Exploiting Preference Rules for Querying Databases

Sergio Greco, Cristian Molinaro, Francesco Parisi

D.E.I.S.
Università della Calabria

{greco, cmolinaro,fparisi}@deis.unical.it
Motivations

Large volume of available data

Expressing preferences on alternative scenarios is natural

Information Filtering and Extraction

Preferences can be exploited to reduce the volume of data presented to the user, thus improving the query answering
Main Idea

• We answer to queries by deriving only supported and preferred information

DB: { beef, red-wine, white-wine }

P: { fruit-salad ← white-wine,
    pie ← red-wine,
    biscuits ← red-wine }

Φ: { red-wine > white-wine ← beef,
    pie > biscuits ← }
Main Idea

- We answer to queries by deriving only supported and preferred information

DB: { beef, red-wine, white-wine }  
beef  red-wine  white-wine

P: { fruit-salad ← white-wine,  
pie ← red-wine,  
biscuits ← red-wine }  

Φ: { red-wine > white-wine ← beef,  
pie > biscuits ← }
Main Idea

- We answer to queries by deriving only supported and preferred information

DB: { beef, red-wine, white-wine }  
bread  red-wine  white-wine

P: { fruit-salad ← white-wine,  
pie ← red-wine,  
biscuits ← red-wine }

Φ: { red-wine > white-wine ← beef,  
pie > biscuits ← }
Main Idea

- We answer to queries by deriving only supported and preferred information

\[ \text{DB: \{ beef, red-wine, white-wine \}} \]
\[ \text{P: \{ fruit-salad } \leftarrow \text{ white-wine,} \]
\[ \quad \text{pie } \leftarrow \text{ red-wine,} \]
\[ \quad \text{biscuits } \leftarrow \text{ red-wine } \}
\[ \Phi: \{ \text{red-wine } \succ \text{ white-wine } \leftarrow \text{ beef,} \]
\[ \quad \text{pie } \succ \text{ biscuits } \leftarrow \} \]
Main Idea

- We answer to queries by deriving only supported and preferred information

\[
\text{DB: } \{ \text{beef}, \text{red-wine}, \text{white-wine} \}
\]
\[
\text{P: } \{ \text{fruit-salad }\leftarrow \text{white-wine}, \\
\text{pie }\leftarrow \text{red-wine}, \\
\text{biscuits }\leftarrow \text{red-wine} \}
\]
\[
\Phi: \{ \text{red-wine }\succ \text{white-wine }\leftarrow \text{beef}, \\
\text{pie }\succ \text{biscuits }\leftarrow \}
\]

Answer=$\{ \text{beef, red-wine, pie} \}$
Preference Rules

• A *preference rule* is of the form

\[ A > C \leftarrow B_1, \ldots, B_m, \ not \ B_{m+1}, \ldots, \ not \ B_n, \ \varnothing \]

– *A* is *preferable to* *C* if the body of the rule is *true*
– *C* is *dominated by* *A* if the body of the rule is *true*

\[
\text{red-wine} > \text{white-wine} \leftarrow \text{beef} \quad \text{beef} \quad \text{red-wine} \quad \text{white-wine}
\]
Preference Rules

• A preference rule is of the form

\[ A > C \leftarrow B_1, \ldots, B_m, \text{not } B_{m+1}, \ldots, \text{not } B_n, \varphi \]

- A is preferable to C if the body of the rule is true
- C is dominated by A if the body of the rule is true
- dominated atoms cannot be used to infer new information

red-wine > white-wine \leftarrow beef

beef red-wine white-wine

white-wine is dominated by red-wine
Preference Rules

- A preference rule is of the form

\[ A > C \leftarrow B_1, \ldots, B_m, \text{not } B_{m+1}, \ldots, \text{not } B_n, \varnothing \]

- \( A \) is preferable to \( C \) if the body of the rule is true
- \( C \) is dominated by \( A \) if the body of the rule is true
- dominated atoms cannot be used to infer new information

\[ P: \{ \text{fruit-salad} \leftarrow \text{white-wine}, \text{pie} \leftarrow \text{red-wine}, \text{biscuits} \leftarrow \text{red-wine} \} \]
Preference Rules

