Foundations of Preference-based Queries

Periklis P. Georgiadis
Dept. of Computer Science,
University of Crete

Motivation

- Preference-based Queries
  - lay users have no clear knowledge of database content and/or schemata
  - they usually describe desirable properties of the objects they would like in the query result
  - system to decide on the best answer approximation and its order, instead of seeking for perfect matches

- Objective vs. Subjective
  - system-oriented queries: actual information resource content
  - user-oriented queries: user perception of content but also of the matching functionality
A student (the user) needs to decide what courses to take and in what order, so that (s)he achieves certain competences. Each course leads to one or more competences. A competence may be acquired through alternative (i.e., disjunctively) courses. Competences may be partially (pre)ordered, representing preference importance on behalf of the user, or an ontology, containing part_of or is_a_prerequisite relations, on behalf of commonly agreed conceptualization. Further partial orders may be composed into a query, reflecting, for instance, the temporal ordering of courses, or a prerequisite relation among the courses themselves.
Curriculum Personalization

![Diagram of relationships between courses and comp C1, C2, C3, C4, C5, C6, C7 and their leads to relationships]

Query: \((\text{comp}4, \leq)\)

Towards Preference-based Queries

- A preference is a partial preorder relation \(\leq\) over a set of objects; by \(x \leq y\) we denote that \(y\) is equally or more preferable than \(x\)
  - considering \(x\) and \(y\) equivalent \((x \sim y)\), iff \(x \leq y\) and \(y \leq x\), allows \(\leq\) to generate a partial order of equivalence classes, a.k.a. alternatives

- A preference-based query over a data collection is a triple \(Q := (q, \leq, k)\), where:
  - \(q\): a regular query providing filtering conditions,
  - \(\leq\): a partial preorder relation over the values of attributes (or relationships),
  - \(k\): a positive integer indicating a top-k answer request
    - \(q\) and \(\leq\) may refer to different attributes or relationships of the filtered data collection
The problem: Given a partial order on attribute values (e.g., competences), create a partial order on the objects of a collection (e.g., courses)

A first approach:

- Exploit inversion of the attribute space to the object space (i.e., functions)
- But, having multi-valued attributes, or 1-n or n-m relationships leads to relations

Thus, we need more elegant semantics of object ordering

- Due to relations (and possible equivalence classes) we need to order sets of objects

Towards Ordering of Object Sets

- Let \( X = \{ x_1, x_2, \ldots, x_n \} \) be a collection of objects. For every object \( x_i \) and attribute (or relationship) \( \text{prop} \), let \( \text{prop}(x_i) = \{ y_1, y_2, \ldots, y_m \} \) be the set of its values (or its related objects) from some sort \( Y \)

- Let \( \leq \) be a partial order relation over \( Y \). Then, we may define four different relations on \( X \) as follows:
  \[
  x \leq_{\forall} x' \text{ iff } \forall y \in \text{prop}(x) \exists y' \in \text{prop}(x') y \leq y'
  \]
  \[
  x \leq_{\exists} x' \text{ iff } \exists y \in \text{prop}(x) \forall y' \in \text{prop}(x') y \leq y'
  \]
  \[
  x \leq_{\forall\forall} x' \text{ iff } \forall y \in \text{prop}(x) \forall y' \in \text{prop}(x') y \leq y'
  \]
  \[
  x \leq_{\exists\exists} x' \text{ iff } \exists y \in \text{prop}(x) \exists y' \in \text{prop}(x') y \leq y'
  \]

- Each of these \( \leq_{**} \) relations is a partial preorder, becoming a partial order of equivalence classes as explained earlier
Curriculum Personalization (Cont.)

Query: \((\text{comp}4, \leq)\)

Composition of Partial Orders

- add term attribute; user prefers courses according to their term sequence
- partial order composition: lexicographic or coordinate-wise order of their product, the former providing an importance or priority on the constituent orders
Lexicographic Composition

\( \leq_{\forall \exists} \leq_{\exists \exists} \leq_{\forall \forall} \leq_{\exists \exists} \)

\[ c_5 \leq c_1, c_5 \leq c_3 \]
\[ c_2 \leq c_1, c_2 \leq c_3 \]
\[ c_1 \leq c_4, c_3 \leq c_4 \]

\[ c_4 \]
\[ \quad \quad \quad c_1 \quad c_3 \]
\[ c_2 \quad c_5 \]

Coordinate-wise Composition

\( \leq_{\forall \exists} \leq_{\exists \exists} \leq_{\forall \forall} \leq_{\exists \exists} \)

\[ c_2 \leq c_3, c_5 \leq c_1 \]

\[ c_4 \]
\[ \quad \quad \quad c_3 \quad c_1 \]
\[ c_2 \quad c_5 \]
Curriculum Personalization (Cont.)

- replace the term attribute, with a prerequisite_courses m-n relationship, which conforms to a partial order

- We consider a bottom element in the prerequisite_courses partial order for those objects that do not have any lower ones defined

Lexicographic Composition

- \( \leq_{\forall \exists} \)
- \( \leq_{\exists \forall} \)
- \( \leq_{\forall \forall} \)
Suppose competence \( \text{comp4} \) did not appear in the data, but still was selected by the use, as it was among those offered by the system; this could be the case where the partial order on competences (also) represented a part_of ontology.

Re-running our example leads to similar results; thus, it is possible to express preferences on values / objects that do not directly appear in the data.

Single multi-valued attribute \( \text{comp} \):

Lexicographic composition \( \text{term} \times \text{comp} \):
More on Curriculum Personalization

- Coordinate-wise composition \( \text{comp} \times \text{term} \)

\[
\begin{align*}
\leq_{\forall \exists} & \quad \leq_{\exists \forall} \\
\leq_{\forall \forall} & \quad \leq_{\exists \exists} \\
\end{align*}
\]

- Lexicographic composition \( \text{prereq\_courses} \times \text{comp} \)

\[
\begin{align*}
\leq_{\forall \exists} & \quad \leq_{\exists \forall} \\
\leq_{\forall \forall} & \quad \leq_{\exists \exists} \\
\end{align*}
\]

Thesis Plan

- Preference model
  - semantics for partially ordered sets of objects
  - operators for preference composition
  - smooth integration of preferences with real data models

- Preference-based query evaluation & optimization
  - intuitive linguistic primitives to capture user preferences
  - avoid multiple scans of the same collection for ordering objects according to the user preferences

- Extensions
  - combining multiple users preferences, preference independence
Related work

A summary of Preference Modeling

Further remarks on the example

- 2 main differences to previous qualitative approaches:
  - we are able to manipulate 1-n or n-m relations
  - we target not just the best answers (maximal elements), but the whole partial order
- We are able to generalize preference composition; yet the semantics of our compositions need further exploration
  - It seems that which composition is best depends on the application and more precisely on the semantics of the involved partial orders
- A note on the preference independence hypothesis;
  - in real life examples, functional dependencies arise, violating this hypothesis (e.g., term order or prerequisite courses’ order are not irrelevant from competence order).
Thank you!
Questions?