Personalization and recommendation of OLAP queries

Patrick Marcel
Université François Rabelais Tours
Computer Science Lab.

Uni. Bologna, 15.02.2011
Outline

- Motivation
- Personalizing OLAP queries
- Recommending OLAP queries
- Perspectives
Motivation

- DB could be more user-friendly
  - See « making DB systems usable », Jagadish, SIGMOD 2007
- Knowledge captured by queries overlooked in DB lifecycle
  - See « A case for a collaborative query management system », Khoussainova & al., CIDR 2009
Motivation (cont'd)

- Are BI tools designed for the decision-maker?
  - « How is BI used in industry », Pedersen, DaWak 2004
  - « Industrializing Data Mining, Challenges and Perspectives », Fogelman-Soulié, ECML/PKDD 2008
  - « Frontiers in BI: Distribution and Personalization », Rizzi, ADBIS 2010
  - « An architecture for ad-hoc and collaborative BI », Berthold & al, EDBT/ICDT Workshops 2010
Personalizing OLAP queries

- PhD Hassina Mouloudi (2007)

- Main publications
  - ACM DOLAP 2005
  - BDA 2006 (French DB conference)
  - Hassina's dissertation (in French)

- Prototype
  - Mobile application for querying a cube with query personalization
    - Mondrian, Oracle, Tomcat, Axis
Motivation

SELECT CROSSJOIN({City.Tours, City.Orleans},
{Category.Members}) ON ROWS
FROM SalesCube
WHERE (Measures.quantity)

Visualization depends on the user's profile

<table>
<thead>
<tr>
<th></th>
<th>Tours</th>
<th>Orleans</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drink</td>
<td>77</td>
<td>25</td>
</tr>
<tr>
<td>Food</td>
<td>89</td>
<td>50</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>2003</th>
<th>2004</th>
<th>2005</th>
<th>2006</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tours</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Drink</td>
<td>54</td>
<td>61</td>
<td>49</td>
<td>32</td>
</tr>
<tr>
<td>Food</td>
<td>55</td>
<td>30</td>
<td>41</td>
<td>32</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tours</th>
<th>2003</th>
<th>2004</th>
<th>2005</th>
<th>2006</th>
</tr>
</thead>
<tbody>
<tr>
<td>Food</td>
<td>77</td>
<td>54</td>
<td>55</td>
<td>33</td>
</tr>
<tr>
<td>Drink</td>
<td>89</td>
<td>61</td>
<td>30</td>
<td>41</td>
</tr>
<tr>
<td>Cloth</td>
<td>56</td>
<td>30</td>
<td>32</td>
<td>60</td>
</tr>
<tr>
<td>Shoes</td>
<td>45</td>
<td>50</td>
<td>32</td>
<td>51</td>
</tr>
</tbody>
</table>
Motivation


Visualization depends on the user's profile

<table>
<thead>
<tr>
<th>Tours</th>
<th>2003</th>
<th>2004</th>
<th>2005</th>
<th>2006</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drink</td>
<td>77</td>
<td>54</td>
<td>55</td>
<td>33</td>
</tr>
<tr>
<td>Food</td>
<td>89</td>
<td>61</td>
<td>30</td>
<td>41</td>
</tr>
<tr>
<td>Orleans Drink</td>
<td>25</td>
<td>50</td>
<td>49</td>
<td>32</td>
</tr>
<tr>
<td>Orleans Food</td>
<td>33</td>
<td>44</td>
<td>59</td>
<td>27</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tours</th>
<th>2003</th>
<th>2004</th>
<th>2005</th>
<th>2005</th>
</tr>
</thead>
<tbody>
<tr>
<td>Food</td>
<td>77</td>
<td>54</td>
<td>55</td>
<td>33</td>
</tr>
<tr>
<td>Drink</td>
<td>89</td>
<td>61</td>
<td>30</td>
<td>41</td>
</tr>
<tr>
<td>Cloth</td>
<td>56</td>
<td>30</td>
<td>32</td>
<td>60</td>
</tr>
<tr>
<td>Shoes</td>
<td>45</td>
<td>50</td>
<td>32</td>
<td>51</td>
</tr>
</tbody>
</table>

a member
Motivation

SELECT CROSSJOIN({City.Tours, City.Orleans},
                {Category.Members}) ON ROWS
FROM SalesCube
WHERE (Measures.quantity)