- A *preference rule* is of the form

\[ A > C \leftarrow B_1, \ldots, B_m, \text{not } B_{m+1}, \ldots, \text{not } B_n, \varphi \]

- *A is preferable to* $C$ if the body of the rule is *true*
- *C is dominated by* $A$ if the body of the rule is *true*
- *dominated atoms* cannot be used to infer new information

$P$: \{ fruit-salad $\leftarrow$ white-wine, 
pie $\leftarrow$ red-wine, 
biscuits $\leftarrow$ red-wine \}
Preference Rules

- A preference rule is of the form
  \[ A > C \leftarrow B_1, \ldots, B_m, \text{not } B_{m+1}, \ldots, \text{not } B_n, \varphi \]

  - \( A \) is preferable to \( C \) if the body of the rule is true
  - \( C \) is dominated by \( A \) if the body of the rule is true
  - dominated atoms cannot be used to infer new information

P: \{ fruit-salad \leftarrow white-wine,
  pie \leftarrow red-wine,
  biscuits \leftarrow red-wine \}
Preferences on Base Atoms

• Preference program $\Phi$

$\Phi: \{ \rho_1 = \text{beef} > \text{fish} \leftarrow, \rho_2 = \text{white-wine} > \text{red-wine} \leftarrow \text{fish}, \rho_3 = \text{red-wine} > \text{white-wine} \leftarrow \text{beef} \}$

– intuitively, the evaluation of $\rho_2$ and $\rho_3$ depends on the evaluation of $\rho_1$

• $\Phi$ is layered as follows:

<table>
<thead>
<tr>
<th>Layer 0</th>
<th>Layer 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>${ \rho_1 }$</td>
<td>${ \rho_2, \rho_3 }$</td>
</tr>
</tbody>
</table>

DB: $\{ \text{beef, fish, red-wine, white-wine} \}$
Preferences on Base Atoms

- Preference program $\Phi$

  $\Phi$: \{ $\rho_1 = \text{beef} > \text{fish} \leftarrow$, \\
  $\rho_2 = \text{white-wine} > \text{red-wine} \leftarrow \text{fish}$, \\
  $\rho_3 = \text{red-wine} > \text{white-wine} \leftarrow \text{beef} \}$

  - intuitively, the evaluation of $\rho_2$ and $\rho_3$ depends on the evaluation of $\rho_1$

- $\Phi$ is layered as follows:

  Layer 0: \{ $\rho_1$ \}  \\
  Layer 1: \{ $\rho_2$, $\rho_3$ \}

DB: \{ beef, fish, red-wine, white-wine \}
Preferences on Base Atoms

• Preference program $\Phi$

\[ \Phi: \{ \rho_1 = \text{beef} > \text{fish} \leftarrow, \]
\[ \rho_2 = \text{white-wine} > \text{red-wine} \leftarrow \text{fish}, \]
\[ \rho_3 = \text{red-wine} > \text{white-wine} \leftarrow \text{beef} \} \]

– intuitively, the evaluation of $\rho_2$ and $\rho_3$ depends on the evaluation of $\rho_1$

• $\Phi$ is layered as follows:

Layer 0: \{ $\rho_1$ \} \rightarrow \text{beef fish red-wine white-wine}

Layer 1: \{ $\rho_2, \rho_3$ \} \rightarrow \text{beef red-wine white-wine}

DB: \{ \text{beef, fish, red-wine, white-wine} \} \quad \text{Answer} = \{ \text{beef, red-wine} \}
Preferences on Base Atoms

• Preference program $\Phi$

$$\Phi: \{ \begin{align*} &\rho_1 = \text{beef} > \text{fish} \leftarrow, \\ &\rho_2 = \text{white-wine} > \text{red-wine} \leftarrow \text{fish}, \\ &\rho_3 = \text{red-wine} > \text{white-wine} \leftarrow \text{beef} \} \end{align*}$$

– intuitively, the evaluation of $\rho_2$ and $\rho_3$ depends on the evaluation of $\rho_1$

• $\Phi$ is layered as follows:

  Layer 0 : $\{ \rho_1 \}$  \quad \longrightarrow \quad \begin{align*} &\text{beef} \times \text{fish} \times \text{red-wine} \times \text{white-wine} \end{align*}

