LSDIS
Large Scale Distributed Information Systems

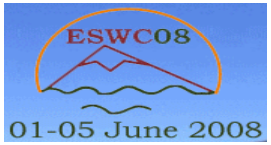
kno.e.sis

University of Georgia
Computer Science Department

Graph Summaries for Subgraph Frequency Estimation

[1]Angela Maduko, [2]Kemafor Anyanwu, [3]Amit Sheth, [4]Paul Schliekelman

[1]LSDIS Lab, University of Georgia
[2]Computer Science Department, North Carolina State University
[3]Kno.e.sis Center, Wright State University
[4]Statistics Department, University of Georgia

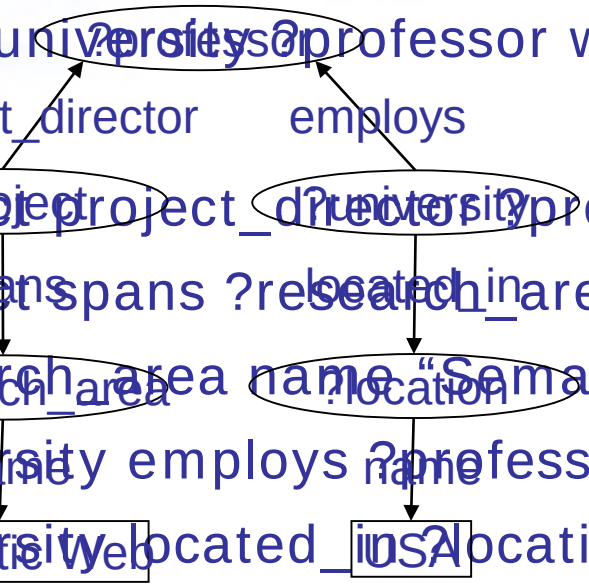


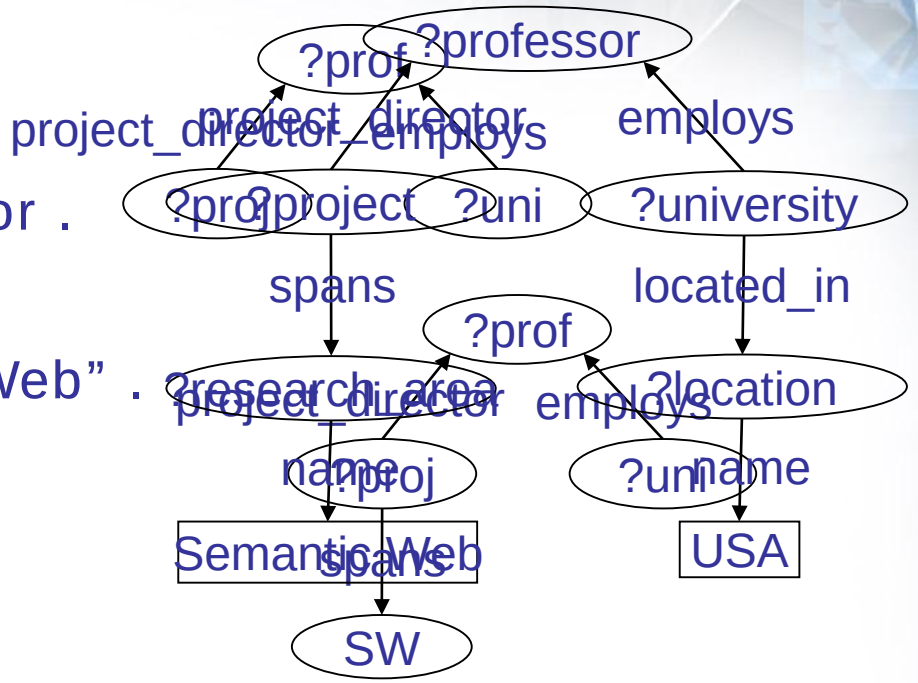
The European Semantic Web Conference,
Tenerife, Spain. June 1 – 5, 2008.