Visualization depends on the user's profile

<table>
<thead>
<tr>
<th>City</th>
<th>2003</th>
<th>2004</th>
<th>2005</th>
<th>2006</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tours</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Drink</td>
<td>77</td>
<td>54</td>
<td>55</td>
<td>33</td>
</tr>
<tr>
<td>Food</td>
<td>89</td>
<td>61</td>
<td>30</td>
<td>41</td>
</tr>
<tr>
<td>Orleans</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Drink</td>
<td>25</td>
<td>50</td>
<td>49</td>
<td>32</td>
</tr>
<tr>
<td>Food</td>
<td>33</td>
<td>44</td>
<td>59</td>
<td>27</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Category</th>
<th>2003</th>
<th>2004</th>
<th>2005</th>
<th>2006</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tours</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Food</td>
<td>77</td>
<td>54</td>
<td>55</td>
<td>33</td>
</tr>
<tr>
<td>Drink</td>
<td>89</td>
<td>61</td>
<td>30</td>
<td>41</td>
</tr>
<tr>
<td>Orleans</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cloth</td>
<td>56</td>
<td>30</td>
<td>32</td>
<td>60</td>
</tr>
<tr>
<td>Shoes</td>
<td>45</td>
<td>50</td>
<td>32</td>
<td>51</td>
</tr>
</tbody>
</table>
Motivation

SELECT CROSSJOIN({City.Tours, City.Orleans},
{Category.Members}) ON ROWS
FROM SalesCube
WHERE (Measures.quantity)

Visualization depends on the user's profile

<table>
<thead>
<tr>
<th></th>
<th>2003</th>
<th>2004</th>
<th>2005</th>
<th>2006</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tours</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Drink</td>
<td>77</td>
<td>54</td>
<td>55</td>
<td>33</td>
</tr>
<tr>
<td>Food</td>
<td>89</td>
<td>61</td>
<td>30</td>
<td>41</td>
</tr>
<tr>
<td>Orleans</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Drink</td>
<td>25</td>
<td>50</td>
<td>49</td>
<td>32</td>
</tr>
<tr>
<td>Food</td>
<td>33</td>
<td>44</td>
<td>59</td>
<td>27</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tours</th>
<th>2003</th>
<th>2004</th>
<th>2005</th>
<th>2005</th>
</tr>
</thead>
<tbody>
<tr>
<td>Food</td>
<td>77</td>
<td>54</td>
<td>55</td>
<td>33</td>
</tr>
<tr>
<td>Drink</td>
<td>89</td>
<td>61</td>
<td>30</td>
<td>41</td>
</tr>
<tr>
<td>Cloth</td>
<td>56</td>
<td>30</td>
<td>32</td>
<td>60</td>
</tr>
<tr>
<td>Shoes</td>
<td>45</td>
<td>50</td>
<td>32</td>
<td>51</td>
</tr>
</tbody>
</table>
Motivation

SELECT CROSSJOIN({City.Tours, City.Orleans},
{Category.Members}) ON ROWS
FROM SalesCube
WHERE (Measures.quantity)

Visualization depends on the user's profile

<table>
<thead>
<tr>
<th>Tours</th>
<th>2003</th>
<th>2004</th>
<th>2005</th>
<th>2006</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drink</td>
<td>77</td>
<td>54</td>
<td>55</td>
<td>33</td>
</tr>
<tr>
<td>Food</td>
<td>89</td>
<td>61</td>
<td>30</td>
<td>41</td>
</tr>
<tr>
<td>Orleans Drink</td>
<td>25</td>
<td>50</td>
<td>49</td>
<td>32</td>
</tr>
<tr>
<td>Orleans Food</td>
<td>33</td>
<td>44</td>
<td>59</td>
<td>27</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tours</th>
<th>2003</th>
<th>2004</th>
<th>2005</th>
<th>2005</th>
</tr>
</thead>
<tbody>
<tr>
<td>Food</td>
<td>77</td>
<td>54</td>
<td>55</td>
<td>33</td>
</tr>
<tr>
<td>Drink</td>
<td>89</td>
<td>61</td>
<td>30</td>
<td>41</td>
</tr>
<tr>
<td>Cloth</td>
<td>56</td>
<td>30</td>
<td>32</td>
<td>60</td>
</tr>
<tr>
<td>Shoes</td>
<td>45</td>
<td>50</td>
<td>32</td>
<td>51</td>
</tr>
</tbody>
</table>

