Profiling Sets for Preference Querying

Xi Zhang and Jan Chomicki

SUNY Buffalo

May 18, 2010

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 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.

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 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])

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 (Chomicki [Cho03])

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,

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

Outline

Profile-based Set Preference

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

Profile-based Set Preference

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

features

Profile-based Set Preference

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

features

preferences over profiles

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

Profile-based Set Preferences

- 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

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
- **profile**
  A vector of k-subset features

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
- **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.

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
(C3)	total cost, # of sci-fi books	(C2)$\triangleright$(C1)

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

$\mathcal{F}_2 \equiv$ `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
(C3)	total cost, # of sci-fi books	(C2)$\triangleright$(C1)

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

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

$$s_1 \gg_{C1} s_2 \Leftrightarrow \mathcal{F}_1(s_1) < \mathcal{F}_1(s_2).$$

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
(C3)	total cost, # of sci-fi books	(C2)$\triangleright$(C1)

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

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

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

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
(C3)	total cost, # of sci-fi books	(C2)$\triangleright$(C1)

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

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

$$
\begin{aligned}
s_1 \gg_{C1} s_2 \Leftrightarrow\ & \mathcal{F}_1(s_1) < \mathcal{F}_1(s_2).\\
s_1 \gg_{C2} s_2 \Leftrightarrow\ & \mathcal{F}_2(s_1) = 1 \wedge \mathcal{F}_2(s_2) \neq 1.\\
s_1 \gg_{C3} s_2 \Leftrightarrow\ & (\mathcal{F}_2(s_1) = 1 \wedge \mathcal{F}_2(s_2) \neq 1)\\
& \vee (\mathcal{F}_2(s_1) = 1 \wedge \mathcal{F}_2(s_2) = 1 \wedge \mathcal{F}_1(s_1) < \mathcal{F}_1(s_2))\\
& \vee (\mathcal{F}_2(s_1) \neq 1 \wedge \mathcal{F}_2(s_2) \neq 1 \wedge \mathcal{F}_1(s_1) < \mathcal{F}_1(s_2)).
\end{aligned}
$$

Outline

Naive Algorithm

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

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 Algorithm

Goal

Generate as few candidate k-subsets as possible

"Superpreference"

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

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

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

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

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

Pruning Condition

Let $cover(t) = \{t' \succ t | t' \succ^+ t\}$.
For every t in the "best" k-subset, $cover(t) < k$.

Find the "Superpreference"

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

Profile preference

$$s_1 \gg_C s_2 \equiv \mathcal{F}_5(s_1) < \mathcal{F}_5(s_2) \wedge \mathcal{F}_6(s_1) > \mathcal{F}_6(s_2)$$

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

Profile preference

$$s_1 \gg_C s_2 \equiv \mathcal{F}_5(s_1) < \mathcal{F}_5(s_2) \wedge \mathcal{F}_6(s_1) > \mathcal{F}_6(s_2)$$

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

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

M-relation

Goal

Avoid redundancy in generating candidate k-subsets

M-relation

Goal

Avoid redundancy in generating candidate k-subsets

Book:

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
a_9	romance	4.0	$20.00	Amazon
a_{10}	history	4.0	$19.00	Amazon

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`

M-relation

Goal

Avoid redundancy in generating candidate k-subsets

Book:

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
a_9	romance	4.0	\$20.00	Amazon
a_{10}	history	4.0	\$19.00	Amazon

Redundancy Example

$$\begin{aligned} & profile_\Gamma(\{a_1, a_2, a_7\}) \\ = \; & profile_\Gamma(\{a_1, a_2, a_9\}) \\ = \; & \{\$15.00, 14\} \end{aligned}$$

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`

M-relation

Goal

Avoid redundancy in generating candidate k-subsets

Book:

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
a_9	romance	4.0	\$20.00	Amazon
a_{10}	history	4.0	\$19.00	Amazon

Redundancy Example

$$\begin{aligned} & profile_\Gamma(\{a_1, a_2, a_7\}) \\ = \; & profile_\Gamma(\{a_1, a_2, a_9\}) \\ = \; & \{\$15.00, 14\} \end{aligned}$$

Exchangeable Tuples a_7, a_9

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

$$profile_\Gamma(s \cup \{a_7\}) = profile_\Gamma(s \cup \{a_9\})$$

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`

M-relation Generation

Book:

Title	Genre	Rating	Price	Vendor
...	...	...	...	...
a_7	biography	4.0	$18.00	Amazon
a_8	sci-fi	3.5	$18.00	Amazon
a_9	romance	4.0	$20.00	Amazon
a_{10}	history	4.0	$19.00	Amazon

