Interactive Data Exploration

Related works

Ali El Adi | Bruno Rubio | Deepak Barua | Hussain Syed

Databases and Information Retrieval Integration Project
Recap – Smart-Drill

**TABLE I: Initial Summary**

<table>
<thead>
<tr>
<th>Store</th>
<th>Product</th>
<th>Region</th>
<th>Count</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>⭐</td>
<td>⭐</td>
<td>⭐</td>
<td>6000</td>
<td>0</td>
</tr>
</tbody>
</table>

**TABLE II: Result After First Smart Drill Down**

<table>
<thead>
<tr>
<th>Store</th>
<th>Product</th>
<th>Region</th>
<th>Count</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>⭐</td>
<td>⭐</td>
<td>⭐</td>
<td>6000</td>
<td>0</td>
</tr>
<tr>
<td>▶ Target</td>
<td>bicycles</td>
<td>⭐</td>
<td>200</td>
<td>2</td>
</tr>
<tr>
<td>▶ ⭐</td>
<td>comforters</td>
<td>MA-3</td>
<td>600</td>
<td>2</td>
</tr>
<tr>
<td>▶ Walmart</td>
<td>⭐</td>
<td>⭐</td>
<td>1000</td>
<td>1</td>
</tr>
</tbody>
</table>
AlphaSum: Size constrained table summarization using value lattices

- K. Selçuk Candan, Huiping Cao, Yan Qi Arizona State Univ. Tempe, USA
- Maria Luisa Sapino Univ. di Torino Torino, Italy
  mlsapino@di.unito.it
Size-Constrained Table Summarization

- AlphaSum: Size-Constrained Table Summarization using Value Lattices
- Table summarization can benefit from knowledge about acceptable value clustering alternatives for clustering the values in the database.
- Summarization of large data tables is required in many scenarios where it is hard to display complete data sets.
Size-Constrained Table Summarization

- For example, table summarization for mobile commerce applications over PDAs, which cannot effectively present a large table of results with their small screens.
- Table summarization is also useful in various other scenarios, where it is hard or highly-impractical to visualize large data sets.
Consider, for example, a scientist exploring the Digital Archaeological Record (tDAR/FICSR).

A digital library which archives and provides access to a large number of (and diverse) data sets, collected by different researchers.
Example

User is interested in summarizing the given table based on the attribute pair, \langle Age, Location \rangle. The summarized table can be visualized in a space that can hold at most 2 tuples.
Example

Can be achieved by clustering rows such that each row in the summary corresponds to at least 3 ($= \lceil 6/2 \rceil$) rows in the original table. It is 3-summarization.

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>1*</td>
<td>Southwest</td>
</tr>
<tr>
<td>-</td>
<td>2*</td>
<td>Maryland</td>
</tr>
</tbody>
</table>

(b) Data table after 3-summarization on $\langle Age, Location \rangle$ using hierarchies in Figure 1
Table 1: (a) A database and (b) A 3-summarized version on the \((Age, Location)\) attribute pair

(a) Hierarchy for \(Age\)  
(b) Hierarchy for \(Location\)

Figure 1: Hierarchy-based value clustering for attribute \(Age\) (a) and \(Location\) (b). The directed edges denote the clustering/summarization direction.
Value hierarchies (such as the one in example) are commonly used for user-driven data exploration within large data sets.

For example, OLAP operators (such as drill-down, navigating from less specific to more detailed data by stepping down on a given hierarchy, and roll-up, which performs aggregation on by climbing up a hierarchy of concepts).
OLAP-based navigation (using drill-down and roll-up operations), on the other hand, does not take into account the visualization real estate (e.g., the number of tuples to be displayed) available for exploration.
As described above, the main goal of AlphaSum is to obtain OLAP-like navigable summaries from large tables. Its goals, however, differ from OLAP in two significant ways:

- First of all, unlike traditional OLAP, AlphaSum takes the visualization into account.
- the maximum number of tuples to be visualized to the user as a highest level summary is an input parameters to AlphaSum
Secondly, AlphaSum recognizes that in many applications (even in OLAP), systems may need to take into account the existence of multiple acceptable value clustering alternatives.

