25th International Workshop on Design, Optimization, Languages and Analytical Processing of Big Data

DOLAP 2023

Assessment Methods for the Interestingness of Cube Queries

Dimos Gkitsakis, Spyridon Kaloudis, Eirini Mouselli, Veronika Peralta, Patrick Marcel and Panos Vassiliadis

Co-financed by Greece and the European Union

D. Gkitsakis was supported by project "Dioni: Computing Infrastructure for Big-Data Processing and Analysis." (MIS No. 5047222), implemented under the Action "Reinforcement of the Research and Innovation Infrastructure", funded by the Operational Programme "Competitiveness, Entrepreneurship and Innovation" (NSRF 2014-2020) and co-financed by Greece and the European Union (European Regional Development Fund).
P. Vassiliadis has been co-financed by the European Regional Development Fund of the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call Research - Create - Innovate (prj code:T2EDK-02848).

Problem and Context

- Given a cube query and prior knowledge (already answered queries or simply user beliefs), how can we assess how interesting a cube query is, based on Interestingness dimensions?
- Context:
  - Cube querying sessions over a multidimensional, hierarchical database
  - The user has prior knowledge about the cube (query history or beliefs)
  - The user devises queries to acquire new information
  - Each query:
    - Is relevant or not with respect to user's information goal
    - Is different or similar to the queries of the history
    - Contradicts or reinforces the user's beliefs
    - Provides new or already seen information
  - Each query is assessed with respect to the dimensions of Relevance, Peculiarity, Surprise, and, Novelty

Importance of Interestingness Assessment

- A-priori evaluation of query Interestingness
  - Selecting queries of high interest out of many candidates for further processing

- A-posteriori evaluation of query Interestingness
  - Analyzing the results of the most interesting queries that have been already executed

Outline

- Related Work
- Multidimensional Data Space
- Interestingness
    - Novelty
    - Relevance
    - Peculiarity
    - Surprise
- Experimental Results
- Conclusion

Related Work

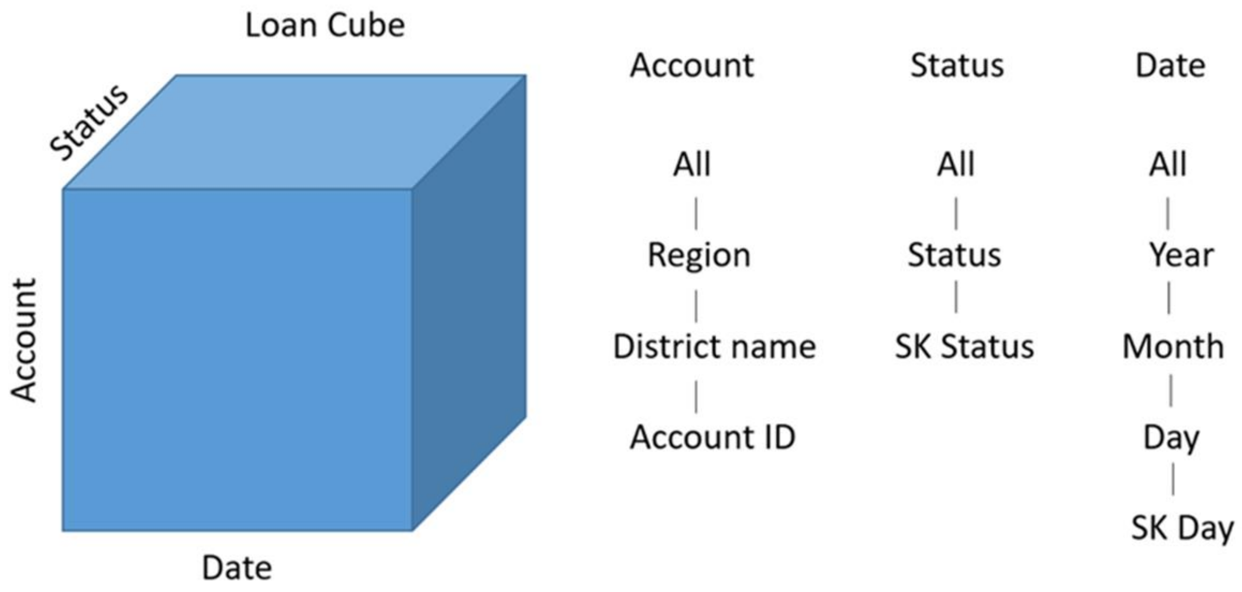
- EDA systems use Interestingness dimensions as metrics, in order to score the insights/highlights/findings that they extract
  - Peculiarity attracts the most attention – different data are more intriguing
  - Novelty is used in order to guarantee that data are new, and move further the exploration
  - Relevance is used in order to characterize data based on how familiar is the user with them
  - Surprise characterizes values that are not shown frequently or challenge user’s prior beliefs
- Cell Interestingness is well addressed, but not enough. We need to assess cube query Interestingness before the query execution too

