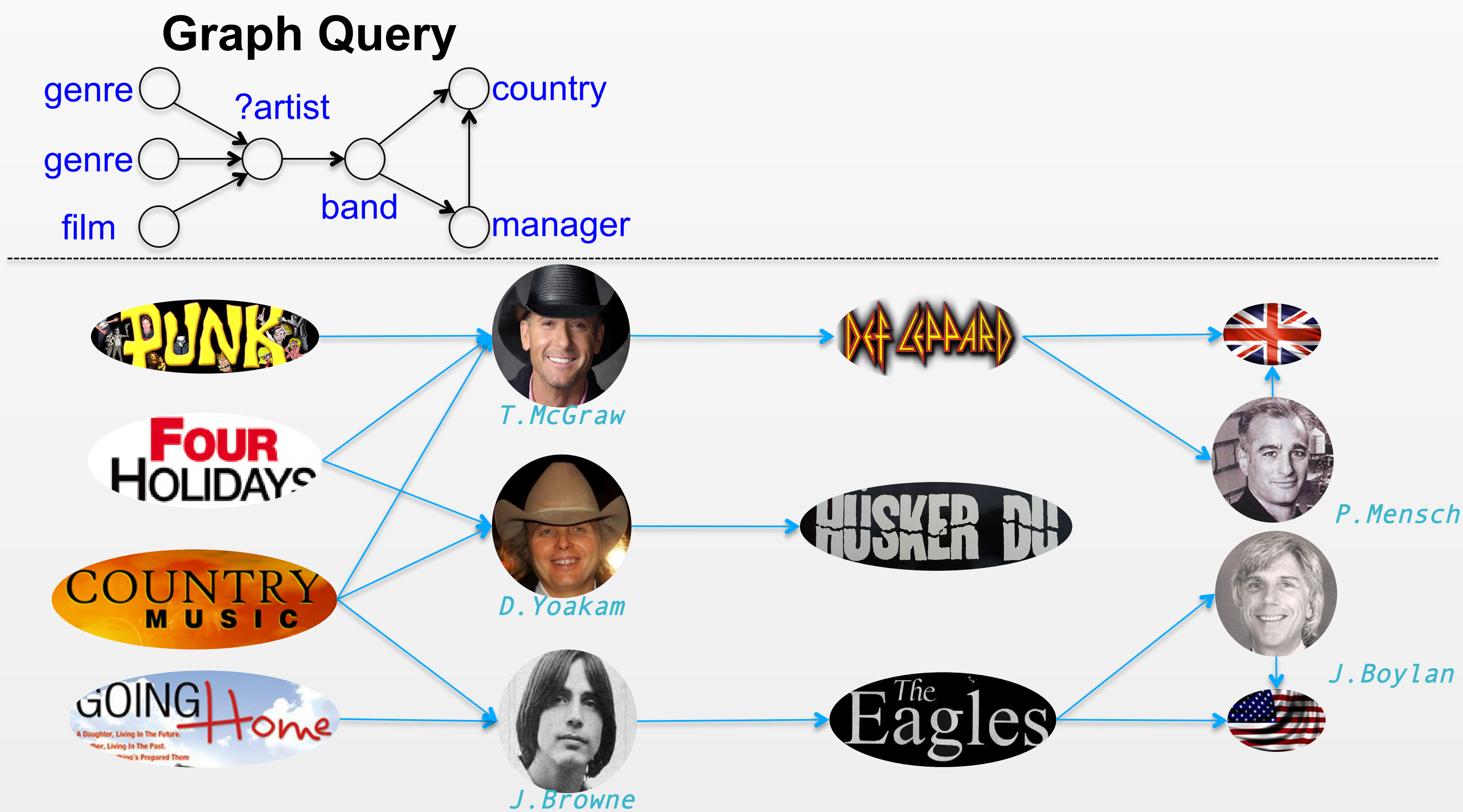
Qi Song[1] Yinghui Wu[1] Xin Luna Dong[2]

[1] Washington State University [2] amazon

Sponsored by: NSF Google Faculty Research Awards

Searching real world graph data

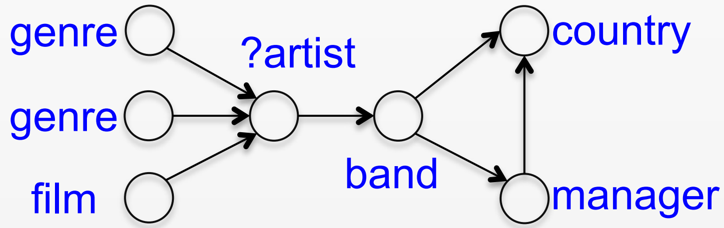
- Knowledge Graph *G*: used to represent knowledge bases
- Graph query *Q*: graph with types on each node



