Foundations of Preferences in Database Systems

Werner Kießling

Institute of Computer Science
University of Augsburg
Foundations of Preferences in Database Systems

Overview

1. Motivation
2. Preference Model
3. Preference Engineering
4. Preference Algebra
5. Preference Query Model
6. Related Work
7. Summary and Ongoing Work

[For technical details: see the proceedings]
Foundations of Preferences in Database Systems

1. Motivation

The importance of preferences:
- Preferences are ubiquitous in all our daily and business lives.
- Importance has been addressed by various scientific communities.
- For database and Internet communities:
  - Personalization of search engines and of Web services
  - Deficiencies: Empty result & flooding effect

Choice of an appropriate preference model:
- Preferences are NOT hard constraints.
- Preferences are NOT necessarily numbers.
- Preferences are NOT necessarily total orders. [Video]
An intuitive notion of preferences:

- Preferences are **personalized wishes**: “I like A better than B”.
- There may exist **incomparable** items.
- “Better than” may be defined **qualitatively** or **quantitatively**.
- Preferences may be complex, covering **multiple attributes**.
- Preferences may come from **different**, even conflicting **sources**.
Foundations of Preferences in Database Systems

2. Preference Model

- **Formal semantics of preferences:**
  Given a set $A$ of attribute names with a domain of values $\text{dom}(A)$.
  - A preference $P$ is a **strict partial order** $P = (A, <_P)$ on $\text{dom}(A)$.
  - $x <_P y$ is interpreted as “**I like $y$ better than $x$**”.

- **Some familiar special cases:**
  - **Chain preference (total order):** for all $x \neq y$: $x <_P y \lor y <_P x$
  - **Anti-chain preference:** $<_P = \emptyset$
  - **Dual preference** $P^\delta = (A, <_{P^\delta}):$ $x <_{P^\delta} y$ iff $y <_P x$
  - **Subset preference** $P^\subseteq = (S, <_P):$ $S \subseteq \text{dom}(A)$
A choice of base preference constructors

- POS(A, POS-set)
- NEG(A, NEG-set)
- POS/NEG(A, POS-set; NEG-set)
- POS/POS(A, POS1-set; POS2-set)
- EXP(A, E-graph)
- AROUND(A, z)
- BETWEEN(A, [low, up])
- LOWEST(A)
- HIGHEST(A)
- SCORE(A, f), f: dom(A) -> R

POS(transmission, {automatic})
NEG(make, {Ferrari})
POS/NEG(color, {yellow}; {gray})
POS/POS(category, {cabriolet}; {roadster})
EXP(color, {(green, yellow), (green, red), (yellow, white)})
AROUND(price, 40000)
BETWEEN(mileage, [20000, 30000])
LOWEST(fuel_consumption)
HIGHEST(horsepower)
x <P y iff f(x) < f(y)
A choice of complex preference constructors

Given \( P_1 = (A_1, \prec P_1) \) and \( P_2 = (A_2, \prec P_2) \); \( A_1 \) and \( A_2 \) may overlap.

- **Pareto preference \( P \):** \( P_1 \) and \( P_2 \) are equally important
  \[ P = (A_1 \cup A_2, \prec P_1 \otimes P_2) \]

- **Prioritized preference \( P \):** \( P_1 \) is more important than \( P_2 \)
  \[ P = (A_1 \cup A_2, \prec P_1 \& P_2) \]

- **Numerical preference \( P \):** Ranking-function \( F: \mathbb{R} \times \mathbb{R} \to \mathbb{R} \)
  \[ P = (A_1 \cup A_2, \prec \text{rank}_F(P_1, P_2)) \]
Inductive construction of preference terms

- Each base, subset or dual preference is a preference term.
- Given preference terms $P_1$ and $P_2$, $P$ is a preference term iff
  - Pareto preference: $P := P_1 \otimes P_2$
  - Prioritized preference: $P := P_1 \& P_2$
  - Numerical preference: $P := \text{rank}_r(P_1, P_2)$
  - Intersection preference: $P := P_1 \smallsetminus P_2$
  - Disjoint union preference: $P := P_1 + P_2$
  - Linear sum preference: $P := P_1 \oplus P_2$

**Theorem:** Each preference term defines a preference
E-commerce scenario

Julia and Leslie want to buy a used car from their car dealer Michael.

