QueRIE: Collaborative Database Exploration

Magdalini Eirinaki\textsuperscript{1}, Suju Abraham\textsuperscript{1}, Neoklis Polyzotis\textsuperscript{2}, Naushin Shaikh\textsuperscript{1}

IEEE Transactions on Knowledge and Data Engineering (TKDE), 26(7), July 2014
Motivation

• Scientific disciplines use relational DBMS for storage and retrieval of information
  – Biologists (e.g. Genome browser)
  – Astronomers (e.g. Skyserver)
  – Chemists (e.g. PubChem)

• Typical users are not SQL experts

• Scientific datasets increase in size
Motivation (cont’d)

“A result of heavy usage has also lead to a lot of tables generated in the warehouse and this has in turn tremendously increased the need for data discovery tools, especially for new users.”

Motivation (cont’d)

• Even when users have the ability to issue complex queries over large data sets, the task of knowledge discovery remains a big challenge as users:
  – may not be familiar with the database schema,
  – may overlook queries that retrieve relevant data, or
  – might not have the required expertise to formulate such queries.

• Moreover, size of data makes an exhaustive exploration of such databases practically infeasible.

Our goal: Assist users in the interactive exploration of a large database
Web Collaborative Filtering

**Example:** Movie Recommendations

If Alice and Bob **both** like movie X and Alice likes movie Y

then

Bob is likely to be interested in seeing movie Y

If Alice and Bob **both** pose similar queries

then

Bob is likely to be interested in other queries posed by Alice
Which parts of the database are interesting to the user?

How do we generate meaningful queries?

How do we define the similarity metric between users?
Roadmap

• Introduction
• The QueRIE Framework & Tuple-based recommendations
• Fragment-based recommendations
• Comparison of approaches
• The QueRIE Prototype
• Conclusions
The QueRIE Framework
Tuple-based recommendations

• (In SSDBM’09, HDMS’09, ICDM’10)
• User profiles (session summaries) represented as weighted vectors – each coordinate corresponding to a distinct database tuple
• Session-tuple matrix used as input to the recommendation algorithm
• Predict using the formula:
  
  \[ S^{pred} = \alpha * S_0 + (1 - \alpha) * \frac{\sum_{1\leq i \leq h} \text{sim}(S, S_i) * S_i}{\sum_{1\leq i \leq h} \text{sim}(S, S_i)} \]

• Recommend queries that retrieve tuples of high predicted weights calculating, for each query Q, the similarity:
  
  \[ \text{sim}(S_Q, S^{pred}) \]
Discussion

• Tuples capture user interest in a very fine level of detail
  – Very high prediction accuracy

• Downsides:
  – Increased complexity as session summaries grow linearly with size of DB
  – Lack of scalability for real-time recommendations due to user-based approach

• Randomized sketching techniques (e.g. MinHash) improve computational efficiency at the cost of precision
Roadmap

• Introduction
• The QueRIE Framework & Tuple-based recommendations
• Fragment-based recommendations
• Comparison of approaches
• The QueRIE Prototype
• Conclusions
Fragment-based recommendations

• Reduce computational cost by:
  – Representing queries in a coarser level of detail
  – Following an item-based CF-like approach

• IDEA: Recommend queries whose syntactical features (fragments) match the queries of the current user
Fragment-based recommendations

---

**Query 1:**
```
SELECT count(*) FROM region WHERE type like 'tiprimary'
```

**Query 2:**
```
SELECT count(distinct id) FROM region WHERE type like 'tiprimary'
```

**Query 3:**
```
SELECT id, count(*) FROM region WHERE type like 'tiprimary'
```

**Query 4:**
```
SELECT id, count(*) FROM region WHERE type like 'tiprimary'
```

**Query 5:**
```
WHERE type like 'tiprimary' GROUP BY id
```

**Query 6:**
```
SELECT id, count(*) FROM region
```

**Query 7:**
```
WHERE type like 'tiprimary' GROUP BY id HAVING count(*) >= 1
```
Fragments

- Relaxed representation
  - E.g. relax all WHERE clauses by converting numerical data and strings to NUM/HEXNUM and STR fragments
  
- Increase cardinality / make input matrix more dense
  - So, for example when two users query the same table and attributes, using slightly different filtering conditions, the algorithm will consider them as similar

<table>
<thead>
<tr>
<th>QUERY</th>
<th>QUERY FRAGMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECT id, count(<em>) FROM region WHERE type like 'primary' GROUP BY id HAVING count(</em>)&gt; 1</td>
<td>ID, COUNT(<em>), REGION, REGION.TYPE PATMATCH, COUNT(</em>) COMPARE NUM.</td>
</tr>
</tbody>
</table>

