Preferences, Queries, Logics

Jan Chomicki
University at Buffalo

DBRank, August 29, 2011

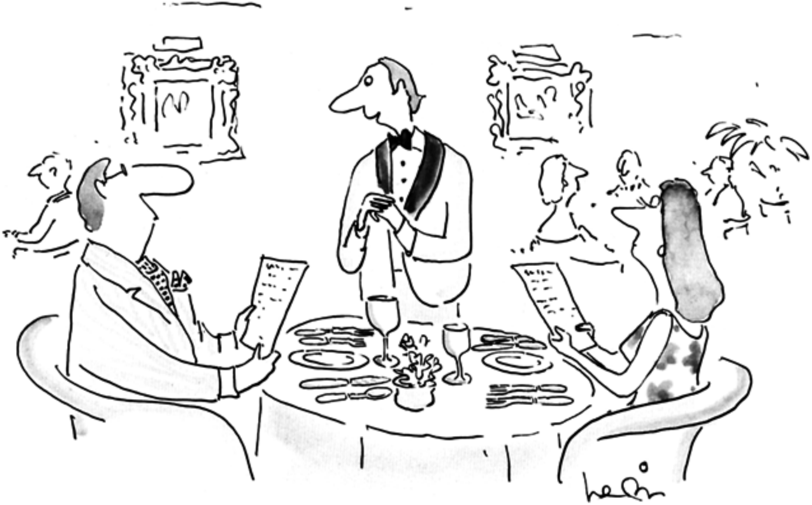
Plan of the talk

1. Preference relations
2. Preference queries
3. Advanced topics

Part I

Preference relations

Motivation



"And what is your preference in wine—single or double figures?"

Preference relations

Universe of objects

- constants: uninterpreted, numbers,...
- individuals (entities)
- tuples
- sets

Preference relations

Universe of objects

- constants: uninterpreted, numbers,...
- individuals (entities)
- tuples
- sets

Preference relation $\succ$

- binary relation between objects
- $x \succ y \equiv x$ *is_better_than* $y \equiv x$ dominates y
- an abstract, uniform way of talking about (relative) desirability, worth, cost, timeliness,..., and their combinations
- preference relations used in preference queries

Buying a car

Buying a car

Salesman: What kind of car do you prefer?

Buying a car

Salesman: What kind of car do you prefer?
Customer: The newer the better, if it is the same make. And cheap, too.

Buying a car

Salesman: What kind of car do you prefer?
Customer: The newer the better, if it is the same make. And cheap, too.
Salesman: Which is more important for you: the age or the price?

Buying a car

Salesman: What kind of car do you prefer?
Customer: The newer the better, if it is the same make. And cheap, too.
Salesman: Which is more important for you: the age or the price?
Customer: The age, definitely.

Buying a car

Salesman: What kind of car do you prefer?
Customer: The newer the better, if it is the same make. And cheap, too.
Salesman: Which is more important for you: the age or the price?
Customer: The age, definitely.
Salesman: Those are the best cars, according to your preferences, that we have in stock.

Buying a car

Salesman: What kind of car do you prefer?
Customer: The newer the better, if it is the same make. And cheap, too.
Salesman: Which is more important for you: the age or the price?
Customer: The age, definitely.
Salesman: Those are the best cars, according to your preferences, that we have in stock.
Customer: Wait...it better be a BMW.

Preferences in perspective

Applications of preferences and preference queries

Preferences in perspective

Applications of preferences and preference queries

1. decision making
2. e-commerce
3. digital libraries
4. personalization

Preferences in perspective

Applications of preferences and preference queries

1. decision making
2. e-commerce
3. digital libraries
4. personalization

Preferences are multi-disciplinary

- economic theory: von Neumann, Arrow, Sen
- philosophy: Aristotle, von Wright
- psychology: Slovic
- artificial intelligence: Boutilier, Brafman
- databases: Kießling, Kossmann

Properties of preference relations

Properties of preference relations

Properties of $\succ$

- irreflexivity: $\forall x.\ x \not\succ x$
- asymmetry: $\forall x, y.\ x \succ y \Rightarrow y \not\succ x$
- transitivity: $\forall x, y, z.\ (x \succ y \wedge y \succ z) \Rightarrow x \succ z$
- negative transitivity: $\forall x, y, z.\ (x \not\succ y \wedge y \not\succ z) \Rightarrow x \not\succ z$
- connectivity: $\forall x, y.\ x \succ y \vee y \succ x \vee x = y$

Properties of preference relations

Properties of $\succ$

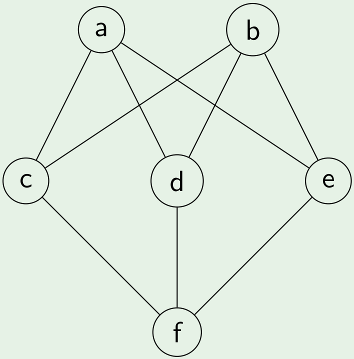
- irreflexivity: $\forall x.\ x \not\succ x$
- asymmetry: $\forall x, y.\ x \succ y \Rightarrow y \not\succ x$
- transitivity: $\forall x, y, z.\ (x \succ y \wedge y \succ z) \Rightarrow x \succ z$
- negative transitivity: $\forall x, y, z.\ (x \not\succ y \wedge y \not\succ z) \Rightarrow x \not\succ z$
- connectivity: $\forall x, y.\ x \succ y \vee y \succ x \vee x = y$