Outline

- Related Work
- Multidimensional Data Space
- Interestingness
  - Novelty
  - Relevance
  - Peculiarity
  - Surprise
- Experimental Results
- Conclusion

Multidimensional Data Space

- We focus on multidimensional, hierarchical data organized in cubes
- Cubes are relevant to the problem, simple, and information-rich
- Cubes are formed in multidimensional spaces, produced by combinations of dimensions and store measures in their cells



Cube Queries

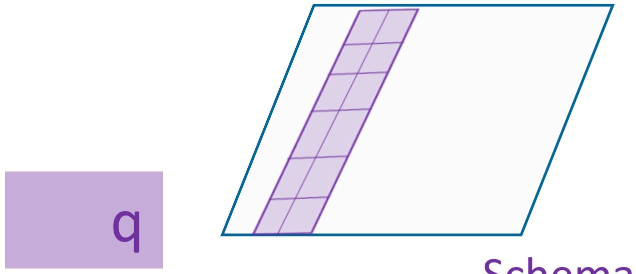
- Dimensions provide context for measures and consist of levels, organized in hierarchies of granularity
- A cube query is a cube too, is specified by
  - a cube over which it is applied,
  - a selection condition, ϕ, a composition of atomic filters for the cube cells,
  - the grouping levels, which determine the detail of the result, and
  - an aggregation over the cube measures

$$q = < C^0, \phi, [L_1, ..., L_n, M_1, ..., M_m], [agg_1(M_1^0), ..., agg_m(M_m^0)] >$$

Detailed Area of Cube Queries

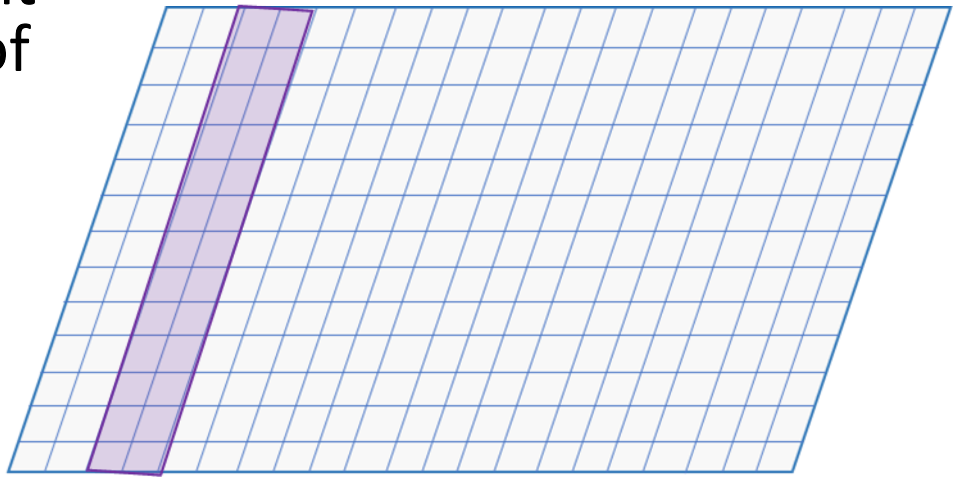
- Detailed Area is the representation of the cells of a query result, in the most detailed levels of their respective dimensions
- A detailed area can be used as common ground for the comparison of cells of different queries that initially were in different levels of detail

