Optimization of Preference Queries

Jan Chomicki
University at Buffalo
http://www.cse.buffalo.edu/~chomicki

Plan of the talk

1. Relational query languages: relational algebra, SQL.
2. Relational query evaluation: data structures, algorithms.
3. Relational query optimization: algebraic, semantic, cost-based.
4. Preference queries: winnow.
5. Preference query evaluation and optimization.
6. Extensions and future research.

Relational algebra

A set of operators on relations that can be nested to form expressions:

- set-theoretic: union, set difference
- *almost* set-theoretic: Cartesian product $R \times S$

$$r \times s = \{t : t[R] \in r \wedge t[S] \in s\}$$

- selection $\sigma_\alpha(R)$

$$\sigma_\alpha(r) = \{t : t \in r \wedge \alpha(t)\}$$

- projection $\pi_X(R)$

$$\pi_X(r) = \{t[X] : t \in r\}$$

- join

$$R \bowtie S = \sigma_{A=B}(R(\ldots, A, \ldots) \times S(\ldots, B, \ldots)).$$

Selection $\sigma_{Price<15}(Book)$:

Book	*Title*	*Vendor*	*Price*
t_1	The Flanders Panel	`amazon.com`	$14.75
t_2	The Flanders Panel	`fatbrain.com`	$13.50
t_3	The Flanders Panel	`bn.com`	$18.80
t_4	Green Guide: Greece	`bn.com`	$17.30

SQL

A hybrid language:

- relational algebra
- relational calculus (= first-order logic)

Basic form:

SELECT $A_1, \ldots, A_n$
FROM $R_1, \ldots, R_k$
WHERE C

This corresponds to the following relational algebra expression:

$$\pi_{A_1,\ldots,A_n}(\sigma_C(R_1 \times \cdots \times R_k))$$

Other features of SQL

Nested subqueries which may include quantification.

Aggregation: `MAX`, `MIN`, `SUM`,...

Grouping.

```
SELECT Title,
  MIN(Price) AS MPrice
FROM Book
GROUP BY Title
```

$\Rightarrow$

Title	*MPrice*
The Flanders Panel	$13.50
Green Guide: Greece	$17.30

Relational query evaluation

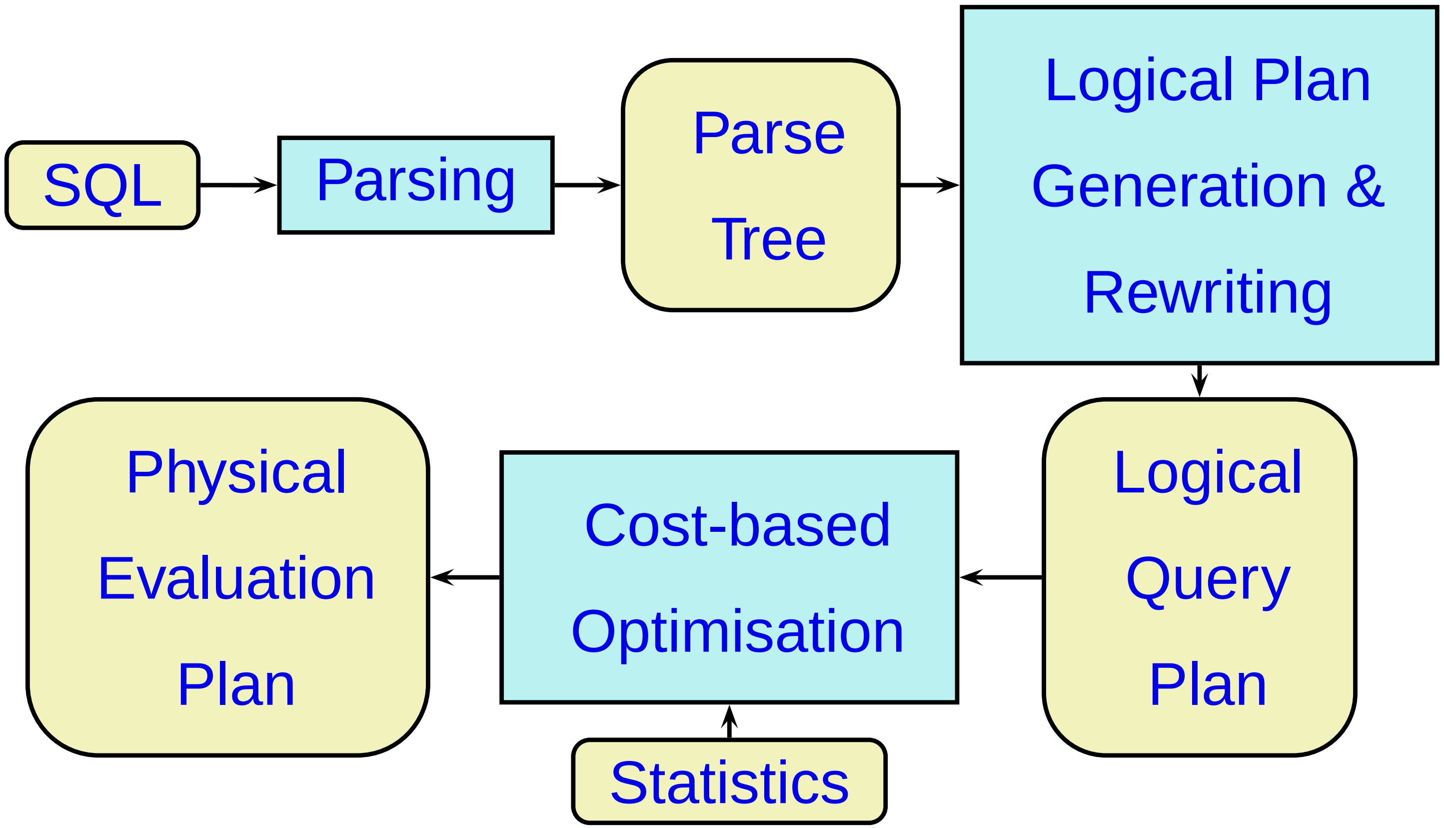
Indexing:

- fast access to individual rows using the values of one or more *index columns*
- speeding-up queries $\sigma_{A=c}$
- special data structures (B-trees) and algorithms (hashing)

Joins:

- various methods: nested loops, hash join, sort-merge join, index join

Query optimization



Algebraic query optimization

Using algebraic laws to rewrite logical query plans.

Pushing selection down:

$$\sigma_\alpha(E_1 \times E_2) = \sigma_\alpha(E_1) \times E_2$$

if F involves only the attributes of E_1.

$$\sigma_\alpha(E_1 \cup E_2) = \sigma_\alpha(E_1) \cup \sigma_\alpha(E_2).$$

$$\sigma_\alpha(E_1 - E_2) = \sigma_\alpha(E_1) - \sigma_\alpha(E_2).$$

Also pushing projections down, join reordering, ...

Semantic query optimization

Using integrity constraints to transform the query.

Various techniques:

- join elimination/introduction
- predicate elimination/introduction
- detecting empty results

Predicate elimination

If we know that faculty members are at least 30 years old, then

```
SELECT Name FROM Faculty WHERE AGE > 25
```

can be rewritten as

```
SELECT Name FROM Faculty
```

Cost-based query optimization

Estimating the cost of physical evaluation plans:

- number of I/O operations (or an approximation)
- based on stored statistics

Enumerating equivalent evaluation plans to find a plan of least cost.

Preference queries

Find the best answers to a query, instead of all the answers.

"Find the lowest price for this book on the Web...

... but also keep in mind my preference for `amazon.com`."

Preferences as first-order formulas

[Chomicki, EDBT'02].

Relation *Book(Title,Vendor,Price)*.

Preference:

$$(i, v, p) \succ_{C_1} (i', v', p') \equiv i = i' \wedge p < p'.$$

Indifference:

$$(i, v, p) \sim_{C_1} (i', v', p') \equiv i \neq i' \vee p = p'.$$

Relational algebra embedding

[Chomicki, EDBT'02; Kiessling, VLDB'02]:

New winnow operator returning the tuples in the given instance that are not dominated by any other tuple in the instance.

Book	*Title*	*Vendor*	*Price*
t_1	The Flanders Panel	`amazon.com`	$14.75
t_2	The Flanders Panel	`fatbrain.com`	$13.50
t_3	The Flanders Panel	`bn.com`	$18.80
t_4	Green Guide: Greece	`bn.com`	$17.30

Definitions

Preference relation: a binary relation $\succ$ between the tuples of a given relation.

