QUERY RECOMMENDATIONS FOR INTERACTIVE DATABASE EXPLORATION

Gloria Chatzopoulou*, UC Riverside
Magdalini Eirinaki, San Jose State Univ
Neoklis Polyzotis, UC Santa Cruz

*work performed while author affiliated with UC Santa Cruz
Motivation

- Scientific disciplines use relational DBMS for storage and retrieval of information
  - Biologists (e.g. UCSC Genome, BMRB)
  - Astronomers (e.g. Skyserver)
  - Chemists (e.g. PubChem)
- DBs are accessible online by users with diverse information needs
- Typical users do interactive exploration
Motivation (cont’d)

- Typical users are not SQL experts
- Scientific datasets increase in size
- Users may miss interesting information
  - They do not write the “right” query
  - They are not aware of all parts of the database

Our goal: Assist users in finding useful information
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 query data X and Alice queries data Y

then

Bob is likely to be interested in querying data Y
System Architecture

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
- QueRIE Recommendation Framework
- Experiments
- Conclusions
Conceptual Framework
### Session Summaries

**Binary Weighting Scheme**

<table>
<thead>
<tr>
<th>R</th>
<th>a</th>
<th>b</th>
<th>L</th>
<th>a</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
<td>3</td>
<td></td>
<td>y</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>s</td>
<td>4</td>
<td></td>
<td>s</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>w</td>
<td>3</td>
<td></td>
<td>s</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>r</td>
<td>2</td>
<td></td>
<td>t</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

q1 = <1,1,0,0,1,1,1,0>  
q2 = <0,1,0,0,0,1,1,0>  
s0 = <1,2,0,0,1,2,2,0>2

q1: \( R \times_{R.a=L.a} L \)  
q2: \( \sigma_{R.b=4} (F \times_{R.a=L.a} L) \)

**Result Weighting Scheme**

| q1 = <0.33,0.33,0.33,0.33,0.33,0.33,0.33,0>  
| q2 = <0,0.50,0,0,0.50,0.50,0,0>  
s0 = <0.33,0.83,0,0,0.33,0.83,0.83,0>
Vector-space similarity functions can be used

- Cosine Similarity

\[ \text{sim}(uA,uB) = \frac{uA \cdot uB}{\|uA\| * \|uB\|} \]

High similarity means that users are interested in the same parts of the database
Predicted Summary

\[ u^{pred} = \alpha \cdot u + (1 - \alpha) \cdot \frac{\sum_{1 \leq i \leq h} \text{sim}(u, u_i) \cdot u_i}{\sum_{1 \leq i \leq h} \text{sim}(u, u_i)} \]

where \( \alpha \) is the “mixing factor” \( \alpha \in [0,1] \)
Generating Recommendations

Use queries of past users

Query Log Data

$\mathbf{q}_1 = \langle 1, 0, 0, \ldots, 0 \rangle$

$\mathbf{q}_2 = \langle 0, 1, 0, \ldots, 0 \rangle$

$\vdots$

$\mathbf{q}_N = \langle 1, 0, 1, \ldots, 1 \rangle$

$\mathbf{u}^{\text{pred}} = \langle 1, 0, 0, \ldots, 0 \rangle$

Similarity Function

$\text{sim}(\mathbf{u}^{\text{pred}}, \mathbf{q}_i)$

Ranking Queries

$\text{rank}(\mathbf{q}_1) = \text{sim}(\mathbf{u}^{\text{pred}}, \mathbf{q}_1)$

$\text{rank}(\mathbf{q}_2) = \text{sim}(\mathbf{u}^{\text{pred}}, \mathbf{q}_2)$

$\vdots$

$\text{rank}(\mathbf{q}_N) = \text{sim}(\mathbf{u}^{\text{pred}}, \mathbf{q}_N)$

Return Top-K Queries
Experimental Setup

- **SkyServer Dataset**

<table>
<thead>
<tr>
<th>Database Size</th>
<th>2.6TB</th>
</tr>
</thead>
<tbody>
<tr>
<td>#Sessions</td>
<td>720</td>
</tr>
<tr>
<td>#Queries</td>
<td>6713</td>
</tr>
<tr>
<td>#Distinct Queries</td>
<td>4037</td>
</tr>
<tr>
<td>Avg. number of queries per session</td>
<td>9.3</td>
</tr>
<tr>
<td>Min. number of queries per session</td>
<td>3</td>
</tr>
</tbody>
</table>

- **Evaluation Metrics: Precision and Recall**
  - **High precision**: most witnesses of the recommended query are witnesses in the actual query.
  - **High Recall**: most witnesses of the actual query are witnesses in the recommended query.
Binary vs Result Weighting Schemes

Binary outperforms Result Weighting Scheme
Effect of mixing factor $\alpha$

Hybrid Collaborative Filtering yields better results
Conclusions

- Scientists need help in exploring databases
- Query recommendations can be an effective tool in guiding exploration
- Collaborative filtering provides a natural method to generate recommendations
- Experiments show promising results on real-world datasets

Ongoing Work:
- Performance improvement
- Use of approximation techniques
Thank you
Top-3 vs Top-5 Binary Weights

The bigger recommendation set the higher accuracy