σ:
Account.ALL∈{ALL}
Date.Year∈ {1996}



σ:
Account.ALL∈{ALL}
Date.Year∈ {1996}

Schema:
[Account.District,
Date.Month],
[AVG(amount)]



Schema:
[Account.Account ID,
Date.SK Day],
[AVG(amount)]

Outline

- Related Work
- Multidimensional Data Space
- <u>Interestingness</u>
  - Novelty
  - Relevance
  - Peculiarity
  - Surprise
- Experimental Results
- Conclusion

Interestingness

- A generic term indicating the extent to which a piece of information is interesting
- Not a single entity, or metric but rather a vector of scores along several dimensions.
    - Relevance: the extent to which a new piece of information (here: the results of the query) are related to the overall information goals, of the user.
    - Surprise: the extent to which the result of the query contradicts, revises, updates the user's prior beliefs.
    - Novelty: the extent to which the information presented to the users is new, and previously unseen to them.
    - Peculiarity: the extent to which the query is different, and not in accordance with the previous queries of a session or history.

Terminology

- Syntactic vs Extensional Assessment: The first is based only on query definition, the second includes the cells of the result
- Same Level vs Detailed Assessment: The first occurs when two assessed cubes are at the same level of aggregation, the second uses their most detailed levels of aggregation as common ground in order to compare their cells
- Full vs Partial Assessment: The first means that the results of the assessment will include a true/false answer, while the second returns a real number in [0.0-1.0]

Outline

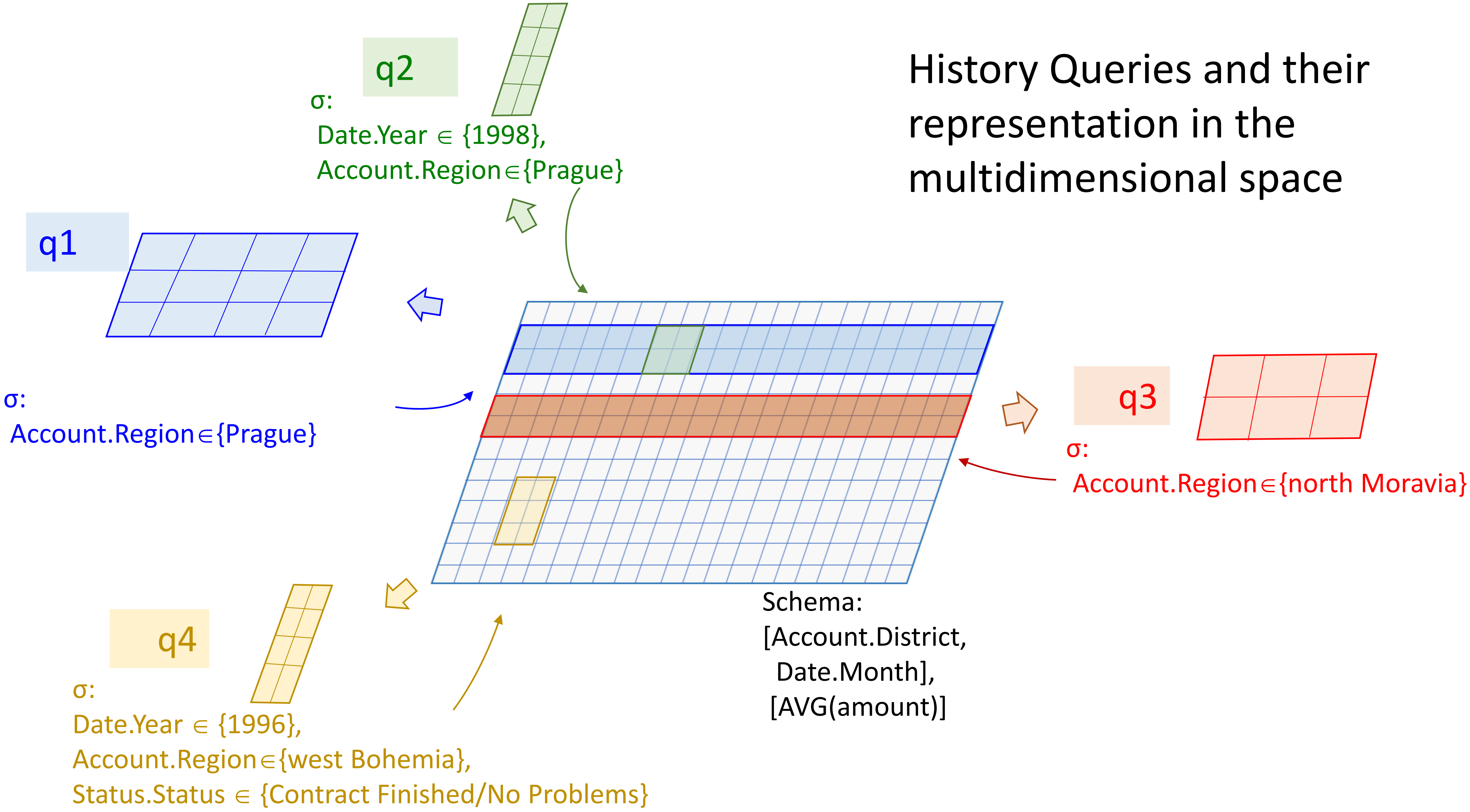
- Related Work
- Multidimensional Data Space
- <u>Interestingness</u>
  - <u>Novelty</u>
  - Relevance
  - Peculiarity
  - Surprise
- Experimental Results
- Conclusion

