Real-Time Query Recommendation

David Gràcia, Jake Edwards, Maximiliano Ariel López and Sergi Nadal

supervised by Dr. Patrick Marcel and Dr. Pedro Furtado

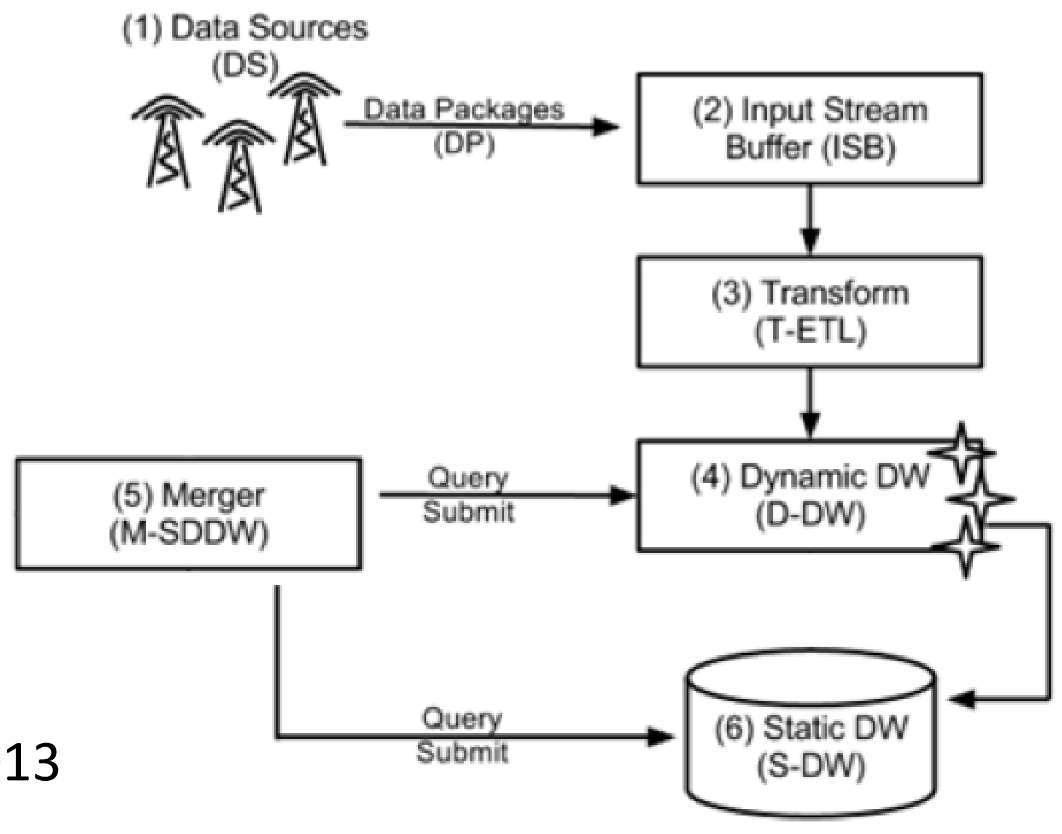
Motivations

- To enable near real-time analysis on recent data, DW architectures have been extended to incorporate real-time functionalities
- Query recommendation techniques for OLAP, leverage past analysis sessions based on query expressions or on historical data
- How can we recommend OLAP queries in basis of real-time information?

Real-Time Data Warehouses

Problem: How to enable access to most recent data?

Solution: Integrated architecture Static-DW + Dynamic-DW.