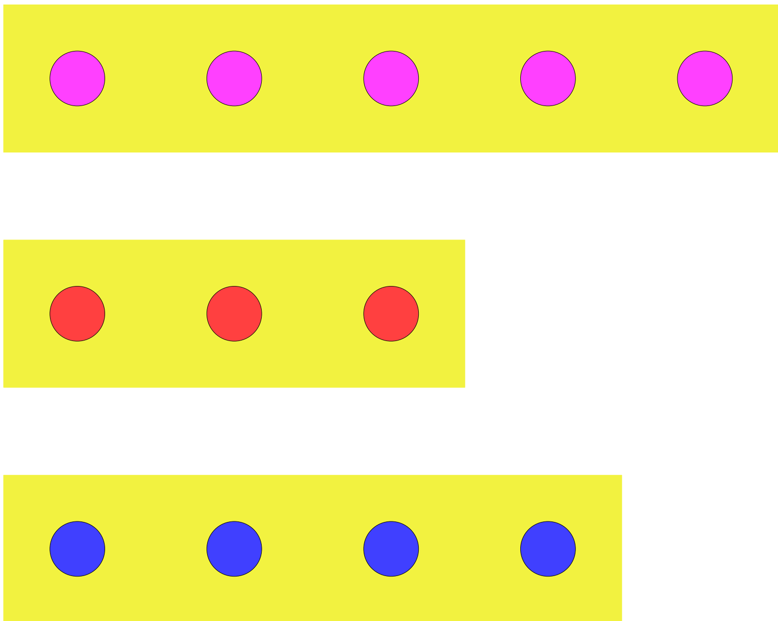
Preference formula: a first-order formula defining a preference relation.

Intrinsic preference formula: the definition uses only built-in predicates.

Typical properties of preference relations: irreflexivity, and transitivity ($\Rightarrow$ strict partial orders), can be effectively checked for intrinsic preference formulas with $=, \neq, <, >, \leq, \geq$.

Weak orders

Weak order: a strict partial order with transitive indifference.



Utility (scoring) functions

An approach grounded in utility theory:

1. construct a real-valued function u such that:

$$t_1 \succ t_2 \equiv u(t_1) > u(t_2)$$

2. return the answers that maximize u in the given instance.

Typically, top K answers are requested.

Properties of scoring functions

+ can be implemented using SQL3 user-defined functions [Agrawal et al, SIGMOD'00] [Hristidis et al., SIGMOD'01]
+ provide an ordering of all the answers
+ capture preference intensity
+ can be numerically aggregated

– need to be hand-crafted for every input
– hard to logically aggregate
– not expressive enough: only weak order pref. relations.

Non-existence of utility functions

	Title	*Vendor*	*Price*
t_1	The Flanders Panel	`amazon.com`	$14.75
t_2	The Flanders Panel	`fatbrain.com`	$13.50
t_3	The Flanders Panel	`bn.com`	$18.80
t_4	Green Guide: Greece	`bn.com`	$17.30

The set of constraints

$$\{u(t_2) > u(t_1) > u(t_3), u(t_4) = u(t_1), u(t_4) = u(t_2)\}$$

is unsatisfiable.

Winnow

Given a preference relation $\succ$ defined using a preference formula C:

$$\omega_C(r) = \{t \in r | \neg \exists t' \in r.\ t' \succ t\}.$$

Example (*"preference for* `amazon.com`*"*):

$$(i, v, p) \succ_2 (i', v', p') \quad \equiv \quad i = i' \wedge v = '\texttt{amazon.com}' \wedge v' \neq '\texttt{amazon.com}'$$

	Title	*Vendor*	*Price*
t_1	The Flanders Panel	`amazon.com`	$14.75
t_2	The Flanders Panel	`fatbrain.com`	$13.50
t_3	The Flanders Panel	`bn.com`	$18.80
t_4	Green Guide: Greece	`bn.com`	$17.30

Applications of winnow

Preference SQL [Kiessling et al, VLDB'02].

Skyline queries [Börzsönyi et al, ICDE'01]:

- find all the tuples that are not dominated by any other tuple in every dimension (Pareto set).

Linear optimization queries:

- find all the tuples that maximize $\sum_{i=1}^{n} a_i x_i$.

Winnow evaluation: BNL

[Börzsönyi et al, ICDE'01].

1. initialize the window W and the temporary file F to empty;
2. repeat the following until the input is empty:
3. for every tuple t in the input:
   - t is dominated by a tuple in $W \Rightarrow$ ignore t,
   - t dominates some tuples in $W \Rightarrow$ eliminate them and insert t into W,
   - t is incomparable with all tuples in $W \Rightarrow$ insert t into W (if there is room), otherwise add t to F;
4. output the tuples from W that were added there when F was empty,
5. make F the input, clear F.

