Profiling Sets for Preference Querying

Xi Zhang and Jan Chomicki

SUNY Buffalo

May 18, 2010
Outline

1. Motivation
2. Profile-based Set Preferences
3. Computing the “Best” Sets
4. Experiments
5. Future Work
Outline

1 Motivation

2 Profile-based Set Preferences

3 Computing the “Best” Sets

4 Experiments

5 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>

She has the following wishes...

(C2) Get one sci-fi book.
(C3) Prioritize (C2) over (C1)
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 wishes...

- (C1) Spend as little money as possible.
- (C2) Get one sci-fi book.
- (C3) Prioritize (C2) over (C1)
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>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

She has the following wishes...

- (C1) Spend as little money as possible. (the cheapest 3 books)
- (C2) Get one sci-fi book.
- (C3) Prioritize (C2) over (C1)
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 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 $R$,

$$\omega_C(r) = \{ t \in r | \neg \exists t' \in r. t' >_C t \}$$
1 Motivation

2 Profile-based Set Preferences

3 Computing the “Best” Sets

4 Experiments

5 Future Work
## Profile-based Set Preference

<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>total cost, # of sci-fi books</td>
<td>(C2) ( \succ ) (C1)</td>
</tr>
</tbody>
</table>
Profile-based Set Preference

<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>total cost, # of sci-fi books</td>
<td>(C2)&gt;(C1)</td>
</tr>
</tbody>
</table>

features
Profile-based Set Preference

<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>total cost, # of sci-fi books</td>
<td>(C2) &gt; (C1)</td>
</tr>
</tbody>
</table>

features over profiles

profile = \langle f_1, f_2, \ldots, f_m \rangle
$k$-subsets

subsets of relation $r$, with fixed cardinality $k$
Profile-based Set Preferences

- **$k$-subsets**
  subsets of relation $r$, with fixed cardinality $k$

- **SQL-based $k$-subset feature**
  A restricted “mini” SQL query over a $k$-subset which returns a scalar value
- **k-subsets**
  subsets of relation $r$, with fixed cardinality $k$

- **SQL-based k-subset feature**
  A restricted “mini” SQL query over a $k$-subset which returns a scalar value

- **profile**
  A vector of $k$-subset features
- **k-subsets**
  subsets of relation \( r \), with *fixed* cardinality \( k \)

- **SQL-based k-subset feature**
  A restricted “mini” SQL query over a \( k \)-subset which returns a scalar value

- **profile**
  A vector of \( k \)-subset features

Given \( k \)-subset feature \( \mathcal{F}_1, \ldots, \mathcal{F}_m \) defined over \( k \)-subsets of relation \( r \)

**Profile-based Set Preference**

A tuple preference over profiles of all \( k \)-subsets of relation \( r \).
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>(C3)</td>
<td>total cost, # of sci-fi books</td>
<td>(C2)≻(C1)</td>
</tr>
</tbody>
</table>

\[
\mathcal{F}_1 \equiv \text{SELECT sum(price) FROM } S \\
\mathcal{F}_2 \equiv \text{SELECT count(title) FROM } S \text{ WHERE genre='sci-fi'}
\]
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>(C3)</td>
<td>total cost, # of sci-fi books</td>
<td>(C2)▷(C1)</td>
</tr>
</tbody>
</table>

\[
\mathcal{F}_1 \equiv \text{SELECT } \text{sum(price)} \text{ FROM } $S \\
\mathcal{F}_2 \equiv \text{SELECT } \text{count(title)} \text{ FROM } $S \text{ WHERE genre='sci-fi'} \\
\]

$s_1 \succ_C s_2 \iff \mathcal{F}_1(s_1) < \mathcal{F}_1(s_2)$. 
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>(C3)</td>
<td>total cost, # of sci-fi books</td>
<td>(C2)⊃(C1)</td>
</tr>
</tbody>
</table>

\[ F_1 \equiv \text{SELECT sum(price) FROM } S \]
\[ F_2 \equiv \text{SELECT count(title) FROM } S \text{ WHERE genre='sci-fi'} \]

\[ s_1 \succ_c s_2 \iff F_1(s_1) < F_1(s_2). \]
\[ s_1 \succ_c s_2 \iff F_2(s_1) = 1 \land F_2(s_2) \neq 1. \]
**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>(C3)</td>
<td>total cost, # of sci-fi books</td>
<td>(C2) ≻ (C1)</td>
</tr>
</tbody>
</table>

\[F_1 \equiv \text{SELECT sum(price) FROM } S\]

\[F_2 \equiv \text{SELECT count(title) FROM } S \text{ WHERE genre='sci-fi'}\]