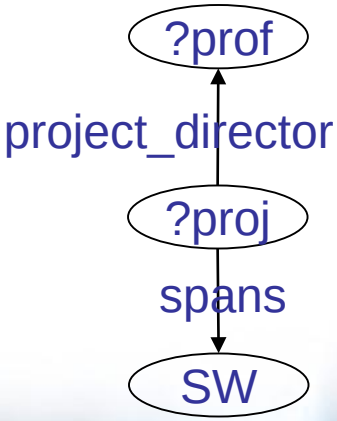
This work is funded by NSF-ITR-IDM Award #0325464 and #071444

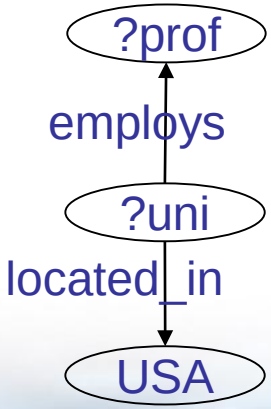
Optimizing Graph Pattern Queries

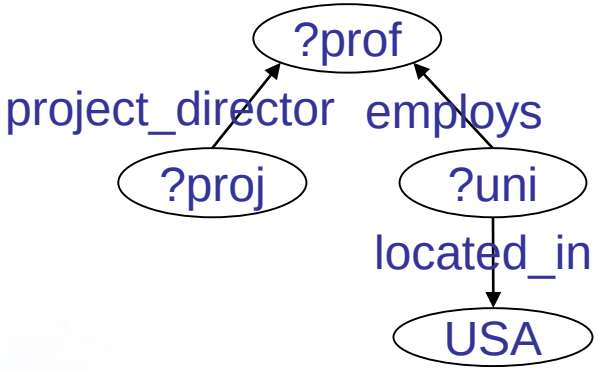
```
Select ?university ?professor where
{
  ?project project_director ?professor .
  ?project spans ?research_area .
  ?research_area name “Semantic Web” .
  ?university employs ?professor .
  ?university located_in ?location .
  ?location name “USA” .
}
```







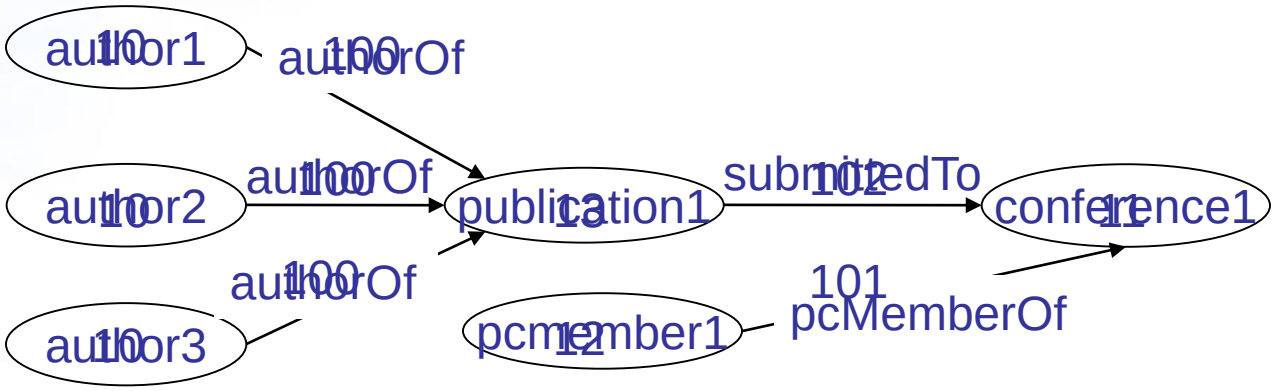




Proposition and Challenges

- Go beyond maintaining statistics of triple patterns to maintaining those of more complex graph patterns
- Number of all graph patterns may be exponential
- Consider graph patterns of up to a fixed length (maxL)
- Representation structure for patterns such that patterns can be
  - Pruned to fit a specified budget, while preserving accuracy of estimates as much as possible
  - Tuned such that certain patterns are favored over others

