Profiling Sets for Preference Querying

Xi Zhang    Jan Chomicki

SUNY at Buffalo

April 24, 2008
1 Motivating Example

2 Profile-based Set Preferences

3 Computing the “Best” Sets

4 Future Work
1 Motivating Example

2 Profile-based Set Preferences

3 Computing the “Best” Sets

4 Future Work
Motivating Example

Alice is buying 3 books as gifts.

<table>
<thead>
<tr>
<th>Title</th>
<th>Genre</th>
<th>Rating</th>
<th>Price</th>
<th>Vendor</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_1$</td>
<td>sci-fi</td>
<td>5.0</td>
<td>$15.00</td>
<td>Amazon</td>
</tr>
<tr>
<td>$a_2$</td>
<td>biography</td>
<td>4.8</td>
<td>$20.00</td>
<td>B&amp;N</td>
</tr>
<tr>
<td>$a_3$</td>
<td>sci-fi</td>
<td>4.5</td>
<td>$25.00</td>
<td>Amazon</td>
</tr>
<tr>
<td>$a_4$</td>
<td>romance</td>
<td>4.4</td>
<td>$10.00</td>
<td>B&amp;N</td>
</tr>
<tr>
<td>$a_5$</td>
<td>sci-fi</td>
<td>4.3</td>
<td>$15.00</td>
<td>Amazon</td>
</tr>
<tr>
<td>$a_6$</td>
<td>romance</td>
<td>4.2</td>
<td>$12.00</td>
<td>B&amp;N</td>
</tr>
<tr>
<td>$a_7$</td>
<td>biography</td>
<td>4.0</td>
<td>$18.00</td>
<td>Amazon</td>
</tr>
<tr>
<td>$a_8$</td>
<td>sci-fi</td>
<td>3.5</td>
<td>$18.00</td>
<td>Amazon</td>
</tr>
</tbody>
</table>
Alice is buying 3 books as gifts. She has the following preferences...

- (C1) Spend as little money as possible.
- (C2) Get one sci-fi book.
- (C3) Use as few vendors as possible.
- (C0) Prioritize (C2) over (C1)

<table>
<thead>
<tr>
<th>Title</th>
<th>Genre</th>
<th>Rating</th>
<th>Price</th>
<th>Vendor</th>
</tr>
</thead>
<tbody>
<tr>
<td>a₁</td>
<td>sci-fi</td>
<td>5.0</td>
<td>$15.00</td>
<td>Amazon</td>
</tr>
<tr>
<td>a₂</td>
<td>biography</td>
<td>4.8</td>
<td>$20.00</td>
<td>B&amp;N</td>
</tr>
<tr>
<td>a₃</td>
<td>sci-fi</td>
<td>4.5</td>
<td>$25.00</td>
<td>Amazon</td>
</tr>
<tr>
<td>a₄</td>
<td>romance</td>
<td>4.4</td>
<td>$10.00</td>
<td>B&amp;N</td>
</tr>
<tr>
<td>a₅</td>
<td>sci-fi</td>
<td>4.3</td>
<td>$15.00</td>
<td>Amazon</td>
</tr>
<tr>
<td>a₆</td>
<td>romance</td>
<td>4.2</td>
<td>$12.00</td>
<td>B&amp;N</td>
</tr>
<tr>
<td>a₇</td>
<td>biography</td>
<td>4.0</td>
<td>$18.00</td>
<td>Amazon</td>
</tr>
<tr>
<td>a₈</td>
<td>sci-fi</td>
<td>3.5</td>
<td>$18.00</td>
<td>Amazon</td>
</tr>
</tbody>
</table>
Motivating Example

Alice is buying 3 books as gifts.

