Personalized Keyword Search

Drawbacks found

28.11.2016

ANNE JERONEN, ARMAND NOUBISIE, YUDIT PONCE
Introduction

Personalized keyword search

Drawbacks

Suggested solution
Introduction

- Georgia Koutrika, Alkis Simitsis, Yannis E. Ioannidis: Précis: The Essence of a Query Answer. ICDE 2006
- Kostas Stefanidis, Marina Drosou, Evaggelia Pitoura: PerK: Personalized Keyword Search in Relational Databases through Preferences. EDBT 2010
Personalized Keyword Search

- **Keyword search**
  - Enables users to find information without knowing the database schema or complicated queries
  - May return huge volumes of data

- **Benefits of personalization**
  - Enables returning personalized results to individual users or groups
  - Enables returning smaller number of results that are also more relevant to the user(s)
Personalized Keyword Search

- Personalization can be implemented in several ways:
  - Weights assigned to the edges of the database graphs (Koutrika et al. 2006)
  - Pairwise preferential ordering (Stefanidis et al. 2010)

- $Q = \{\text{western}\}$:
- $cp1 = \{\text{western}, \text{C. Eastwood} \succ \text{J. Wayne}\}$
Koutrika et al. (2006): Weights assigned to edges of the database graph.

• **Weights.** A weight, $w \in [0,1]$ , is assigned to each edge of the graph showing the significance of the bond between the corresponding nodes.

**Movie Database Schema:**

<table>
<thead>
<tr>
<th>Table</th>
<th>Columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>THEATRE</td>
<td>(tid, name, phone, region)</td>
</tr>
<tr>
<td>PLAY</td>
<td>(tid, mid, date)</td>
</tr>
<tr>
<td>MOVIE</td>
<td>(mid, title, year, did)</td>
</tr>
<tr>
<td>GENRE</td>
<td>(mid, genre)</td>
</tr>
</tbody>
</table>

**Movie Database Graph:**

• $w = 1$ strong relationship: if one node appears in an answer, then the other node appear as well.

• $w = 0$, occurrence of one node in an answer does not imply occurrence of the other one.
A directed join edge expresses the dependence of the left part of the join on the right part.

- Movies and genres are related but one may consider that genres are more dependent on movies than the other way around.
- For instance, the weight of the edge from GENRE to MOVIE is 1, while the weight of the edge from MOVIE to GENRE is 0.9.

answer regarding a genre

always contain

related movies

answer regarding a movie

not necessary contain

related genre
Koutrika et al. (2006)

PERSONALIZED KEYWORD SEARCH

CONSTRAINTS

DEGREE CONSTRAINT
1. The maximum number of attributes
2. The minimum weight on a relationship

CARDINALITY CONSTRAINT
1. The maximum number of tuples in the query result
Koutrika et al. (2006): Drawbacks

1. The maximum number of attributes
2. The minimum weight on a relationship

Q = \{Tampere\}
1. The maximum number of attributes = 2
2. The Minimum Weight = 1

Which ones will be selected?
Proposed Solution

Attribute Preferences: setting priorities among the attributes to be displayed in the query result.

{region, name, phone}, 1

Q = {Tampere}
attributes = 2
weight = 1

<table>
<thead>
<tr>
<th>region</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1</td>
<td>Tampere</td>
</tr>
<tr>
<td>t2</td>
<td>Tampere</td>
</tr>
<tr>
<td>t3</td>
<td>Helsinki</td>
</tr>
<tr>
<td>t4</td>
<td>Pori</td>
</tr>
</tbody>
</table>

THEATER
The maximum number of tuples in the query result

\[ \text{Q} = \{ \text{Tampere} \} \]

1. The maximum number of tuples = 10

**CARDINALITY CONSTRAINT**

**THEATER**

<table>
<thead>
<tr>
<th>tid</th>
<th>region</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1</td>
<td>Tampere</td>
<td>Cine Atlas</td>
</tr>
<tr>
<td>t2</td>
<td>Tampere</td>
<td>Plevna</td>
</tr>
<tr>
<td>t3</td>
<td>Helsinki</td>
<td>Tennispalatsi</td>
</tr>
<tr>
<td>t4</td>
<td>Tampere</td>
<td>Promenadi</td>
</tr>
<tr>
<td>t5</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>t20</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Which ones will be selected
To submit a query and keep only the top tuples, whose number is determined based on the cardinality constraint.

```
SELECT top 10 FROM THEATER WHERE..... (10 cardinality constraint)
```

