Preference Representation using Higher-Order LP

Antonis Troumpoukis
NCSR “Demokritos”

November 3, 2023

Outline

- Preferences
- Preferences in Databases
- Preferences in Higher-Order Logic Programming
- WIP

Preferences

- are ubiquitous in real life, are being studied in many sciences
- as **soft constraints**
  - too many hard constraints $\rightsquigarrow$ empty result set
  - too few hard constraints $\rightsquigarrow$ “needle in a haystack”
- are **comparative** in nature
  - in many cases, a “preference function” cannot be easily defined
  - Users rarely want to express their preference using numbers

Examples

- “I **want** to fly from Athens to Rome and I **prefer** to fly with Realiable Airlines”
- “For my tomorrow flight, I would **prefer** an aisle seat to a window seat”

Preferences lifecycle

- Preference aquisition
- Preference modelling
- Preference representation
- Preference reasoning
- Preference revision

Qualitative Preferences

Quantitative vs. Qualitative preferences

Main approaches for representing preferences:

- *The quantitative approach:* as degrees of interest (“My preference in science-fiction books is 0.8 while in novels 0.4”).
- *The qualitative approach:* by direct comparisons (“I like action movies more that comedies”).

Qualitative approach

- more general (preference function cannot always be defined)
- more intuitive (“humans are rarely willing to express their preferences directly in terms of a value function.”)

Preference relations

Universe of objects

- constants: uninterpreted, numbers, ...
- entities
- database tuples
- sets

Preference relation

- **binary** relation between objects
- $x \succ y \equiv x$ is_preferred_to $y \equiv x$ is_better_than $y \equiv x$ dominates y
- abstract, uniform way of talking about (relative) desirability, worth, cost, ...
- preference relations used in **preference queries**

Properties of preference relations

Incomparability (or indifference) $x \sim y \equiv x \not\succ y \ \wedge \ y \not\succ x$

Properties of $\succ$

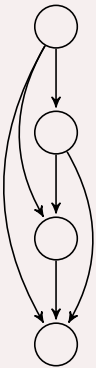
1. irreflexivity: $\forall x. \quad x \not\succ x$
2. asymmetry: $\forall x, y. \quad x \succ y \implies y \not\succ x$
3. transitivity: $\forall x, y, z. \quad (x \succ y) \ \wedge \ (y \succ z) \implies x \succ z$
4. transitivity of incomparability: $\forall x, y, z. \quad (x \sim y) \ \wedge \ (y \sim z) \implies x \sim z$
5. connectivity: $\forall x, y. \quad (x \succ y) \ \wedge \ (y \succ x) \ \wedge \ (x = y)$

Orders

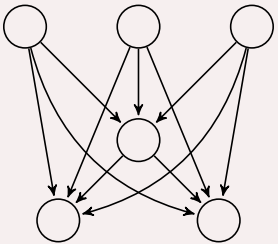
- strict partial order (SPO): irreflexive, asymmetric and transitive
- weak order (WO): incomparability-transitive SPO
- total order: connected SPO

Orders

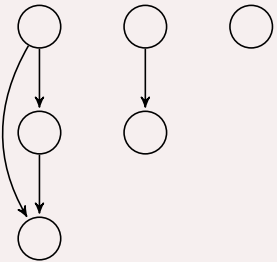
TO



WO



SPO



Qualitative Preferences in Databases

Qualitative Preference Representation in Databases

The qualitative approach in databases

An influential work in this area is: J. Chomicki, "*Preference Formulas in Relational Queries*", ACM TODS, 2003.

Main ideas:

- Preferences between tuples are specified using *binary preference relations*, defined using logic formulas.
- A new relational algebra operator is introduced, that eliminates from its argument relation the less preferred tuples according to the given preference relation.