- Julia has a complex customer preference Q1:

  P1 := POS/POS(category, \{cabriolet\}; \{roadster\})
  P2 := POS(transmission, \{automatic\})
  P3 := AROUND(horsepower, 100)
  P4 := LOWEST(price)
  P5 := NEG(color, \{gray\})

  Q1 = (\{color, category, transmission, horsepower, price\}, <Q1)
  := P5 & ((P1 \odot P2 \odot P3) & P4)
Foundations of Preferences in Database Systems

Preference Engineering (cont’d)

- Michael has **domain knowledge** about cars:
  
  \[ P_6 := \text{HIGHEST(year-of-construction)} \]

- Michael has his own **vendor preferences**, of course:
  
  \[ P_7 := \text{HIGHEST(commission)} \]

- Michael is a fair play guy, performing this **query expansion**:
  
  \[ Q_2 = \{ \text{color, category, transmission, horsepower, price, year-of-construction, commission} \}, <Q_2) \]
  
  \[ := (Q_1 \& P_6) \& P_7 \]
  
  \[ = ((P_5 \& ((P_1 \& P_2 \& P_3) \& P_4)) \& P_6) \& P_7 \]
Now Leslie enters the scene. She has a **different** color taste:

\[ P_8 := \text{POS/NEG}(\text{color}, \{\text{blue}\}; \{\text{gray, red}\}) \]

Julia still insists on her color preference \( P_5 \).

Leslie convinces Julia that money should matter as much as color.

Q1 **adapted** to this new situation reads as follows:

\[
Q_1^* = (\{\text{color, category, transmission, horsepower, price}\}, <Q_1^*)
\]

\[
:= (P_5 \otimes P_8 \otimes P_4) \& (P_1 \otimes P_2 \otimes P_3)
\]

Finally Michael expands \( Q_2^* \) ...

... and the story might end that everybody is happy with the query result ...
4. Preference Algebra

- Commutative and associative laws for preference terms
- More basic laws for preference terms

- "Discrimination" theorem for $P_1 \& P_2$ ("more important than")
  - $P_1 \& P_2 \equiv P_1 + (A_1\leftrightarrow & P_2)$
  - $P_1 \& P_2 \equiv P_1$ if $P_1 = (A, <P_1)$ and $P_2 = (A, <P_2)$

- "Non-discrimination" theorem for $P_1 \otimes P_2$ ("equally important")
  - $P_1 \otimes P_2 \equiv (P_1 \& P_2) \diamond (P_2 \& P_1)$
  - $P_1 \otimes P_2 \equiv P_1 \diamond P_2$ if $P_1 = (A, <P_1)$ and $P_2 = (A, <P_2)$
5. Preference Query Model

Preferences and database query languages

- **The Best-Matches-Only (BMO) query model:**
  - Preferences are *soft selection* conditions (not every wish can become true).
  - **Perfect choices** are the *maximal elements* of a given preference $P$.
  - BMO query result: Retrieve *perfect choices*, if present in the database. Otherwise deliver *best-matching alternatives*, but nothing worse.

- **Preference query:** (declarative semantics)
  Given $P = (A, <P)$ and a relation $R$, $R[A] \subseteq \text{dom}(A)$. Consider $P^R = (R[A], <P)$:
  - $\sigma[P](R) := \{ t \in R \mid t[A] \in \text{max}(P^R) \}$
  - $\sigma[P \text{ groupby } A](R) := \sigma[A \leftrightarrow & P](R)$
Properties of BMO

- **Non-monotonicity of BMO queries:**
  - Being ‘better than’ is **not** a property of a single value.
  - BMO query results adapt to the **quality** of data in the database.

