Towards a Decision Query Language

Jan Chomicki

University at Buffalo

http://www.cse.buffalo.edu/~chomicki

Decisions, decisions,...

- decision scope: What kind of car? For how much?
- desirability: I prefer German cars.
- uncertainty: Will it be available in this area?

Some requirements are hard, others are soft.

What is required

Languages in which possible choices and decision criteria of agents can be formulated.

Essential features:

- data and queries
- constraints
- preferences
- *uncertainty, risk,...*

Preferences

Ordering the choices in terms of:

- desirability, coolness, ...
- reliability
- cost, convenience
- timeliness...

Two options:

- binary preference relations: what's better
- numeric utility functions: scores.

Many different preference relations

Between two hawks, which flies the higher pitch;
Between two dogs, which hath the deeper mouth;
Between two blades, which bears the better temper;
Between two horses, which doth bear him best;
Between two girls, which hath the merriest eye.

W. Shakespeare, King Henry VI.

Decision querying

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`."

What to do with the obtained information is not addressed.

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:

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

Utility functions?

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

Application scenarios

E-commerce:

- B2C: comparison shopping
- B2B: e-procurement (Cosima [Kiessling, CEC'04])

Personalization:

- personalized query results [Koutrika et al. ICDE'04]
- personalized interaction

Configuration:

- "soft" constraints

Plan of the talk

1. Preference relations vs. utility functions.
2. Query languages and query classes.
3. Preference query evaluation.
4. Preference query optimization.
5. Current work.
6. Future work.

Definitions

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

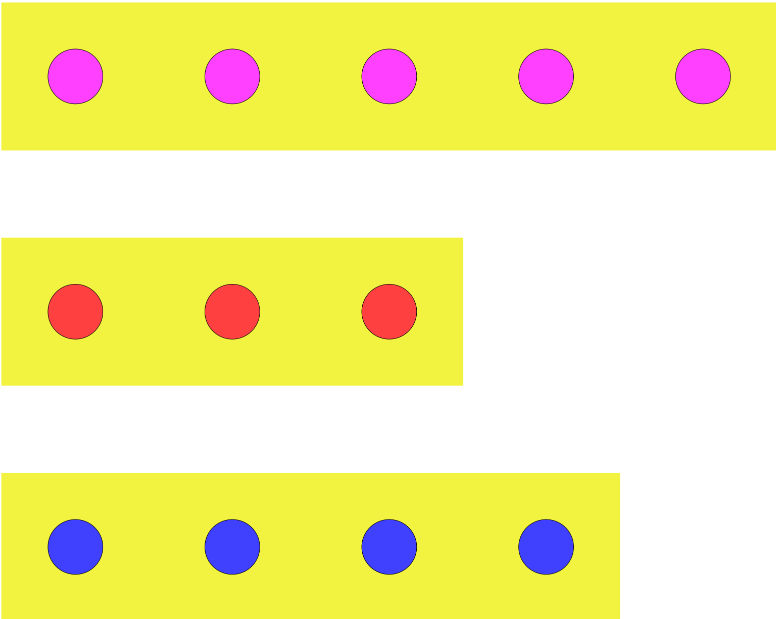
Preference formula: a first-order formula defining a preference relation.

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`*"*):

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

	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

Preference SQL

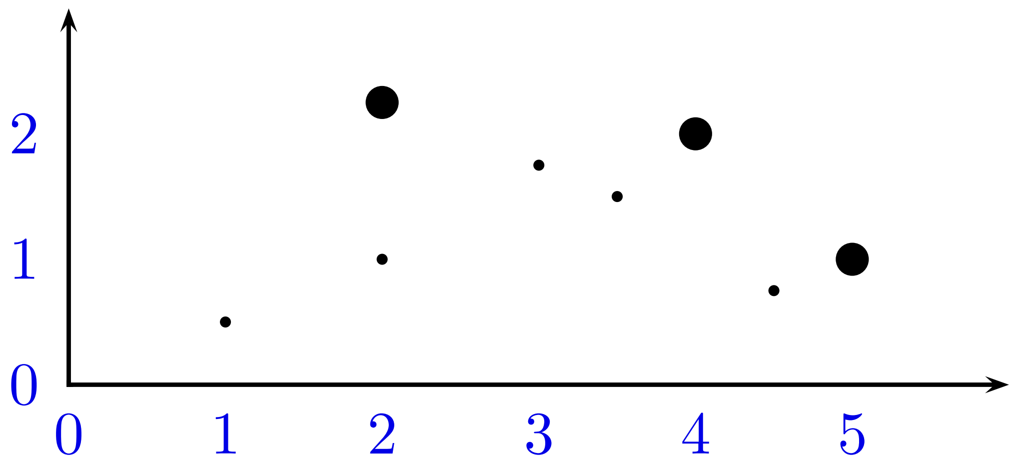
[Kiessling et al., VLDB 2002]:

- atomic and composite preference specifications
- winnow but no logical framework
- implementation: Preference SQL compiled to SQL
- deployed applications: personalized search engines and shopping agents

```
SELECT * FROM Book
PREFERRING MIN(Price)
GROUPING Title
```

Skyline queries

Find all the tuples that are not dominated by any other tuple in every dimension [Börzsönyi et al, ICDE'01] (Pareto set).



Skylines contain maxima of monotone scoring functions.

Skyline in SQL

```
SELECT ... FROM ... WHERE ...
  GROUP BY ... HAVING ...
  SKYLINE OF A1[MIN|MAX|DIFF],..., An[MIN|MAX|DIFF]
```

Skyline:

```
SKYLINE OF A DIFF, B MAX, C MIN
```

maps to the preference formula:

$$(x, y, z) \succ (x', y', z') \equiv x = x' \wedge y \geq y' \wedge z \leq z' \wedge (y > y' \vee z < z').$$

Winnow evaluation

General methods:

- translation to relational algebra/SQL (Preference SQL [Kiessling et al, VLDB'02])
- BNL: Block-Nested-Loops [Börzsönyi et al, ICDE'01]
- β-tree [Torlone, Ciaccia, SEBD'03]

Many special methods for computing skylines.

BNL

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.

Optimization of preference queries

Algebraic query optimization.

Semantic query optimization.

Algebraic laws [Chomicki, TODS'03]

Commutativity with selection:

If the formula

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

is valid, then for every r

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

Under 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} but $\sigma_{Price>20}$ does not.

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, ICDT'95]:

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

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

Current work: preference revision

Example:

- relation $Car(Make, Year)$
- preference: within each make, prefer a more recent car:

$$(m, y) \succ_{C_1} (m', y') \equiv m = m' \wedge y > y'.$$

	Make	*Year*
t_1	BMW	2002
t_2	BMW	1997
t_3	Dodge	1997

Perhaps this is not quite what the user had in mind?

Revise the original preference with a preference for BMW cars:

$$(m, y) \succ_{C_2} (m', y') \equiv m = ''\mathrm{BMW}'' \wedge m' \neq ''\mathrm{BMW}'' \wedge y = y'.$$

	Make	*Year*
t_1	BMW	2002
t_2	BMW	1997
t_3	Dodge	1997

The revised preference is the transitive closure of the union of $\succ_{C_1}$ and $\succ_{C_2}$.

Varieties of preference revision

Monotonic revision:

- original preferences are not retracted
- constructed using union

Nonmonotonic revision:

- conflict between preferences $\Rightarrow$ conflict resolution
- constructed using prioritized or Pareto composition

Preference revision vs. belief revision

Preference revision	Belief revision
First-order	Propositional
Revising a single relation	Revising a theory
Preserving order axioms	Axiomatic properties of operators

Current work: preferences between sets

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

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

Design query language extensions:

- sets first-class citizens
- set preference relations: logic, aggregation
- set winnow
- efficient implementation

Future work

Preference modelling and management:

- elicitation: how to construct preference formulas?
- aggregation
- modelling risk and uncertainty

Decision components:

- preferences between actions and plans: workflows, ECA
- preferences between E-services

Preferences for XML?

Papers

1. J. Chomicki “Querying with Intrinsic Preferences,” *EDBT’02*.

2. J. Chomicki “Preference Formulas in Relational Queries,” *ACM Transactions on Database Systems*, December 2003.

3. J. Chomicki ”Semantic Optimization of Preference Queries,” *CDB’04*; full version to appear in *Information Systems*.

4. J. Chomicki “Iterative Modification and Incremental Evaluation of Preference Queries,” *FOIKS’06*.

5. J. Chomicki, P. Godfrey, J. Gryz, D. Liang “Skyline with Presorting,” Poster at *ICDE’03*.

6. D. Mindolin, J. Chomicki “Constrained Preference Contraction,” in preparation.

Towards a multi-disciplinary perspective

Preferences are studied in AI, databases, philosophy, decision/voting/social choice theory, e-commerce,...

Interdisciplinary meetings:

- PREFS'02 (AAAI'02)
- Dagstuhl seminar 04271 (2004)
- PREFS'05 (IJCAI'05)
- PREFS'06 (ECAI'06)