How to (repeatedly) change preferences

Jan Chomicki
University at Buffalo

* FOIKS’06, AMAI
Preference relations

- Binary relations between tuples
- Abstract way to capture a variety of criteria: desirability, relative value, quality, timeliness…
- More general than numeric scoring functions

<table>
<thead>
<tr>
<th>Make</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>VW</td>
<td>2002</td>
</tr>
<tr>
<td>VW</td>
<td>1998</td>
</tr>
<tr>
<td>Kia</td>
<td>1998</td>
</tr>
</tbody>
</table>

within each make, prefer more recent cars
Preference queries

- **Winnow**: In a given table, find the best elements according to a given preference relation.

<table>
<thead>
<tr>
<th>Make</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>VW</td>
<td>2002</td>
</tr>
<tr>
<td>VW</td>
<td>1998</td>
</tr>
<tr>
<td>Kia</td>
<td>1998</td>
</tr>
</tbody>
</table>

within each make, prefer a more recent car

Too many results…
Query modification via preference revision

<table>
<thead>
<tr>
<th>Make</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>VW</td>
<td>2002</td>
</tr>
<tr>
<td>VW</td>
<td>1998</td>
</tr>
<tr>
<td>Kia</td>
<td>1998</td>
</tr>
</tbody>
</table>

within each make, prefer a more recent car among cars of the same production year, prefer VW

- Objectives:
  - Preference composition operators
  - Minimal change to preferences
  - Preservation of order properties
Overview

- Preference representation
- Order axioms
- Preference revision
- Incremental evaluation of preference queries
- Related work
- Conclusions and future work
Preference relations

Preference relation
- binary relation (possibly infinite)
- represented by a quantifier-free first-order formula

within each make, prefer more recent cars:

\[(m, y) \succ (m', y') \equiv (m = m' \land y > y')\]

Winnow operator

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

Used to select the best tuples
Order axioms ORD

- **Strict Partial Order (SPO) = transitivity + irreflexivity**
  - Preference SQL
  - winnow is nonempty
  - efficient algorithms for winnow (BNL,…)
  - incremental query evaluation

- **Weak Order (WO) = SPO + negative transitivity:**
  \[
  \forall x, y, z. (x \not\succ y \land y \not\succ z) \rightarrow x \not\succ z
  \]
  - often representable with a utility function
  - single pass winnow evaluation
Composing preference relations

**Union**

\[ t (\succ_1 \cup \succ_2) s \iff t \succ_1 s \lor t \succ_2 s \]

**Prioritized composition**

\[ t (\succ_1 \triangleright \succ_2) s \iff t \succ_1 s \land (s \not\succ_1 t \land t \succ_2 s) \]

**Pareto composition**

\[ t (\succ_1 \otimes \succ_2) s \iff (s \not\succ_2 t \land t \succ_1 s) \lor (s \not\succ_1 t \land t \succ_2 s) \]

**Transitive closure**

\[ (t,s) \in \text{TC}(\succ) \iff t \succ^n s \text{ for some } n > 0 \]
Preference revisions

Preference relation $\succ$
Revising pref. relation $\succ_0$
Composition operator $\theta$
Order axioms ORD
$\succ$ and $\succ_0$ satisfy ORD

ORD $\theta$-revision of $\succ$ with $\succ_0$

Preference relation $\succ'$:
- minimally different from $\succ$
- contains $\succ_0 \theta \succ$
- satisfies ORD
Conflicts and SPO revisions

0-conflict

1-conflict

2-conflict

solved by ▷

solved by ⊗

no SPO θ-revision
0-conflict

1-conflict

2-conflict

∪

B

⊗
Is lack of conflict sufficient?

Interval Order (IO) = SPO + \( \forall x,y,z,w. (x \triangleright y \land z \triangleright w) \rightarrow (x \triangleright w \lor z \triangleright y) \)

- \( \triangleright, \triangleright_0 \) satisfy SPO
  - no 0-conflicts
  - \( \triangleright \) or \( \triangleright_0 \) is IO

- \( \triangleright, \triangleright_0 \) satisfy SPO
  - no 1-conflicts
  - \( \triangleright_0 \) is IO

\[ \triangleright' = \text{TC}(\triangleright \cup \triangleright_0) \text{ is an SPO} \cup \text{-revision} \]

\[ \triangleright' = \text{TC}(\triangleright_0 \triangleright) \text{ is an SPO} \triangleright \text{-revision} \]

No conflicts

However, no SPO revision!
within each make, prefer more recent cars:
\[(m,y) \succ (m',y') \equiv (m = m' \land y > y')\]

among cars produced in 1999, prefer VW:
\[(m,y) \succ_0 (m',y') \equiv m = vw \land m' \neq vw \land y = y' = 1999\]

\[TC(\succ_0 \cup \succ)\]

\[(m,y) \succ' (m',y') \equiv m = m' \land y > y' \lor m = vw \land m' \neq vw \land y \geq 1999 \land y' \leq 1999\]
WO revisions and utility functions

1. $\succ, \succ_0$ satisfy WO no 0-conflicts

2. $\succ'$ = $\succ_0 \cup \succ$ is a WO $\cup$-revision

3. $\succ'$ may be not representable with a utility function

4. $\succ$ represented with $u(x)$
   $\succ_0$ represented with $u_0(x)$

5. $u'(x) = a \cdot u(x) + b \cdot u_0(x) + c$
   $a, b > 0$
Incremental evaluation: preference revision
\(\succ\) : within each make, prefer more recent cars

\(\succ_0\) : among cars produced in 1999, prefer VW

\(\omega_{\succ}\)

\(\omega_{TC(\succ \cup \succ_0)}\)
Incremental evaluation: tuple insertion
Preference vs. belief revision

Preference revision
- First-order
- Revising a single, finitely representable relation
- Preserving order axioms

Belief revision
- Propositional
- Revising a theory
- Axiomatic properties of BR operators
Related work

  - preferences = sets of ground formulas
  - preference revision ∼ belief revision
  - no focus on construction of revisions, SPO/WO preservation
  - preference contraction, domain expansion/shrinking

  - revising finite ranking with new information
  - new ranking can be computed in a simple way

  - revision and contraction of finite WO preferences with single pairs $t \succ_0 s$
Summary and future work

Summary:

- Preference query modification through preference revision
- Preference revision using composition
- Closure of SPO and WO under revisions
- Incremental evaluation of preference queries

Future work:

- Integrating with relational query evaluation and optimization
- General revision language
- Preference contraction (query result too small)
- Preference elicitation