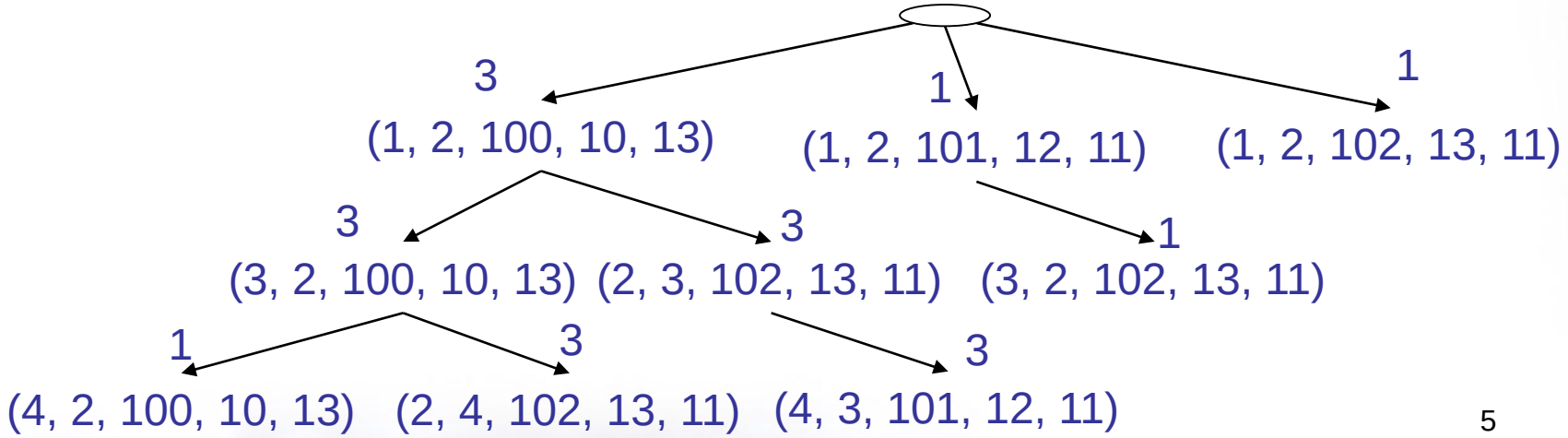
Canonical Label – DFS Coding(Yan et al)



(1, 2, 100, 10, 13)	3
(1, 2, 101, 12, 11)	1
(1, 2, 102, 13, 11)	1
(1, 2, 100, 10, 13) (3, 2, 100, 10, 13)	3
(1, 2, 100, 10, 13) (2, 3, 102, 13, 11)	3
(1, 2, 101, 12, 11) (3, 2, 102, 13, 11)	1
(1, 2, 100, 10, 13) (3, 2, 100, 10, 13) (4, 2, 100, 10, 13)	1
(1, 2, 100, 10, 13) (3, 2, 100, 10, 13) (2, 4, 102, 13, 11)	3
(1, 2, 100, 10, 13) (2, 3, 102, 13, 11) (4, 3, 101, 12, 11)	3

Pattern Tree (P-Tree)

(1, 2, 100, 10, 13)	3
(1, 2, 101, 12, 11)	1
(1, 2, 102, 13, 11)	1
(1, 2, 100, 10, 13) (3, 2, 100, 10, 13)	3
(1, 2, 100, 10, 13) (2, 3, 102, 13, 11)	3
(1, 2, 101, 12, 11) (3, 2, 102, 13, 11)	1
(1, 2, 100, 10, 13) (3, 2, 100, 10, 13) (4, 2, 100, 10, 13)	1
(1, 2, 100, 10, 13) (3, 2, 100, 10, 13) (2, 4, 102, 13, 11)	3
(1, 2, 100, 10, 13) (2, 3, 102, 13, 11) (4, 3, 101, 12, 11)	3