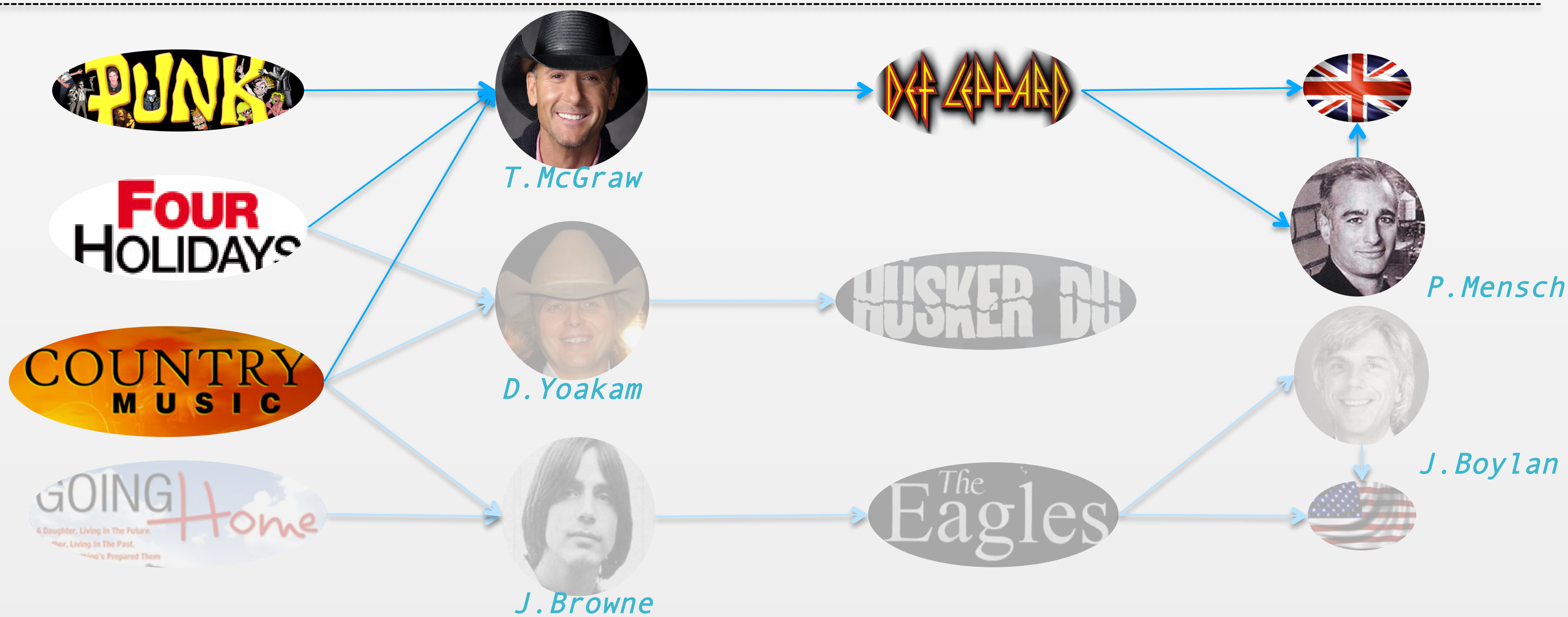
Searching real world graph data

- Knowledge Graph *G*: used to represent knowledge bases
- Graph query *Q*: graph with types on each node
- Answer *Q(G)*: the set of entities with certain type in the subgraphs of *G* that are isomorphic to *Q*.
- Challenges: usability & scalability