Orders

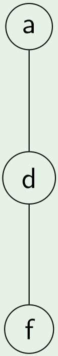
- strict partial order (SPO): irreflexive and transitive
- weak order (WO): negatively transitive SPO
- total order: connected SPO

Weak and total orders

Weak order



Total order



Order properties of preference relations

Order properties of preference relations

Irreflexivity, asymmetry: uncontroversial.

Order properties of preference relations

Irreflexivity, asymmetry: uncontroversial.

Transitivity:

- captures rationality of preference
- not always guaranteed: voting paradoxes
- helps with preference querying

Order properties of preference relations

Irreflexivity, asymmetry: uncontroversial.

Transitivity:

- captures rationality of preference
- not always guaranteed: voting paradoxes
- helps with preference querying

Negative transitivity:

- scoring functions represent weak orders

Order properties of preference relations

Irreflexivity, asymmetry: uncontroversial.

Transitivity:

- captures rationality of preference
- not always guaranteed: voting paradoxes
- helps with preference querying

Negative transitivity:

- scoring functions represent weak orders

We assume that preference relations are SPOs.

Not every SPO is a WO

Not every SPO is a WO

Indifference $\sim$

$$x \sim y \equiv x \not\succ y \wedge y \not\succ x.$$

Not every SPO is a WO

Indifference $\sim$

$$x \sim y \equiv x \not\succ y \wedge y \not\succ x.$$

Canonical example

$mazda \succ kia,\ mazda \sim^i vw,\ kia \sim^i vw$

Not every SPO is a WO

Indifference $\sim$

$$x \sim y \equiv x \not\succ y \wedge y \not\succ x.$$

Canonical example

$mazda \succ kia,\ mazda \sim^i vw,\ kia \sim^i vw$

Violation of negative transitivity

$mazda \not\succ vw,\ vw \not\succ kia,\ mazda \succ kia$

Preference specification

Preference specification

Explicit preference relations

Finite sets of pairs: bmw $\succ$ mazda, mazda $\succ$ kia,...

Preference specification

Explicit preference relations

Finite sets of pairs: bmw $\succ$ mazda, mazda $\succ$ kia,...

Implicit preference relations

- can be infinite but finitely representable
- defined using logic formulas in some constraint theory:

Preference specification

Explicit preference relations

Finite sets of pairs: bmw $\succ$ mazda, mazda $\succ$ kia,...

Implicit preference relations

- can be infinite but finitely representable
- defined using logic formulas in some constraint theory:

$$(m_1, y_1, p_1) \succ_1 (m_2, y_2, p_2) \equiv y_1 > y_2 \vee (y_1 = y_2 \wedge p_1 < p_2)$$

for relation $Car(Make, Year, Price)$.

Preference specification

Explicit preference relations

Finite sets of pairs: bmw $\succ$ mazda, mazda $\succ$ kia,...

Implicit preference relations

- can be infinite but finitely representable
- defined using logic formulas in some constraint theory:

$$(m_1, y_1, p_1) \succ_1 (m_2, y_2, p_2) \equiv y_1 > y_2 \vee (y_1 = y_2 \wedge p_1 < p_2)$$

  for relation $Car(Make, Year, Price)$.
- defined using preference constructors (Preference SQL)

Preference specification

Explicit preference relations

Finite sets of pairs: bmw $\succ$ mazda, mazda $\succ$ kia,...

Implicit preference relations

- can be infinite but finitely representable
- defined using logic formulas in some constraint theory:

$$(m_1, y_1, p_1) \succ_1 (m_2, y_2, p_2) \equiv y_1 > y_2 \vee (y_1 = y_2 \wedge p_1 < p_2)$$

  for relation $Car(Make, Year, Price)$.
- defined using preference constructors (Preference SQL)
- defined using real-valued scoring functions:

Preference specification

Explicit preference relations

Finite sets of pairs: bmw $\succ$ mazda, mazda $\succ$ kia,...

Implicit preference relations

- can be infinite but finitely representable
- defined using logic formulas in some constraint theory:

$$(m_1, y_1, p_1) \succ_1 (m_2, y_2, p_2) \equiv y_1 > y_2 \vee (y_1 = y_2 \wedge p_1 < p_2)$$

  for relation $Car(Make, Year, Price)$.
- defined using preference constructors (Preference SQL)
- defined using real-valued scoring functions: $F(m, y, p) = \alpha \cdot y + \beta \cdot p$

Preference specification

Explicit preference relations

Finite sets of pairs: bmw $\succ$ mazda, mazda $\succ$ kia,...