Novelty

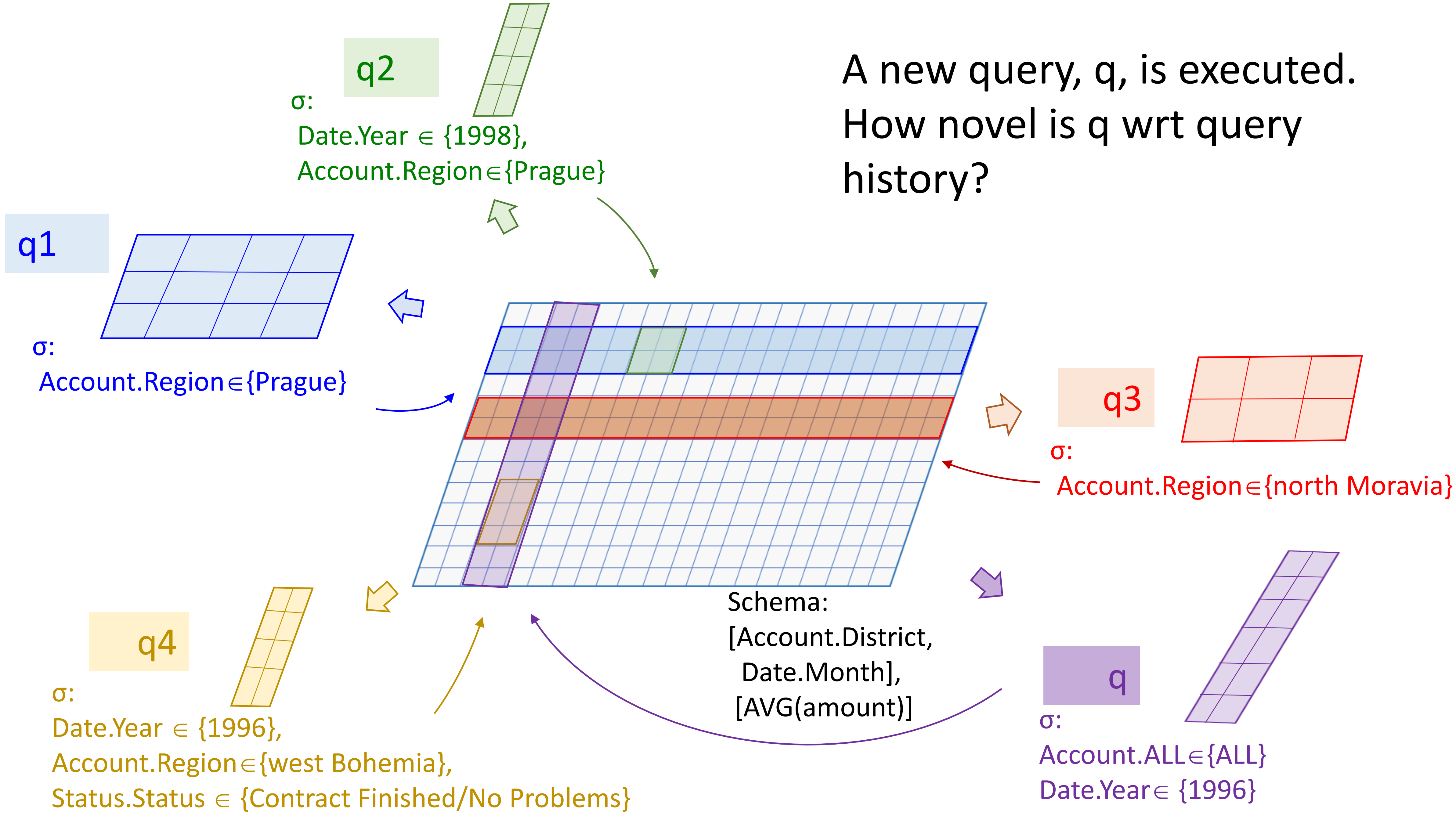
- Novelty assesses the amount of previously unknown information produced by a query.