<table>
<thead>
<tr>
<th>Title</th>
<th>Genre</th>
<th>Rating</th>
<th>Price</th>
<th>Vendor</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_1$</td>
<td>sci-fi</td>
<td>5.0</td>
<td>$15.00</td>
<td>Amazon</td>
</tr>
<tr>
<td>$a_2$</td>
<td>biography</td>
<td>4.8</td>
<td>$20.00</td>
<td>B&amp;N</td>
</tr>
<tr>
<td>$a_3$</td>
<td>sci-fi</td>
<td>4.5</td>
<td>$25.00</td>
<td>Amazon</td>
</tr>
<tr>
<td>$a_4$</td>
<td>romance</td>
<td>4.4</td>
<td>$10.00</td>
<td>B&amp;N</td>
</tr>
<tr>
<td>$a_5$</td>
<td>sci-fi</td>
<td>4.3</td>
<td>$15.00</td>
<td>Amazon</td>
</tr>
<tr>
<td>$a_6$</td>
<td>romance</td>
<td>4.2</td>
<td>$12.00</td>
<td>B&amp;N</td>
</tr>
<tr>
<td>$a_7$</td>
<td>biography</td>
<td>4.0</td>
<td>$18.00</td>
<td>Amazon</td>
</tr>
<tr>
<td>$a_8$</td>
<td>sci-fi</td>
<td>3.5</td>
<td>$18.00</td>
<td>Amazon</td>
</tr>
</tbody>
</table>

She has the following preferences...

- (C1) Spend as little money as possible. (the cheapest 3 books)
- (C2) Get one sci-fi book.
- (C3) Use as few vendors as possible.
- (C0) Prioritize (C2) over (C1)
Challenge

How to handle preferences over sets of homogeneous objects?

- *Homogeneous*: a collection of books, a set of faculty candidates
- *Heterogeneous*: a travel package
Challenge

How to handle preferences over sets of homogeneous objects?

- *Homogeneous*: a collection of books, a set of faculty candidates
- *Heterogeneous*: a travel package

Basic idea [Binshtok et al., AAAI’07]:

- Capture set preference as tuple preference
Outline

1. Motivating Example
2. Profile-based Set Preferences
3. Computing the “Best” Sets
4. Future Work
**Set Preferences**

**k-subsets**: \(k\)-element subsets of a given relation \(r\)

<table>
<thead>
<tr>
<th>Set Pref.</th>
<th>Quantity of Interest</th>
<th>Desired Value or Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>(C1)</td>
<td>total cost</td>
<td>&lt;</td>
</tr>
<tr>
<td>(C2)</td>
<td># of sci-fi books</td>
<td>1</td>
</tr>
<tr>
<td>(C3)</td>
<td># of distinct vendors</td>
<td>&lt;</td>
</tr>
</tbody>
</table>
**Set Preferences**

**k-subsets:** $k$-element subsets of a given relation $r$

**Simple Set Preference**

<table>
<thead>
<tr>
<th>Set Pref.</th>
<th>Quantity of Interest</th>
<th>Desired Value or Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>(C1)</td>
<td>total cost</td>
<td>$&lt;$</td>
</tr>
<tr>
<td>(C2)</td>
<td># of sci-fi books</td>
<td>1</td>
</tr>
<tr>
<td>(C3)</td>
<td># of distinct vendors</td>
<td>$&lt;$</td>
</tr>
</tbody>
</table>

**Complex Set Preference (C0)**

(C2) is more important than (C1):

prioritized composition of (C2) and (C1)

i.e. $(C2) \succ (C1)$
<table>
<thead>
<tr>
<th>Set Pref.</th>
<th>Quantities of Interest</th>
<th>Desired Value or Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>(C1)</td>
<td>total cost</td>
<td>&lt;</td>
</tr>
<tr>
<td>(C2)</td>
<td># of sci-fi books</td>
<td>1</td>
</tr>
<tr>
<td>(C3)</td>
<td># of distinct vendors</td>
<td>&lt;</td>
</tr>
<tr>
<td>(C0)</td>
<td>total cost, # of sci-fi books</td>
<td>(C2)&gt;(C1)</td>
</tr>
</tbody>
</table>
## Preferences over Profiles