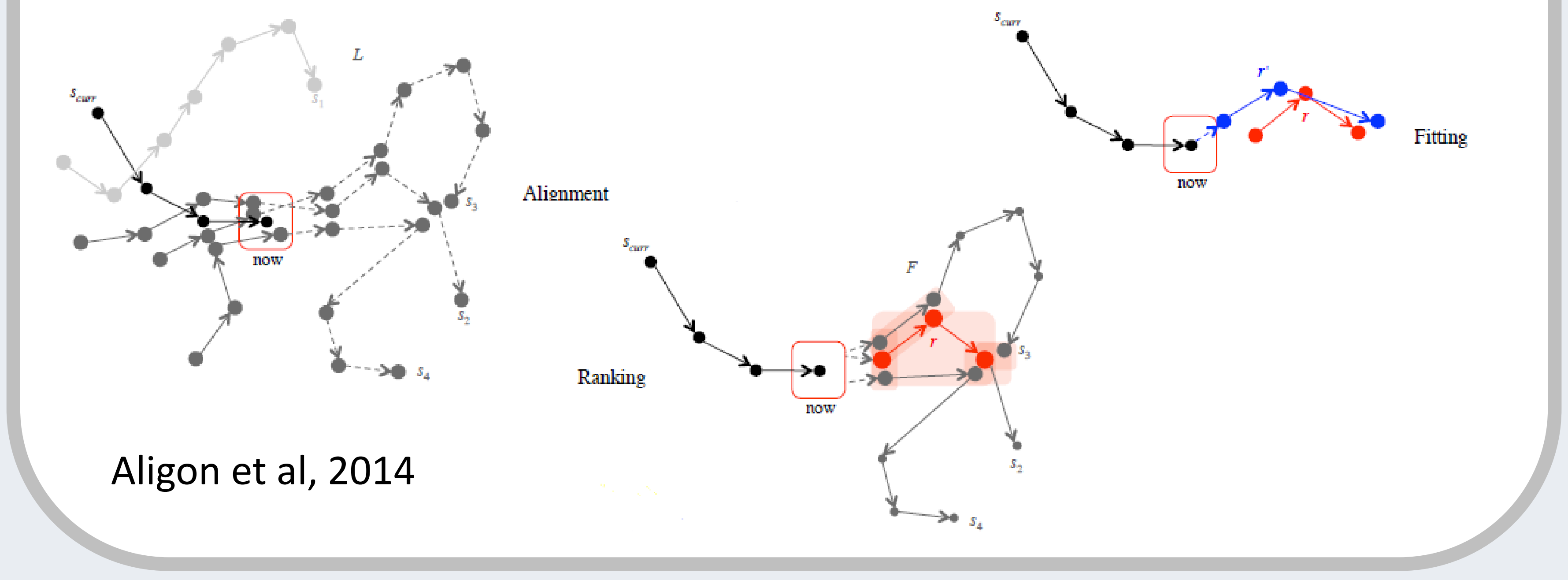


Furtado et al, 2013

OLAP Query Recommendation

Problem: How to recommend OLAP queries?

Solution: Extract and transform queries from the log of previous sessions, based on sessions similarity.



Aligon et al, 2014

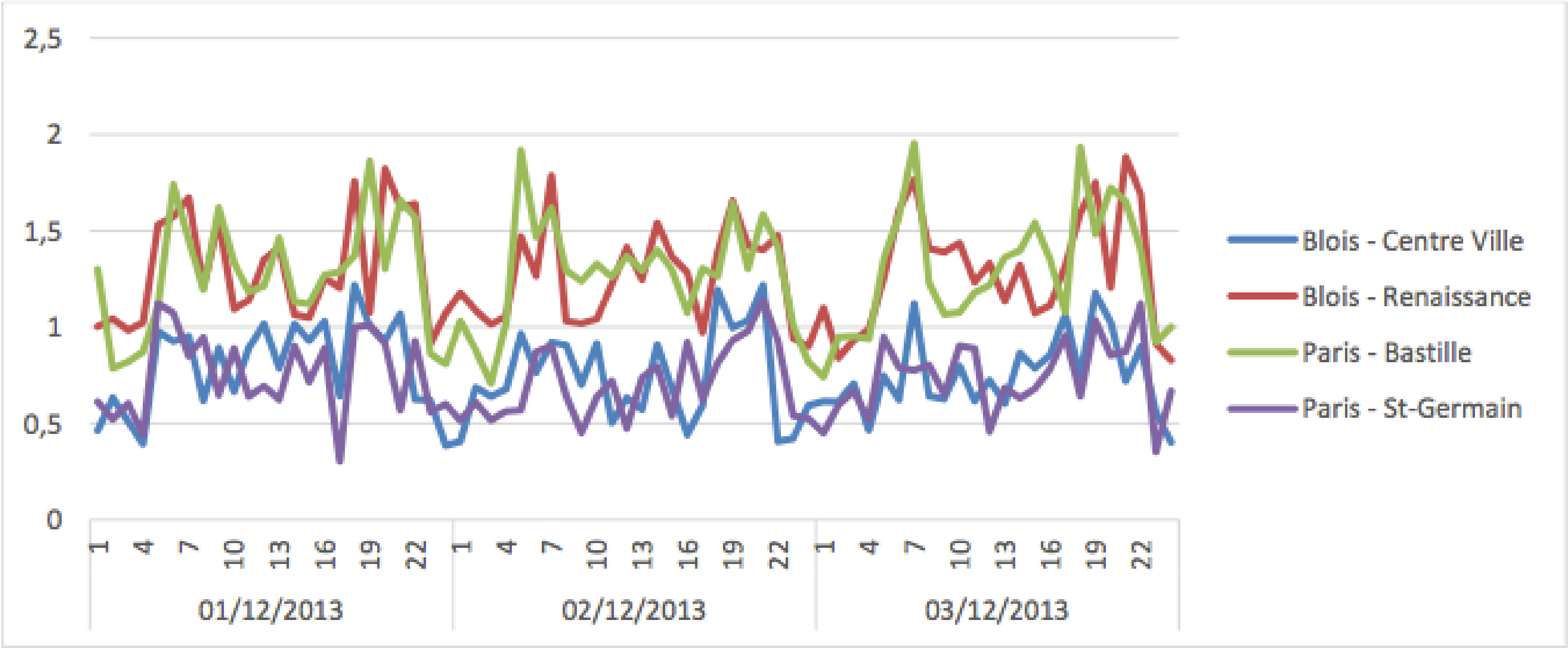
Real-Time Query Recommendation Algorithm