Movie preferences

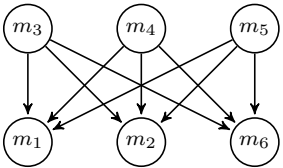
ID	Name	Genre	Runtime (min)	Rating
m_1	The Godfather	drama	175	9.2
m_2	The Irishman	drama	189	8.5
m_3	Goodfellas	drama	146	8.7
m_4	The Exorcist	horror	117	8.2
m_5	The Shining	horror	142	8.8
m_6	Inception	scifi	152	8.8

Example Preferences

1. Prefer movies that their runtime are less than 150min.
2. Prefer drama movies over scifi movies over horror movies.
3. Given two movies of the same genre, prefer the one with the highest rating.

Movie preferences (cont.)

ID	Name	Genre	Runtime (min)	Rating
m_1	The Godfather	drama	175	9.2
m_2	The Irishman	drama	189	8.5
m_3	Goodfellas	drama	146	8.7
m_4	The Exorcist	horror	117	8.2
m_5	The Shining	horror	142	8.8
m_6	Inception	scifi	152	8.8



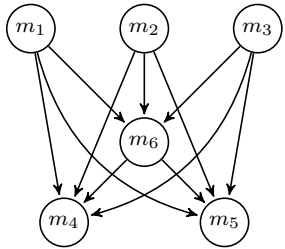
Example

Prefer movies that their runtime are less than 150min.

$$t \succ_1 t' \equiv (\mathsf{runtime}(t) < 150) \wedge (\mathsf{runtime}(t') \geq 150)$$

Movie preferences (cont.)

ID	Name	Genre	Runtime (min)	Rating
m_1	The Godfather	drama	175	9.2
m_2	The Irishman	drama	189	8.5
m_3	Goodfellas	drama	146	8.7
m_4	The Exorcist	horror	117	8.2
m_5	The Shining	horror	142	8.8
m_6	Inception	scifi	152	8.8



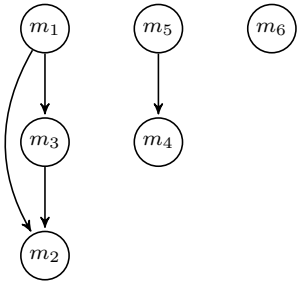
Example

Prefer drama movies over scifi movies over horror movies.

$$
\begin{aligned}
t \succ_2 t' \quad \equiv \quad & \big((\mathsf{genre}(t) = \mathsf{drama}) \wedge (\mathsf{genre}(t') = \mathsf{scifi})\big) \vee \\
& \big((\mathsf{genre}(t) = \mathsf{scifi}) \wedge (\mathsf{genre}(t') = \mathsf{horror})\big) \vee \\
& \big((\mathsf{genre}(t) = \mathsf{drama}) \wedge (\mathsf{genre}(t') = \mathsf{horror})\big)
\end{aligned}
$$

Movie preferences (cont.)

ID	Name	Genre	Runtime (min)	Rating
m_1	The Godfather	drama	175	9.2
m_2	The Irishman	drama	189	8.5
m_3	Goodfellas	drama	146	8.7
m_4	The Exorcist	horror	117	8.2
m_5	The Shining	horror	142	8.8
m_6	Inception	scifi	152	8.8



Example

Given two movies of the same genre, prefer the one with the highest rating.

$$t \succ_3 t' \equiv (\mathsf{genre}(t) = \mathsf{genre}(t')) \wedge (\mathsf{rating}(t) > \mathsf{rating}(t'))$$

Preference operators

- In order to select the **most preferred** elements of a relation we use the operator **winnow**.
- Was defined in the context of relational databases

Winnow

Given a preference relation $\succ_C$ over a relation R, we define:

$$w_{\succ_C}(R) = \{t \in R : \neg\exists t' \in R \text{ such that } t' \succ_C t\}$$

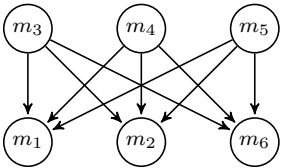
Other preference operators

- Return all elements that appear "at level" n
- Rank all elements according to their "level"
- Return all elements that are dominated by at most k other elements

Movie preferences (cont.)