<table>
<thead>
<tr>
<th>Set Pref.</th>
<th>Quantities of Interest</th>
<th>Desired Value or Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>(C1)</td>
<td>total cost</td>
<td>&lt;</td>
</tr>
<tr>
<td>(C2)</td>
<td># of sci-fi books</td>
<td>1</td>
</tr>
<tr>
<td>(C3)</td>
<td># of distinct vendors</td>
<td>&lt;</td>
</tr>
<tr>
<td>(C0)</td>
<td>total cost, # of sci-fi books</td>
<td>(C2) &gt; (C1)</td>
</tr>
</tbody>
</table>

features
### Preferences over Profiles

<table>
<thead>
<tr>
<th>Set Pref.</th>
<th>Quantities of Interest</th>
<th>Desired Value or Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>(C1)</td>
<td>total cost</td>
<td>&lt;</td>
</tr>
<tr>
<td>(C2)</td>
<td># of sci-fi books</td>
<td>1</td>
</tr>
<tr>
<td>(C3)</td>
<td># of distinct vendors</td>
<td>&lt;</td>
</tr>
<tr>
<td>(C0)</td>
<td>total cost, # of sci-fi books</td>
<td>(C2)(\succ)(C1)</td>
</tr>
</tbody>
</table>

#### Example:

\[
\text{profile} = \langle f_1, f_2, \ldots, f_m \rangle
\]

**features**

**preferences over profiles**
Definition (SQL-based $k$-subset Feature)

Parameterized feature

$$A \equiv \text{SELECT expr FROM } S \text{ WHERE condition}$$

where $S$ is a set variable over any $k$-subset of relation $r$.

**Requirement**: the query is *categorical*. 

---

Xi Zhang, Jan Chomicki (SUNY at Buffalo)  
Profiling Sets for Pref. Querying  
April 24, 2008  9 / 20
Definition (SQL-based \( k \)-subset Feature)

Parameterized feature

\[
\mathcal{A} \equiv \text{SELECT} \ expr \ \text{FROM} \ S \ \text{WHERE} \ \text{condition}
\]

where \( S \) is a set variable over any \( k \)-subset of relation \( r \).

**Requirement**: the query is *categorical*.

Definition (Profile)

\[
\text{profile}(s) = \langle \mathcal{A}_1(s), \ldots, \mathcal{A}_m(s) \rangle
\]

where \( s \) is any \( k \)-subset of relation \( r \).
### Example

<table>
<thead>
<tr>
<th>Set Pref.</th>
<th>Quantities of Interest</th>
<th>Desired Value or Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>(C1)</td>
<td>total cost</td>
<td>&lt;</td>
</tr>
<tr>
<td>(C2)</td>
<td># of sci-fi books</td>
<td>1</td>
</tr>
<tr>
<td>(C3)</td>
<td># of distinct vendors</td>
<td>&lt;</td>
</tr>
<tr>
<td>(C0)</td>
<td>total cost, # of sci-fi books</td>
<td>(C2)&gt;(C1)</td>
</tr>
</tbody>
</table>

\[ A_1 \equiv \text{SELECT sum(price) FROM S} \]
### Example

<table>
<thead>
<tr>
<th>Set Pref.</th>
<th>Quantities of Interest</th>
<th>Desired Value or Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>(C1)</td>
<td>total cost</td>
<td>&lt;</td>
</tr>
<tr>
<td>(C2)</td>
<td># of sci-fi books</td>
<td>1</td>
</tr>
<tr>
<td>(C3)</td>
<td># of distinct vendors</td>
<td>&lt;</td>
</tr>
<tr>
<td>(C0)</td>
<td>total cost, # of sci-fi books</td>
<td>(C2) ⪰ (C1)</td>
</tr>
</tbody>
</table>

$$A_1 \equiv \text{SELECT } \text{sum(price)} \text{ FROM } S$$

$$A_2 \equiv \text{SELECT } \text{count(title)} \text{ FROM } S \text{ WHERE } \text{genre=’sci-fi’}$$
### Example