Implicit preference relations

- can be infinite but finitely representable
- defined using logic formulas in some constraint theory:

$$(m_1, y_1, p_1) \succ_1 (m_2, y_2, p_2) \equiv y_1 > y_2 \vee (y_1 = y_2 \wedge p_1 < p_2)$$

  for relation $Car(Make, Year, Price)$.
- defined using preference constructors (Preference SQL)
- defined using real-valued scoring functions: $F(m, y, p) = \alpha \cdot y + \beta \cdot p$
  $(m_1, y_1, p_1) \succ_2 (m_2, y_2, p_2) \equiv F(m_1, y_1, p_1) > F(m_2, y_2, p_2)$

Logic formulas

Logic formulas

The language of logic formulas

- constants
- object (tuple) attributes
- comparison operators: $=, \neq, <, >, \ldots$
- arithmetic operators: $+, \cdot, \ldots$
- Boolean connectives: $\neg$, $\wedge$, $\vee$
- quantifiers:
  - $\forall, \exists$
  - usually can be eliminated (quantifier elimination)
- no database relations

Representability

Representability

Definition

A scoring function f represents a preference relation $\succ$ if for all x, y

$$x \succ y \equiv f(x) > f(y).$$

Representability

Definition

A scoring function f represents a preference relation $\succ$ if for all x, y

$$x \succ y \equiv f(x) > f(y).$$

Necessary condition for representability

The preference relation $\succ$ is a weak order.

Representability

Definition

A scoring function f represents a preference relation $\succ$ if for all x, y

$$x \succ y \equiv f(x) > f(y).$$

Necessary condition for representability

The preference relation $\succ$ is a weak order.

Sufficient condition for representability

- $\succ$ is a weak order
- the domain is countable or some continuity conditions are satisfied (studied in decision theory)

Preference constructors [Kießling, 2002]

Preference constructors [Kießling, 2002]

Good values

Prefer $v \in S_1$ over $v \notin S_1$.

Preference constructors [Kießling, 2002]

Good values

Prefer $v \in S_1$ over $v \notin S_1$.

`POS(Make,{mazda,vw})`

Preference constructors [Kießling, 2002]

Good values

Prefer $v \in S_1$ over $v \notin S_1$.

Bad values

Prefer $v \notin S_1$ over $v \in S_1$.

`POS(Make,{mazda,vw})`

Preference constructors [Kießling, 2002]

Good values

Prefer $v \in S_1$ over $v \notin S_1$.

`POS(Make,{mazda,vw})`

Bad values

Prefer $v \notin S_1$ over $v \in S_1$.

`NEG(Make,{yugo})`

Preference constructors [Kießling, 2002]

Good values

Prefer $v \in S_1$ over $v \notin S_1$.

`POS(Make,{mazda,vw})`

Bad values

Prefer $v \notin S_1$ over $v \in S_1$.

`NEG(Make,{yugo})`

Explicit preference

Preference encoded by a finite directed graph.

Preference constructors [Kießling, 2002]

Good values

Prefer $v \in S_1$ over $v \notin S_1$.

`POS(Make,{mazda,vw})`

Bad values

Prefer $v \notin S_1$ over $v \in S_1$.

`NEG(Make,{yugo})`

Explicit preference

Preference encoded by a finite directed graph.

```
EXP(Make,{(bmw,ford),...,
          (mazda,kia)})
```

Preference constructors [Kießling, 2002]

Good values

Prefer $v \in S_1$ over $v \notin S_1$.

`POS(Make,{mazda,vw})`

Bad values

Prefer $v \notin S_1$ over $v \in S_1$.

`NEG(Make,{yugo})`

Explicit preference

Preference encoded by a finite directed graph.

```
EXP(Make,{(bmw,ford),...,
          (mazda,kia)})
```

Value comparison

Prefer larger/smaller values.

Preference constructors [Kießling, 2002]

Good values

Prefer $v \in S_1$ over $v \notin S_1$.

`POS(Make,{mazda,vw})`

Bad values

Prefer $v \notin S_1$ over $v \in S_1$.

`NEG(Make,{yugo})`

Explicit preference

Preference encoded by a finite directed graph.

```
EXP(Make,{(bmw,ford),...,
          (mazda,kia)})
```

Value comparison

Prefer larger/smaller values.

```
HIGHEST(Year)
LOWEST(Price)
```

Preference constructors [Kießling, 2002]

Good values

Prefer $v \in S_1$ over $v \notin S_1$.

`POS(Make,{mazda,vw})`

Bad values

Prefer $v \notin S_1$ over $v \in S_1$.

`NEG(Make,{yugo})`

Explicit preference

Preference encoded by a finite directed graph.

```
EXP(Make,{(bmw,ford),...,
          (mazda,kia)})
```

Value comparison

Prefer larger/smaller values.

```
HIGHEST(Year)
LOWEST(Price)
```

Distance

Prefer values closer to v_0.