- Novelty is mostly related (a) to query history, and (b) to registered values for beliefs with confidence below a certain threshold.

History Queries and their representation in the multidimensional space



A new query, q, is executed. How novel is q wrt query history?



Novelty in the presence of query history

- Assessing the novelty of a cube query q assuming a query history $Q = \{q1, \ldots, qn\}$ exists.
- Detailed Assessment of Novelty
  - Partial Detailed Extensional Novelty. The fraction of the detailed cells of q, which are not covered by the detailed areas of history queries, over the entire detailed area of q.
- Same-Level Assessment of Novelty
  - Full Syntactic Same-Level Novelty. If a query with identical syntax with q is found in the query history, the algorithm returns 0 (not novel), otherwise returns 1.

Algorithm 1: Cell-based extensional enumeration of covered detailed cells

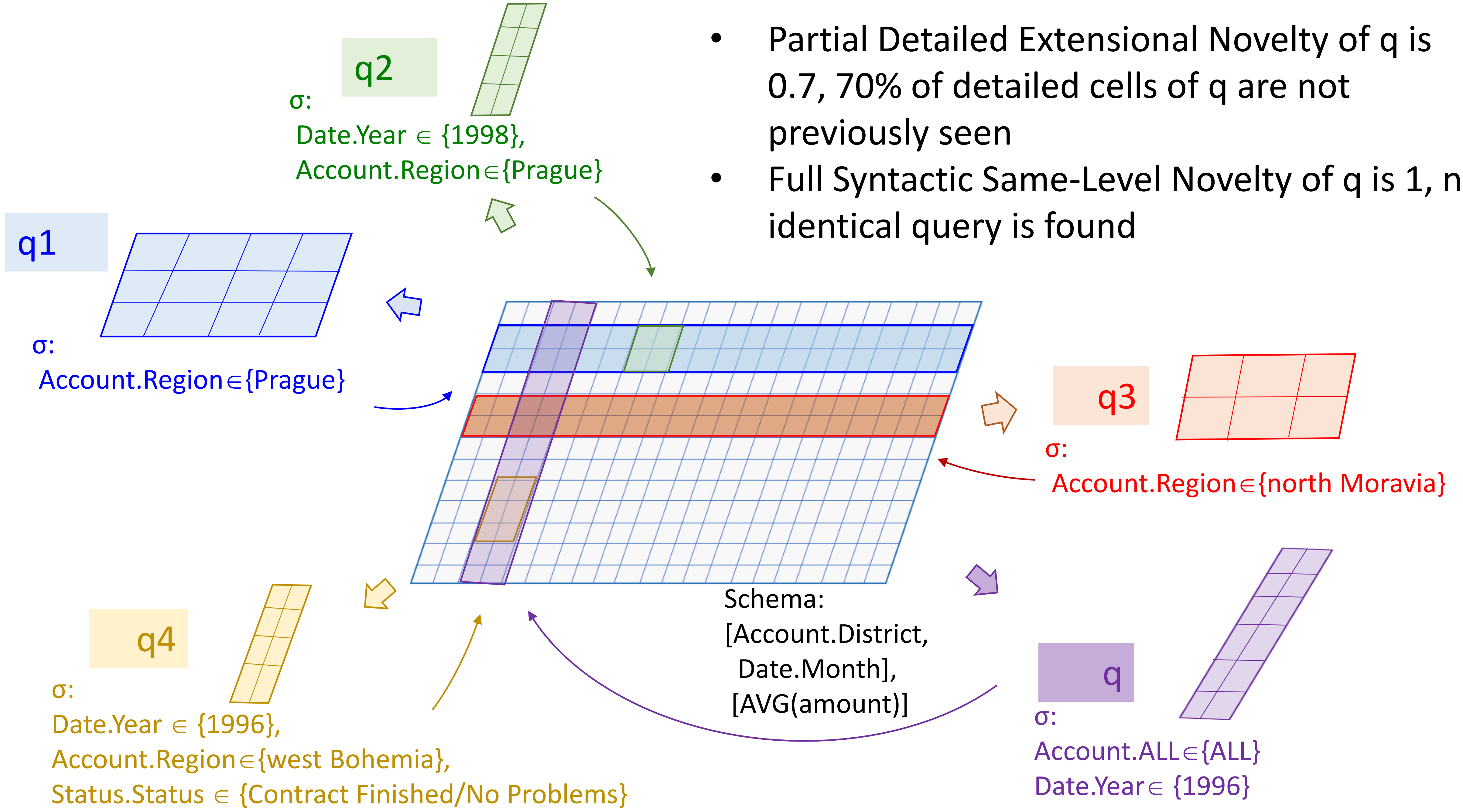
Input: A query q; the query history Q expressed as a set of queries q_i