Result of $w_{\succ_1}$(movie):

ID	Name	Genre	Runtime (min)	Rating
m_1	The Godfather	drama	175	9.2
m_2	The Irishman	drama	189	8.5
m_3	Goodfellas	drama	146	8.7
m_4	The Exorcist	horror	117	8.2
m_5	The Shining	horror	142	8.8
m_6	Inception	scifi	152	8.8



Example

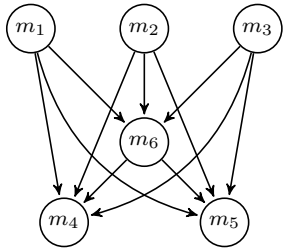
Prefer movies that their runtime are less than 150min.

$$t \succ_1 t' \equiv (\mathsf{runtime}(t) < 150) \wedge (\mathsf{runtime}(t') \geq 150)$$

Movie preferences (cont.)

Result of $w_{\succ_2}(\text{movie})$:

ID	Name	Genre	Runtime (min)	Rating
m_1	The Godfather	drama	175	9.2
m_2	The Irishman	drama	189	8.5
m_3	Goodfellas	drama	146	8.7
m_4	The Exorcist	horror	117	8.2
m_5	The Shining	horror	142	8.8
m_6	Inception	scifi	152	8.8



Example

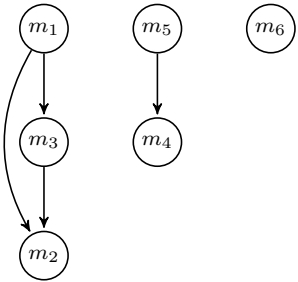
Prefer drama movies over scifi movies over horror movies.

$$
\begin{aligned}
t \succ_2 t' \quad \equiv \quad & \big((\text{genre}(t) = \text{drama}) \wedge (\text{genre}(t') = \text{scifi})\big) \vee \\
& \big((\text{genre}(t) = \text{scifi}) \wedge (\text{genre}(t') = \text{horror})\big) \vee \\
& \big((\text{genre}(t) = \text{drama}) \wedge (\text{genre}(t') = \text{horror})\big)
\end{aligned}
$$

Movie preferences (cont.)

Result of $w_{\succ_3}$(movie):

ID	Name	Genre	Runtime (min)	Rating
m_1	The Godfather	drama	175	9.2
m_2	The Irishman	drama	189	8.5
m_3	Goodfellas	drama	146	8.7
m_4	The Exorcist	horror	117	8.2
m_5	The Shining	horror	142	8.8
m_6	Inception	scifi	152	8.8



Example

Given two movies of the same genre, prefer the one with the highest rating.

$$t \succ_3 t' \equiv (\text{genre}(t) = \text{genre}(t')) \wedge (\text{rating}(t) > \text{rating}(t'))$$

Preference composition

Prioritized composition

- **Prioritized composition** $\succ_{C_1 \triangleright C_2}$ of two preference relations C_1 and C_2
- "prefer according to C_1 and if it is **inapplicable**, then prefer according to C_2"

$$x \succ_{C_1 \triangleright C_2} y \equiv (x \succ_{C_1} y) \vee ((x \sim_{C_1} y) \wedge (x \succ_{C_2} y)),$$

Pareto composition

- **Pareto composition** $\succ_{C_1 \otimes C_2}$ of two preference relations C_1 and C_2
- "prefer according to both C_1 and C_2 with **equal importance**"

$$x \succ_{C_1 \otimes C_2} y \equiv \big((x \succ_{C_1} y) \wedge (y \not\succ_{C_2} x)\big) \vee \big((x \succ_{C_2} y) \wedge (y \not\succ_{C_1} x)\big)$$

Movie preferences (cont.)