- **Adaptive AND/ OR-like filter effect of BMO queries:**
  - Nuisances from the **empty result** are **defeated**: Implicit query relaxation
  - Nuisances from the **flooding effect** are **defeated**: On-the-fly filtering of worse results
  - Tedious **parametric search** and boolean **expert search** are **obsolete**.
Decomposition laws

- **Decomposition of prioritized queries:**
  \[ \sigma[P1 \& P2](R) = \sigma[P1](R) \cap \sigma[P2 \text{ groupby } A1](R) \]
  \[ \sigma[P1 \& P2](R) = \sigma[P2](\sigma[P1](R)) , \text{ if } P1 \text{ is a chain} \quad \text{["cascade of preferences"]} \]

- **Decomposition of Pareto queries:**
  \[ \sigma[P1 \otimes P2](R) = (\sigma[P1](R) \cap \sigma[P2 \text{ groupby } A1](R)) \cup \]
  \[ (\sigma[P2](R) \cap \sigma[P1 \text{ groupby } A2](R)) \cup \]
  \[ YY(P1 \& P2, P2 \& P1)^R \]
6. Related Work

- **Original idea for preferences in deductive database systems:**
  - Datalog_S (Kießling, Güntzer; 1994)
  - Subsumption lattices (Köstler, Kießling, Thöne, Güntzer; 1995)

- **Preference SQL:** (Kießling, Köstler; 1997)
  - Extension of Standard SQL by a preferring-clause

- **Preference XPATH:** (Kießling, Hafenrichter, Fischer, Holland; 2001)
  Implementation of proposed preference model for XML databases.
Related Work (cont’d)

- **Sample Preference SQL query:** [See my talk on Preference SQL, Session Industry 6]

  ```sql
  SELECT * FROM used_cars WHERE make = 'Opel'
  PREFERING (category = 'cabriolet' ELSE category <> 'roadster')
  AND price AROUND 40000 AND HIGHEST(power)
  AND mileage BETWEEN 20000, 30000;
  ```

- **Sample Preference XPATH query:**

  ```xpath
  /CARS/CAR [ (@fuel_economy) highest and (@mileage) lowest
  prior to (@color) in ("black", "white") and
  (@price) around 10000
  ]#
  ```
Foundations of Preferences in Database Systems

Related Work (cont’d)

- **The SKYLINE operator:** (Borzsonyi, Kossmann, Stocker; 2001)
  
  Restricted form of Pareto queries \( P = P_1 \otimes P_2 \otimes \ldots \otimes P_n \), each \( P_i \) must be a **LOWEST** or **HIGHEST** preference (hence a chain).

- **The ranked query model (“top-k”):**
  
  - SQL/MM multi-feature queries, SQL text extenders / cartridges
  - XML full-text queries (Theobald, Weikum; 2000)
  
    \[ P = \text{rank}_F(\text{SCORE}(A_1, f_1), \ldots, \text{SCORE}(A_n, f_n)) \]

- **The framework of Chomicki:** (2002)
  
  Preferences as strict partial orders; & preference constructor; “winnow” operator
7. Summary and Ongoing Work

Foundations of preferences in database systems:

- “Foundation matters“.
- This framework unifies and extends existing approaches.
- Various implementations exist. (Performance was NOT the focus here.)

Highlights:

- Preferences as strict partial orders enable an intuitive semantics, covering qualitative as well as numerical methods.
- An extensible repertoire of powerful preference constructors serves a wide spectrum of database applications.
- A systematic approach towards preference engineering is enabled.
- Preferences as soft constraints and the BMO query model.
- Preference algebra and decomposition laws.
"It’s a Preference World"

Preference research at the Univ. of Augsburg:

- Automation of B2B sales processes: P-Bargainer, P-Agent
- Preference J2EE application server: P-Services
- Preference engineering: P-News
- Preference maintenance: P-Miner, P-Repository
- Preference query languages: Preference XPATH
- Preference performance: P-Optimizer, SR-Combine

Funding: DFG, FORSIP (www.forsip.de)

Forsip

[Bavarian Research Cooperation for Situated, Individualized and Personalized Man-Machine Interaction]