Preference constructors [Kießling, 2002]

Good values

Prefer $v \in S_1$ over $v \notin S_1$.

```
POS(Make,{mazda,vw})
```

Bad values

Prefer $v \notin S_1$ over $v \in S_1$.

```
NEG(Make,{yugo})
```

Explicit preference

Preference encoded by a finite directed graph.

```
EXP(Make,{(bmw,ford),...,
          (mazda,kia)})
```

Value comparison

Prefer larger/smaller values.

```
HIGHEST(Year)
LOWEST(Price)
```

Distance

Prefer values closer to v_0.

```
AROUND(Price,12K)
```

Combining preferences

Combining preferences

Preference composition

- combining preferences about objects of the same kind
- dimensionality is not increased
- representing preference aggregation, revision, ...

Combining preferences

Preference composition

- combining preferences about objects of the same kind
- dimensionality is not increased
- representing preference aggregation, revision, ...

Preference accumulation

- defining preferences over objects in terms of preferences over simpler objects
- dimensionality is increased (preferences over Cartesian product).

Combining preferences: composition

Combining preferences: composition

Boolean composition

$$x \succ^{\cup} y \equiv x \succ_1 y \;\vee\; x \succ_2 y$$

and similarly for $\cap$.

Combining preferences: composition

Boolean composition

$$x \succ^{\cup} y \equiv x \succ_1 y \;\; \vee \;\; x \succ_2 y$$

and similarly for $\cap$.

Prioritized composition

$$x \succ^{lex} y \equiv x \succ_1 y \vee (y \not\succ_1 x \wedge x \succ_2 y).$$

Combining preferences: composition

Boolean composition

$$x \succ^{\cup} y \equiv x \succ_1 y \ \vee \ x \succ_2 y$$

and similarly for $\cap$.

Prioritized composition

$$x \succ^{lex} y \equiv x \succ_1 y \vee (y \not\succ_1 x \wedge x \succ_2 y).$$

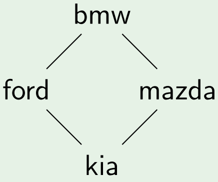
Pareto composition

$$x \succ^{Par} y \equiv (x \succ_1 y \wedge y \not\succ_2 x) \vee (x \succ_2 y \wedge y \not\succ_1 x).$$

Preference composition

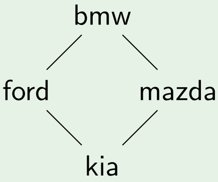
Preference composition

Preference relation $\succ_1$

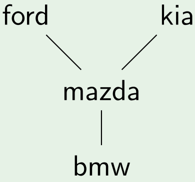


Preference composition

Preference relation $\succ_1$

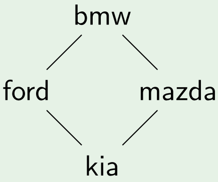


Preference relation $\succ_2$

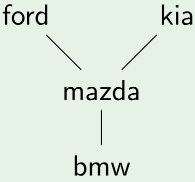


Preference composition

Preference relation $\succ_1$



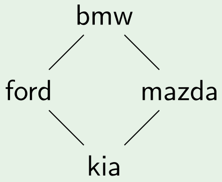
Preference relation $\succ_2$



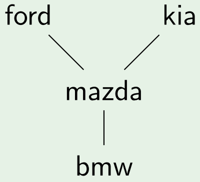
Prioritized composition

Preference composition

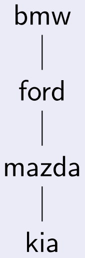
Preference relation $\succ_1$



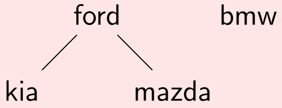
Preference relation $\succ_2$



Prioritized composition



Pareto composition



Combining preferences: accumulation [Kiessling, 2002]

Combining preferences: accumulation [Kiessling, 2002]

Prioritized accumulation: $\succ^{pr} = (\succ_1 \,\&\, \succ_2)$

$$(x_1, x_2) \succ^{pr} (y_1, y_2) \equiv x_1 \succ_1 y_1 \vee (x_1 = y_1 \wedge x_2 \succ_2 y_2).$$

Combining preferences: accumulation [Kiessling, 2002]

Prioritized accumulation: $\succ^{pr} = (\succ_1 \,\&\, \succ_2)$

$$(x_1, x_2) \succ^{pr} (y_1, y_2) \equiv x_1 \succ_1 y_1 \vee (x_1 = y_1 \wedge x_2 \succ_2 y_2).$$

Pareto accumulation: $\succ^{pa} = (\succ_1 \otimes \succ_2)$

$$(x_1, x_2) \succ^{pa} (y_1, y_2) \equiv (x_1 \succ_1 y_1 \wedge x_2 \succeq_2 y_2) \vee (x_1 \succeq_1 y_1 \wedge x_2 \succ_2 y_2).$$

Combining preferences: accumulation [Kiessling, 2002]

Prioritized accumulation: $\succ^{pr} = (\succ_1 \& \succ_2)$

$$(x_1, x_2) \succ^{pr} (y_1, y_2) \equiv x_1 \succ_1 y_1 \vee (x_1 = y_1 \wedge x_2 \succ_2 y_2).$$