a query
Motivation

```
SELECT CROSSJOIN({City.Tours, City.Orleans}, {Category.Members}) ON ROWS
FROM SalesCube
WHERE (Measures.quantity)
```

Visualization depends on the user’s profile

<table>
<thead>
<tr>
<th></th>
<th>2003</th>
<th>2004</th>
<th>2005</th>
<th>2006</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tours</td>
<td>Drink</td>
<td>77</td>
<td>54</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td>Food</td>
<td>89</td>
<td>61</td>
<td>30</td>
</tr>
<tr>
<td>Orleans</td>
<td>Drink</td>
<td>25</td>
<td>50</td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>Food</td>
<td>33</td>
<td>44</td>
<td>59</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tours</th>
<th>2003</th>
<th>2004</th>
<th>2005</th>
<th>2005</th>
</tr>
</thead>
<tbody>
<tr>
<td>Food</td>
<td>77</td>
<td>54</td>
<td>55</td>
<td>33</td>
</tr>
<tr>
<td>Drink</td>
<td>89</td>
<td>61</td>
<td>30</td>
<td>41</td>
</tr>
<tr>
<td>Cloth</td>
<td>56</td>
<td>50</td>
<td>30</td>
<td>41</td>
</tr>
<tr>
<td>Shoes</td>
<td>45</td>
<td>50</td>
<td>32</td>
<td>51</td>
</tr>
</tbody>
</table>

a query result
The problem

• Given
  – An MDX query \( q \)
  – User preferences \( P \)
  – A Visualization constraint \( v \)

• Find a preferred query \( q' \)
  – Included in \( q \)
  – Nearest to \( q \) satisfying \( v \)
  – The most interesting w.r.t \( P \)

Formally, compute \( q' = \max_{<P} \{q'' \subseteq q \mid v(q'') = \text{true} \} \)
A short introduction to preferences
Preference Relation

• General Hypothesis
  - I like “a” better than “b” if (a > b) where “>” is a partial ordering
  - I cannot compare a and b (a ~ b)

• Partial Ordering (PO)
  - A binary relation “≥” over a set O which is
    - Reflexive: (a ≥ a)
    - Antisymmetric: If (a ≥ b) and (b ≥ a) then (a = b)
    - Transitive: If (a ≥ b) and (b ≥ c) then (a ≥ c)
Strict and Total Order

• **Strict Partial Order (SPO)**
  - A binary relation “>” over a set O which is
    - **Irreflexive:** \( \neg(a > a) \)
    - **Asymmetric:** If \( a \neq b \) and \( a > b \) then \( \neg(b > a) \)
    - **Transitive:** If \( a > b \) and \( b > c \) then \( a > c \)
  - Let R be a PO and R’ defined by:
    - \( (a R’ b) \) iff \( (a R b) \) and \( a \neq b \)
    - \( R’ \) is a SPO

• **Total Order (TO)**
  - A partial order “>” such that
    - **For every a and b,** \( (a > b) \) or \( (b > a) \)
Preference Composition

• Single-dimensional
  - Let R be a relation
  - Let \( >_1 \) and \( >_2 \) be two preference relations over R
  - How to combine \( >_1 \) and \( >_2 \) to define a new preference relation over R?