Graph Query

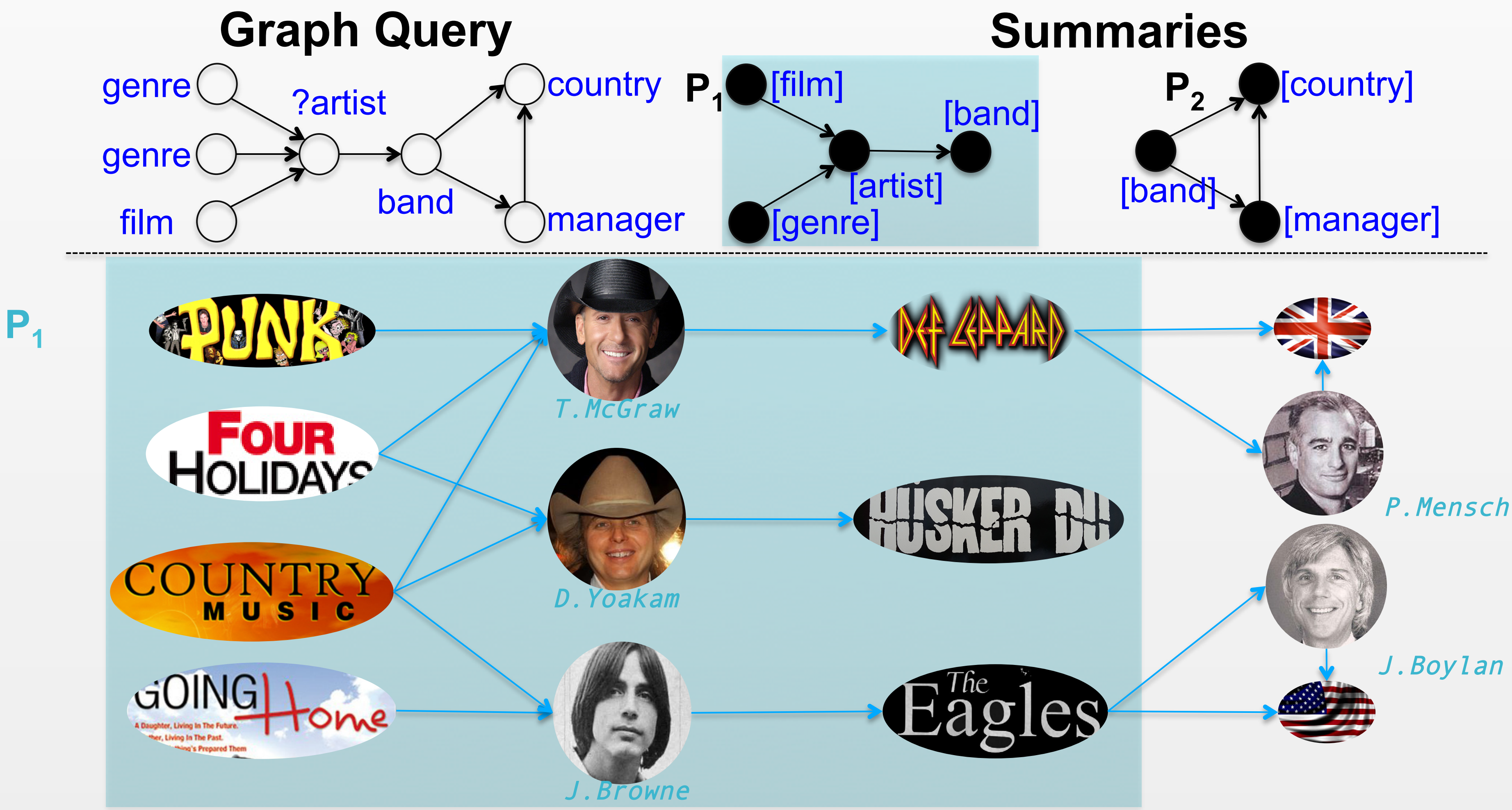


Answer



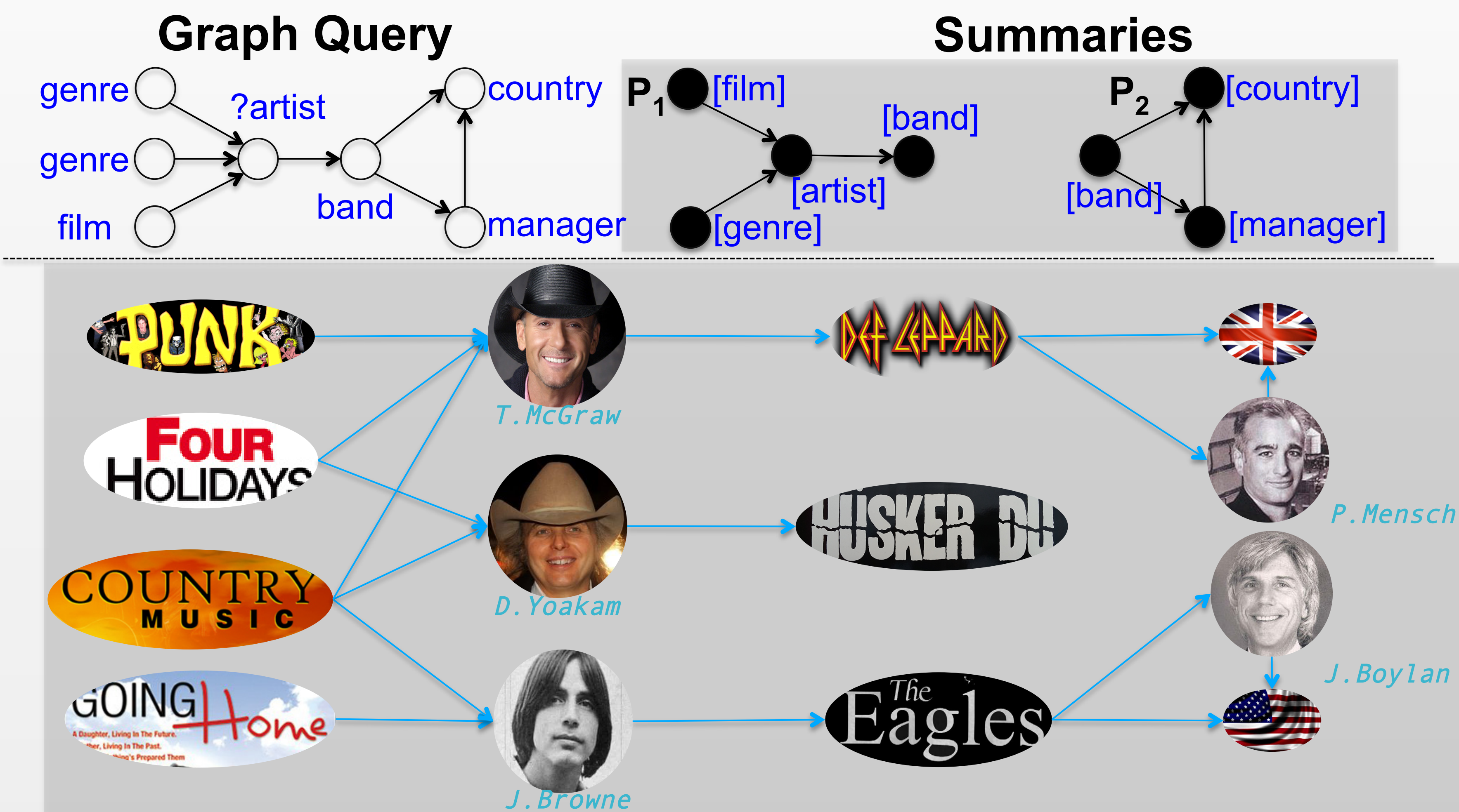
Use summarization to facilitate query evaluation

- Graph summarization: describe the data graph with a small amount of information



Use summarization to facilitate query evaluation

- Graph summarization: describe the data graph with a small amount of information
- Summary based query evaluation: Query Q can be answered by accessing only the entities summarized by “relevant” patterns