Pareto accumulation: $\succ^{pa} = (\succ_1 \otimes \succ_2)$

$$(x_1, x_2) \succ^{pa} (y_1, y_2) \equiv (x_1 \succ_1 y_1 \wedge x_2 \succeq_2 y_2) \vee (x_1 \succeq_1 y_1 \wedge x_2 \succ_2 y_2).$$

Properties

- closure
- associativity
- commutativity of Pareto accumulation

Skylines

Skylines

Skyline

Given single-attribute total preference relations $\succ_{A_1}, \ldots, \succ_{A_n}$ for a relational schema $R(A_1, \ldots, A_n)$, the skyline preference relation $\succ^{sky}$ is defined as

$$\succ^{sky} = \succ_{A_1} \otimes \succ_{A_2} \otimes \cdots \otimes \succ_{A_n} .$$

Unfolding the definition

$$(x_1, \ldots, x_n) \succ^{sky} (y_1, \ldots, y_n) \equiv \bigwedge_i x_i \succeq_{A_i} y_i \wedge \bigvee_i x_i \succ_{A_i} y_i .$$

Skyline in Euclidean space

Skyline in Euclidean space

Two-dimensional Euclidean space

$$(x_1, x_2) \succ^{sky} (y_1, y_2) \equiv x_1 \geqslant y_1 \wedge x_2 > y_2 \vee x_1 > y_1 \wedge x_2 \geqslant y_2$$

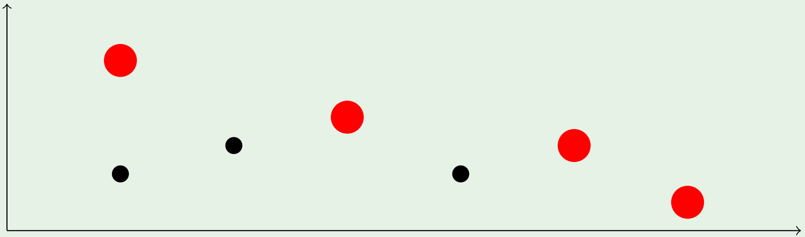
Skyline consists of $\succ^{sky}$-maximal vectors

Skyline in Euclidean space

Two-dimensional Euclidean space

$$(x_1, x_2) \succ^{sky} (y_1, y_2) \equiv x_1 \geqslant y_1 \wedge x_2 > y_2 \vee x_1 > y_1 \wedge x_2 \geqslant y_2$$

Skyline consists of $\succ^{sky}$-maximal vectors



Skyline properties

Skyline properties

Invariance

A skyline preference relation is unaffacted by scaling or shifting in any dimension.

Skyline properties

Invariance

A skyline preference relation is unaffacted by scaling or shifting in any dimension.

Maxima

A skyline consists of the maxima of monotonic scoring functions.

Skyline properties

Invariance

A skyline preference relation is unaffacted by scaling or shifting in any dimension.

Maxima

A skyline consists of the maxima of monotonic scoring functions.

Skyline is not a weak order

$$(2,0) \not\succ_{sky} (0,2), (0,2) \not\succ_{sky} (1,0), (2,0) \succ_{sky} (1,0)$$

Skyline in SQL

Skyline in SQL

Grouping

Designating attributes not used in comparisons (DIFF).

Example

```
SELECT * FROM Car
SKYLINE Price MIN,
        Year MAX,
        Make DIFF
```

Part II

Preference queries

Winnow [Ch., 2002]

Winnow [Ch., 2002]

Winnow

- new relational algebra operator ω (other names: Best, BMO)
- retrieves the non-dominated (best) elements in a database relation
- can be expressed in terms of other operators

Winnow [Ch., 2002]

Winnow

- new relational algebra operator ω (other names: Best, BMO)
- retrieves the non-dominated (best) elements in a database relation
- can be expressed in terms of other operators

Definition

Given a preference relation $\succ$ and a database relation r:

$$\omega_{\succ}(r) = \{t \in r \mid \neg\exists t' \in r.\ t' \succ t\}.$$

Winnow [Ch., 2002]

Winnow

- new relational algebra operator ω (other names: Best, BMO)
- retrieves the non-dominated (best) elements in a database relation
- can be expressed in terms of other operators

Definition

Given a preference relation $\succ$ and a database relation r:

$$\omega_{\succ}(r) = \{t \in r \mid \neg \exists t' \in r.\ t' \succ t\}.$$

Notation: If a preference relation $\succ_C$ is defined using a formula C, then we write $\omega_C(r)$, instead of $\omega_{\succ_C}(r)$.

Winnow [Ch., 2002]

Winnow

- new relational algebra operator ω (other names: Best, BMO)
- retrieves the non-dominated (best) elements in a database relation
- can be expressed in terms of other operators

Definition

Given a preference relation $\succ$ and a database relation r:

$$\omega_{\succ}(r) = \{t \in r \mid \neg \exists t' \in r.\ t' \succ t\}.$$

Notation: If a preference relation $\succ_C$ is defined using a formula C, then we write $\omega_C(r)$, instead of $\omega_{\succ_C}(r)$.

Skyline query

$\omega_{\succ_{sky}}(r)$ computes the set of maximal vectors in r (the skyline set).