Output: The subset of the cells of q^0, say q^{cov} that are also part of the union of the results of the queries in Q, i.e., the union of q_i^0, and its complement q^{nov}

```
begin
    produce q^0.cells
    produce q_i^0.cells for all q_i
    populate the hashmap(cell signature) Q^0 ← ∪_i q_i^0.cells
    q^{cov^0} ← ∅
    q^{nov^0} ← q^0.cells
    forall c^0 ∈ q^0.cells do
        if c^0 ∈ Q^0 then
            remove c^0 from q^{nov^0} and add it to q^{cov^0}
        end
    end
    return q^{cov^0}, q^{nov^0}
end
```

$$PartialDetailedExtensionalNovelty = \frac{|q^{nov^0}|}{|q^{nov^0}| \cup |q^{cov^0}|}$$

- Partial Detailed Extensional Novelty of q is 0.7, 70% of detailed cells of q are not previously seen
- Full Syntactic Same-Level Novelty of q is 1, no identical query is found



Novelty in the presence of user’s beliefs

- There is no explicit knowledge about the query history
- We have beliefs, estimations of probabilities about the distribution of values for some cells
- For example, assume the user beliefs:

  - $p(sales \in [100..200) \mid city = Athens, year = 2020) = 30\%$
  - $p(sales \in [80..100) \mid city = Athens, year = 2020) = 70\%$

- For these probabilities, we set a threshold Π (e.g., Π=50%)
- Estimations of probabilities that exceed or are equal to Π are named Π-known

Novelty in the presence of user's beliefs

- Cells that are covered by a Π-known belief are considered "known"
- Partial Detailed Extensional Belief Novelty. When a detailed cell of q is also "known" is considered not novel. Belief Novelty is expressed by the ratio of the detailed, not covered (i.e., novel) cells of q over the entire detailed area of q.

Algorithm 2: Partial Extensional Detailed Belief-Based Enumeration Of Covered Cells