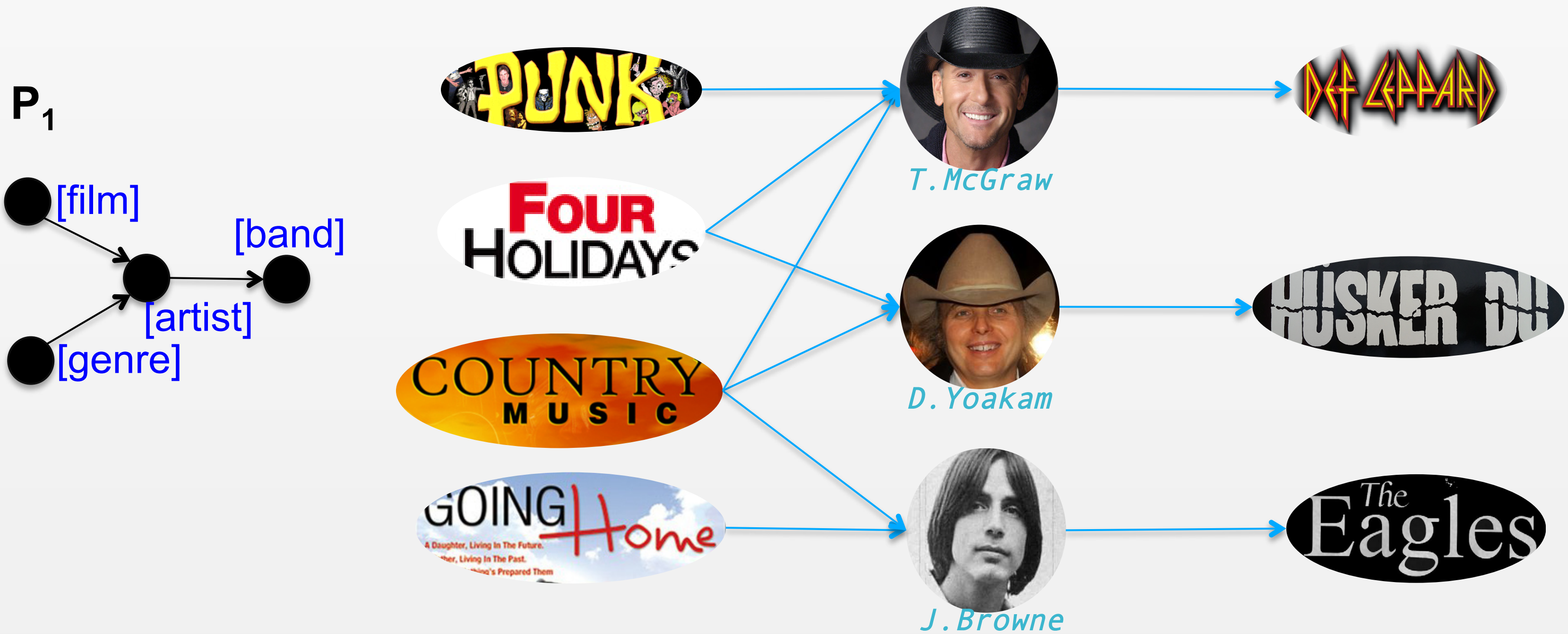


Use summarization to facilitate query evaluation

- How to construct summaries in a schema-less KG?
  - Traditional isomorphism based frequent pattern mining may not work
  - D-summaries: summarize similar entities up to a bounded hop d
- How to leverage the summaries to support KG search?
  - How to measure the quality of KG summarization
  - Diversified graph summarization problem and approximate algorithms

D-summaries

- Subgraph isomorphism VS **d-hop dual simulation**
  - Relax 1-1 to many-many relation
  - Bounded match with hop d
  - Dual-simulation: parent-children matching
  - Quadratic time solvable



Diversified knowledge graph summarization

- Problem definition:
  - Given: knowledge graph G, integers k and d
  - Output: a set of k d-summaries that maximizes the bi-criteria quality function.

- Objective function

$$F(S_G) = (1-\alpha)\sum_{P_i \in S_G} \boxed{I(P_i)} + \frac{\alpha}{card(S_G)-1}\sum_{P_i \neq P_j \in S_G} \boxed{diff(P_i, P_j)}$$