Algebraic laws [Chomicki, TODS'03]

Commutativity with selection:

If the formula

$$\forall t_1, t_2.\ (\alpha(t_2) \wedge \gamma(t_1, t_2)) \Rightarrow \alpha(t_1)$$

is valid, then for every r

$$\sigma_\alpha(\omega_\gamma(r)) = \omega_\gamma(\sigma_\alpha(r)).$$

Example

The preference relation

$$(i, v, p) \succ_{C_1} (i', v', p') \equiv i = i' \wedge p < p'.$$

The selection $\sigma_{Price<20}$ commutes with ω_{C_1} because

$$\forall p, p', i, i'[(p' < 15 \wedge i = i' \wedge p < p') \Rightarrow p < 15]$$

is a valid formula.

The selection $\sigma_{Price>20}$ does not commute with ω_{C_1} because

$$\forall p, p', i, i'[(p' > 15 \wedge i = i' \wedge p < p') \Rightarrow p > 15]$$

is not a valid formula.

Distributivity over Cartesian product: For every r_1 and r_2

$$\omega_C(r_1 \times r_2) = \omega_C(r_1) \times r_2.$$

Commutativity of winnow: If $C_1(t_1, t_2) \Rightarrow C_2(t_1, t_2)$ and $\succ_{C_1}$ and $\succ_{C_2}$ are strict partial orders, then for all finite instances r:

$$\omega_{C_1}(\omega_{C_2}(r)) = \omega_{C_2}(\omega_{C_1}(r)) = \omega_{C_2}(r).$$

Also commutativity with projection.

Semantic query optimization

[Chomicki, CDB'04].

Using information about integrity constraints to:

- eliminate redundant occurrences of winnow.
- make more efficient computation of winnow possible.

Eliminating redundancy: Given a set of integrity constraints F, ω_C is redundant w.r.t. F iff F entails the formula

$$\forall t_1, t_2.\ R(t_1) \wedge R(t_2) \Rightarrow t_1 \sim_C t_2.$$

Integrity constraints

Constraint-generating dependencies (CGDs) [Baudinet et al, JCSS'99]:

$$\forall t_1 . \ldots . \forall t_n . \, [R(t_1) \wedge \cdots \wedge R(t_n) \wedge \gamma(t_1, \ldots t_n)] \Rightarrow \gamma'(t_1, \ldots t_n).$$

Entailment is decidable for CGDs by reduction to the validity of $\forall$-formulas in the constraint theory.

Example

Relation *Book(Title,Vendor,Price)*.

For the preference relation

$$(i, v, p) \succ_{C_1} (i', v', p') \equiv i = i' \wedge p < p'$$

$\omega_{C_1}(Book)$ is redundant w.r.t. FD *Title* $\rightarrow$ *Price*, because the formula

$$(i_1 \neq i_2 \vee p_1 = p_2) \wedge i_1 = i_2 \wedge p_1 < p_2$$

is unsatisfiable.

Cost-based optimization

Little known about result size estimates for preference queries. For skylines [Buchta, 1989; Godfrey, FOIKS'04]:

> The expected cardinality of a d-dimensional skyline of n tuples is equal to $H_{d-1,n}$, the $d-1$-order harmonic of n (under attribute independence).

Asymptotically: $H_{d,n} \in \Theta((\ln n)^d / d!)$.

Some values:

$$
\begin{aligned}
H_{2,10^6} &= 104 \\
H_{6,10^6} &= 14,087
\end{aligned}
$$

Extension: extrinsic preference

Extrinsic preference relation: depends not only on the components of the tuples being compared but also on other factors:

- the presence or absence of other tuples in the database
- computed or aggregate values.

Solution: winnow + SQL.

Preference for a lower total cost of a book (including shipping and handling).

Vendor	*SH*
`amazon.com`	$6.99
`fatbrain.com`	$3.99
`bn.com`	$5.99

Apply winnow to the following view:

```
CREATE VIEW TotalCost(Title, Vendor, Cost) AS
SELECT Book.Title, Book.Vendor, Book.Price + SHCosts.SH
FROM Book, SHCosts WHERE Book.Vendor = SHCosts.Vendor
```

Problem: computing Cartesian products.

Extension: preferences between sets

A best set does not necessarily consist of the best individuals:

- bundling [Chang et al, EC'03]
- complementarity
- diversity $\Rightarrow$ College Admissions Problem

Design query language extensions in which:

- sets are first-class citizens: powerset? nondeterminism?
- solutions can be constrained
- set winnow is available.

Future work

Preference management:

- elicitation: how to construct preference formulas?
- aggregation

Decision components:

- preferences between actions: workflows, ECA systems
- preferences between E-services

Preferences for XML?