Profiling Sets for Preference Querying

Xi Zhang   Jan Chomicki

SUNY at Buffalo

April 24, 2008

Outline

Outline

Motivating Example

Alice is buying 3 books as gifts.

Title	Genre	Rating	Price	Vendor
a_1	sci-fi	5.0	$15.00	Amazon
a_2	biography	4.8	$20.00	B&N
a_3	sci-fi	4.5	$25.00	Amazon
a_4	romance	4.4	$10.00	B&N
a_5	sci-fi	4.3	$15.00	Amazon
a_6	romance	4.2	$12.00	B&N
a_7	biography	4.0	$18.00	Amazon
a_8	sci-fi	3.5	$18.00	Amazon

Motivating Example

Alice is buying 3 books as gifts.

Title	Genre	Rating	Price	Vendor
a_1	sci-fi	5.0	$15.00	Amazon
a_2	biography	4.8	$20.00	B&N
a_3	sci-fi	4.5	$25.00	Amazon
a_4	romance	4.4	$10.00	B&N
a_5	sci-fi	4.3	$15.00	Amazon
a_6	romance	4.2	$12.00	B&N
a_7	biography	4.0	$18.00	Amazon
a_8	sci-fi	3.5	$18.00	Amazon

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)

Motivating Example

Alice is buying 3 books as gifts.

Title	Genre	Rating	Price	Vendor
a_1	sci-fi	5.0	$15.00	Amazon
a_2	biography	4.8	$20.00	B&N
a_3	sci-fi	4.5	$25.00	Amazon
a_4	romance	4.4	$10.00	B&N
a_5	sci-fi	4.3	$15.00	Amazon
a_6	romance	4.2	$12.00	B&N
a_7	biography	4.0	$18.00	Amazon
a_8	sci-fi	3.5	$18.00	Amazon

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

Set Preferences

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

Set Pref.	Quantity of Interest	Desired Value or Order
(C1)	total cost	$<$
(C2)	# of sci-fi books	1
(C3)	# of distinct vendors	$<$

Set Preferences

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

Simple Set Preference

Set Pref.	Quantity of Interest	Desired Value or Order
(C1)	total cost	$<$
(C2)	# of sci-fi books	1
(C3)	# of distinct vendors	$<$

Complex Set Preference (C0)

(C2) is more important than (C1):

prioritized composition of (C2) and (C1)
i.e. (C2) $\rhd$ (C1)

Preferences over Profiles

Set Pref.	Quantities of Interest	Desired Value or Order
(C1)	total cost	$<$
(C2)	# of sci-fi books	1
(C3)	# of distinct vendors	$<$
(C0)	total cost, # of sci-fi books	(C2)$\rhd$(C1)

Preferences over Profiles

Set Pref.	Quantities of Interest	Desired Value or Order
(C1)	total cost	$<$
(C2)	# of sci-fi books	1
(C3)	# of distinct vendors	$<$
(C0)	total cost, # of sci-fi books	(C2)$\rhd$(C1)

$\downarrow$

features

Preferences over Profiles

Set Pref.	Quantities of Interest	Desired Value or Order
(C1)	total cost	$<$
(C2)	# of sci-fi books	1
(C3)	# of distinct vendors	$<$
(C0)	total cost, # of sci-fi books	(C2)$\rhd$(C1)

↓ features

↓ preferences over profiles

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

Profiling Subsets

Definition (SQL-based k-subset Feature)

Parameterized feature

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

where S is a set variable over any k-subset of relation r.
Requirement: the query is *categorical*.

Profiling Subsets

Definition (SQL-based *k*-subset Feature)

Parameterized feature

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

where S is a set variable over any k-subset of relation r.
Requirement: the query is *categorical*.

Definition (Profile)

$$\mathit{profile}(s) = \langle \mathcal{A}_1(s), \ldots, \mathcal{A}_m(s) \rangle$$

where s is any k-subset of relation r.

Example

Set Pref.	Quantities of Interest	Desired Value or Order
(C1)	total cost	$<$
(C2)	# of sci-fi books	1
(C3)	# of distinct vendors	$<$
(C0)	total cost, # of sci-fi books	(C2)$\rhd$(C1)

$\mathcal{A}_1 \equiv$ `SELECT sum(price) FROM S`

Example

Set Pref.	Quantities of Interest	Desired Value or Order
(C1)	total cost	$<$
(C2)	# of sci-fi books	1
(C3)	# of distinct vendors	$<$
(C0)	total cost, # of sci-fi books	(C2)$\rhd$(C1)

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

Example

Set Pref.	Quantities of Interest	Desired Value or Order
(C1)	total cost	$<$
(C2)	# of sci-fi books	1
(C3)	# of distinct vendors	$<$
(C0)	total cost, # of sci-fi books	(C2)$\rhd$(C1)

$\mathcal{A}_1 \equiv$ `SELECT sum(price) FROM S`

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

$\mathcal{A}_3 \equiv$ `SELECT count(DISTINCT vendor) FROM S`

Tuple Preference - Winnow [Chomicki03]

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

$$C(t_1, t_2) \Leftrightarrow t_1 \succ_C t_2$$

Tuple Preference - Winnow [Chomicki03]

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

$$C(t_1, t_2) \Leftrightarrow t_1 \succ_C t_2$$

Definition (Winnow Operator)

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

$$\omega_C(r) = \{t \in r | \neg \exists t' \in r . t' \succ_C t\}$$

Example

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

Set Pref.	Quantities of Interest	Desired Value or Order
(C1)	total cost	$<$
(C2)	# of sci-fi books	1
(C0)	total cost, # of sci-fi books	(C2)$\rhd$(C1)