Informativeness

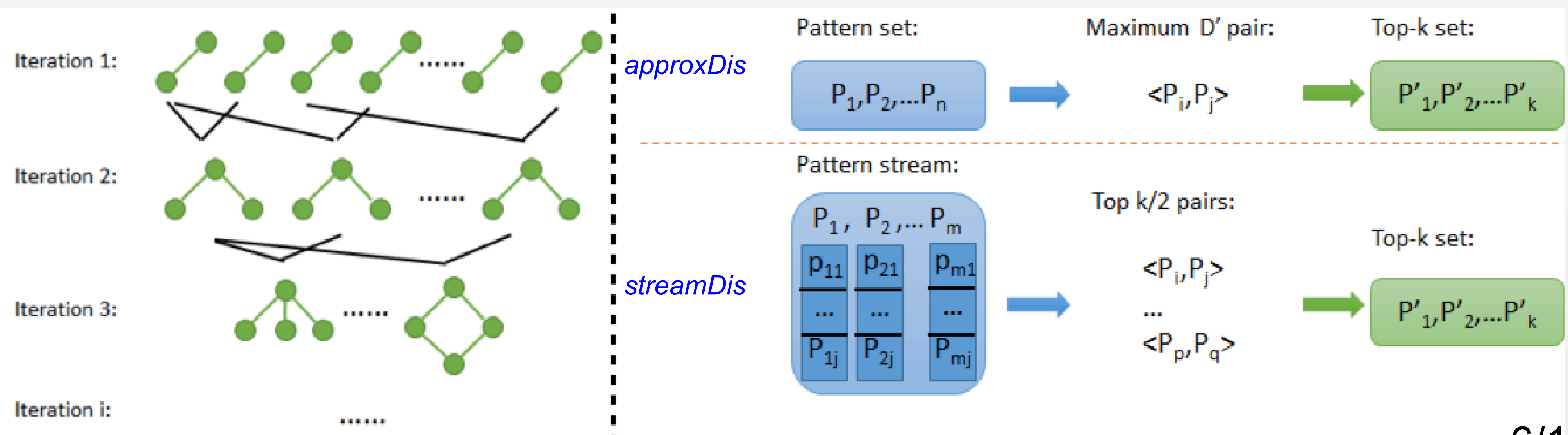
Difference

Diversified knowledge graph summarization

- 2-approximation algorithm *approxDis*:
  - Mining frequent patterns based on d-similarity
  - Calculate pair-wise score and select top score pairs
  - Have to wait until all frequent patterns are generated
- Anytime algorithm *streamDis*:
  - Maintain a cache during pattern mining

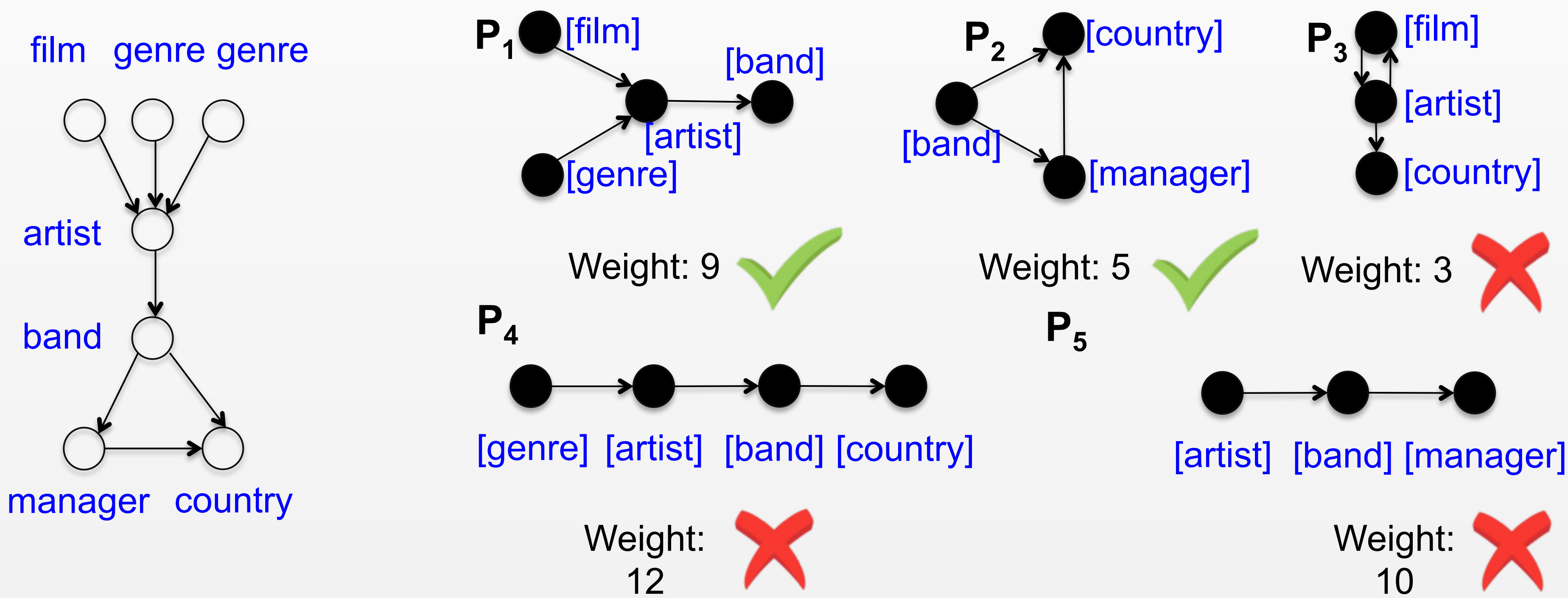
$$O(N_t * b_p(b_p + |V|)(b_p + |E|) + \frac{k}{2} N_t^2)$$

  - Can be interrupted at any time
  - Maintain 2-approximation (better than pure heuristic)



“Summaries + Δ“ scheme for query evaluation

- Pattern selection
  - Iteratively selects a view with minimum weight



- Query answering *evalSum*: “Summaries + Δ”