- Rapid detection of anomalies in real-time data.
- The more anomalous a data value, the more interesting it is assumed to be to the user.
- Anomalous data can be detected by using baselines such as mean and standard deviation of measures.

Baselines

Problem: How to efficiently compare real-time with historical information in order to detect anomalies?

Solution: Baselines summarize historical data for different dimensions. Each point is a baseline cell that contains statistical measures.



Prototype

Problem: How to evaluate and validate recommendations?

Solution: Using a prototype to perform experiments.

· OLAP Metadata dynamic detection

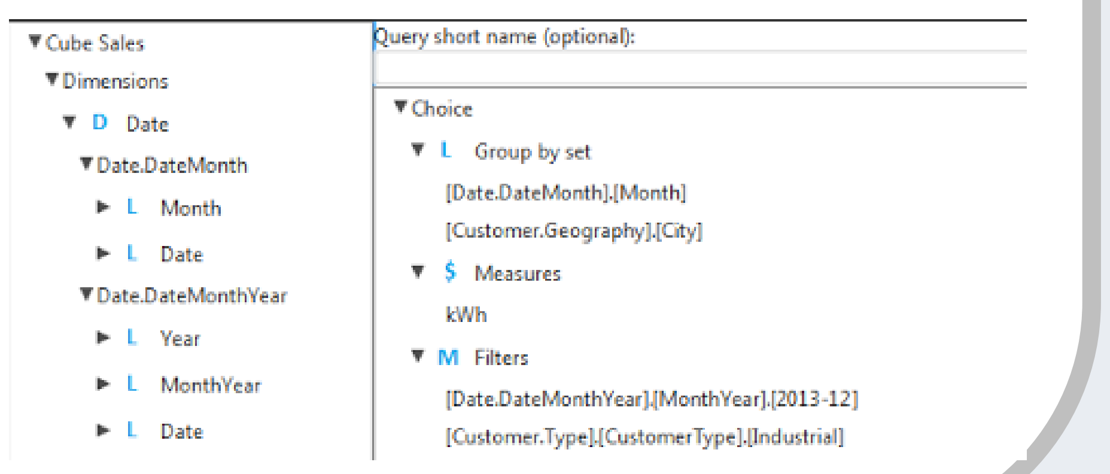
· Query design with drag-and-drop mouse gestures

· Evaluation of queries in Real-Time and/or Historical cubes

· Logs: addition of single items in existing or new sessions

· Baselines: identification, persistence and exploration

· Execution of finding, ranking and mutating of candidates steps



▼Cube Sales
▼Dimensions
▼ D Date
▼Date.DateMonth
► L Month
► L Date
▼Date.DateMonthYear
► L Year
► L MonthYear
► L Date

Query short name (optional):

▼Choice
▼ L Group by set
[Date.DateMonth].[Month]
[Customer.Geography].[City]
▼ $ Measures
kWh
▼ M Filters
[Date.DateMonthYear].[MonthYear].[2013-12]
[Customer.Type].[CustomerType].[Industrial]

Workflow

Problem: How to adapt a current query recommendation algorithm in order to take advantage of baselines?