Estimation from the P-Tree

- Patterns of length at most maxL
  - Traverse the tree, matching labels on query pattern to node labels
- For a pattern P′ of length k, k > maxL
  - Partition into non-disjoint patterns of length maxL, P'_1, P'_1, …, $P'_{k-maxL+1}$
  - P'_i intersects P'_{i+1} in all but one edge
  - Combine frequency of partitions under the conditional independence assumption

Pruning the P-Tree

- Estimation value of patterns(nodes) in the P-Tree
  - Number of children that can be estimated within some error bound
  - Entropy of the frequency distribution of its children
- Prune children of nodes with larger estimation values

Tuning the P-Tree

- Observed value
  - Assume importance threshold is given, we measure as a function of the number of patterns that are less important
- Final value then combines estimation and observed values
- Combination is such that the final value of any important node always exceeds that of an unimportant one

Maximal Dependence Tree (MD-Tree)

- Maximal Dependence Tree (MD-Tree)
  - Tree representation of a statistical model of patterns cardinalities
- Base MD-Tree – Independence assumption
  - Edge patterns occur independently on any position in patterns of a given length
- Refined MD-Tree – Single point of dependence assumption
  - For patterns of a given length, there exists a position that exerts the most influence on the occurrence of edge patterns on others
- Complete MD-Tree – Completely Refined MD-Tree

Estimation from the MD-Tree

- Patterns of length at most maxL, we combine statistics from the MD-Tree
  - Under the independence assumption
  - Under the single point of dependence assumption
- Patterns of length k, k > maxL
  - Partition into non-disjoint patterns of length maxL as before
  - Estimate using conditional independence

Pruning the MD-Tree

- Explore the space between the base and Complete MD-Tree
- Pick the MD-Tree that
  - best fits the budget
  - favors subtrees with wider deviations from the estimation assumptions

Tuning the MD-Tree

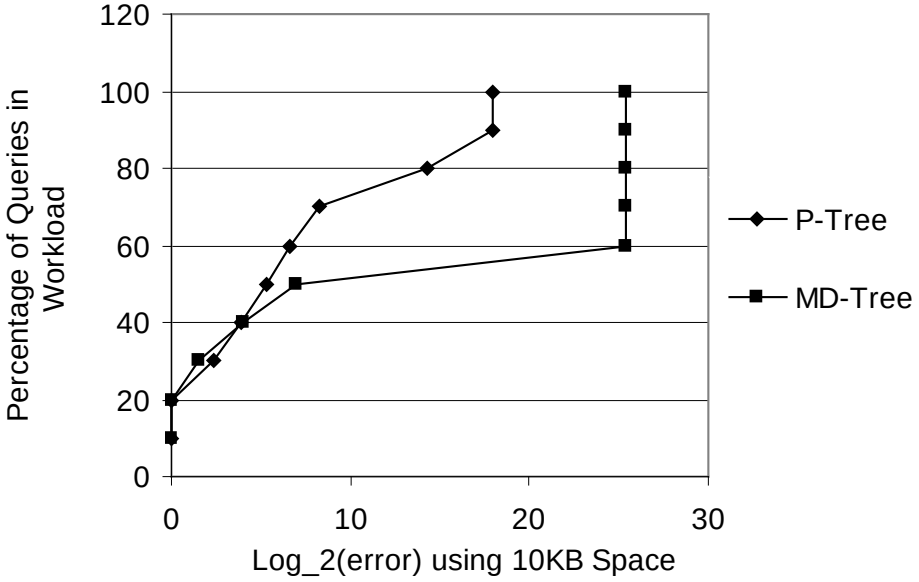
- Pick the MD-Tree that
  - best fits the budget
  - favors subtrees with wider deviations from the estimation assumption
  - favors subtrees created from a larger number of important patterns