  Layer 1 : $\{ \rho_2 , \rho_3 \}$  \quad \longrightarrow \quad \begin{align*} &\text{beef} \quad \text{red-wine} \quad \text{white-wine} \times \times \times \end{align*}

• It is possible to define sufficient conditions which guarantee that the set of preference rules $\Phi$ can be partitioned into layers
General Preferences

• Preferences on both base and derived atoms

• Stratified semantics
  – a program P is partitioned into strata
  – preference rules are associated with strata of P
  – for each stratum of P, its preference rules are divided into layers
  – P is evaluated by computing one stratum at a time
    • for each stratum of P, the associated preference rules are applied one layer at a time
General Preferences

• (Stratified) Datalog program P

P: Lunch (X) ← Menu (X)
Dinner (X) ← Menu (X), not Lunch (X)
Dinner (fruit-salad) ← Dinner (white-wine)
Dinner (ice-cream) ← Dinner (white-wine)
Dinner (pie) ← Dinner (red-wine).

• Preference program Φ

Φ: ρ₁ = Lunch (beef) > Lunch (fish) ←,
   ρ₂ = Lunch (red-wine) > Lunch (white-wine) ← Lunch (beef)
   ρ₃ = Lunch (white-wine) > Lunch (red-wine) ← Lunch (fish)
   ρ₄ = Dinner (fruit-salad) > Dinner (ice-cream) ← Dinner (fish)
   ρ₅ = Dinner (ice-cream) > Dinner (fruit-salad) ← Dinner (beef)
General Preferences

• (Stratified) Datalog program $P$

\[ P: \]
\[ \text{Lunch (X)} \leftarrow \text{Menu (X)} \quad \text{Stratum } S_1 \]
\[ \text{Dinner (X)} \leftarrow \text{Menu (X)}, \text{not Lunch (X)} \]
\[ \text{Dinner (fruit-salad)} \leftarrow \text{Dinner (white-wine)} \quad \text{Stratum } S_2 \]
\[ \text{Dinner (ice-cream)} \leftarrow \text{Dinner (white-wine)} \]
\[ \text{Dinner (pie)} \leftarrow \text{Dinner (red-wine)}. \]

• Preference program $\Phi$

\[ \Phi: \rho_1 = \text{Lunch (beef)} > \text{Lunch (fish)} \leftarrow, \]
\[ \rho_2 = \text{Lunch (red-wine)} > \text{Lunch (white-wine)} \leftarrow \text{Lunch (beef)} \]
\[ \rho_3 = \text{Lunch (white-wine)} > \text{Lunch (red-wine)} \leftarrow \text{Lunch (fish)} \]
\[ \rho_4 = \text{Dinner (fruit-salad)} > \text{Dinner (ice-cream)} \leftarrow \text{Dinner (fish)} \]
\[ \rho_5 = \text{Dinner (ice-cream)} > \text{Dinner (fruit-salad)} \leftarrow \text{Dinner (beef)} \]
General Preferences

• (Stratified) Datalog program $P$

\[ P: \text{Lunch} (X) \leftarrow \text{Menu} (X) \text{ Stratum } S_1 \]

Dinner (X) $\leftarrow$ Menu (X), not Lunch (X)
Dinner (fruit-salad) $\leftarrow$ Dinner (white-wine)
Dinner (ice-cream) $\leftarrow$ Dinner (white-wine)
Dinner (pie) $\leftarrow$ Dinner (red-wine).

• Preference program $\Phi$

\[ \Phi: \rho_1 = \text{Lunch (beef)} > \text{Lunch (fish)} \leftarrow, \]
\[ \rho_2 = \text{Lunch (red-wine)} > \text{Lunch (white-wine)} \leftarrow \text{Lunch (beef)} \]
\[ \rho_3 = \text{Lunch (white-wine)} > \text{Lunch (red-wine)} \leftarrow \text{Lunch (fish)} \]
\[ \rho_4 = \text{Dinner (fruit-salad)} > \text{Dinner (ice-cream)} \leftarrow \text{Dinner (fish)} \]
\[ \rho_5 = \text{Dinner (ice-cream)} > \text{Dinner (fruit-salad)} \leftarrow \text{Dinner (beef)} \]

preferences on atoms defined by $S_1$
General Preferences

• (Stratified) Datalog program P

P: Lunch (X) ← Menu (X)
Dinner (X) ← Menu (X), not Lunch (X)
Dinner (fruit-salad) ← Dinner (white-wine)
Dinner (ice-cream) ← Dinner (white-wine)
Dinner (pie) ← Dinner (red-wine).