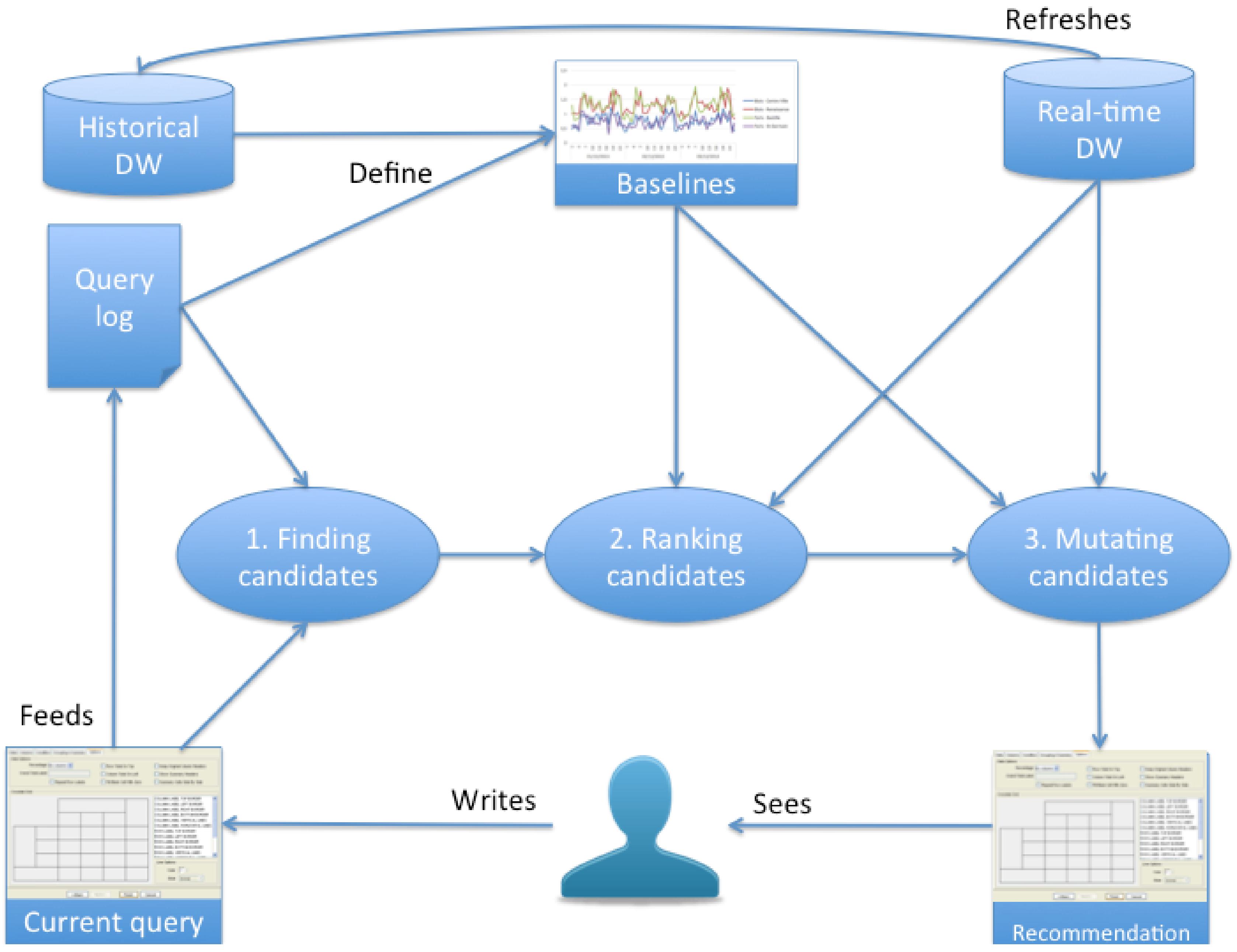
Solution:

- Offline process to calculate baselines based on historical data
- Three steps algorithm, given a current query:

1) Finding candidates: Look for similar candidates

2) Ranking candidates: Sort the candidates based on interestingness

3) Mutating candidates: Perform mutations to the ranked query


Refreshes
Historical DW
Define
Baselines
Real-time DW
Query log
1. Finding candidates
2. Ranking candidates
3. Mutating candidates
Feeds
Writes
Sees
Current query
Recommendation