- Orthogonal decision to our framework
Fragment-based recommendations

- (VLDB’10, PersDB’10, TKDE’13)
- User profiles (session summaries) represented as weighted vectors – each coordinate corresponding to a distinct query fragment
- user-fragment matrix used as input to the recommendation algorithm
- Predict using the formula:

\[
S^{\text{pred}}[\phi] = \frac{\sum_{\rho \in R} S_0[\rho] \cdot \text{sim}(\rho, \phi)}{\sum_{\rho \in R} \text{sim}(\rho, \phi)}
\]

- Recommend queries have high overlap with the predicted session calculating, for each query Q, the similarity:

\[
\text{sim}(S_Q, S^{\text{pred}})
\]
Roadmap

• Introduction
• The QueRIE Framework & Tuple-based recommendations
• Fragment-based recommendations
• Comparison of approaches
• The QueRIE Prototype
• Conclusions
Experimental Setup

- **SkyServer Dataset**
  - Database Size: 2.6TB
  - #Sessions (training/test): 720 (412/45)
  - #Queries: 6713
  - #Distinct Queries: 4037
  - #Distinct witnesses: 13,607,430
  - #Distinct query fragments: 3212
  - Avg. number of queries per session: 9.3
  - Min. number of queries per session: 4

- **Default parameter values (set by experimentation)**
  - Top-k fragments: 5
  - Top-m recommendations: 5
  - alpha (“mixing” factor): 0.5
  - Weighting scheme/similarity: Weighted/cosine
Methodology

• Generate top-m recommendations for each session in test set.
• Calculate precision, recall, F-score using 10-fold cross-validation
• Report
  – Maximum recall
  – Precision for query with max. recall
  – Avg. precision and recall

\[ P = \frac{|F_r \cap F_u|}{|F_r|} \quad R = \frac{|F_r \cap F_u|}{|F_u|} \]
## Comparison

<table>
<thead>
<tr>
<th></th>
<th>Fragment-based</th>
<th>Tuple-based</th>
<th>Tuple-based using MinHash synopses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information representation</td>
<td>coarse</td>
<td>detailed</td>
<td>coarse</td>
</tr>
<tr>
<td>Phase 1: Session summaries’ formation</td>
<td>Offline</td>
<td>Offline</td>
<td>Offline</td>
</tr>
<tr>
<td>Phase 2: Similarities - $S^\text{pred}$ calculation</td>
<td>Offline ($\text{sim}(\rho, \phi)$) - Online</td>
<td>Online ($\text{sim}(S_i, S_0)$) - Online</td>
<td>Online ($\text{sim}(h(S_i), h(S_0))$) - Online</td>
</tr>
<tr>
<td>Phase 3: Retrieval of similar queries</td>
<td>Online</td>
<td>Online</td>
<td>Online</td>
</tr>
<tr>
<td>Time to generate recommendations after query is posted</td>
<td>&lt; 6 sec</td>
<td>&gt; 5 min</td>
<td>&lt; 6 sec</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Fragment-based</th>
<th>Tuple-based with MinHash synopses</th>
</tr>
</thead>
<tbody>
<tr>
<td>% of sessions with precision = 1</td>
<td>50%</td>
<td>35%</td>
</tr>
<tr>
<td>% of sessions with recall = 1</td>
<td>55%</td>
<td>40%</td>
</tr>
<tr>
<td>% of sessions with precision $\geq 0.5$</td>
<td>85%</td>
<td>60%</td>
</tr>
<tr>
<td>% of sessions with recall $\geq 0.5$</td>
<td>75%</td>
<td>65%</td>
</tr>
<tr>
<td>% of sessions with average precision $\geq 0.5$</td>
<td>65%</td>
<td>40%</td>
</tr>
<tr>
<td>% of sessions with average recall $\geq 0.5$</td>
<td>45%</td>
<td>50%</td>
</tr>
</tbody>
</table>
Roadmap

• Introduction
• The QueRIE Framework & Tuple-based recommendations
• Fragment-based recommendations
• Comparison of approaches
• The QueRIE Prototype
• Conclusions
QueRIE Prototype
(ICDM’09, VLDB’10)

Query Results

Select top 10 objid, ra, dec, u, g, r, i, z from photoobj where run = 2566 and rerun = 40 and camcol = 3 and field = 176