Stratum $S_2$

• Preference program $\Phi$

$\Phi$: $\rho_1$ = Lunch (beef) > Lunch (fish) ←,
$\rho_2$ = Lunch (red-wine) > Lunch (white-wine) ← Lunch (beef)
$\rho_3$ = Lunch (white-wine) > Lunch (red-wine) ← Lunch (fish)
$\rho_4$ = Dinner (fruit-salad) > Dinner (ice-cream) ← Dinner (fish) preferences on atoms
$\rho_5$ = Dinner (ice-cream) > Dinner (fruit-salad) ← Dinner (beef) defined by $S_2$
General Preferences

- (Stratified) Datalog program $P$

\[
P: \text{Lunch (X)} \leftarrow \text{Menu (X)} \\
\text{Dinner (X)} \leftarrow \text{Menu (X)}, \text{not Lunch (X)} \\
\text{Dinner (fruit-salad)} \leftarrow \text{Dinner (white-wine)} \\
\text{Dinner (ice-cream)} \leftarrow \text{Dinner (white-wine)} \\
\text{Dinner (pie)} \leftarrow \text{Dinner (red-wine)}.
\]

- Preference program $\Phi$

\[
\Phi: \rho_1 = \text{Lunch (beef)} > \text{Lunch (fish)} \leftarrow, \\
\rho_2 = \text{Lunch (red-wine)} > \text{Lunch (white-wine)} \leftarrow \text{Lunch (beef)} \\
\rho_3 = \text{Lunch (white-wine)} > \text{Lunch (red-wine)} \leftarrow \text{Lunch (fish)} \\
\rho_4 = \text{Dinner (fruit-salad)} > \text{Dinner (ice-cream)} \leftarrow \text{Dinner (fish)} \\
\rho_5 = \text{Dinner (ice-cream)} > \text{Dinner (fruit-salad)} \leftarrow \text{Dinner (beef)}
\]

DB: \{ \text{Menu (beef), Menu (fish), Menu (red-wine), Menu (white-wine)} \}
General Preferences

• (Stratified) Datalog program $P$

$P$: $\text{Lunch} (X) \leftarrow \text{Menu} (X)$

Dinner (X) $\leftarrow$ Menu (X), not Lunch (X)
Dinner (fruit-salad) $\leftarrow$ Dinner (white-wine)
Dinner (ice-cream) $\leftarrow$ Dinner (white-wine)
Dinner (pie) $\leftarrow$ Dinner (red-wine).

• Preference program $\Phi$

$\Phi$: $\rho_1 = \text{Lunch (beef)} > \text{Lunch (fish)}$ \leftarrow,
$\rho_2 = \text{Lunch (red-wine)} > \text{Lunch (white-wine)}$ \leftarrow Lunch (beef)
$\rho_3 = \text{Lunch (white-wine)} > \text{Lunch (red-wine)}$ \leftarrow Lunch (fish)
$\rho_4 = \text{Dinner (fruit-salad)} > \text{Dinner (ice-cream)}$ \leftarrow Dinner (fish)
$\rho_5 = \text{Dinner (ice-cream)} > \text{Dinner (fruit-salad)}$ \leftarrow Dinner (beef)

DB: $\{ \text{Menu (beef)}, \text{Menu (fish)}, \text{Menu (red-wine)}, \text{Menu (white-wine)} \}$
General Preferences

• (Stratified) Datalog program P

P:
- Lunch (X) ← Menu (X)
- Dinner (X) ← Menu (X), not Lunch (X)
- Dinner (fruit-salad) ← Dinner (white-wine)
- Dinner (ice-cream) ← Dinner (white-wine)
- Dinner (pie) ← Dinner (red-wine).