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`

M-relation Generation SQL

```
SELECT CASE WHEN r.genre='sci-fi' THEN r.price ELSE 0 END AS A5,
       r.rating AS A6, count(*) AS Acnt FROM r
GROUP BY A5, A6
```

M-relation Generation

Book:

Title	Genre	Rating	Price	Vendor
...	...	...	...	...
a_7	biography	4.0	$18.00	Amazon
a_8	sci-fi	3.5	$18.00	Amazon
a_9	romance	4.0	$20.00	Amazon
a_{10}	history	4.0	$19.00	Amazon

M-relation:

	A_5	A_6	A_{cnt}
...	...	...	...
$m_{7,9,10}$	$0.00	4.0	3
m_8	$18.00	3.5	1

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

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

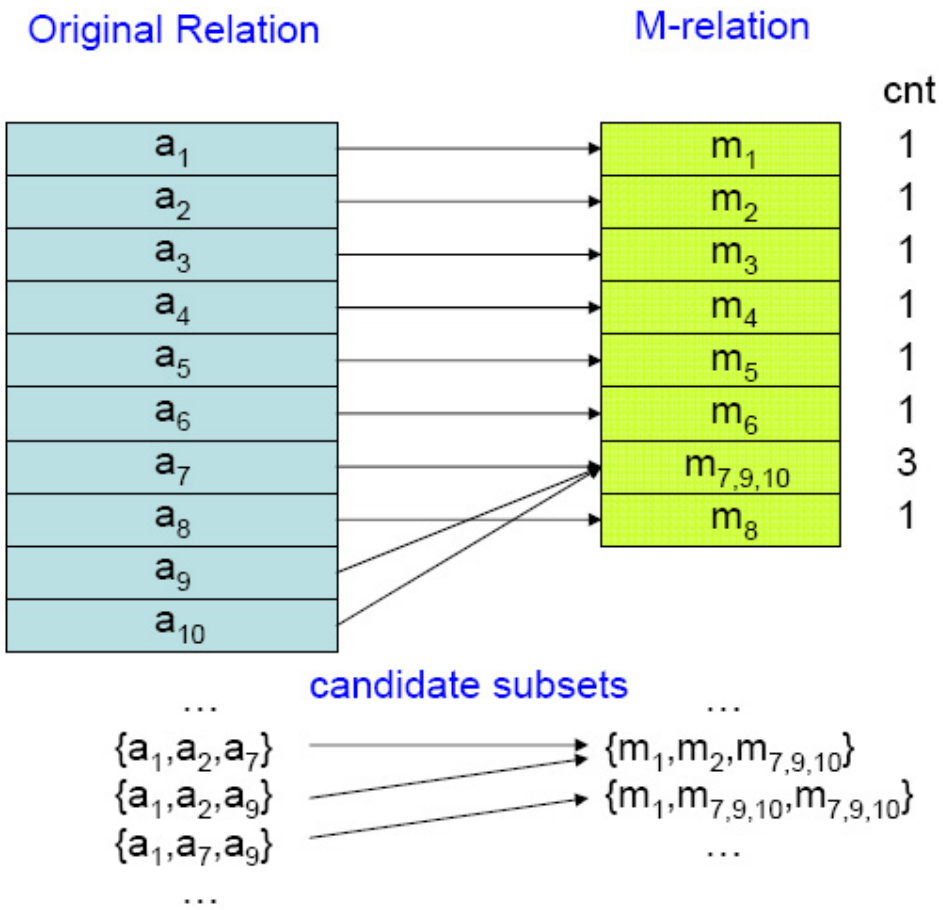
$\mathcal{F}_6 \equiv$ `SELECT sum(rating) FROM $S`

M-relation Generation SQL

```
SELECT CASE WHEN r.genre='sci-fi' THEN r.price ELSE 0 END AS A5,
       r.rating AS A6, count(*) AS Acnt FROM r
GROUP BY A5, A6
```

Set Preference via M-relations

- Set preference over the original relations ⇒ set preference over its M-relation



Hybrid

Hybrid Algorithms

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

Outline

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**

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

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

$\mathcal{F}_9 \equiv$ `SELECT sum(rating) FROM $S WHERE genre='sci-fi'`

$\mathcal{F}_{10} \equiv$ `SELECT sum(price) FROM $S`

$\mathcal{F}_{11} \equiv$ `SELECT count(title) FROM $S WHERE genre='sci-fi' and price < 20.00`