• Multi-dimensional
  - Let \( R_1 \) and \( R_2 \) be two relations
  - Let \( >_1 \) and \( >_2 \) be preference relations over \( R_1 \) and \( R_2 \)
  - How to combine \( >_1 \) and \( >_2 \) to define a preference relation over \( R_1 \times R_2 \)?
• **Boolean Composition**
  - **Intersection**: \( R = (>_1 \cap >_2) \)
    - \( t R t' \) if \( t >_1 t' \) and \( t >_2 t' \)
  - **Union**: \( R = (>_1 \cup >_2) \)
    - \( t R t' \) if \( t >_1 t' \) or \( t >_2 t' \)

• **Prioritized Composition**: \( R = (>_1 \uparrow >_2) \)
  - \( t R t' \) if \( t >_1 t' \) or
    - \( \neg (t' >_1 t) \) and \( t >_2 t' \)

• **Pareto Composition**: \( R = (>_1 \otimes >_2) \)
  - \( t R t' \) if \( ((t >_1 t') \) and \( (t >_2 t' \) or \( t \sim_2 t' \)) or
    - \( ((t >_2 t') \) and \( (t >_1 t' \) or \( t \sim_1 t' \)) \)
• **Lexicographic Composition** : \( R = (>_1 \uparrow >_2) \)
  
  \( (t_1, t_2) \preceq (t'_1, t'_2) \) if 
  \( (t_1 >_1 t'_1) \) or 
  
  \( (t_1 \sim_1 t'_1) \) and \( (t_2 >_2 t'_2) \)

• **Pareto Composition** : \( R = (>_1 \otimes >_2) \)
  
  \( (t_1, t_2) \preceq (t'_1, t'_2) \) if 
  \( (t_1 >_1 t'_1) \) and \( (t_2 >_2 t'_2) \) or 
  
  \( (t_1 >_1 t'_1) \) and \( (t_2 \sim_2 t'_2) \) or 
  
  \( (t_1 \sim_1 t'_1) \) and \( (t_2 >_2 t'_2) \)
Preservation Properties

<table>
<thead>
<tr>
<th></th>
<th>Union</th>
<th>Intersection</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strict Partial Order</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Total Order</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Lexicographic composition preserves

- Strict partial order
- Total order
Ordering Sets of Objects

• Given
  • A set of objects $O$
  • A preference $>_O$ relation over $O$

• Problem
  • How to define a preference relation $>_S$ over $\mathcal{P}(O)$?
  • Example of solution
    – Let $X$ and $Y$ be two subsets of $O$
    – I like $X$ better than $Y$, denoted $(X >_S Y)$, if for every $y \in Y$, there exists $x \in X$ such that $(x >_O y)$
    – If $>_O$ is a PO over $O$, then $>_S$ is a PO over $\mathcal{P}(O)$
Simple Example

X ⊃ Y

\[ O_1 \leftarrow O_2 \leftarrow O_3 \]
\[ O_4 \leftarrow O_5 \leftarrow O_6 \]
Going back to our problem

User preferences

query

Personalize the query

Personazlized query

The cube to be queried

Inspired by Koutrika and Ioannidis, ICDE 2004
Preferences

• Given
  – A partial ordering over dimensions
    – \( D < D' \) if \( D' \) is preferred to \( D \)
  – For all dimension \( D \), a partial ordering over members
    – \( m <_D m' \) if \( m' \) is preferred to \( m \)

• An ordering over cell references is deduced
  – \( M \) is the set of preferred dimensions where \( r \) and \( r' \) differ
  – \( r < r' \) if for all dimension \( D \) in \( M \) \( r(D) <_D r'(D) \)
    – Lexicographic composition
Ordering over queries

• The ordering over references gives a partial ordering over queries
  • \( q < q' \) if
    • They have the same schema
    • For all references of \( q \) not in \( q' \) there exists a preferred reference in \( q' \) not in \( q \)
Ordering over queries

• The ordering over queries is total if the ordering over references is total

• The ordering is such that if $q \subseteq q'$ then $q < q'$
  • Larger queries are preferred…
  • But they have to fit on the screen!
    • Hence the visualization constraint
    • E.g., no more than 2 axes with 4 positions each
Example of preferred query

Since the user profile contains
Location < Product, Product < Time
2005 < 2006, Food < Drink