<table>
<thead>
<tr>
<th>Set Pref.</th>
<th>Quantities of Interest</th>
<th>Desired Value or Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>(C1)</td>
<td>total cost</td>
<td>&lt;</td>
</tr>
<tr>
<td>(C2)</td>
<td># of sci-fi books</td>
<td>1</td>
</tr>
<tr>
<td>(C3)</td>
<td># of distinct vendors</td>
<td>&lt;</td>
</tr>
<tr>
<td>(C0)</td>
<td>total cost, # of sci-fi books</td>
<td>(C2)&gt;(C1)</td>
</tr>
</tbody>
</table>

\[
A_1 \equiv \text{SELECT sum(price) FROM S}
\]

\[
A_2 \equiv \text{SELECT count(title) FROM S WHERE genre='sci-fi'}
\]

\[
A_3 \equiv \text{SELECT count(DISTINCT vendor) FROM S}
\]
Definition (Tuple Preference)

Given a relation schema $R = \langle A_1, \ldots, A_m \rangle$, a tuple preference is defined by a first order formula $C$ if

$$C(t_1, t_2) \iff t_1 >_C t_2$$
Definition (Tuple Preference)

Given a relation schema \( R = \langle A_1, \ldots, A_m \rangle \), a tuple preference is defined by a \textit{first order formula} \( C \) if

\[
C(t_1, t_2) \iff t_1 >_C t_2
\]

Definition (Winnow Operator)

Winnow operator \( \omega_C(R) \) is defined by tuple preference \( >_C \) if for every instance \( r \) of \( R \),

\[
\omega_C(r) = \{ t \in r | \neg \exists t' \in r. t' >_C t \}
\]
Set preferences are defined by tuple preferences over profiles

<table>
<thead>
<tr>
<th>Set Pref.</th>
<th>Quantities of Interest</th>
<th>Desired Value or Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>(C1)</td>
<td>total cost</td>
<td>&lt;</td>
</tr>
<tr>
<td>(C2)</td>
<td># of sci-fi books</td>
<td>1</td>
</tr>
<tr>
<td>(C0)</td>
<td>total cost, # of sci-fi books</td>
<td>(C2)&gt;(C1)</td>
</tr>
</tbody>
</table>

\[ A_1 \equiv \text{SELECT} \ \text{sum}(\text{price}) \ \text{FROM} \ S \]
\[ A_2 \equiv \text{SELECT} \ \text{count}(\text{title}) \ \text{FROM} \ S \ \text{WHERE} \ \text{genre}=\text{’sci-fi’} \]
Set preferences are defined by *tuple* preferences over *profiles*

<table>
<thead>
<tr>
<th>Set Pref.</th>
<th>Quantities of Interest</th>
<th>Desired Value or Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>(C1)</td>
<td>total cost</td>
<td>&lt;</td>
</tr>
<tr>
<td>(C2)</td>
<td># of sci-fi books</td>
<td>1</td>
</tr>
<tr>
<td>(C0)</td>
<td>total cost, # of sci-fi books</td>
<td>(C2)&gt;(C1)</td>
</tr>
</tbody>
</table>

\[ A_1 \equiv \text{SELECT } \sum \text{price} \text{ FROM } S \]
\[ A_2 \equiv \text{SELECT } \text{count(title)} \text{ FROM } S \text{ WHERE genre='sci-fi'} \]

