SPROUT\(^2\): A Squared Query Engine for Uncertain Web Data

Robert Fink\(^1\), Andrew Hogue\(^2\), Dan Olteanu\(^1\), Swaroop Rath\(^1\)

\(^1\)Department of Computer Science, University of Oxford, Oxford, OX1 3QD, UK \(^2\)Google Inc., New York, NY, USA

Query Answering in SPROUT

Query evaluation in two steps (logically distinct but possibly intertwined):

1. Compute the tuples in the query result together with their lineage
   The lineage of a result tuple \( t \) is a propositional formula over the tuples in the input database and says which input tuples must be present in order for the query to return \( t \).

2. Compute the probabilities of result tuples by incremental lineage compilation:
   - Independent or \((e \land v\) or syntactically independent) \( P((e \lor v)) = 1 - (1 - P(e))(1 - P(v)) \)
   - Independent \((e \land v)\) or syntactically independent \( P((e \lor v)) = P(e)P(v) + P(e)P(v)P(e)P(v) \)
   - Shannon expansion \((v) \text{ is a variable in } e\) \( P((e \lor v)) = P(e)P(v) + P(e)P(v)P(e)P(v) \)

Example: Relational division query "Which supplier stocks all products?"

Novel Techniques for Exact and Approximate Probability Computation

- Incremental decomposition of lineage into d-trees using the above three rules
- After each decomposition step, compute rough lower and upper bounds on the probabilities of the residual formulas at the leaves of the decomposition tree
  - Approach 1: Lower bound is the largest probability of a clause in \( e \); Upper bound is the sum of probabilities of all clauses in \( e \).
  - Approach 2: Compute read-once formulas, whose probabilities represent lower and upper bounds.
  - Using the bounds at the leaves, compute lower and upper bounds for the whole lineage
  - Stop when the desired precision is reached or the time budget is exhausted
  - Underlying idea: Leaves deeper in the d-tree contribute little to the overall probability mass, hence a good approximation can be found quickly

Complete decomposition

- Corresponds to exact probability computation
- Can be done in polynomial time for tractable query & data instances
  - propositional languages
  - relational algebra queries without repeating symbols and with read-once lineage
  - a class of conjunctive queries with inequality \((\cdot \neq \cdot)\) joins

Partial decomposition

- Corresponds to approximate probability computation with error guarantees
- Applicable for hard query and data instances

Example: Efficient computation of bounds that are read-once formulas

- Left: original formula; middle: lower bound; right: upper bound
- Lower/Upper bounds obtained by setting the marked literals to false/true

New Book on Probabilistic Databases