Indeed:

(2005,Food,Tours,quantity) < (2006,Drink,Tours,quantity)
(2005,Drink,Tours,quantity) < (2006,Drink,Tours,quantity)
Personalizing

User query ➔ Personnalization engine ➔ Query processor ➔ Fact table

User profil ➔ Dimension tables

Result ← Personnalization engine ← Query processor ← Fact table
Personalizing OLAP queries

- **Context**
  - Dimension tables in main memory
  - No access to the fact table

- **Principle**
  - Compute sets of positions in the resulting crosstab
    - Largest possible
    - Visualizable w.r.t. the visualization constraint
    - Corresponding to the preferred facts
  - Compute the structures of the crosstabs
Example of personalization (1)

The query:

```sql
SELECT CROSSJOIN({City.Tours, City.Orleans},
                 {Category.Members}) ON ROWS
FROM SalesCube
WHERE (Measures.quantity)
```

Preferences:

- Time < Location and Product < Location
- 2002 < 2003 < 2004 < 2005 < 2006
- Electronics < shoes < cloth < food < drink
- Quantity < price

Constraint: 2 axes, no more than 4 positions on each axis
Example of personalization (2)

<table>
<thead>
<tr>
<th>Drink</th>
<th>Orleans Tours</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2006</td>
</tr>
</tbody>
</table>

Step 1
The most preferred references
Example of personalization (3)

<table>
<thead>
<tr>
<th></th>
<th>2006</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drink</td>
<td>Orleans</td>
</tr>
<tr>
<td>Tours</td>
<td></td>
</tr>
</tbody>
</table>

Step 2
The second most preferred references

<table>
<thead>
<tr>
<th></th>
<th>2006</th>
<th>2005</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drink</td>
<td>Orleans</td>
<td></td>
</tr>
<tr>
<td>Tours</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Food</td>
<td>Orleans</td>
<td></td>
</tr>
<tr>
<td>Tours</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example of personalization (4)

Step 3: the next most preferred references
But the selected references have to satisfy the visualization constraint
Example of personalization (5)

Finally, one of the constructed query is

```
SELECT CROSSJOIN({City.Tours, City.Orleans},
    {Category.Food, Category.drink}) ON ROWS
FROM SalesCube
WHERE (Measures.quantity)
```

<table>
<thead>
<tr>
<th></th>
<th>2003</th>
<th>2004</th>
<th>2005</th>
<th>2006</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tours</td>
<td>Drink</td>
<td>77</td>
<td>54</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td>Food</td>
<td>89</td>
<td>61</td>
<td>30</td>
</tr>
<tr>
<td>Orleans</td>
<td>Drink</td>
<td>25</td>
<td>50</td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>Food</td>
<td>33</td>
<td>44</td>
<td>59</td>
</tr>
</tbody>
</table>
Obtaining a total order

• Assume a partial order
  • e.g., over members

• Assume a total order imposed by the device
  • E.g., the classical lexicographic order over strings

• Combine them with prioritized composition
  • The result is a total order over members
Speedup
Recommending OLAP queries

- PhD Elsa Negre (2009)
- Main publications
  - ACM DOLAP 2008
  - DaWaK 2009
  - ACM DOLAP 2009
  - Int. Journal of DW and mining (to appear)
- Prototype
  - Various methods for OLAP query recommendation
    - Mondrian, MySql
What is recommending?

- The matrix is huge and sparse
- How to fill the holes?

<table>
<thead>
<tr>
<th>interest</th>
<th>Item 1</th>
<th>Item 2</th>
<th>Item 3</th>
<th>...</th>
<th>Item m</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>0.3</td>
<td></td>
<td>0.9</td>
<td>...</td>
<td>0.7</td>
</tr>
<tr>
<td>User 2</td>
<td></td>
<td>0.4</td>
<td>0.8</td>
<td>...</td>
<td>0.6</td>
</tr>
<tr>
<td>User 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>User n</td>
<td>0.9</td>
<td>0.5</td>
<td></td>
<td>...</td>
<td>0.2</td>
</tr>
</tbody>
</table>
Collaborative filtering approach