$\mathcal{F}_{12} \equiv$ `SELECT sum(rating) FROM $S WHERE rating >= 4.0`

Dataset and Set Preferences

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

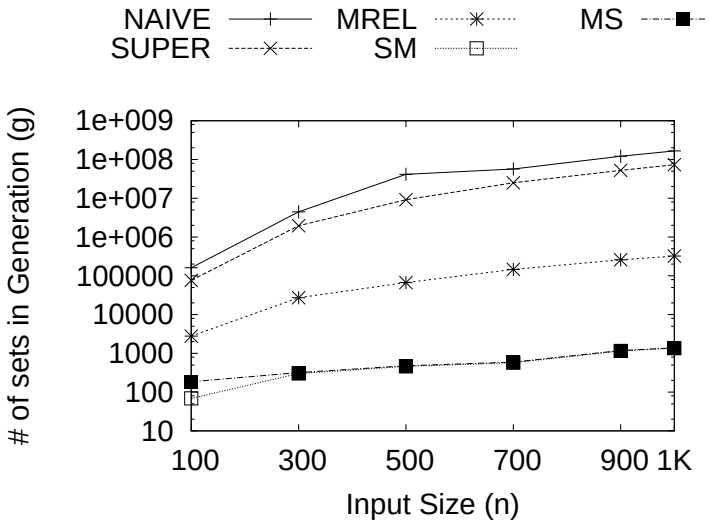
$\mathcal{F}_5 \equiv$ `SELECT sum(price) FROM $S WHERE genre='sci-fi'`
$\mathcal{F}_6 \equiv$ `SELECT sum(rating) FROM $S`
$\mathcal{F}_9 \equiv$ `SELECT sum(rating) FROM $S WHERE genre='sci-fi'`
$\mathcal{F}_{10} \equiv$ `SELECT sum(price) FROM $S`
$\mathcal{F}_{11} \equiv$ `SELECT count(title) FROM $S WHERE genre='sci-fi' and price < 20.00`
$\mathcal{F}_{12} \equiv$ `SELECT sum(rating) FROM $S WHERE rating >= 4.0`

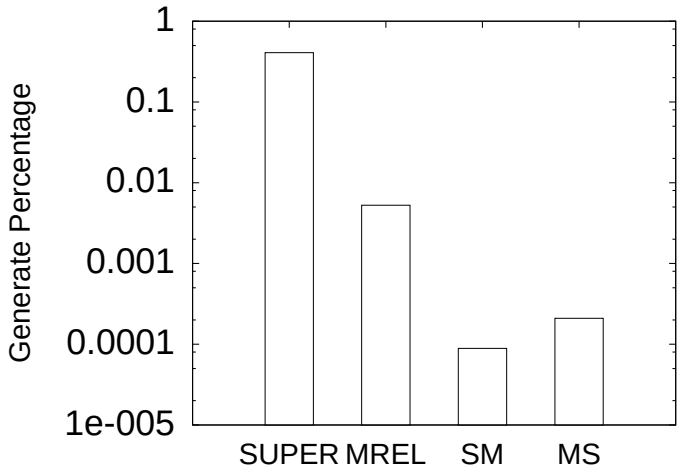
- **Set Preferences**

Set Pref. Name	Profile Schema Γ	Profile Pref. Formula C
SP1	$\langle \mathcal{F}_5, \mathcal{F}_6 \rangle$	$\mathcal{F}_5(s_1) < \mathcal{F}_5(s_2) \wedge \mathcal{F}_6(s_1) > \mathcal{F}_6(s_2)$
SP2	$\langle \mathcal{F}_9, \mathcal{F}_{10} \rangle$	$\mathcal{F}_9(s_1) > \mathcal{F}_9(s_2) \wedge \mathcal{F}_{10}(s_1) < \mathcal{F}_{10}(s_2)$
SP3	$\langle \mathcal{F}_{11}, \mathcal{F}_{12} \rangle$	$\mathcal{F}_{11}(s_1) > \mathcal{F}_{11}(s_2) \wedge \mathcal{F}_{12}(s_1) > \mathcal{F}_{12}(s_2)$