Example of winnow

Example of winnow

Relation $Car(Make, Year, Price)$

Preference relation:

$$(m, y, p) \succ_1 (m', y', p') \equiv y > y' \vee (y = y' \wedge p < p').$$

Example of winnow

Relation $Car(Make, Year, Price)$

Preference relation:

$$(m, y, p) \succ_1 (m', y', p') \equiv y > y' \vee (y = y' \wedge p < p').$$

Make	Year	Price
mazda	2009	20K
ford	2009	15K
ford	2007	12K

Example of winnow

Relation $Car(Make, Year, Price)$

Preference relation:

$$(m, y, p) \succ_1 (m', y', p') \equiv y > y' \vee (y = y' \wedge p < p').$$

Make	Year	Price
mazda	2009	20K
ford	2009	15K
ford	2007	12K

Generalizations of winnow

Generalizations of winnow

Iterating winnow

$$\omega^0_{\succ}(r) = \emptyset$$

$$\omega^{n+1}_{\succ}(r) = \omega_{\succ}(r - \bigcup_{0 \leqslant i \leqslant n} \omega^i_{\succ}(r))$$

Generalizations of winnow

Iterating winnow

$$\omega^0_\succ(r) = \varnothing$$

$$\omega^{n+1}_\succ(r) = \omega_\succ(r - \bigcup_{0 \leqslant i \leqslant n} \omega^i_\succ(r))$$

Ranking

Rank tuples by their minimum distance from a winnow tuple:

$$\eta_\succ(r) = \{(t, i) \mid t \in \omega^i_C(r)\}.$$

Generalizations of winnow

Iterating winnow

$$\omega^0_\succ(r) = \emptyset$$

$$\omega^{n+1}_\succ(r) = \omega_\succ(r - \bigcup_{0 \leqslant i \leqslant n} \omega^i_\succ(r))$$

Ranking

Rank tuples by their minimum distance from a winnow tuple:

$$\eta_\succ(r) = \{(t, i) \mid t \in \omega^i_C(r)\}.$$

k-band

Return the tuples dominated by at most k tuples:

$$\omega_\succ(r) = \{t \in r \mid \#\{t' \in r \mid t' \succ t\} \leqslant k\}.$$

Preference SQL

Preference SQL

The language

- basic preference constructors
- Pareto/prioritized accumulation
- new SQL clause `PREFERRING`
- groupwise preferences
- native implementation

Preference SQL

The language

- basic preference constructors
- Pareto/prioritized accumulation
- new SQL clause `PREFERRING`
- groupwise preferences
- native implementation

Winnow in Preference SQL

```
SELECT * FROM Car
PREFERRING HIGHEST(Year)
    CASCADE LOWEST(Price)
```

Algebraic laws [Ch., 2002; Kießling, 2002]

Algebraic laws [Ch., 2002; Kießling, 2002]

Commutativity of winnow 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

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

Algebraic laws [Ch., 2002; Kießling, 2002]

Commutativity of winnow 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

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

Under the preference relation

$$(m, y, p) \succ_{C_1} (m', y', p') \equiv y > y' \wedge p \leqslant p' \vee y \geqslant y' \wedge p < p'$$

the selection $\sigma_{Price<20K}$ commutes with ω_{C_1} but $\sigma_{Price>20K}$ does not.

Semantic query optimization [Ch., 2004]

Semantic query optimization [Ch., 2004]

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 implies the formula

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

Integrity constraints

Integrity constraints

Constraint-generating dependencies (CGD) [Baudinet et al., 1995]

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

Integrity constraints

Constraint-generating dependencies (CGD) [Baudinet et al., 1995]

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

CGD entailment

Decidable by reduction to the validity of ∀-formulas in the constraint theory (assuming the theory is decidable).

Part III

Advanced topics

Preference modification

Preference modification

Goal

Given a preference relation $\succ$ and additional preference or indifference information $\mathcal{I}$, construct a new preference relation $\succ'$ whose contents depend on $\succ$ and $\mathcal{I}$.

Preference modification

Goal

Given a preference relation $\succ$ and additional preference or indifference information $\mathcal{I}$, construct a new preference relation $\succ'$ whose contents depend on $\succ$ and $\mathcal{I}$.

General postulates

- fulfillment: the new information $\mathcal{I}$ should be completely incorporated into $\succ'$
- minimal change: $\succ'$ should be as close to $\succ$ as possible
- closure:
  - order-theoretic (SPO, WO) properties of $\succ$ should be preserved in $\succ'$
  - finiteness or finite representability of $\succ$ should also be preserved in $\succ'$

Preference revision [Ch., 2007]

Preference revision [Ch., 2007]

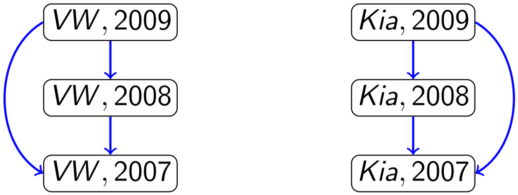
Setting

- new information: revising preference relation $\succ_0$
- composition operator θ: union, prioritized or Pareto composition
- composition eliminates (some) preference conflicts
- additional assumptions: interval orders
- $\succ' = TC(\succ_0 \, \theta \succ)$ to guarantee SPO