$\mathcal{A}_1 \equiv$ `SELECT sum(price) FROM S`

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

Example

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

Set Pref.	Quantities of Interest	Desired Value or Order
(C1)	total cost	$<$
(C2)	# of sci-fi books	1
(C0)	total cost, # of sci-fi books	(C2)$\rhd$(C1)

$\mathcal{A}_1 \equiv$ `SELECT sum(price) FROM S`

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

$$s_1 \ggcurly_{C1} s_2 \Leftrightarrow \quad \mathcal{A}_1(s_1) < \mathcal{A}_1(s_2).$$

Example

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

Set Pref.	Quantities of Interest	Desired Value or Order
(C1)	total cost	$<$
(C2)	# of sci-fi books	1
(C0)	total cost, # of sci-fi books	(C2)$\rhd$(C1)

$\mathcal{A}_1 \equiv$ `SELECT sum(price) FROM S`

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

$$s_1 \succ_{C1} s_2 \Leftrightarrow \mathcal{A}_1(s_1) < \mathcal{A}_1(s_2).$$
$$s_1 \succ_{C2} s_2 \Leftrightarrow \mathcal{A}_2(s_1) = 1 \wedge \mathcal{A}_2(s_2) \neq 1.$$

Example

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

Set Pref.	Quantities of Interest	Desired Value or Order
(C1)	total cost	$<$
(C2)	# of sci-fi books	1
(C0)	total cost, # of sci-fi books	(C2)$\rhd$(C1)

$\mathcal{A}_1 \equiv$ `SELECT sum(price) FROM S`

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

$$s_1 \succ_{C1} s_2 \Leftrightarrow \mathcal{A}_1(s_1) < \mathcal{A}_1(s_2).$$

$$s_1 \succ_{C2} s_2 \Leftrightarrow \mathcal{A}_2(s_1) = 1 \wedge \mathcal{A}_2(s_2) \neq 1.$$

$$\begin{aligned} s_1 \succ_{C0} s_2 \Leftrightarrow\ & (\mathcal{A}_2(s_1) = 1 \wedge \mathcal{A}_2(s_2) \neq 1) \\ & \vee (\mathcal{A}_2(s_1) = 1 \wedge \mathcal{A}_2(s_2) = 1 \wedge \mathcal{A}_1(s_1) < \mathcal{A}_1(s_2)) \\ & \vee (\mathcal{A}_2(s_1) \neq 1 \wedge \mathcal{A}_2(s_2) \neq 1 \wedge \mathcal{A}_1(s_1) < \mathcal{A}_1(s_2)). \end{aligned}$$

Outline

Naive Algorithm

- Generate all k-subsets of relation r and compute the set γ 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 γ'.

Naive Algorithm

- Generate all k-subsets of relation r and compute the set γ 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 γ'.

Too many k-subsets!

Conditions for Early Pruning

"Superpreference"

Find a "superpreference" $(\succ^+)$ over the relation r, such that

$$t_1 \succ^+ t_2 \Leftrightarrow \quad s' \cup \{t_1\} \ggcurly_C s' \cup \{t_2\}.$$

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

Conditions for Early Pruning

"Superpreference"

Find a "superpreference" ($\succ^+$) over the relation r, such that

$$t_1 \succ^+ t_2 \Leftrightarrow \quad s' \cup \{t_1\} \gg_C s' \cup \{t_2\}.$$

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

Additive k-subset Feature $\mathcal{A}$

- $\mathcal{A}$ is well-defined for the domain of *(k-1)-subsets*(r), and
- for any subset $s' \in$ *(k-1)-subsets*(r), and any $t \in r \wedge t \notin s'$,

$$\mathcal{A}(s' \cup \{t\}) = \mathcal{A}(s') + f(t)$$

where f is a function of t only.

Pruning Theorem

Theorem

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

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

where $\theta \in \{=, \neq, <, >, \leqslant, \geqslant\}$ *and* $\mathcal{A}_{ij}$ *is an additive k-subset feature, then superpreference* C^+ *exists and can be constructed systematically.*

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)$\cap$(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

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

$\mathcal{A}_6 \equiv$ `SELECT avg(rating) FROM S`

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

$\mathcal{A}_5 \equiv$ `SELECT sum(price) FROM S WHERE genre='sci-fi'`
$\mathcal{A}_6 \equiv$ `SELECT avg(rating) FROM S`

Profile preference

$$s_1 \ggcurly_C s_2 \equiv \mathcal{A}_5(s_1) < \mathcal{A}_5(s_2) \wedge \mathcal{A}_6(s_1) > \mathcal{A}_6(s_2)$$

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

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

$\mathcal{A}_6 \equiv$ `SELECT avg(rating) FROM S`

Profile preference

$$s_1 \ggcurly_C s_2 \equiv \mathcal{A}_5(s_1) < \mathcal{A}_5(s_2) \wedge \mathcal{A}_6(s_1) > \mathcal{A}_6(s_2)$$

"Superpreference" formula C^+ (assuming $price > 0$)

$$t_1 \succ_{C^+} t_2 \quad \equiv \quad t_1.rating > t_2.rating \wedge t_2.genre = \text{'sci-fi'} \\ \wedge (t_1.price < t_2.price \vee t_1.genre \neq \text{'sci-fi'}).$$

Heuristic Algorithm

- Initialize $r' = \varnothing$.
- Find the "most promising" tuples based on superpreference C^+

$$\text{repeat } r' := r' \cup \omega_{C^+}(r - r') \text{ until } |r'| \geqslant k.$$

- Generate all k-subsets of relation r' and compute the set γ 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 γ'.

Outline

Future Work

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