<table>
<thead>
<tr>
<th>interest</th>
<th>Item 1</th>
<th>Item 2</th>
<th>Item 3</th>
<th>...</th>
<th>Item m</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>0.3</td>
<td></td>
<td>0.9</td>
<td>...</td>
<td>0.7</td>
</tr>
<tr>
<td>User 2</td>
<td></td>
<td>0.4</td>
<td>0.8</td>
<td>...</td>
<td>0.6</td>
</tr>
<tr>
<td>User 3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User n</td>
<td>0.9</td>
<td>0.5</td>
<td></td>
<td>...</td>
<td>0.2</td>
</tr>
</tbody>
</table>

- Consider each user as a vector
- Compute similarity between users
  - e.g., Cosine(user 1, user 2)
- Compute item score based on the scores of similar users
  - e.g., with weighted average
- Recommend highly rated items
Our approach
Distances proposed

• Between positions in the cube
  – Hamming
  – Based on shortest path

• Between queries
  – Based on differences in dimension
  – Hausdorff

• Between sessions
  – Based on the subsequence
  – Edit distance
Distances proposed

• Between positions in the cube
  – Hamming
  – Based on shortest path

• Between queries
  – Based on differences in dimension
  – Hausdorff

• Between sessions
  – Based on the subsequence
  – Edit distance

More distances?
See Baikousi, Rogkakos, Vassiliadis. « Similarity Measures for Multidimensional Data », ICDE 2011
Distance between references

How far is (Pepsi, July08, Paris) to (Soda, 2008, North)?

Hamming says very far

Shortest path says not that far
Distance between queries

How far is to?

Members in the dimensions say quite far

Hausdorff says not that far
Hausdorff distance

\[ d_H(q_1, q_2) = \max \left\{ \max_{r_1 \in q_1} \min_{r_2 \in q_2} d_{\text{references}}(r_1, r_2), \right. \\
\left. \max_{r_2 \in q_2} \min_{r_1 \in q_1} d_{\text{references}}(r_1, r_2) \right\} \]
Distance between sessions

How far is coke aug08 to coke aug08?

First sequence is not a subsequence of the second one, so distance is very far.
Distance between sessions

How far is

To

I can transform the first sequence into the second one by editing it:

- transforming the first two queries
- Adding a third query

so distance is not that far
Levenstein (or edit) distance

Operations are:

- Add a letter (cost 1)
- Remove a letter (cost 1)
- Modify a letter (cost 1)
Architecture of the prototype
Experiments

• Cube
  – Foodmart (Mondrian sample cube)

• Log generator
  – Max 300 cells per MDX query
  – X sessions
  – At most Y queries/session
  – Z dimensions for navigation (density)
  – Logs of up to 3000 queries
Session generation

1. Generate X sessions of a given size
   1. Construct first query and evaluate it
   2. Detect an interesting pair of cells
   3. Apply one of Sarawagi’s operators to find interesting rollups or drilldowns
   4. Construct the next query from the result of the operator
   5. Evaluate it and Goto 2

See Sarawagi’s papers at VLDB 1999-2001
10 fold cross validation

Initial Log → Log

Current sessions of size n

\[ s_c = q_1 \rightarrow \ldots \rightarrow q_{n-1} \rightarrow q_{\text{expected}} \]

\[ s_c = q_1 \rightarrow \ldots \rightarrow q_{n-1} \rightarrow q_{\text{recommended}} = q_{\text{expected}} \]

recommend
Effectiveness

\[ E = \text{Members of the expected query} \]
\[ R = \text{Members of the recommended query} \]

\[ \text{Precision} = \frac{\text{Intersect}}{R} \]
\[ \text{Recall} = \frac{\text{Intersect}}{E} \]
\[ \text{F-measure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \]
Perspectives

- Joint work with UBO
  - Extraction of preferences for OLAP query personalization
- Joint work with UPC Barcelona
  - Leveraging logical connections between queries for OLAP query recommendations
- Contributions to a collaborative query management system for OLAP
  - Ongoing PhD
Thanks for your attention

Any question?