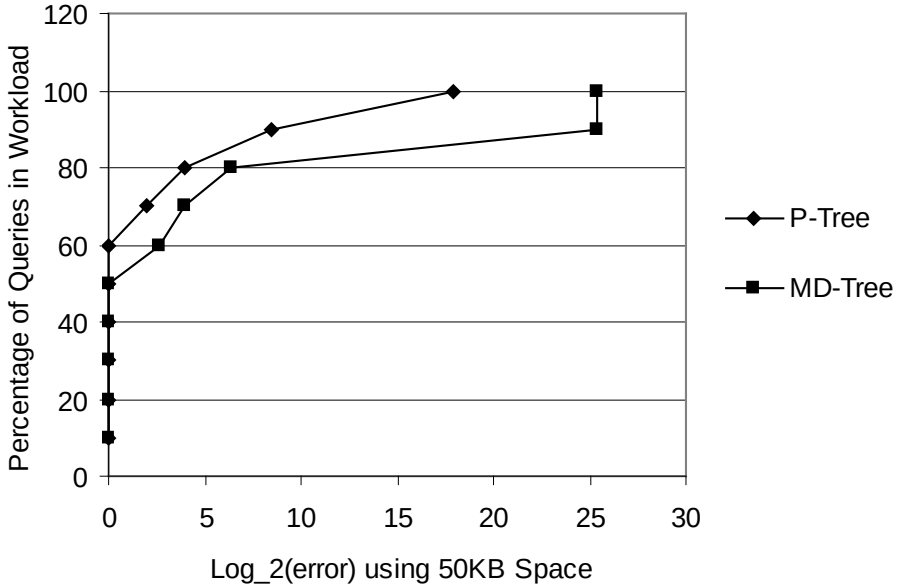
Evaluation

- SwetoDBLP – from LSDIS, part of the DBLP (RDF) enhanced to include more relationships amongst entities. Follows a Zipfian distribution
- TOntogen – from LSDIS, random node-degree distribution

	SwetoDBLP	TOntoGen
Number of nodes	1037856	200001
Number of edges	848839	749825
Number of unique edge labels	87	9
Size of patterns of length ≤ 3 (bytes)	6036340	10890
Size of unpruned P-Tree (bytes)	245000 (95% reduction)	4916(55% reduction)
Size of unpruned MD-Tree (bytes)	259200 (95% reduction)	7554(31% reduction)