• Preference program Φ

Φ:
- ρ₁ = Lunch (beef) > Lunch (fish)
- ρ₂ = Lunch (red-wine) > Lunch (white-wine) ← Lunch (beef)
- ρ₃ = Lunch (white-wine) > Lunch (red-wine) ← Lunch (fish)
- ρ₄ = Dinner (fruit-salad) > Dinner (ice-cream) ← Dinner (fish)
- ρ₅ = Dinner (ice-cream) > Dinner (fruit-salad) ← Dinner (beef)

Layer 0: { ρ₁ }
Layer 1: { ρ₂ , ρ₃ }

DB: { Menu (beef), Menu (fish), Menu (red-wine), Menu (white-wine) }
General Preferences

• **(Stratified) Datalog program P**

P: Lunch (X) ← Menu (X)

| Dinner (X) ← Menu (X), not Lunch (X) |
| Dinner (fruit-salad) ← Dinner (white-wine) |
| Dinner (ice-cream) ← Dinner (white-wine) |
| Dinner (pie) ← Dinner (red-wine). |

<table>
<thead>
<tr>
<th>Menu</th>
<th>beef</th>
<th>fish</th>
<th>red-wine</th>
<th>white-wine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lunch</td>
<td></td>
<td></td>
<td></td>
<td>(X)</td>
</tr>
<tr>
<td>Dinner</td>
<td>fish</td>
<td>white-wine</td>
<td>fruit-salad</td>
<td>ice-cream</td>
</tr>
</tbody>
</table>

• **Preference program Φ**

Φ: ρ₁ = Lunch (beef) > Lunch (fish) ←,
ρ₂ = Lunch (red-wine) > Lunch (white-wine) ← Lunch (beef)
ρ₃ = Lunch (white-wine) > Lunch (red-wine) ← Lunch (fish)
ρ₄ = Dinner (fruit-salad) > Dinner (ice-cream) ← Dinner (fish)
ρ₅ = Dinner (ice-cream) > Dinner (fruit-salad) ← Dinner (beef)

DB: { Menu (beef), Menu (fish), Menu (red-wine), Menu (white-wine) }
General Preferences

• (Stratified) Datalog program $P$

$$P: \text{Lunch}(X) \leftarrow \text{Menu}(X)$$
$$\quad \text{Dinner}(X) \leftarrow \text{Menu}(X), \text{not Lunch}(X)$$
$$\quad \text{Dinner}(\text{fruit-salad}) \leftarrow \text{Dinner}(\text{white-wine})$$
$$\quad \text{Dinner}(\text{ice-cream}) \leftarrow \text{Dinner}(\text{white-wine})$$
$$\quad \text{Dinner}(\text{pie}) \leftarrow \text{Dinner}(\text{red-wine}).$$

• Preference program $\Phi$

$$\Phi: \rho_1 = \text{Lunch}(\text{beef}) > \text{Lunch}(\text{fish}) \leftarrow ,$$
$$\quad \rho_2 = \text{Lunch}(\text{red-wine}) > \text{Lunch}(\text{white-wine}) \leftarrow \text{Lunch}(\text{beef})$$
$$\quad \rho_3 = \text{Lunch}(\text{white-wine}) > \text{Lunch}(\text{red-wine}) \leftarrow \text{Lunch}(\text{fish})$$
$$\quad \rho_4 = \text{Dinner}(\text{fruit-salad}) > \text{Dinner}(\text{ice-cream}) \leftarrow \text{Dinner}(\text{fish})$$
$$\quad \rho_5 = \text{Dinner}(\text{ice-cream}) > \text{Dinner}(\text{fruit-salad}) \leftarrow \text{Dinner}(\text{beef}).$$

Layer 0: $\{ \rho_4, \rho_5 \}$

DB: $\{ \text{Menu}(\text{beef}), \text{Menu}(\text{fish}), \text{Menu}(\text{red-wine}), \text{Menu}(\text{white-wine}) \}$
General Preferences

- (Stratified) Datalog program $P$

$P$: Lunch ($X$) ← Menu ($X$)
   Dinner ($X$) ← Menu ($X$), not Lunch ($X$)
   Dinner (fruit-salad) ← Dinner (white-wine)
   Dinner (ice-cream) ← Dinner (white-wine)
   Dinner (pie) ← Dinner (red-wine).