Result of $w_{\succ_{1\triangleright 3}}$(movie):

ID	Name	Genre	Runtime (min)	Rating
m_1	The Godfather	drama	175	9.2
m_2	The Irishman	drama	189	8.5
m_3	Goodfellas	drama	146	8.7
m_4	The Exorcist	horror	117	8.2
m_5	The Shining	horror	142	8.8
m_6	Inception	scifi	152	8.8

Example

1. Prefer movies that their runtime are less than 150min.
2. Prefer drama movies over scifi movies over horror movies.
3. Given two movies of the same genre, prefer the one with the highest rating.

*. Prioritize 1. over 3.

Movie preferences (cont.)

Result of $w_{\succ_{1\otimes 2}}$(movie):

ID	Name	Genre	Runtime (min)	Rating
m_1	The Godfather	drama	175	9.2
m_2	The Irishman	drama	189	8.5
m_3	Goodfellas	drama	146	8.7
m_4	The Exorcist	horror	117	8.2
m_5	The Shining	horror	142	8.8
m_6	Inception	scifi	152	8.8

Example

1. Prefer movies that their runtime are less than 150min.
2. Prefer drama movies over scifi movies over horror movies.
3. Given two movies of the same genre, prefer the one with the highest rating.

*. Prefer w.r.t. 1. and 2. with equal importance

Movie preferences (cont.)

Result of $w_{\succ_{1\otimes 2}}(\text{movie} - \{\text{m}_3\})$:

ID	Name	Genre	Runtime (min)	Rating
m_1	The Godfather	drama	175	9.2
m_2	The Irishman	drama	189	8.5
m_4	The Exorcist	horror	117	8.2
m_5	The Shining	horror	142	8.8
m_6	Inception	scifi	152	8.8

Example

1. Prefer movies that their runtime are less than 150min.
2. Prefer drama movies over scifi movies over horror movies.
3. Given two movies of the same genre, prefer the one with the highest rating.

*. Prefer w.r.t. 1. and 2. with equal importance

Preference Scores

Definition

A scoring function $f : R \to \mathbb{R}$ **represents** a preference relation $\succ_C$ over R if for all $\mathbf{x}, \mathbf{y} \in R$:

$$\mathbf{x} \succ_C \mathbf{y} \equiv f(\mathbf{x}) > f(\mathbf{y}),$$

Examples

1. Prefer movies according to their IMDB rating.
   $f_1(t) = \text{imdb\_rating}(t)$
2. Prefer according to both IMDB and RT ratings; more important is RT.
   $f_2(t) = 0.3 \cdot \text{imdb\_rating}(t) + 0.7 \cdot \text{rt\_rating}(t)$

Preference Scores (cont.)

- Not every preference relation can be expressed using a scoring function. Example:

Lexicographic order in $\mathbb{R} \times \mathbb{R}$

$$(x_1, y_1) \succ_{lo} (x_2, y_2) \equiv (x_1 > x_2) \vee \big((x_1 = x_2) \wedge (y_1 > y_2)\big)$$

- Intuition: The first dimension is **infinitely more important** than the second.

Theorem

There **does not exist** a function $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ such that for all $x_1, x_2, y_1, y_2 \in \mathbb{R}$:

$$(x_1, y_1) \succ_{lo} (x_2, y_2) \iff f(x_1, y_1) > f(x_2, y_2).$$

Preference Representation in the Semantic Web

The SPREFQL language (Troumpoukis et al., 2017)

- extension of SPARQL with a solution modifier to select the most preferred solutions
- SPREFQL queries can be rewritten into SPARQL 1.1
- syntax allows to recognize opportunities to apply specialized algorithms

```
SELECT ?film ?genre ?runtime WHERE {
  ?film a :film.
  ?film :genre ?genre.
  ?film :runtime ?runtime.
}
PREFER (?film1 ?genre1 ?runtime1)
TO     (?film2 ?genre2 ?runtime2)
IF (?genre1 = ?genre2 && ?runtime1 > ?runtime2)
```

Representing Preferences in Higher-Order Logic Programming

Representing Preferences in Higher-Order Logic Programming

Intuition

- Preferences are binary **relations**
- Operators over preferences are operators that take **relations as parameters** (e.g. winnow).
- Preference relations and preference operators can be expressed in the **same language** using Higher-Order Logic Programming.

