Confidence-Aware Join Algorithms

Parag Agrawal, Jennifer Widom

Uncertain Databases

Name	SSN
Bob	123

0.8 ⋈

SSN	Employer
123	Stanford

0.6

Name	Employer
Bob	Stanford

0.48

- Tuples have confidences
- Result confidence computation
  - Combine input confidences
  - Multiplication if independent

High confidence result tuples are more important

Threshold | Top-*k* | Sorted

on result tuple confidence

Problem

Efficient algorithms for join queries

Minimize IO cost

Traditional Solution

- Leave it to the optimizer
  - Treat confidence values as another column
  - Result confidence computed in query
  - Threshold in WHERE clause

```
SELECT R.A, S.C, R.conf * S.conf as conf
FROM R, S
WHERE R.B = S.B AND R.conf * S.conf > threshold
```

  - Sorted using ORDER-BY
  - Top-*k* using ORDER-BY and LIMIT

We can do better!

Exploit Monotonicity

of combining function like NRA Algorithm, Rank-Aware Joins

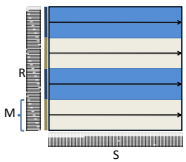
Assume sorted access by confidence

Limited Memory

in contrast to previous work

Join Visualization

- Nested Block Join
- Memory size : M
- Repeat
  - *Load* part of R into memory
  - *Scan* S and evaluate
- *Explore* cross-product

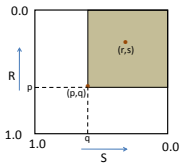


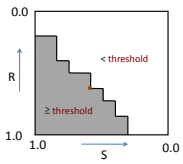
- IO Cost
- Number of tuples read
  - Load
  - Scan

Cost : |R| + (|R| ÷ M)•M

Observation

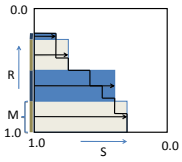
Monotonicity : C(p, q) ≥ C(r, s) if p ≥ r and q ≥ s



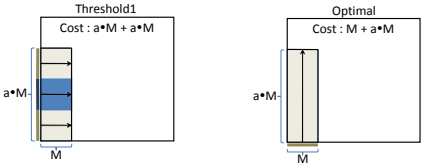


Threshold1

- Explore area under the stair
- IO Cost : less than 2 times optimal
  - Assuming no indexes

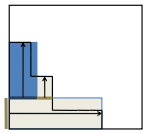


Bad Case



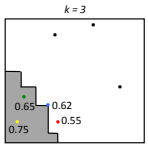
Threshold2

- Picks relation to load in each step
  - Longer scan
- Optimality ratio : 3/2
  - Close to optimal if large area under the stair



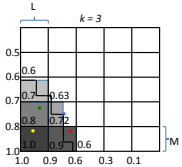
Top-*k*

- Threshold = confidence of k^{th} tuple
- Explore area under the stair
- Threshold value is not known



Explore in approximate order of result confidence

- Scan length L
- Explore blocks (M x L)
  - In order of max confidence
- Top-*k* tuples maintained during algorithm
- Treat confidence of current k^{th} tuple as threshold
- Exit when stair explored
- Optimality ratio : 3
  - 2 if large area under the stair



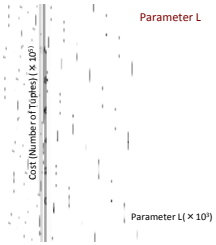
Top-*k* tuples in memory

	Intermediate	Final
1	0.75	0.75
2	0.65	0.65
3	0.55	0.62

Experiments

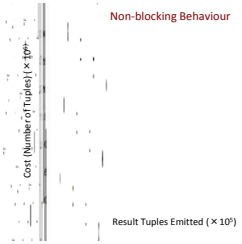
- Synthetic data-sets
  - 1M – 10M tuples in each relation
  - Various confidence distributions
- Algorithms perform well
  - Not affected by confidence distributions
  - Results in paper

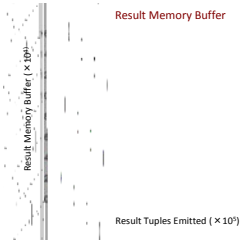
Top-*k*



Sorted

- Explores M x L blocks like top-*k*
- Result memory buffer (priority queue)
- Emit a result tuple after exploring corresponding stair
- Non-blocking operator





Future Work

- Use as operator in query plans
  - Parameter L (block size)
  - Cost estimation
  - Memory allocation
- Non-independence
  - Interval approximations
  - Monte-Carlo simulations