Supporting Ranking Queries in RDBMS

The Context: AIMing to the top

The ultimate goal of the AIM project --
Supporting ranking in data retrieval

The Problem: Supporting Ranking in RDBMS

- Support ranking as a first-class query type
- Integrate ranking with Boolean query constructs

Our Challenges:

- RDBMS treats ranking as second class:
  - Monolithic ranking component
  - Processed after Boolean component

Naive materialize-then-sort scheme:
- Only 5 top results are requested, whole results are scored and ordered;
- Scan and join unnecessary tuples;
- Ranking predicates can be expensive: cheap(h.price); online source related(m.collection, “dinosaur”); IR close(h.add, “O’Hare airport”); querying geographical data

Our Insight: Splitting and Interleaving

- Support ranking as a first-class query type in RDBMS: splitting ranking predicates.
- Integrate ranking with traditional Boolean query constructs: interleaving ranking predicates with other operations.

Our Principle: Ranking Principle

Tuples should be processed in the order of their upper-bound scores with respect to the evaluated predicates.

Our Solution: RankSQL [SIGMOD’05]

Foundation: Rank-Relational Algebra
Data model: ranking
Operators: new and augmented
Algebraic laws
Impact: Query Engine
- Executor: Physical operators implementation
  - Incremental plans
- Optimizer: Plan enumeration
  - Cost estimation

Task 1: Algebraic Foundation

Rank-Relational Algebra
- Data Model: Rank-Relation Ss
- S (membership); P (order) : evaluated predicates
- Operators:
  - Order dimension: μ (rank) operator
  - Membership dimension: augmented rank-aware operators σ ∩ n → σ
  - Satisfying interleaving requirement.
- Algebraic laws: e.g., σ(μ(Sp)) = μ(σ(Sp))

Task 2: Rank-Aware Query Optimizer

Two-Dimensional Plan Enumeration:
Sampling-Based Cardinality Estimation:

Contributions: Summary

For supporting ranking in RDBMS, we developed:
- Key insight: Splitting and interleaving
- Fundamental Principle: Ranking principle
- Algebraic foundation: Rank-Relation
- New and augmented operators
- Algebraic laws
- Optimization Framework: Two-dimensional plan enumeration
  - Sampling-based cardinality estimation