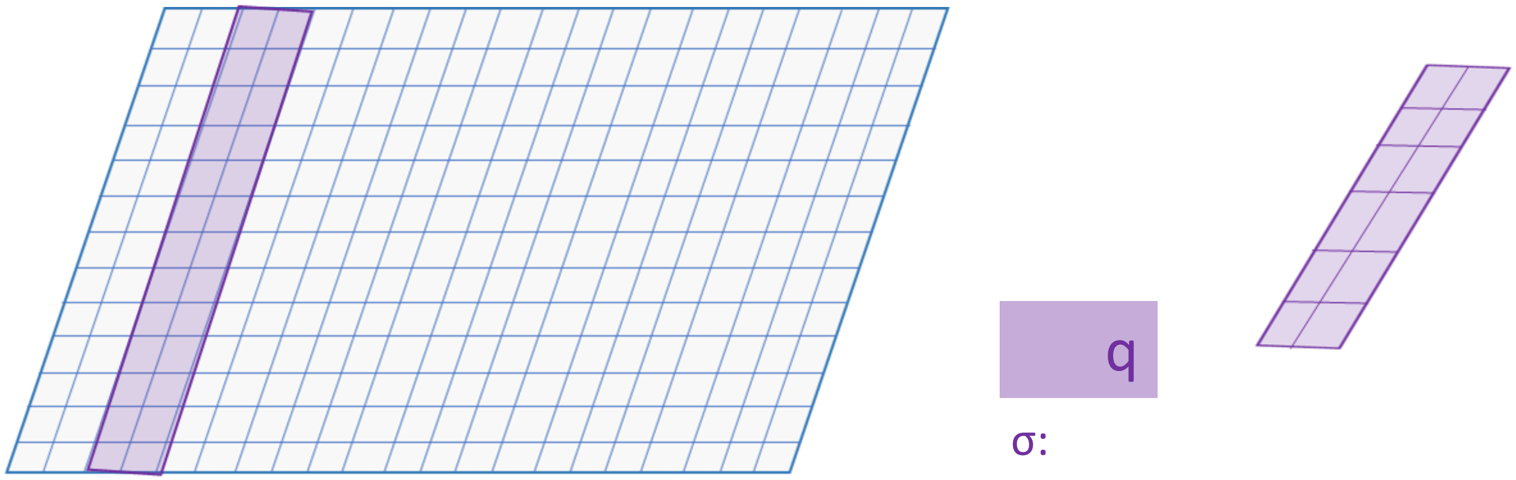
Input: A query q; a set of beliefs B over a set of cells C^B at the most detailed level; a threshold Π for deciding if a cell is eligible for being novel

Output: The subset of the cells of q^0, say q^{cov^0} that are also part of the space the beliefs cover, as well as its complement q^{nov^0}

begin
  produce $q^0.cells$
  $q^{cov^0} \leftarrow \emptyset$
  $q^{nov^0} \leftarrow q^0.cells$
  $C^\star \leftarrow$ the subset of C^B for which there exists a known belief, i.e., $\{c \mid c \in C^B, \exists\, p(M \in m|c) \in B, p(M \in m|c) \geq \Pi\}$
  forall $c^0 \in q^0.cells$ **do**
    if $c^{0^+} \in C^{\star^+}$ **then**
      remove c^0 from q^{nov^0} and add it to q^{cov^0}
    end
  end
  return q^{cov^0}, q^{nov^0}
end

Novelty in the presence of user’s beliefs

- The Belief Novelty for the query q of the previous example is 0.97, indicating high novelty based on the user’s beliefs



σ:
Account.ALL∈{ALL}
Date.Year∈ {1996}

Schema:
[Account.District,
Date.Month],
[AVG(amount)]

Outline

- Related Work
- Multidimensional Data Space
- <u>Interestingness</u>
  - Novelty
  - <u>Relevance</u>
  - Peculiarity
  - Surprise
- Experimental Results
- Conclusion

