Preference Queries over Sets

Xi Zhang       Jan Chomicki

SUNY at Buffalo

April 15, 2011
Tuple Preferences

Well known preferences: top-$k$, skyline etc.

Set Preferences

Preferences between sets of tuples.
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>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>

... ... ... ... ...

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)
Tuple Preference - Winnow (Chomicki [Cho03])

- Tuple Preference: $t_1$ is preferred to $t_2$

$$t_1 >_C t_2 \iff t_1.rating = 'sci-fi' \land t_1.price < t_2.price$$
Tuple Preference: \( t_1 \) is preferred to \( t_2 \)

\[
t_1 >_C t_2 \iff t_1.\text{rating} = '\text{sci-fi}' \land t_1.\text{price} < t_2.\text{price}
\]

Winnow Operator \( \omega_C \): return all tuples in a database relation \( r \) for which there is no preferred tuple in \( r \)

\[
\omega_C(r) = \{ t \in r \mid \neg \exists t' \in r. t' >_C t \}
\]
Tuple Preference - Winnow (Chomicki [Cho03])

- Tuple Preference: \( t_1 \) is preferred to \( t_2 \)
  \[ t_1 >_C t_2 \iff t_1.rating = 'sci-fi' \land t_1.price < t_2.price \]

- Winnow Operator \( \omega_C \): return all tuples in a database relation \( r \) for which there is no preferred tuple in \( r \)
  \[ \omega_C(r) = \{ t \in r \mid \neg \exists t' \in r. t' >_C t \} \]

- Set Preference - tuple set \( \{ t_1, t_2, \ldots, t_k \} \) is preferred to tuple set \( \{ t'_1, t'_2, \ldots, t'_k \} \)
- Fixed cardinality \( (k) \) assumption
**Profile-based Set Preference**

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

<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)$\triangleright$(C1)</td>
</tr>
</tbody>
</table>
$k$-subsets: subsets of relation $r$, with fixed cardinality $k$

<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)$\triangleright$(C1)</td>
</tr>
</tbody>
</table>

features $\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_m$
**Profile-based Set Preference**

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

<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)$\triangleright$(C1)</td>
</tr>
</tbody>
</table>

features $F_1, F_2, \ldots, F_m$

profile $= \langle F_1, F_2, \ldots, F_m \rangle$
Features: mini-SQL Queries

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

Features: mini-SQL Queries

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

Set Preferences

\[ s_1 \succ C_1 s_2 \iff F_1(s_1) < F_1(s_2). \]
\[ s_1 \succ C_2 s_2 \iff F_2(s_1) = 1 \land F_2(s_2) \neq 1. \]
\[ s_1 \succ C_3 s_2 \iff (F_2(s_1) = 1 \land F_2(s_2) \neq 1) \]
\[ \lor (F_2(s_1) = 1 \land F_2(s_2) = 1 \land F_1(s_1) < F_1(s_2)) \]
\[ \lor (F_2(s_1) \neq 1 \land F_2(s_2) \neq 1 \land F_1(s_1) < F_1(s_2)). \]
Assume additive features in this talk.

Efficient optimizations exist for additive features.
Additive Features

- Assume additive features in this talk.
- Efficient optimizations exist for additive features.

Additive Feature

A feature $\mathcal{F}$ is additive iff for every subset $s$ of relation $r$, and every $t \in r - s$

$$\mathcal{F}(s \cup \{t\}) = \mathcal{F}(s) + f(t)$$

$$\mathcal{F}(\{t\}) = f(t)$$
Computing the “Best” Sets

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.
Computing the “Best” Sets

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!
Computing the “Best” Sets

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!

Example

$k = 3, |r| = 1000 \implies \binom{1000}{3} = 166167000$ candidate subsets
Superpreference Optimization

Goal: Generate as few candidate $k$-subsets as possible
Superpreference Optimization

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 $cover\ p t q t t_1 P r | t_1 >^+ t_2$, i.e. all tuples preferred to $t_1$.

d $t P s, cover\ p t q s s \not\in s$ is not a best subset.