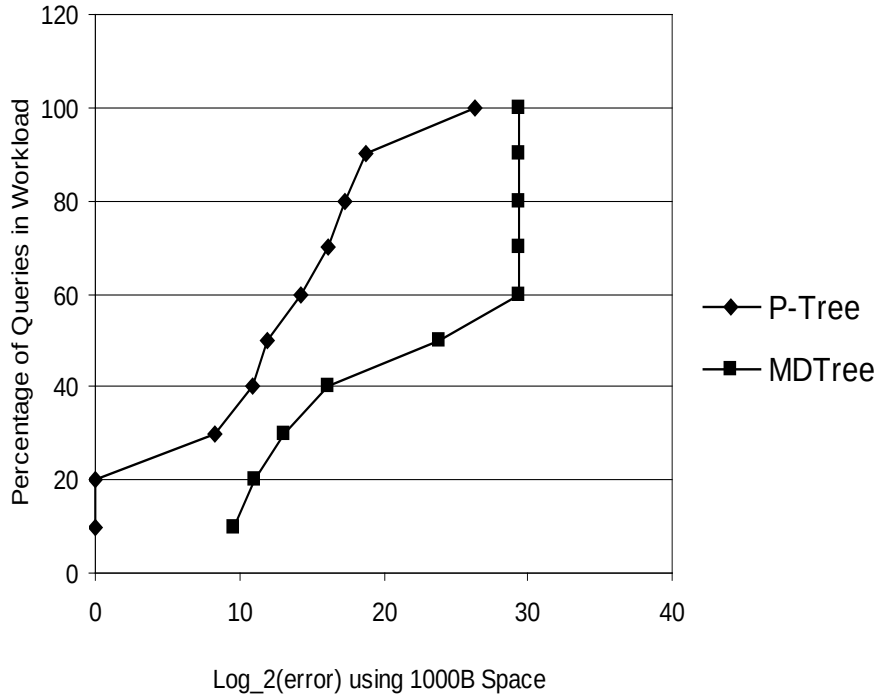
SwetoDBLP

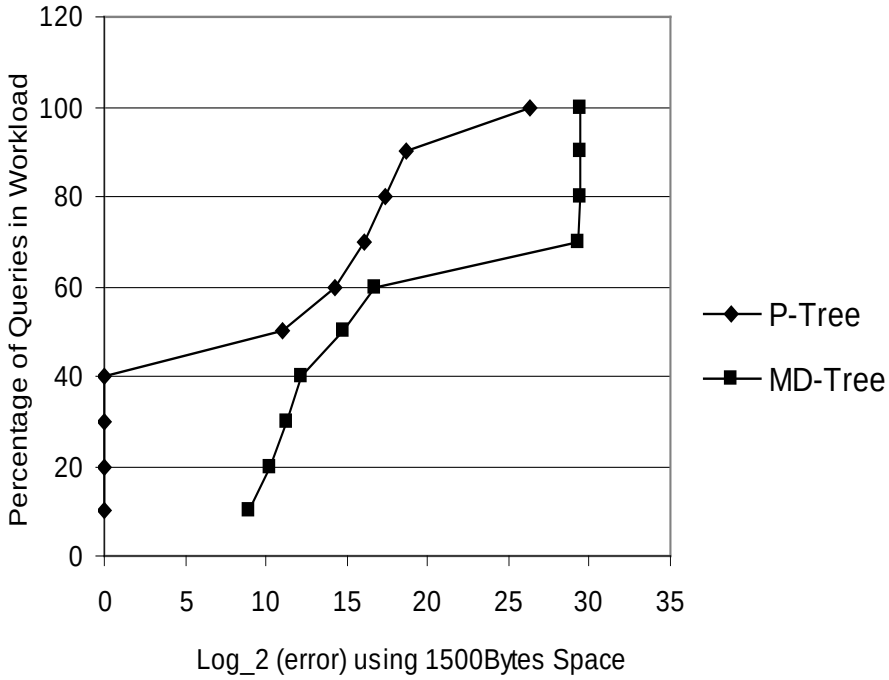




- 10KB summary – 4% of original P-Tree/MD-Tree with 20% of queries in workload estimated with 0 error and 40% ≤ 32.
- 50KB summary – 20% of original P-Tree/MD-Tree with 50% of queries in workload estimated with 0 error and 70% ≤ 32