Experimental study

- Datasets: real-world and synthetic knowledge graphs
  - Yago: 1.54M nodes, 2.37M edges, 324k labels
  - DBPedia: 4.86M nodes, 15M edges, 676 labels
  - Freebase: 40M nodes, 63M edges, 9630 labels
  - BSBM: up to 60M nodes, 152M edges and 3080 labels

- Algorithms:
  - Summarization: *approxDis*, *streamDis* and its counterpart *heuDis*, *GRAMI*[*]
  - Query evaluation: *evalSum*, *evalRnd* (performs random selection), *evalGRAMI* (employs FPGs mined by GRAMI), *evalNo* (directly employ subgraph isomorphism algorithm)

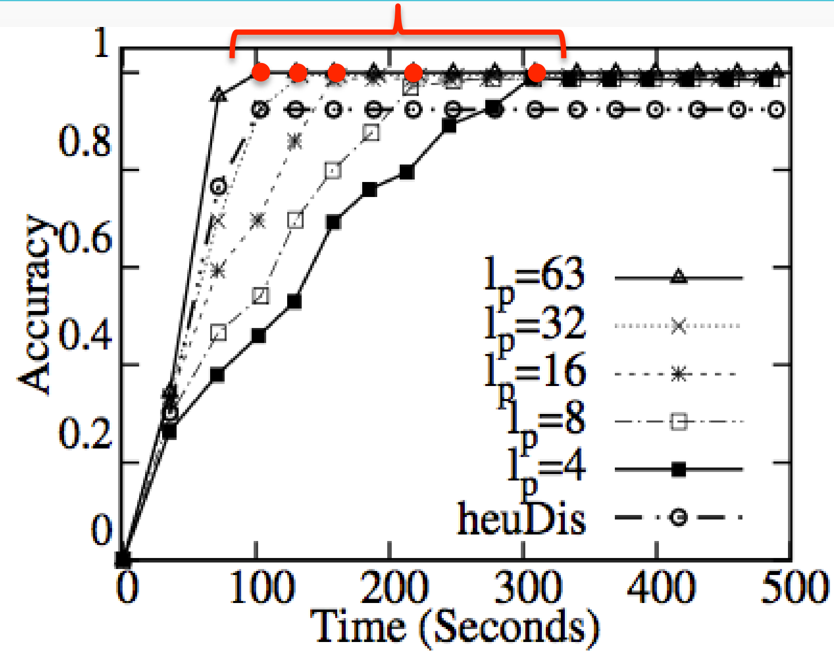
[*] M. Elseidy, E. Abdelhamid, S. Skiadopoulos, and P. Kalnis.GRAMI: frequent subgraph and pattern mining in a single large graph. *PVLDB*, 7(7):517–528, 2014.

Source code: https://github.com/songqi1990/KnowGraphSum

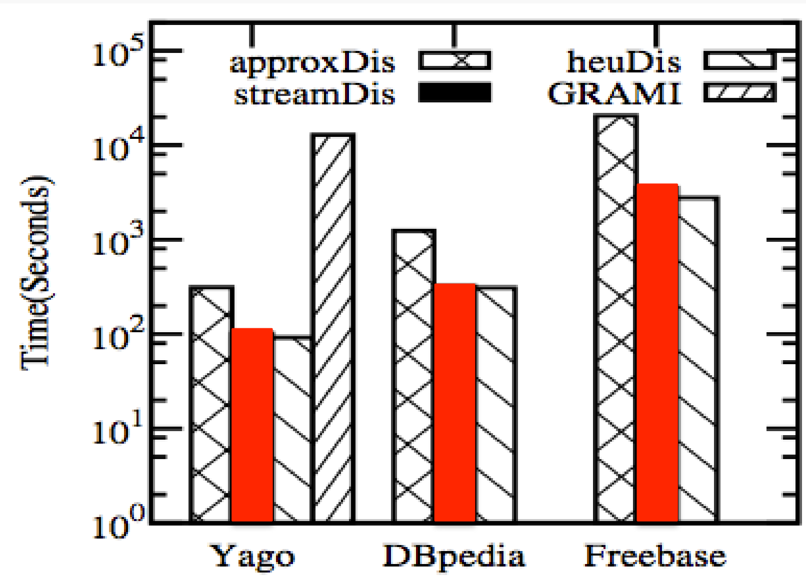
Effectiveness of summary discovery

- Faster convergence with larger cache size
- Cache size in general small to guarantee fast convergence.