Representing base relations

Example

```
movie(theGodfather).
movie(theIrishman).
movie(goodfellas).
movie(theExorcist).
movie(theShining).
movie(inception).

% and for each movie, specify its genre, runtime, and rating
genre(theGodfather, drama).
runtime(theGodfather, 175).
rating(theGodfather, 9.2).

% ...
```

Representing preferences

Example

```
c1_pref(X,Y) :- runtime(X,N), runtime(Y,M), N < 150, M >= 150.
```

- Prefer X from Y if the runtime of X is less than 150 and the runtime of Y is greater than 150.

Example

```
c2_pref(X,Y) :- genre(X,drama), genre(Y,scifi).
c2_pref(X,Y) :- genre(X,drama), genre(Y,horror).
c2_pref(X,Y) :- genre(X,scifi), genre(Y,horror).
```

- Prefer X from Y if the genre of X is drama and the genre of Y is scifi or horror, or if the genre of X is scifi and the genre of Y is horror.

Representing preferences (cont.)

Example

```
c3_pref(X,Y) :- genre(X,D), genre(Y,D),
                rating(X,N), rating(Y,M), N > M.
```

- Prefer X from Y if they have the same genre and the rating of X is greater than that of Y.

Example

```
winnow(C,R)(X) :- R(X), \+ bypassed(C,R,X).
bypassed(C,R,X) :- R(Z), C(Z,X).
```

- An element X of a relation R is a "best" element according to a preference relation C if it is not "bypassed".
- An element X is "bypassed" if there exists an element Z of a relation R that is preferred from X according to a preference relation C.

Representing preferences (cont.)

Example

```
prioritized(C1,C2)(X,Y) :- C1(X,Y).
prioritized(C1,C2)(X,Y) :- indifferent(C1)(X,Y), C2(X,Y).
indifferent(C)(X,Y) :- \+ C(X,Y), \+ C(Y,X).
```

- `X` is preferred from `Y` according to a the prioritized composition of `C1` and `C2` if `X` is preferred according to `C1`, or if they are indifferent and `X` is preferred acording to `C2`.

Example

```
pareto(C1,C2)(X,Y) :- C1(X,Y), \+ C2(Y,X).
pareto(C1,C2)(X,Y) :- C2(X,Y), \+ C1(Y,X).
```

- `X` is preferred from `Y` according to a the Pareto composition of `C1` and `C2` if `X` is preferred according to `C1`, and `Y` is not preferred according to `C2` or the other way around.

Preference queries

Example

```
% Preference queries that correspond to the examples shown
% in previous slides

?- winnow(c1_pref, movie)(X).
?- winnow(c2_pref, movie)(X).
?- winnow(c3_pref, movie)(X).

?- winnow(prioritized(c1_pref,c3_pref), movie)(X).
?- winnow(pareto(c1_pref,c2_pref), movie)(X).
?- winnow(pareto(c1_pref,c2_pref), minus(movie,goodfellas))(X).
```

3-element set preferences

ID	Name	Genre	Runtime (min)	Rating
m_1	The Godfather	drama	175	9.2
m_2	The Irishman	drama	189	8.5
m_3	Goodfellas	drama	146	8.7
m_4	The Exorcist	horror	117	8.2
m_5	The Shining	horror	142	8.8
m_6	Inception	scifi	152	8.8

ID	Total Runtime (min)	Has scifi?
$\{m_1, m_2, m_3\}$	510	no
$\{m_1, m_2, m_4\}$	481	no
$\{m_1, m_2, m_5\}$	506	no
$\{m_1, m_2, m_6\}$	516	yes
...	...	...
$\{m_4, m_5, m_6\}$	411	yes

Examples

1. I want to watch three movies and I prefer the total runtime be the lowest possible.
2. I want to watch three movies and I prefer to watch at least one scifi movie.
3. Prioritize 2 over 1.