Preference revision [Ch., 2007]

Setting

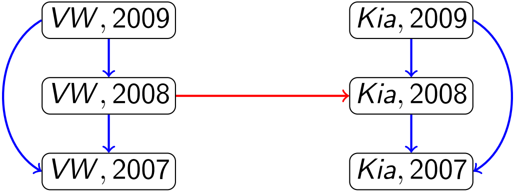
- new information: revising preference relation $\succ_0$
- composition operator θ: union, prioritized or Pareto composition
- composition eliminates (some) preference conflicts
- additional assumptions: interval orders
- $\succ' = TC(\succ_0 \, \theta \, \succ)$ to guarantee SPO



Preference revision [Ch., 2007]

Setting

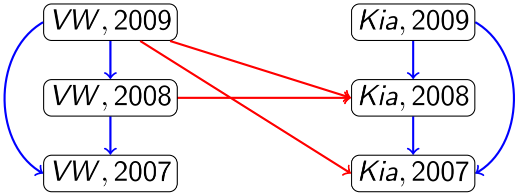
- new information: revising preference relation $\succ_0$
- composition operator θ: union, prioritized or Pareto composition
- composition eliminates (some) preference conflicts
- additional assumptions: interval orders
- $\succ' = TC(\succ_0 \; \theta \succ)$ to guarantee SPO



Preference revision [Ch., 2007]

Setting

- new information: revising preference relation $\succ_0$
- composition operator θ: union, prioritized or Pareto composition
- composition eliminates (some) preference conflicts
- additional assumptions: interval orders
- $\succ' = TC(\succ_0 \; \theta \succ)$ to guarantee SPO



Preference contraction [Mindolin, Ch., 2011]

Preference contraction [Mindolin, Ch., 2011]

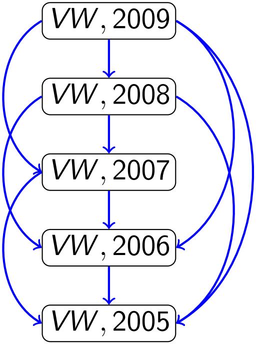
Setting

- new information: contractor relation CON
- $\succ'$: maximal subset of $\succ$ disjoint with CON

Preference contraction [Mindolin, Ch., 2011]

Setting

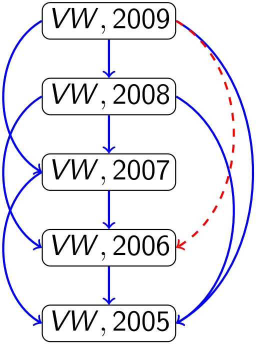
- new information: contractor relation CON
- $\succ'$: maximal subset of $\succ$ disjoint with CON



Preference contraction [Mindolin, Ch., 2011]

Setting

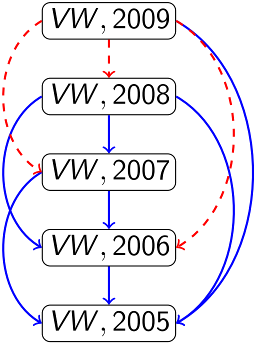
- new information: contractor relation CON
- $\succ'$: maximal subset of $\succ$ disjoint with CON



Preference contraction [Mindolin, Ch., 2011]

Setting

- new information: contractor relation CON
- $\succ'$: maximal subset of $\succ$ disjoint with CON



Substitutability [Balke et al., 2006]

Substitutability [Balke et al., 2006]

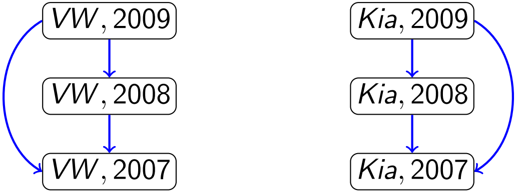
Setting

- new information: set of indifference pairs
- additional preferences are added to achieve object substitutability

Substitutability [Balke et al., 2006]

Setting

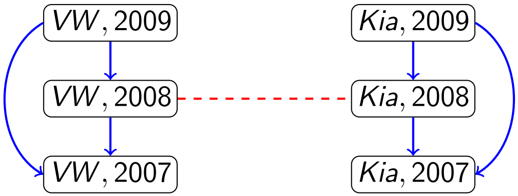
- new information: set of indifference pairs
- additional preferences are added to achieve object substitutability



Substitutability [Balke et al., 2006]

Setting

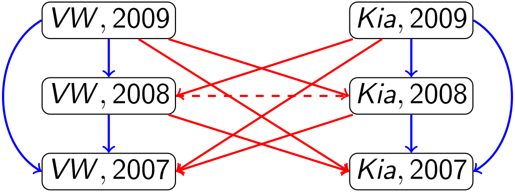
- new information: set of indifference pairs
- additional preferences are added to achieve object substitutability



Substitutability [Balke et al., 2006]