TOntogen

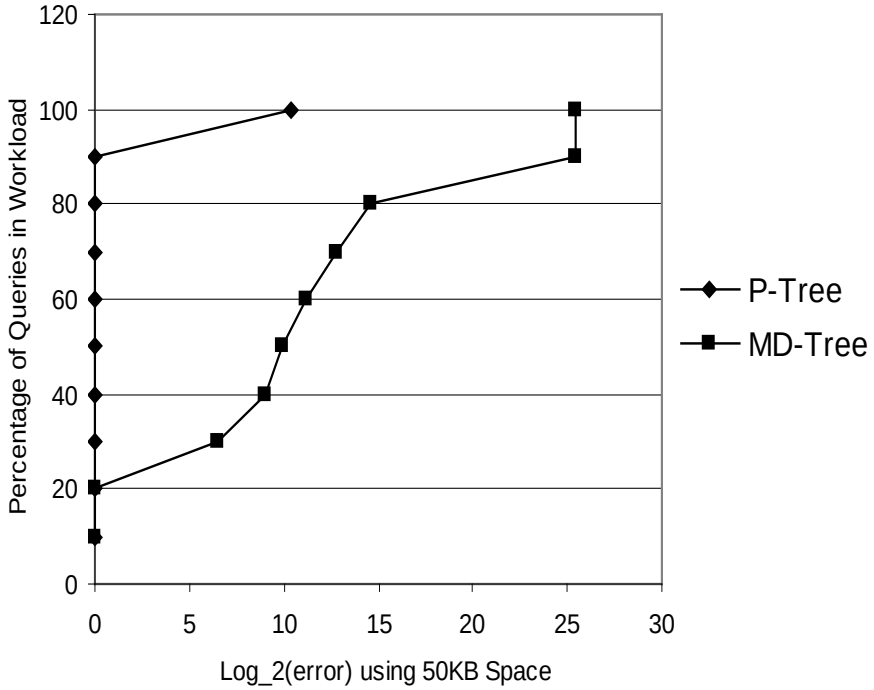




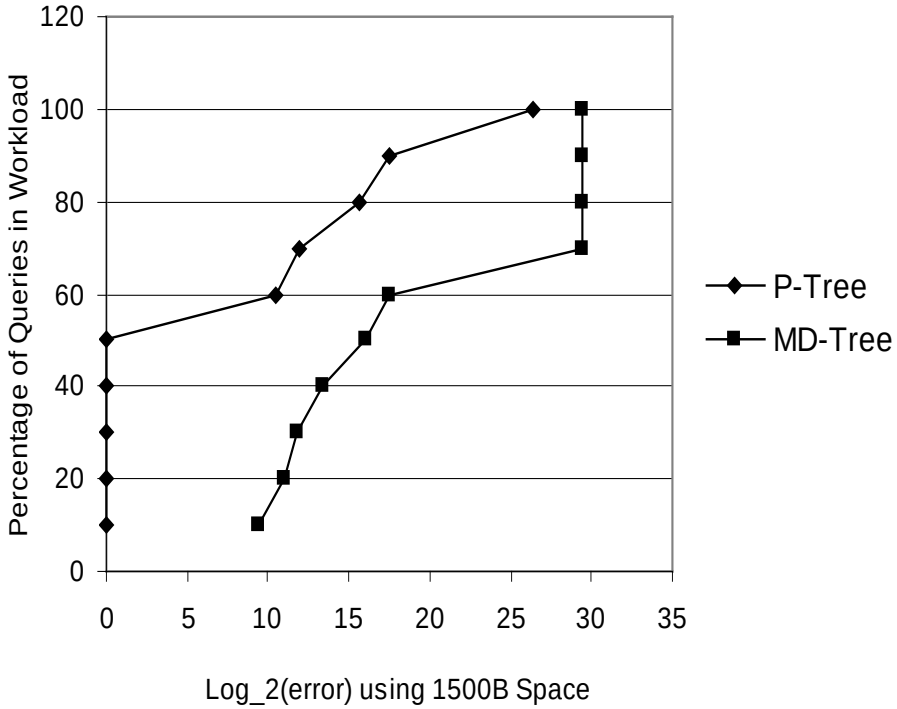
- 1000 bytes summary – 20% of original P-Tree/MD-Tree
  - P-Tree: 20% of queries in workload are estimated with 0 error and 25% ≤ 32
- 1500 bytes summary – 30% of original P-Tree/MD-Tree
  - p-Tree: 40% of queries in workload are estimated with 0 error and 45% ≤ 32

Summaries tuned for Frequent Patterns

SwetoDBLP



TOntoGen



- P-Tree is more amenable to tuning than MD-Tree, with 90% of all queries in workload estimated with 0 error for SwetoDBLP dataset and 50% estimated with 0 error for the Military dataset

Conclusion

- Frequency of graph patterns are useful for query optimization
- Two representation structures, P-Tree and MD-Tree
  - With pruning to fit a specified budget
  - Tuning to favor certain patterns
- Although P-Tree exhibits better performance in terms of accuracy of estimates, in more recent experiments, MD-Tree performed equally well for optimizing graph pattern queries in almost all tested cases
- Expensive discovery of patterns is done offline as a pre-processing step

Future Work

- A comprehensive evaluation of the effectiveness of our summaries for query processing
- More compact data structure to reduce the space overhead of the MD-Tree
- Estimating patterns in graphs whose nodes/edges may be arranged in subsumption hierarchies.
- Extend to gracefully accommodate updates to the data graph into the summaries.