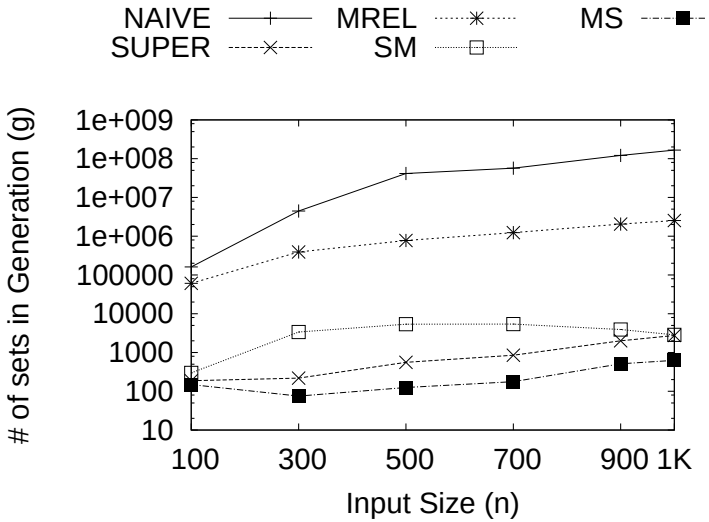
Performance of Different Algorithms

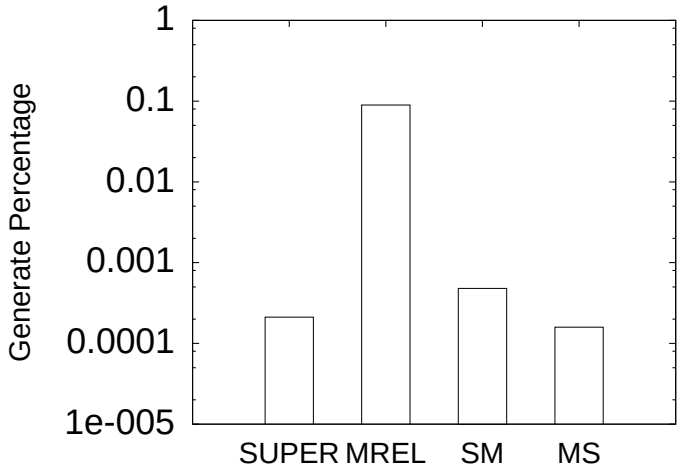
Set Pref 1





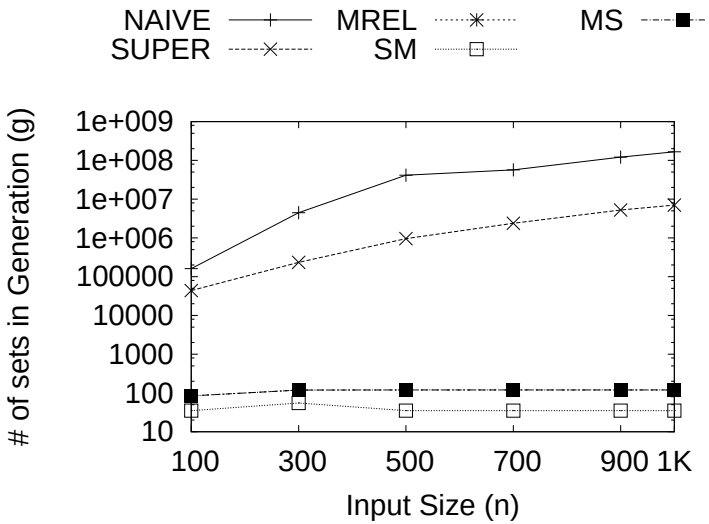
Set Pref 2

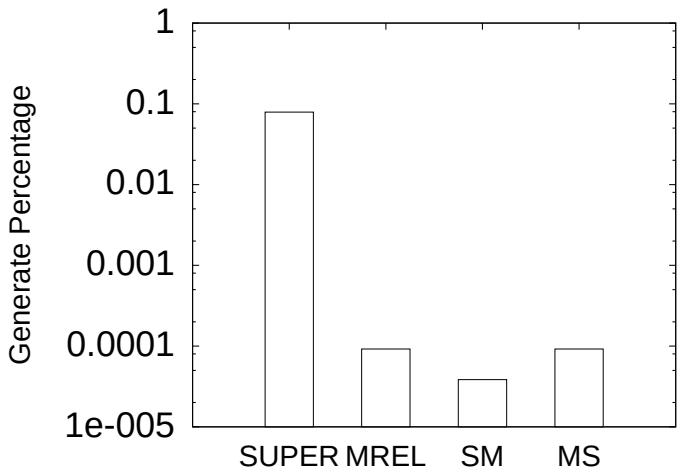




Performance of Different Algorithms

Set
Pref 3





Related Work

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

Outline

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

References I


Maxim Binshtok, Ronen I. Brafman, Solomon Eyal Shimony, Ajay Mani, and Craig Boutilier.
Computing optimal subsets.
In *AAAI*, pages 1231–1236, 2007.


Jan Chomicki.
Preference formulas in relational queries.
ACM Trans. Database Syst., 28(4):427–466, 2003.


Marie desJardins and Kiri Wagstaff.
DD-pref: A language for expressing preferences over sets.
In *AAAI*, pages 620–626, 2005.

References II

Sudipto Guha, Dimitrios Gunopulos, Nick Koudas, Divesh Srivastava, and Michail Vlachos.
Efficient approximation of optimization queries under parametric aggregation constraints.
In *VLDB*, pages 778–789, 2003.

Thank you!