- Orders of magnitude faster than GRAMI

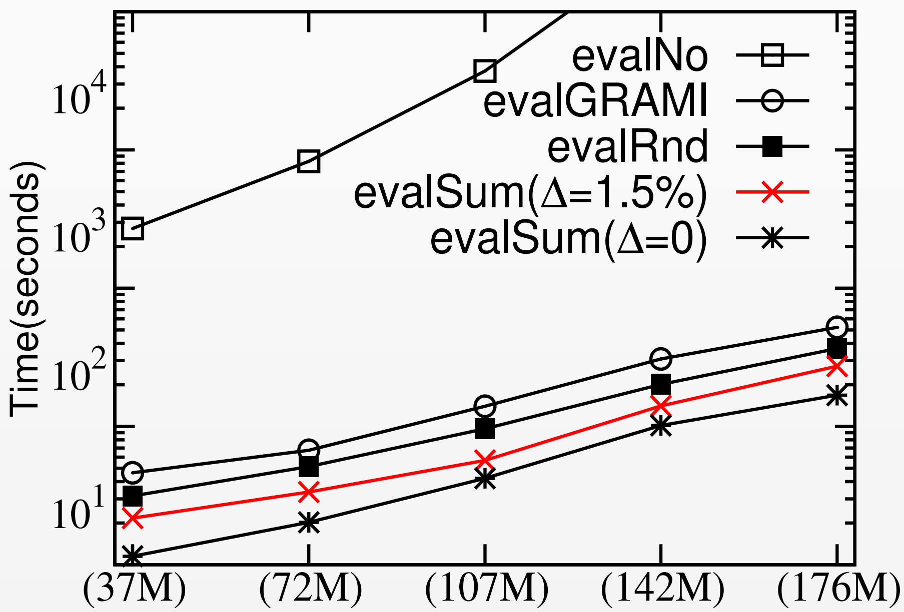


(a) streamDis: quality vs. time

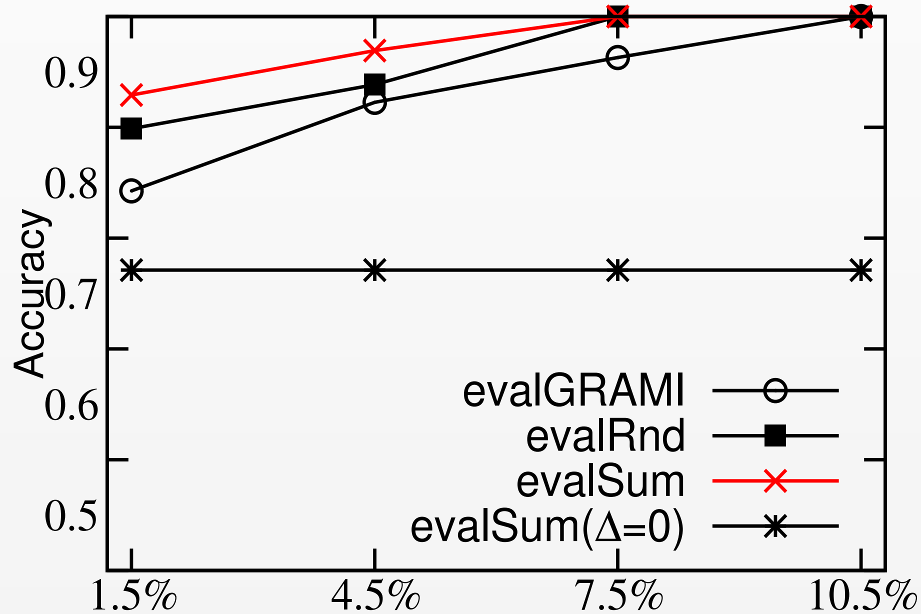


(b) Real-world datasets

Effectiveness of *evalSum*



Time *w.r.t* |G|



Accuracy *w.r.t* Δ

40 times faster than *evalNo*

Little additional cost (Δ ≤ 5% of graph size) to find exact answers.

Conclusion and future work

- Mining Summaries for Knowledge Graph Search:
  - We proposed a class of d-summaries
  - We developed feasible summary mining algorithms and efficient query evaluation algorithm
  - We show that our algorithms efficiently generate concise summaries that significantly reduces query evaluation cost
- Future work
  - Distributed query evaluation over different information source
  - Query suggestion, data integration, knowledge fusion using views

Thanks!