\[s_1 \gg_{C_1} s_2 \iff F_1(s_1) < F_1(s_2).\]

\[s_1 \gg_{C_2} s_2 \iff F_2(s_1) = 1 \land F_2(s_2) \neq 1.\]

\[s_1 \gg_{C_3} s_2 \iff (F_2(s_1) = 1 \land F_2(s_2) \neq 1)\]

\[
\begin{align*}
&\vee (F_2(s_1) = 1 \land F_2(s_2) = 1 \land F_1(s_1) < F_1(s_2)) \\
&\vee (F_2(s_1) \neq 1 \land F_2(s_2) \neq 1 \land F_1(s_1) < F_1(s_2)).
\end{align*}
\]
Outline

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

- Generate all $k$-subsets of relation $r$ and compute their profiles.
- Run the winnow operator over all the profiles and get the “best” profiles.
Naive Algorithm

- Generate all $k$-subsets of relation $r$ and compute their profiles.
- Run the winnow operator over all the profiles and get the “best” profiles.

Too many candidate $k$-subsets!
Superpreference Algorithm

Goal

Generate as few candidate $k$-subsets as possible

"Superpreference"

Find a "superpreference" ($\preceq$) over the relation $r$, such that $t_1 \preceq t_2$ if $s_1 \cap t_1 \subseteq s_1 \cap t_2$ for every $(k-1)$-subset $s_1$ of $r$ containing neither $t_1$ nor $t_2$.

Pruning Condition

Let $p(t_1, t_2)$ be $t_1 \preceq t_2$.

For every $t$ in the "best" $k$-subset, cover $p(t, q)$.
Superpreference Algorithm

Goal

Generate as few candidate $k$-subsets as possible

“Superpreference”

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

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

for every (k-1)-subset $s'$ of $r$ containing neither $t_1$ nor $t_2$. 

Superpreference Algorithm

Goal
Generate as few candidate $k$-subsets as possible

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

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

for every $(k-1)$-subset $s'$ of $r$ containing neither $t_1$ nor $t_2$.

Pruning Condition
Let $cover(t) = \{t' > t | t' >^+ t\}$.
For every $t$ in the “best” $k$-subset, $cover(t) < k$. 

Theorem ("Superpreference" Construction)

If the set preference contains

- additive $k$-subset features only, and
- can be rewritten as a constant-free DNF formula

then the superpreference $>^+$ can be defined by a first-order formula which is independent of $k$. 
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.
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
\[ F_5 \equiv \text{SELECT sum(price) FROM } S \text{ WHERE genre='sci-fi'} \]
\[ F_6 \equiv \text{SELECT avg(rating) FROM } S \]
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
\(\mathcal{F}_5 \equiv \text{SELECT } \text{sum(price)} \text{ FROM } S \text{ WHERE genre='sci-fi'}\)
\(\mathcal{F}_6 \equiv \text{SELECT } \text{avg(rating)} \text{ FROM } S\)

Profile preference

\[ s_1 \succ_C s_2 \equiv \mathcal{F}_5(s_1) < \mathcal{F}_5(s_2) \land \mathcal{F}_6(s_1) > \mathcal{F}_6(s_2) \]
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
\[ F_5 \equiv \text{SELECT } \text{sum(price)} \text{ FROM } $S \text{ WHERE } \text{genre='sci-fi'} \]
\[ F_6 \equiv \text{SELECT } \text{avg(rating)} \text{ FROM } $S \]

Profile preference

\[ s_1 \succ_C s_2 \equiv F_5(s_1) < F_5(s_2) \land F_6(s_1) > F_6(s_2) \]

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

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

Avoid redundancy in generating candidate $k$-subsets
**Goal**

Avoid redundancy in generating candidate \( k \)-subsets

**Book:**

<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>
<tr>
<td>( a_9 )</td>
<td>romance</td>
<td>4.0</td>
<td>$20.00</td>
<td>Amazon</td>
</tr>
<tr>
<td>( a_{10} )</td>
<td>history</td>
<td>4.0</td>
<td>$19.00</td>
<td>Amazon</td>
</tr>
</tbody>
</table>

**Profile** \( \Gamma = \{ F_5, F_6 \} \)

\[
F_5 \equiv \text{SELECT sum(price) FROM } S \text{ WHERE genre='sci-fi'}
\]

\[
F_6 \equiv \text{SELECT sum(rating) FROM } S
\]
M-relation

Goal

Avoid redundancy in generating candidate $k$-subsets

Book:

<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>$18.00</td>
<td>Amazon</td>
</tr>
<tr>
<td>$a_9$</td>
<td>romance</td>
<td>4.0</td>
<td>$20.00</td>
<td>Amazon</td>
</tr>
<tr>
<td>$a_{10}$</td>
<td>history</td>
<td>4.0</td>
<td>$19.00</td>
<td>Amazon</td>
</tr>
</tbody>
</table>

Profile $\Gamma = \{\mathcal{F}_5, \mathcal{F}_6\}$

$\mathcal{F}_5 \equiv$ SELECT sum(price) FROM $S$ WHERE genre='sci-fi'

$\mathcal{F}_6 \equiv$ SELECT sum(rating) FROM $S$

Redundancy Example

$\text{profile}_\Gamma(\{a_1, a_2, a_7\})$

$= \text{profile}_\Gamma(\{a_1, a_2, a_9\})$

$= \{15.00, 14\}$
M-relation

Goal
Avoid redundancy in generating candidate $k$-subsets

Book:

<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>$18.00</td>
<td>Amazon</td>
</tr>
<tr>
<td>$a_9$</td>
<td>romance</td>
<td>4.0</td>
<td>$20.00</td>
<td>Amazon</td>
</tr>
<tr>
<td>$a_{10}$</td>
<td>history</td>
<td>4.0</td>
<td>$19.00</td>
<td>Amazon</td>
</tr>
</tbody>
</table>

Redundancy Example

$\text{profile}_\Gamma(\{a_1, a_2, a_7\})$

$=$

$\text{profile}_\Gamma(\{a_1, a_2, a_9\})$

$=$

$\{15.00, 14\}$

Exchangeable Tuples $a_7, a_9$

For any 2-subset $s$ of $Book\setminus\{a_7, a_9\}$

$\text{profile}_\Gamma(s \cup \{a_7\}) = \text{profile}_\Gamma(s \cup \{a_9\})$

Profile $\Gamma = \{F_5, F_6\}$

$F_5 \equiv \text{SELECT sum(price) FROM } S \text{ WHERE genre='sci-fi'}$

$F_6 \equiv \text{SELECT sum(rating) FROM } S$
Book:

<table>
<thead>
<tr>
<th>Title</th>
<th>Genre</th>
<th>Rating</th>
<th>Price</th>
<th>Vendor</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</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>
<tr>
<td>a₉</td>
<td>romance</td>
<td>4.0</td>
<td>$20.00</td>
<td>Amazon</td>
</tr>
<tr>
<td>a₁₀</td>
<td>history</td>
<td>4.0</td>
<td>$19.00</td>
<td>Amazon</td>
</tr>
</tbody>
</table>

Profile \( \Gamma = \{F⃗{5}, F⃗{6}\} \)

\[ F⃗{5} \equiv \text{SELECT sum(price) FROM } S \text{ WHERE } \text{genre='sci-fi'} \]
\[ F⃗{6} \equiv \text{SELECT sum(rating) FROM } S \]

**M-relation Generation SQL**

```
SELECT CASE WHEN r.genre='sci-fi' THEN r.price ELSE 0 END AS A_5,
           r.rating AS A_6,
           count(*) AS A_cnt
FROM r
GROUP BY A_5, A_6
```
Book:

<table>
<thead>
<tr>
<th>Title</th>
<th>Genre</th>
<th>Rating</th>
<th>Price</th>
<th>Vendor</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</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>
<tr>
<td>a9</td>
<td>romance</td>
<td>4.0</td>
<td>$20.00</td>
<td>Amazon</td>
</tr>
<tr>
<td>a10</td>
<td>history</td>
<td>4.0</td>
<td>$19.00</td>
<td>Amazon</td>
</tr>
</tbody>
</table>

Profile $\Gamma = \{F_5, F_6\}$

$F_5 \equiv \text{SELECT } \text{sum(price)} \text{ FROM } S \text{ WHERE genre='sci-fi'}$

$F_6 \equiv \text{SELECT } \text{sum(rating)} \text{ FROM } S$

M-relation Generation SQL

```
SELECT CASE WHEN r.genre='sci-fi' THEN r.price ELSE 0 END AS A_5,
           r.rating AS A_6,
           count(*) AS A_cnt
FROM r
GROUP BY A_5, A_6
```
Set Preference via M-relations

- Set preference over the original relations $\Rightarrow$ set preference over its M-relation

![Diagram showing original relation and M-relation with candidate subsets and counts.](image-url)
Hybrid Algorithms

- SM: Superpreference followed by M-relation
- MS: M-relation followed by Superpreference
Outline