For example, value lattice shown on next slide provides more clustering alternatives than the simpler hierarchy in the initial example.
Figure 4: Lattice-based value clustering graphs for the attribute *Age*. 
Figure 2: $\alpha$-summarization helps reduce the table size for quick exploration and enable the user explore the summary as in OLAP
What is DWARF

Dwarf is a patented (US Patent 7,133,876) highly compressed structure for computing, storing, and querying Data Cubes. It is a highly compressed structure with reduction reaching 1:1,000,000 depending on the data distribution.

The most important aspect of this patented Dwarf technology is that its data fusion (prefix and suffix redundancy elimination) is discovered and eliminated BEFORE the cube is computed and this explains the dramatic reduction in compute time.

The term “Dwarf” is used in analogy to dwarf stars that have a very large condensed mass but occupy very small space.
Sample Input Data

Legend: ('Country Dimension', 'City Dimension', 'Station Dimension', 'Measure')

('Ireland', 'Dublin', 'Fenian St', '3')
('France', 'Amiens', 'Bd Maignan Larviere', '2')
('Ireland', 'Dublin', 'City Quay', '0')
('France', 'Paris', 'Champs Elysses', '9')
DWARF Cube Data Structure
### DWARF Schema

#### A: DWARF_Schema column family

<table>
<thead>
<tr>
<th>id</th>
<th>node_count</th>
<th>cell_count</th>
<th>size_as_mb</th>
<th>entry_node_id</th>
<th>is_cube</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>int</td>
<td>int</td>
<td>int</td>
<td>int</td>
<td>bool</td>
</tr>
</tbody>
</table>

#### B: DWARF_Node schema

<table>
<thead>
<tr>
<th>id</th>
<th>parentIds</th>
<th>childrenIds</th>
<th>root</th>
<th>schema_id</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>set&lt;int&gt;</td>
<td>set&lt;int&gt;</td>
<td>boolean</td>
<td>int</td>
</tr>
</tbody>
</table>

#### C: DWARF_Cell schema

<table>
<thead>
<tr>
<th>id</th>
<th>key</th>
<th>measure</th>
<th>parentNode</th>
<th>pointerNode</th>
<th>leaf</th>
<th>schema_id</th>
<th>dimension_table_name</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>text</td>
<td>int</td>
<td>int</td>
<td>int</td>
<td>boolean</td>
<td>int</td>
<td>text</td>
</tr>
</tbody>
</table>
MySQL-Min has best storage performance of all schemas

Table 4: DWARF storage performance

<table>
<thead>
<tr>
<th>Size (MB) use to store a DWARF cube</th>
<th>Day</th>
<th>Week</th>
<th>Month</th>
<th>TMonth</th>
<th>SMonth</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySQL-DWARF</td>
<td>2</td>
<td>20</td>
<td>80</td>
<td>169</td>
<td>424</td>
</tr>
<tr>
<td>MySQL-Min</td>
<td>&lt; 1</td>
<td>8</td>
<td>33</td>
<td>70</td>
<td>178</td>
</tr>
<tr>
<td>NoSQL-DWARF</td>
<td>&lt; 1</td>
<td>9</td>
<td>35</td>
<td>73</td>
<td>182</td>
</tr>
<tr>
<td>NoSQL-Min</td>
<td>&lt; 1</td>
<td>11</td>
<td>45</td>
<td>96</td>
<td>243</td>
</tr>
</tbody>
</table>
NoSQL-DWARF has best time performance of all schemas

Table 5: DWARF storage time performance

<table>
<thead>
<tr>
<th></th>
<th>Day</th>
<th>Week</th>
<th>Month</th>
<th>TMonth</th>
<th>SMonth</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySQL-DWARF</td>
<td>1768</td>
<td>12501</td>
<td>47247</td>
<td>100466</td>
<td>255098</td>
</tr>
<tr>
<td>MySQL-Min</td>
<td>1107</td>
<td>5955</td>
<td>22243</td>
<td>47936</td>
<td>121221</td>
</tr>
<tr>
<td><strong>NoSQL-DWARF</strong></td>
<td><strong>927</strong></td>
<td><strong>4368</strong></td>
<td><strong>15955</strong></td>
<td><strong>34203</strong></td>
<td><strong>89257</strong></td>
</tr>
<tr>
<td>NoSQL-Min</td>
<td>5699</td>
<td>57153</td>
<td>222044</td>
<td>484498</td>
<td>1219887</td>
</tr>
</tbody>
</table>
One of the Query Steering modes:

Auto Steering
System automatically recommends queries through the user profile.