Setting

- new information: set of indifference pairs
- additional preferences are added to achieve object substitutability



Preferences over finite sets

Preferences over finite sets

Set preferences

Induced:

$$X \ggcurly Y \equiv \forall x \in X.\ \exists y \in Y.\ x \succ y$$

Aggregate:

$$X \ggcurly Y \equiv sum_A(X) > sum_A(Y)$$

Preferences over finite sets

Set preferences

Induced:

$$X \ggcurly Y \equiv \forall x \in X.\ \exists y \in Y.\ x \succ y$$

Aggregate:

$$X \ggcurly Y \equiv sum_A(X) > sum_A(Y)$$

Set preference queries

- find the best subsets of a given set
- restrictions on cardinality

Preferences over set profiles [Zhang, Ch., 2011]

Preferences over set profiles [Zhang, Ch., 2011]

Name	Area	Rating
Newton	Physics	9
Einstein	Physics	10
Gargamel	Alchemy	0

Preferences over set profiles [Zhang, Ch., 2011]

Name	Area	Rating
Newton	Physics	9
Einstein	Physics	10
Gargamel	Alchemy	0

Preferences

- 2-element subsets
- P_1: at most one physicist
- P_2: higher total rating
- P_1 more important than P_2

Set profile (F_1, F_2)

F_1(S) $\equiv$ `SELECT COUNT(Name) FROM S WHERE Area='Physics'`

F_2(S) $\equiv$ `SELECT SUM(Rating) FROM S`

Set profile (F_1, F_2)

$$F_1(\mathtt{S}) \equiv \mathtt{SELECT\ COUNT(Name)\ FROM\ S\ WHERE\ Area='Physics'}$$
$$F_2(\mathtt{S}) \equiv \mathtt{SELECT\ SUM(Rating)\ FROM\ S}$$

Set preference relations

$$X \succ\!\!\succ_1 Y \equiv F_1(X) \leqslant 1 \wedge F_1(Y) > 1$$
$$X \succ\!\!\succ_2 Y \equiv F_2(X) > F_2(Y)$$

Set profile (F_1, F_2)

$$F_1(\mathtt{S}) \equiv \mathtt{SELECT\ COUNT(Name)\ FROM\ S\ WHERE\ Area='Physics'}$$
$$F_2(\mathtt{S}) \equiv \mathtt{SELECT\ SUM(Rating)\ FROM\ S}$$

Set preference relations

$$X \succ_1 Y \equiv F_1(X) \leqslant 1 \wedge F_1(Y) > 1$$
$$X \succ_2 Y \equiv F_2(X) > F_2(Y)$$

Prioritized composition

$$X \succ Y \equiv X \succ_1 Y \vee (Y \not\succ_1 X \wedge X \succ_2 Y)$$

Prospective research topics

Prospective research topics

Definability

- of scoring functions representing preference relations
- of CP-nets and other graphical models of preferences

Prospective research topics

Definability

- of scoring functions representing preference relations
- of CP-nets and other graphical models of preferences

Extrinsic preference relations

Preference relations that are not fully defined by tuple contents:

$$x \succ y \equiv \exists n_1, n_2.\ Dissatisfied(x, n_1) \wedge Dissatisfied(y, n_2) \wedge n_1 < n_2.$$

Prospective research topics

Definability

- of scoring functions representing preference relations
- of CP-nets and other graphical models of preferences

Extrinsic preference relations

Preference relations that are not fully defined by tuple contents:

$$x \succ y \equiv \exists n_1, n_2.\ Dissatisfied(x, n_1) \wedge Dissatisfied(y, n_2) \wedge n_1 < n_2.$$

Incomplete preferences

- tuple scores and probabilities [Soliman et al., 2007]
- uncertain tuple scores
- disjunctive preferences: $a \succ b \vee a \succ c$

Prospective applications

Prospective applications

Databases

- preference queries as decision components: workflows, event systems
- personalization of query results

Prospective applications

Databases

- preference queries as decision components: workflows, event systems
- personalization of query results

Multi-agent systems

- conflict resolution
- negotiating joint preferences and decisions
- negotiation preferences

Prospective applications

Databases

- preference queries as decision components: workflows, event systems
- personalization of query results

Multi-agent systems

- conflict resolution
- negotiating joint preferences and decisions
- negotiation preferences

Social media

- preference similarity and stability
- preference aggregation

Preference queries vs. Top-K queries

Preference queries vs. Top-K queries

Preference queries	**Top-K queries**
Binary preference relations	Scoring functions
Clear declarative reading	"Mysterious" formulation Nondeterminism
No relational data model extension	Rank-relations [Li et al., 2005]
Structured data	Structured and unstructured data
Manual construction	Automatic construction
Preference SQL	Mainstream DBMS

Acknowledgments

- Paolo Ciaccia
- Parke Godfrey
- Jarek Gryz
- Werner Kießling
- Denis Mindolin
- Sławek Staworko
- Xi Zhang

Companion paper

Jan Chomicki, Logical Foundations of Preference Queries, *Data Engineering Bulletin*, 34(2), 2011.