- Preference program $\Phi$

$\Phi$: $\rho_1 =$ Lunch (beef) > Lunch (fish) ←,
   $\rho_2 =$ Lunch (red-wine) > Lunch (white-wine) ← Lunch (beef)
   $\rho_3 =$ Lunch (white-wine) > Lunch (red-wine) ← Lunch (fish)
   $\rho_4 =$ Dinner (fruit-salad) > Dinner (ice-cream) ← Dinner (fish)
   $\rho_5 =$ Dinner (ice-cream) > Dinner (fruit-salad) ← Dinner (beef)

The answer to the prioritized query $<$ Dinner, $P$, $\Phi$ $>$ is
   $\{ $ Dinner (fish), Dinner (white-wine), Dinner (fruit-salad) $\}$
Well-Formed Queries

- A prioritized query \(<q, P, \Phi>\) is well-formed if
  - \(\Phi\) is layered, and
  - for each \(A > C \leftarrow B_1, \ldots, B_m, \text{not } B_{m+1}, \ldots, \text{not } B_n\), it holds that \(A, B_1, B_m\) do not depend on \(C\) in \(P\)

```
DB: { white-wine, red-wine }
P: { beef \leftarrow white-wine }
\Phi: { red-wine > white-wine \leftarrow beef }
```
Complexity Result

Let DB be a database and Q = < q, P, Φ > be a well-formed prioritized query.

The computational complexity of evaluating Q on DB is polynomial time.
Conclusions

• We have presented prioritized queries
  – preferences can be defined on both base and derived atoms

• A stratified semantics for prioritized queries has been introduced

• The computational complexity of evaluating prioritized queries is still polynomial
Thank you!

...any questions?
Layers

• A (ground) preference program $\Phi$ is layered if it is possible to partition $\Phi$ into $n$ layers as follows:
  – for each atom $C$ such that there is no rule $A > C \leftarrow B_1, \ldots, B_m, \text{not } B_{m+1}, \ldots, \text{not } B_n$, $\text{layer}(C)=0$;
  – for each atom $C$ such that there is a rule $A > C \leftarrow B_1, \ldots, B_m, \text{not } B_{m+1}, \ldots, \text{not } B_n$, $\text{layer}(C) > \max \{ \text{layer}(B_1), \ldots, \text{layer}(B_n), 0 \}$ and $\text{layer}(C) \geq \text{layer}(A)$;
  – the layer of a preference rule $A > C \leftarrow B_1, \ldots, B_m, \text{not } B_{m+1}, \ldots, \text{not } B_n$, is $\text{layer}(C)$;
  – $\Phi[i]$ consists of all preference rules having layer $i$

• It is possible to define sufficient conditions which guarantee that the set of preference rules $\Phi$ can be partitioned into layers
Prioritized query

- A prioritized query is a triplet \(<q, P, \Phi>\),
  - \(q\) is a predicate symbol denoting the output relation,
  - \(P\) is a (stratified) Datalog program
  - \(\Phi\) is a preference program
Well-Formed Queries

- A prioritized query \(<q, P, \Phi>\) is well formed if
  - the ground transitive closure of \(\Phi\) is layered, and
  - for each \(A > C \leftarrow B_1, \ldots, B_m, \text{not } B_{m+1}, \ldots, \text{not } B_n\), it holds that \(A, B_1, B_m\) do not depend on \(C\) in \(P\)

Proof:

\[\text{DB: } \{\text{white-wine}, \text{red-wine}\}\]
\[\text{P: } \{\text{beef} \leftarrow \text{white-wine}\}\]
\[\Phi: \{\text{red-wine} \succ \text{white-wine} \leftarrow \text{beef}\}\]
Naive Translation

Φ: { red-wine > white-wine ← beef
    white-wine > red-wine ← fish }

white-wine’ ← white-wine, not X
X ← red-wine’, beef’
red-wine’ ← red-wine
beef’ ← beef
red-wine’ ← red-wine, not Y
Y ← white-wine’, fish’
white-wine’ ← white-wine
fish’ ← fish

results in a non-stratified program

white-wine’
not
X
Y
not
red-wine’