A new framework is proposed to address how the system should discover relevant data.

**Automatic Interactive Data Exploration (AIDE)**

AIDE

An iterative exploration model that:

• Retrieves objects to be extracted and labeled by the user
• Minimizes the number of samples presented to the user

How to do it?

By integrating machine learning techniques with classification algorithms and decision tree systems.
AIDE Framework Overview
AIDE Framework steps with examples

CLINICAL TRIALS

1. Sample Extraction
   Records of clinical trials are extracted, along with its attributes (e.g. year, outcome, patience age, medication dosage, etc).

2. User Relevance Feedback
   Each sample is labeled as interesting or not. Users can also mark “similar” samples and modify their feedback on previously seen samples.
3. **Data Classification**
   
   Samples are used to train a classification model that characterizes user’s interests and predicts which clinical trials are relevant based on the feedback collected so far.

4. **Space Exploration**
   
   Feedbacks and current user model are used to identify promising sample areas and retrieve next sample from dataset.
5. Query Formulation

By utilizing decision tree classifiers to identify linear patterns of user interests, a classification model is produced and then “translated” into a query expression.

Let’s assume a decision tree classifier that predicts relevant and irrelevant clinical trials objects based on the attributes age and dosage:
select * from table where 
(age <= 20 and dosage > 10 and dosage <= 15) 
or 
(age > 20 and age <= 40 and dosage >= 0 and dosage <= 10)
AIDE’s effectiveness

Goal: maximize $F$-measure of the final decision tree $C$ on the total data space $T$, defined as:

$$F(T) = \frac{2 \times \text{precision}(T) \times \text{recall}(T)}{\text{precision}(T) + \text{recall}(T)}$$

where:

$$\text{precision} = \frac{tp}{tp + fp} \quad \text{recall} = \frac{tp}{tp + fn}$$

The perfect precision value of 1.0 means that every object characterized as relevant by the decision tree is indeed relevant.
Space Exploration Overview

AIDE Framework

**Goal:**
- Discover *relevant area(s)*
- Formulate user queries to select *relevant area(s)*
3 Exploration phases:

1. Relevant Object Discovery
2. Misclassified Exploitation
3. Boundary Exploitation

Assumption:
The user interests are captured by range queries, i.e., relevant objects are clustered in one or more areas in the data space.
1. **Relevant Object Discovery**

**Output:** *A sample from diverse data areas for reviewing by the user.*

**Steps:**
- Each normalized attribute domain is divided into equal width ranges. → Grid cells
- One data object is retrieved from each non-empty cells (near the center of the cell). → Sample

*After the review: A Classifier is created/updated*
2. **Misclassified Exploitation**

The diagram illustrates the concept of misclassified exploitation in a classification task.

- **Irrelevant** area: may contain false negative objects.
- **Relevant** areas:
  - Irrelevant areas
  - Relevant areas

The classifier divides the attribute space into relevant and irrelevant areas based on the normalized domain of attributes A and B.
2. **Misclassified Exploitation**

Aim: Examine further « irrelevant » area(s)

![Diagram showing relevant and irrelevant areas](image)
2. **Misclassified Exploitation**

Aim: Examine further «irrelevant» area(s)

*In a lower exploration level*

- Irrelevant areas
- Relevant areas

! may contain false negative objects!
2. **Misclassified Exploitation**

Aim: Examine further « irrelevant » area(s) in a lower exploration level

Next iteration: the classifier is updated and includes new relevant areas

---

**Diagram Details**

- `x`: irrelevant sample
- `o`: relevant sample
- `diamond`: cell center
- `δ = 50` ($β=2$)

**Legend**

- Relevant areas
- Irrelevant areas

**Classifier**

- Relevant areas
- Irrelevant areas
3. **Boundary Exploitation**

- Need to redefine boundaries (remove false positives)
  
→ Varying the sampling area