Systematic Construction

Possible if all the features are additive.
Superpreference Optimization

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 $cover(t) = \{t' \in r | t' >^+ t\}$, i.e. all tuples preferred to $t$.

$$\exists t \in s, cover(t) \not\subseteq s \Rightarrow s \text{ is not a best subset.}$$
**Superpreference Optimization**

**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 $t_2$.

**Pruning Condition**

Let $cover(t) = \{t' \in r | t' >^+ t\}$, i.e. all tuples preferred to $t$.

$$ \exists t \in s, cover(t) \nsubseteq s \Rightarrow s \text{ is not a best subset}. $$

**Systematic Construction**

Possible if all the features are additive.
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:*

$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:

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

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

Profile preference:

\(s_1 \gg_C s_2 \equiv F_5(s_1) < F_5(s_2) \land F_6(s_1) > F_6(s_2)\)
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:

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

\[ F_6 \equiv \text{SELECT} \ \text{avg}(\text{rating}) \ \text{FROM} \ \$S \]

Profile preference:

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

“Superpreference” formula (assuming price > 0)

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

Book:

<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>
<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>

Redundancy Example

\[
\text{profile}_\Gamma(a_1, a_2, a_7) = \text{profile}_\Gamma(a_1, a_2, a_9) = \langle 15.00, 13.8 \rangle
\]

Exchangeable Tuples \(a_7, a_9\)
For any 2-subset \(s\) of \(\text{Book}\backslash\{a_7, a_9\}\)

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

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

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

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

#### 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>

#### M-relation:

<table>
<thead>
<tr>
<th></th>
<th>A5</th>
<th>A6</th>
<th>A_{cnt}</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>---------</td>
</tr>
<tr>
<td>m_{7,9,10}</td>
<td>$0.00</td>
<td>4.0</td>
<td>3</td>
</tr>
<tr>
<td>m_{8}</td>
<td>$18.00</td>
<td>3.5</td>
<td>1</td>
</tr>
</tbody>
</table>

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

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

$F_6 \equiv$ SELECT sum(rating) FROM $S$

#### M-relation Generation SQL

```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-relation

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

![Diagram showing original relation and M-relation with k-subsets and k-multisubsets]
Conclusions and Future Work

Conclusions

- A formal “logic + SQL” framework for specifying restricted set preferences and implementing set preference queries.
- Effective optimizations yielding improvements of several orders of magnitude.

Future Work

- Query optimization for non-additive features.
- Set preference elicitation.
- Embedding “best-subset” generation in relational query languages.
- Additional set ranking or browsing techniques for result navigation.
- Relaxing the fixed cardinality assumption: Superpreference depends on it assumption while M-relation does not.
Conclusions

- A formal “logic + SQL” framework for specifying restricted set preferences and implementing set preference queries.
- Effective optimizations yielding improvements of several orders of magnitude.

Future Work

- Query optimization for non-additive features.
- Set preference elicitation.
- Embedding “best-subset” generation in relational query languages.
- Additional set ranking or browsing techniques for result navigation.
- Relaxing the fixed cardinality assumption: Superpreference depends on it assumption while M-relation does not.
Thank you!
### Dataset
- 8,000 book quotes from Amazon
- Schema: \(\langle \text{title, genre, rating, price, vendor} \rangle\)

### Features
- \(F_5 \equiv \text{SELECT sum(price) FROM } S \text{ WHERE genre='sci-fi'}\)
- \(F_6 \equiv \text{SELECT sum(rating) FROM } S\)
- \(F_7 \equiv \text{SELECT sum(rating) FROM } S \text{ WHERE genre='sci-fi'}\)
- \(F_8 \equiv \text{SELECT sum(price) FROM } S\)
- \(F_9 \equiv \text{SELECT count(title) FROM } S \text{ WHERE genre='sci-fi' and price < 20.00}\)
- \(F_{10} \equiv \text{SELECT sum(rating) FROM } S \text{ WHERE rating } \geq 4.0\)

### Set Preferences

<table>
<thead>
<tr>
<th>Set Pref. Name</th>
<th>Profile Schema (\Gamma)</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

![Graph showing performance comparison]

Xi Zhang, Jan Chomicki (SUNY at Buffalo)
Preference Queries over Sets
April 15, 2011 18 / 21
Related Work

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

- Binshtok et al. [BBS$^+$07]
  - Problem: find a optimal subset of items
  - Set property: predicate
  - Set preference: TCP-net or scoring function
  - Consider subsets of any cardinality, in the fixed-cardinality case, it is subsumed by our framework

- desJardins and Wagstaff [dW05]
  - Consider fixed-cardinality set preference
  - Consider two set features: diversity and depth
Maxim Binshtok, Ronen I. Brafman, Solomon Eyal Shimony, Ajay Mani, and Craig Boutilier.
Computing optimal subsets.

Jan Chomicki.
Preference formulas in relational queries.

Marie desJardins and Kiri Wagstaff.
DD-pref: A language for expressing preferences over sets.
Sudipto Guha, Dimitrios Gunopulos, Nick Koudas, Divesh Srivastava, and Michail Vlachos.
Efficient approximation of optimization queries under parametric aggregation constraints.