Proposed Solution

- **Adding Preferences:**
  1. **Quantitative approaches, Degree of Interest:** Preference(Condition, Score)
     - `theater.name = 'Plevna', 0.9`
     - `theater.name = 'Promenadi', 0.5`

```
Plevna is prefered over Promenadi
```

2. **Contextual Preference:** (Contextual, Preference): Internal Preference
   - **In the region Tampere, I prefer the theater whose name is ‘Plevna’**
   - `{name = ‘Plevna’ > name = ‘Promenadi’ | region = ‘Tampere’}`
Drawbacks: Pre-definition

- Personalized keyword search is based on pre-defined conditions
  
  (\{thriller\}, G. Oldman ≻ W. Allen) or (\{comedy\}, W. Allen ≻ G. Oldman)

- Finding new and interesting search results may be difficult if they are ruled out by the existing preferences.
  
  - How to find new data that doesn't have any similarity with the pre-defined preferences?
Drawbacks: Pre-definition
- Proposed Solution

- Relaxation of context parameters and allocation of multi-keyword choices
  - Extending the context of the preference model by allowing more keywords in the choice part.
For a profile $P$ and query $Q$, the associated set of preferences $P_Q$ may be empty (no existing preferences for $Q$). In this case, we can use those preferences whose context is more general than $Q$ to induce a preferential ordering.

(RELAXED CONTEXT).

Given a query $Q$ and a profile $P$, a set $C \subset Q$ is a relaxed context for $Q$ in $P$, if and only if,

(i) $\exists (C, \text{choice}) \in P$

and

(ii) $\nexists (C', \text{choice}') \in P$

such that, $C' \subset Q$ and $C \subset C'$.

Figure 3: Context lattice of preferences.
Multi-keywords Choices

Our model of preferences supports choices between two keywords. One may think of more complex preferences of the form \((C, \text{choice})\), where \(C \subseteq W\) and \(\text{choice} = (w_{l1} \land \ldots \land w_{lx}) \succ (w_{r1} \land \ldots \land w_{ry})\), \(w_{lj}, w_{rz}, 1 \leq j \leq x, 1 \leq z \leq y, \in W\). We shall refer to such preferences as composite contextual keyword preferences.

Methods and Techniques such as:

- Sharing JTS Algorithm: performs only a small fraction of the join operations.
- Top-k JTTs algorithm: Algorithm combines four metrics in determining the top-k results for a query \(q\), namely, preferential dominance, degree of relevance, coverage and diversity

Allow to evaluate the performance of the relaxation and the adding of new keywords in the preference definition.
Drawbacks: Cold Start Problem

- How are preferences created
  - Explicitly provided by the user
  - Created by a designer for groups of users
  - Automatically constructed based on previous user interactions
- More data = more information for constructing preferences
Drawbacks: Cold Start Problem - Suggested Solution

- Relaxing context parameters (Stefanidis et al. 2010)
  - Context-free parameters may be applied when more specific, contextual parameters do not exist
  - Priority is given to popular keywords over less popular ones
    - For example, after user has performed first queries, we see a preference for 
      \{(\text{action}), \ T. \ Cruise \succ \ B. \ Pitt\}
      
      Q1 (action, B. Pitt)
      Q2 (action, T. Cruise)
      Q3 (action, T. Cruise)
      Q4 (drama, B. Pitt)
  
- This solution does not apply for the approach in the Koutrika et al. 2006 paper as there are no contextual parameters to relax
Drawbacks: Changing Preferences

- User preferences may change
  - Slowly over time
    - Normal changes in tastes as people age
  - Suddenly
    - A life-changing event such as having a child
  - Periodically
    - Summer or winter, alone or with friends
- How to adapt to these changes and change the user preferences accordingly?
Drawbacks: Changing Preferences
- Suggested Solution

- Sudden changes are caught quickly
- Using time as an external context enables catching periodic changes but not slow changes in preferences
- Preferences could be limited to include an exact amount of data, or data from a specific time period.
  - Preferences from the previous 1000 searches
    - Problem: Users that run several queries using different keywords
  - Preferences from the searches made in the previous month
    - Problem: Users that make searches rarely
  - A combination of these?
Thank you!

Next week: Related Works