1 Motivation

2 Profile-based Set Preferences

3 Computing the “Best” Sets

4 Experiments

5 Future Work
Dataset and Set Preferences

- **Dataset**
  - 8000 book quotes from Amazon
  - Schema: \(\langle title, genre, rating, price, vendor \rangle\)
Dataset and Set Preferences

- **Dataset**
  - 8000 book quotes from Amazon
  - Schema: $\langle title, genre, rating, price, vendor \rangle$

- **Features**
  
  \[ F_5 \equiv \text{SELECT sum(price) FROM } S \text{ WHERE genre='sci-fi'} \]
  \[ F_6 \equiv \text{SELECT sum(rating) FROM } S \]
  \[ F_9 \equiv \text{SELECT sum(rating) FROM } S \text{ WHERE genre='sci-fi'} \]
  \[ F_{10} \equiv \text{SELECT sum(price) FROM } S \]
  \[ F_{11} \equiv \text{SELECT count(title) FROM } S \text{ WHERE genre='sci-fi'} \text{ and price < 20.00} \]
  \[ F_{12} \equiv \text{SELECT sum(rating) FROM } S \text{ WHERE rating } \geq 4.0 \]
Dataset and Set Preferences

- **Dataset**
  - 8000 book quotes from Amazon
  - Schema: \( \langle \text{title, genre, rating, price, vendor} \rangle \)

- **Features**

  \[ F_5 \equiv \text{SELECT} \ \text{sum(price)} \ \text{FROM} \ S \ \text{WHERE} \ \text{genre} = '\text{sci-fi}' \]
  \[ F_6 \equiv \text{SELECT} \ \text{sum(rating)} \ \text{FROM} \ S \]
  \[ F_9 \equiv \text{SELECT} \ \text{sum(rating)} \ \text{FROM} \ S \ \text{WHERE} \ \text{genre} = '\text{sci-fi}' \]
  \[ F_{10} \equiv \text{SELECT} \ \text{sum(price)} \ \text{FROM} \ S \]
  \[ F_{11} \equiv \text{SELECT} \ \text{count(title)} \ \text{FROM} \ S \ \text{WHERE} \ \text{genre} = '\text{sci-fi}' \ \text{and} \ \text{price} < 20.00 \]
  \[ F_{12} \equiv \text{SELECT} \ \text{sum(rating)} \ \text{FROM} \ S \ \text{WHERE} \ \text{rating} \geq 4.0 \]

- **Set Preferences**

<table>
<thead>
<tr>
<th>Set Pref. Name</th>
<th>Profile Schema</th>
<th>Profile Pref. Formula C</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP1</td>
<td>( \langle F_5, F_6 \rangle )</td>
<td>( F_5(s_1) &lt; F_5(s_2) \land F_6(s_1) &gt; F_6(s_2) )</td>
</tr>
<tr>
<td>SP2</td>
<td>( \langle F_9, F_{10} \rangle )</td>
<td>( F_9(s_1) &gt; F_9(s_2) \land F_{10}(s_1) &lt; F_{10}(s_2) )</td>
</tr>
<tr>
<td>SP3</td>
<td>( \langle F_{11}, F_{12} \rangle )</td>
<td>( F_{11}(s_1) &gt; F_{11}(s_2) \land F_{12}(s_1) &gt; F_{12}(s_2) )</td>
</tr>
</tbody>
</table>
Performance of Different Algorithms

Set
Pref 1

Set
Pref 2
Performance of Different Algorithms

Set Pref 3

# of sets in Generation (g)
Input Size (n)
NAIVE
SUPER
MREL
SM
MS

Generate Percentage

Xi Zhang and Jan Chomicki (SUNY Buffalo)
Related Work

- Binshtok et al. [BBS\textsuperscript{+}07]
  - Problem: find an optimal subset of items
  - Set property: predicate
  - Set preference: TCP-net or scoring function
  - Subsets of any cardinality, subsumed by our framework in the fixed-cardinality case

- desJardins and Wagstaff [dW05]
  - Fixed-cardinality set preference
  - Two set features: diversity and depth

- Guha et al. [GGK\textsuperscript{+}03]
  - Problem: find an optimal subset of tuples
  - Set property: $aggr(A) < parameter$
  - Set preference: objective function $\min / \max$
Future Work

- Query optimization for non-additive features
- Set preference elicitation
- Embedding “best-subset” generation in relational query language
- Additional set ranking or browsing techniques for navigation of results


Sudipto Guha, Dimitrios Gunopulos, Nick Koudas, Divesh Srivastava, and Michail Vlachos.
Efficient approximation of optimization queries under parametric aggregation constraints.
Thank you!