\[ s_1 \gg_{C_1} s_2 \iff A_1(s_1) < A_1(s_2). \]
Example

Set preferences are defined by *tuple* preferences over *profiles*

<table>
<thead>
<tr>
<th>Set Pref.</th>
<th>Quantities of Interest</th>
<th>Desired Value or Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>(C1)</td>
<td>total cost</td>
<td>&lt;</td>
</tr>
<tr>
<td>(C2)</td>
<td># of sci-fi books</td>
<td>1</td>
</tr>
<tr>
<td>(C0)</td>
<td>total cost, # of sci-fi books</td>
<td>(C2)&gt;(C1)</td>
</tr>
</tbody>
</table>

\[
\mathcal{A}_1 \equiv \text{SELECT } \text{sum(price)} \text{ FROM S}
\]

\[
\mathcal{A}_2 \equiv \text{SELECT } \text{count(title)} \text{ FROM S WHERE genre='sci-fi'}
\]

\[
s_1 \succ_{C1} s_2 \iff \mathcal{A}_1(s_1) < \mathcal{A}_1(s_2).
\]

\[
s_1 \succ_{C2} s_2 \iff \mathcal{A}_2(s_1) = 1 \land \mathcal{A}_2(s_2) \neq 1.
\]
Set preferences are defined by *tuple* preferences over *profiles*

<table>
<thead>
<tr>
<th>Set Pref.</th>
<th>Quantities of Interest</th>
<th>Desired Value or Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>(C1)</td>
<td>total cost</td>
<td>&lt;</td>
</tr>
<tr>
<td>(C2)</td>
<td># of sci-fi books</td>
<td>1</td>
</tr>
<tr>
<td>(C0)</td>
<td>total cost, # of sci-fi books</td>
<td>(C2)≻(C1)</td>
</tr>
</tbody>
</table>

\[ A_1 \equiv \text{SELECT } \text{sum(price)} \text{ FROM } S \]
\[ A_2 \equiv \text{SELECT } \text{count(title)} \text{ FROM } S \text{ WHERE } \text{genre='sci-fi'} \]

\[ s_1 \succ C_1 s_2 \iff A_1(s_1) < A_1(s_2). \]
\[ s_1 \succ C_2 s_2 \iff A_2(s_1) = 1 \land A_2(s_2) \neq 1. \]
\[ s_1 \succ C_0 s_2 \iff (A_2(s_1) = 1 \land A_2(s_2) \neq 1) \]
\[ \lor (A_2(s_1) \neq 1 \land A_2(s_2) \neq 1 \land A_1(s_1) < A_1(s_2)) \]
\[ \lor (A_2(s_1) \neq 1 \land A_2(s_2) \neq 1 \land A_1(s_1) < A_1(s_2)). \]
Outline

1. Motivating Example
2. Profile-based Set Preferences
3. Computing the “Best” Sets
4. Future Work
Naive Algorithm

- Generate all $k$-subsets of relation $r$ and compute the set $\gamma$ of their profiles.
- Run the winnow operator over all the profiles and get the “best” profiles
  \[ \gamma' = \omega_C(\gamma). \]
- Get all subsets corresponding to the “best” profiles in $\gamma'$. 
- Generate all $k$-subsets of relation $r$ and compute the set $\gamma$ of their profiles.
- Run the winnow operator over all the profiles and get the “best” profiles
  \[ \gamma' = \omega_c(\gamma). \]
- Get all subsets corresponding to the “best” profiles in $\gamma'$.

Too many $k$-subsets!
“Superpreference”

Find a “superpreference” ($>^+$) over the relation $r$, such that

$$t_1 >^+ t_2 \iff s' \cup \{t_1\} \succ_{C} s' \cup \{t_2\}.$$ 

where $s'$ is any $(k-1)$-subset that contains neither $t_1$ nor $t_2$. 
### Conditions for Early Pruning

#### “Superpreference”

Find a “superpreference” \((>^+)\) over the relation \(r\), such that

\[
t_1 >^+ t_2 \Leftrightarrow s' \cup \{t_1\} \gg_C s' \cup \{t_2\}.
\]

where \(s'\) is any \((k-1)\)-subset that contains neither \(t_1\) nor \(t_2\).