<table>
<thead>
<tr>
<th>objid</th>
<th>ra</th>
<th>dec</th>
<th>u</th>
<th>g</th>
<th>r</th>
<th>i</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>38770773880079197</td>
<td>339.651042</td>
<td>13.40994045</td>
<td>-999</td>
<td>-999</td>
<td>-999</td>
<td>-999</td>
<td>-999</td>
</tr>
<tr>
<td>38770773880078748</td>
<td>339.67940086</td>
<td>13.16823604</td>
<td>-999</td>
<td>-999</td>
<td>-999</td>
<td>-999</td>
<td>-999</td>
</tr>
<tr>
<td>38770773880078762</td>
<td>339.64822203</td>
<td>13.40984306</td>
<td>-999</td>
<td>-999</td>
<td>-999</td>
<td>-999</td>
<td>-999</td>
</tr>
<tr>
<td>38770773880078713</td>
<td>339.66976523</td>
<td>13.40984305</td>
<td>-999</td>
<td>-999</td>
<td>-999</td>
<td>-999</td>
<td>-999</td>
</tr>
<tr>
<td>38770773880078719</td>
<td>339.58683802</td>
<td>13.39943876</td>
<td>-999</td>
<td>-999</td>
<td>-999</td>
<td>-999</td>
<td>-999</td>
</tr>
<tr>
<td>38770773880078720</td>
<td>339.65153411</td>
<td>13.17769085</td>
<td>-999</td>
<td>-999</td>
<td>-999</td>
<td>-999</td>
<td>-999</td>
</tr>
<tr>
<td>38770773880078759</td>
<td>339.54411796</td>
<td>13.14130356</td>
<td>-999</td>
<td>-999</td>
<td>-999</td>
<td>-999</td>
<td>-999</td>
</tr>
<tr>
<td>38770773880078767</td>
<td>339.64822486</td>
<td>13.40497266</td>
<td>-999</td>
<td>-999</td>
<td>-999</td>
<td>-999</td>
<td>-999</td>
</tr>
<tr>
<td>38770773880078757</td>
<td>339.60565329</td>
<td>13.17386310</td>
<td>-999</td>
<td>-999</td>
<td>-999</td>
<td>-999</td>
<td>-999</td>
</tr>
<tr>
<td>38770773880078712</td>
<td>339.60780512</td>
<td>13.40319833</td>
<td>-999</td>
<td>-999</td>
<td>-999</td>
<td>-999</td>
<td>-999</td>
</tr>
</tbody>
</table>

1 of 1

Recommended Queries:

Select top 1000 objid, ra, dec, u, g, r, i, z from photoobj where u between 0 and 10.6 and g between 6 and 20

Note: Snapshot of the results for selected recommended query is shown below:

<table>
<thead>
<tr>
<th>objid</th>
<th>ra</th>
<th>dec</th>
<th>u</th>
<th>g</th>
<th>r</th>
<th>i</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>758826253680042304</td>
<td>50.80146876</td>
<td>76.9533619</td>
<td>19.5459</td>
<td>18.121929</td>
<td>17.404572</td>
<td>17.066448</td>
<td>16.914612</td>
</tr>
<tr>
<td>758826253680042304</td>
<td>51.09964384</td>
<td>77.0655908</td>
<td>18.814566</td>
<td>17.439357</td>
<td>16.865353</td>
<td>16.587626</td>
<td>16.449738</td>
</tr>
<tr>
<td>758826253680042304</td>
<td>51.39736992</td>
<td>76.9976585</td>
<td>18.367891</td>
<td>17.530790</td>
<td>16.863512</td>
<td>16.584689</td>
<td>16.457417</td>
</tr>
<tr>
<td>758826253680042304</td>
<td>50.41136753</td>
<td>77.0190968</td>
<td>19.543238</td>
<td>17.956791</td>
<td>17.199127</td>
<td>16.853485</td>
<td>16.490905</td>
</tr>
<tr>
<td>758826253680042304</td>
<td>50.39784082</td>
<td>77.0856121</td>
<td>19.415105</td>
<td>18.064713</td>
<td>17.432316</td>
<td>17.122215</td>
<td>16.982015</td>
</tr>
</tbody>
</table>
Roadmap

• Introduction
• The QueRIE Framework & Tuple-based recommendations
• Fragment-based recommendations
• Comparison of approaches
• The QueRIE Prototype
• Conclusions
Conclusions

• Scalability is no.1 priority in such systems, even at the loss of accuracy
• Fragment-based approach performed better than approximation of tuple-based approach
• Latent factor CF models on fragments are more scalable with similar levels of prediction accuracy
• Framework can be employed by friendlier UIs for an even better user experience
Thank you!
Related publications

• G. Chatzopoulou, M. Eirinaki, and N. Polyzotis, “Collaborative filtering for interactive database exploration,” in SSDBM ’09
• S. Mittal, J. S. V. Varman, G. Chatzopoulou, M. Eirinaki, and N. Polyzotis, “QueRIE: A Recommender System supporting Interactive Database Exploration,” in ICDM’09 (ICDM’10 proceedings)
• M. Eirinaki, S. Abraham, N. Polyzotis, N. Shaikh, “QueRIE: Collaborative Database Exploration”, IEEE Transactions on Knowledge and Data Engineering (TKDE), 26(7), July 2014