Mimir: ETL Made On-Demand

Poonam Kumari, William Spoth, Aaron Huber, Jon Logan, Lisa Lu, Oliver Kennedy

Alumni: Arindam Nandi, Niccolo Meneghetti, Vinayak Karuppanasamy, Jacob Varghese, Ying Yang

The ODIn Lab @ UB

Motivation

Efficient analytics depends on accurate, reliable, high-quality information. However, raw data is messy.

1. Upfront cleaning: clean all messy data before analysis. Drawbacks: Unnecessary processing of unused data.
2. Inline cleaning: clean all messy data when analyzing. Drawbacks: (1) Unnecessary processing of unused data. (2) Duplication of work.
3. On-demand cleaning: delay the cleaning process until needed and clean incrementally. Advantages: Time and cost efficient compared to 1 and 2. We need a general on-demand cleaning framework.

Example

Alice is an analyst from HappyBuy. She wants to explore the ratings of HappyBuy products.

<table>
<thead>
<tr>
<th>Product</th>
<th>Rating</th>
<th>Review Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phone 1</td>
<td>4.5</td>
<td>120</td>
</tr>
<tr>
<td>Phone 2</td>
<td>4.0</td>
<td>60</td>
</tr>
<tr>
<td>TV 1</td>
<td>3.5</td>
<td>80</td>
</tr>
<tr>
<td>TV 2</td>
<td>3.0</td>
<td>40</td>
</tr>
</tbody>
</table>

I am interested in phones and TVs and other product with good ratings.

SELECT p.pid, p.category, r.rating, r.review_ct FROM Product p, Rating r WHERE p.category IN ('phone', 'TV') OR r.rating > 4

Lenses

Domain repair lens

CREATE LENS SaneProduct AS SELECT * FROM Product USING ORACLE REPAIR (category string NOT NULL, brand string NOT NULL);

Lenses make best use of source data and make a best-effort guess using the learnt model.

Schema matching lens

CREATE LENS MatchedRatings AS SELECT * FROM Ratings1 USING SCHEMA_MATCHING ( pid string, ...,

Cells in a generalized C-Table can have arbitrary expressions.

JSON shredder lens

Build functional dependency between columns, this allows us to group columns into 'entities' (parent column) that contain attributes (children columns).

Mapping entities to other entities allows us to preform schema matching on entity selection. This allows simple queries to analyze wide data sets.

User Interface

We aim to design a user interface for presenting query results with query-level uncertainty, optimizing for three objectives:

- Familiarity
- Effectiveness
- Efficiency

The two primary questions that we sought to answer for each of the representations of uncertainty were:

- Is the representation effective at communicating uncertainty?
- What is the cognitive burden of interpreting a representation?

A total of 22 participants drawn from the entire student body of the University at Buffalo participated.

- Time taken to interpret uncertainty is consistent across all forms except for Tolance for CS students.
- Non-CS background participants displayed a quicker decision compared to CS participants in case of asterisk, colored Text and color coding representations. The comparison might suggest that being familiar with the representation (tolerance bands and ranges) reduces the cognitive burden of interpreting uncertainty.
- As a result of this study, we showed that users made rational decisions more quickly with low-bandwidth uncertainty representations like red text or red backgrounds.

Feedback

We use cost of perfect information (CPI) to rank the uncertainties.

CPI is calculated, as follows:

- Alice: I want to improve the result quality.
- Alice: No.
- Alice: Oh, that is good enough, fine then and Thank you.

Generic Schemes For Metadata Propagation

- Propagating deterministic metadata at the query level
- Avoids changing Mimir query annotation
- Allows analyst to propagate information through Mimir queries to determine data correlations

Probabilistic-System Catalog

- Schema-level Information Presentation Responsiveness To UI
- Clearly represents data schema level information to user
- Allows responsiveness to feedback generated by user
- Trivializes JSON data as it changes, indicating nested JSON data
- Represents changes as possible schemata and ease code that a user may wish to work on based on current task
- Possible probabilistic information is retained by Mimir to create best guess assumptions of data

Contributions

We propose Mimir to provide:

- Lenses: A structure to represent different kinds of messy data in a uniform way.
- Analysis: presenting (uncertain) query results to user.
- Feedback: improving the data quality in a cost efficient way.

Mimir is supported by NSF Award #1640864, NPS Award #0024-16-1-0022, and gifts from Oracle.

http://ads.unb.ca/~lu/odinlab/research/Mimir