#### Additive \(k\)-subset Feature \(A\)

- \(A\) is well-defined for the domain of \((k-1)\)-subsets\((r)\), and
- for any subset \(s' \in (k-1)\)-subsets\((r)\), and any \(t \in r \land t \notin s'\),

\[
A(s' \cup \{t\}) = A(s') + f(t)
\]

where \(f\) is a function of \(t\) only.
Pruning Theorem

Theorem

If the profile preference formula $C$ can be rewritten as a DNF formula

$$\bigvee_{i=1}^{n} \bigwedge_{j=1}^{m} (A_{ij}(s_1) \theta A_{ij}(s_2))$$

where $\theta \in \{=, \neq, <, >, \leq, \geq\}$ and $A_{ij}$ is an additive $k$-subset feature, then superpreference $C^+$ exists and can be constructed systematically.
Example - “Superpreference”

*Set preference: (C5) ∩ (C6)*

(C5) Alice wants to spend as little money as possible on sci-fi books.

(C6) Alice wants the average rating of books to be as high as possible.
Example - “Superpreference”

Set preference: \((C5) \cap (C6)\)

(C5) Alice wants to spend as little money as possible on sci-fi books.
(C6) Alice wants the average rating of books to be as high as possible.

Features
\[ A_5 \equiv \text{SELECT } \text{sum(price)} \text{ FROM } S \text{ WHERE genre='sci-fi'} \]
\[ A_6 \equiv \text{SELECT } \text{avg(rating)} \text{ FROM } S \]
Example - “Superpreference”

Set preference: \((C5) \cap (C6)\)

(C5) Alice wants to spend as little money as possible on sci-fi books.
(C6) Alice wants the average rating of books to be as high as possible.

Features
\[ A_5 \equiv \text{SELECT } \text{sum(price)} \text{ FROM } S \text{ WHERE genre='sci-fi'} \]
\[ A_6 \equiv \text{SELECT } \text{avg(rating)} \text{ FROM } S \]

Profile preference

\[ s_1 \gg_C s_2 \equiv A_5(s_1) < A_5(s_2) \land A_6(s_1) > A_6(s_2) \]
Example - “Superpreference”

Set preference: (C5) ∩ (C6)
(C5) Alice wants to spend as little money as possible on sci-fi books.
(C6) Alice wants the average rating of books to be as high as possible.

Features
\[ A_5 \equiv \text{SELECT } \text{sum(price)} \text{ FROM } S \text{ WHERE genre='sci-fi'} \]
\[ A_6 \equiv \text{SELECT } \text{avg(rating)} \text{ FROM } S \]

Profile preference

\[ s_1 \succ_C s_2 \equiv A_5(s_1) < A_5(s_2) \land A_6(s_1) > A_6(s_2) \]

“Superpreference” formula \( C^+ \) (assuming \( \text{price} > 0 \))

\[ t_1 >_{C^+} t_2 \equiv t_1.rating > t_2.rating \land t_2.genre = 'sci-fi' \land (t_1.price < t_2.price \lor t_1.genre \neq 'sci-fi'). \]
Heuristic Algorithm

- Initialize $r' = \emptyset$.
- Find the “most promising” tuples based on superpreference $C^+$
  
  \[
  \text{repeat } r' := r' \cup \omega_{C^+}(r - r') \text{ until } |r'| \geq k.
  \]

- Generate all $k$-subsets of relation $r'$ and compute the set $\gamma$ of their profiles.
- Run the winnow operator over all the profiles and get the “best” profiles
  
  \[
  \gamma' = \omega_C(\gamma).
  \]
- Get all subsets corresponding to the “best” profiles in $\gamma'$. 
1 Motivating Example

2 Profile-based Set Preferences

3 Computing the “Best” Sets

4 Future Work
Future Work

- Expressive power
- Preference query optimization techniques
- Query categoricity
- Integrity constraints
- Finding interesting sets