Representing set preferences in Higher-Order Logic Programming

Intuition

- Sets are **relations**.
- Operators over preferences over sets are operators that take **relations over relations as parameters**.
- Set preferences can be represented elegantly in Higher-Order Logic Programming.

Representing set preferences

Example

```
runtime_sum(R,0) :- empty(R).
runtime_sum(R,N) :- R(X), runtime(X,M),
                    runtime_sum(minus(R,X),K),
                    N is M + K.

spref_1(S,Q) :- runtime_sum(S,N), runtime_sum(Q,M), N < M.
```

- Prefer `S` from `Q` if the sum of of the runtimes of all elements of `S` is less than that of `Q`.
- The `runtime_sum` has a similar structure to `size`.

Representing set preferences

Example

```
has_scifi(S) :- S(X), genre(X,scifi).

spref_2(S,Q) :- has_scifi(S), \+ has_scifi(Q).
```

- Prefer `S` from `Q` if `S` contains an element `X` that its genre is scifi, and `Q` does not contain such an element `X`.

Remark

- The definitions of remaining preference operators (`winnow`, `prioritized`, `pareto`, etc.) remain the same in the case of set preferences.

Set preference queries

- We can define `subset3(R)(S)` for generating the relation of all candidate sets of `R`.

Example

```
% Preference queries that correspond to the previous example

?- winnow(spref_1, subset3(movie))(X).
?- winnow(spref_2, subset3(movie))(X).

?- winnow(prioritized(spref_1,spref_2), subset3(movie))(X).
```

WIP

Preference Use Cases?

- Lack of use cases and datasets where preferences are more complex than looking for
  - a cheap hotel that is close to the seaside;
  - or multi-faceted book/movie recommendations
- Databases literature: emphasises efficient retrieval
- ???


Giga Campus
by Vodafone & Demokritos

Lefkippos Parking Data

- A log of timestamped occupied/available status for each of the 30 parking slots of the Lefkippos car park
  - 288844 log lines
  - containing 2845 instances of free-to-occupied transitions
  - during the period 1 Jan 2022 - 30 Apr 2022
- 53968 preferance pairs of a slot having been preferenced over each one of the other slots available at that time
- Tried to update the above statistcs with new data since May
  - Sensors stopped reacting some time in early May
  - Log lines are being produced, but stuck to the same value

WIP: Manual rules

- Encode some base preference relations (using common sense): e.g., given two slots:
  - prefer the one closest to the entrance (pref_distance)
  - prefer the one that has empty slots available on either side (pref_sparse)
  - prefer the one first seen while entering the parking (pref_precedes)
- Construct combinations using previously defined preference relations and preference compositions (e.g., prioritized) and calculate positive and negative preference coverage:

Preference	POS cover	NEG cover
pref_distance	0.45	0.50
pref_sparse	0.19	0.22
pref_precedes	0.36	0.38
prioritized(pref_distance ,...)	0.46	0.52
prioritized(pref_sparse ,...)	0.43	0.46
prioritized(pref_precedes,...)	0.45	0.45

WIP: ILP

- Encode base preference relations, preference compositions, and context predicates as background knowledge (and transform it in Prolog)
- WIP: Use Aleph to learn a theory (based on positive, negative preference examples)

```
[theory]

[Rule 2] [Pos cover = 15212 Neg cover = 13225]
target(A,B,C) :—
  cons_pref(distance,D), apply(D,B,A,C).

Accuracy = 0.5182621962427852
[Training set summary] [[15212,13225,39190,41177]]
[time taken] [24063.435770162]
[total clauses constructed] [235588]
```

Thank you for your attention