Relevance

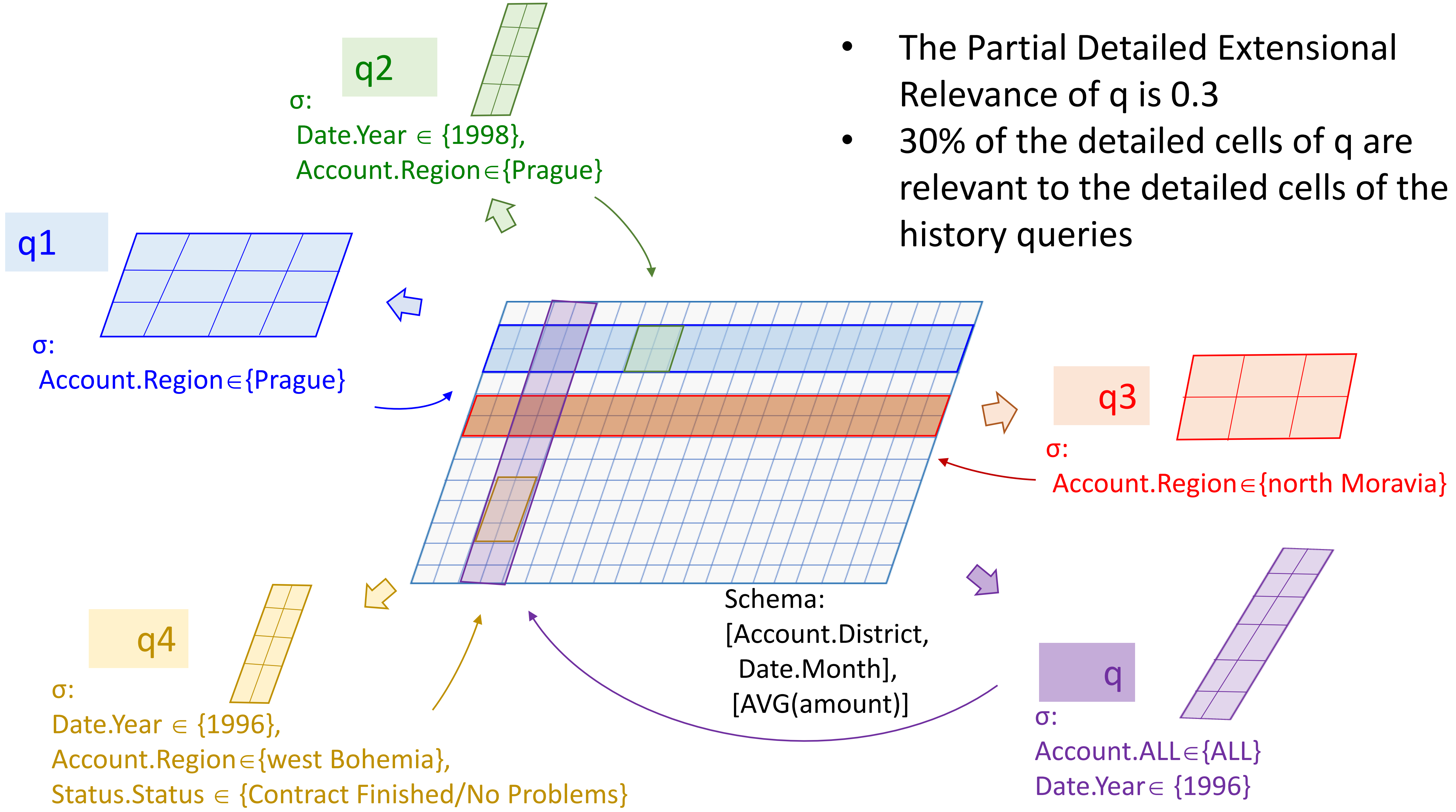
- Relevance is a dimension that pertains to retaining focus towards a specific information goal (or a set of them)
- In the case where the goal is given by user, Relevance is calculated simply by comparing an under-question query and the user-specified goal
- In the case where the goal is not given, the goal has to be inferred from collateral profile information
  - We use the query history $Q = \{q1, \dots, qn\}$ which provides a space of data already seen in the session and which are therefore considered relevant to the current querying session

Relevance in the absence of information goal

- Partial Detailed Extensional Relevance. The algorithm returns the fraction of the detailed cells of q, which are covered (therefore, relevant) by the detailed areas of history queries, over the total detailed cells of q.

$$PartialDetailedExtensionalRelevance = \frac{|q^{cov^0}|}{|q^0|}$$

- The Partial Detailed Extensional Relevance of q is 0.3
- 30% of the detailed cells of q are relevant to the detailed cells of the history queries



Relevance in the absence of information goal

- Partial Same Level Extensional Relevance. In the case that, the under-question query q and some queries in Q are in the same level, the algorithm performs a partial check between the cells of q and the cells of the history queries with the same level with q and returning the ratio of the cells of their intersection to the total number of q cells.

- Both Detailed and Same Level Relevance algorithms have Syntactic equivalents, that compare queries syntax, not cells of their results.

Outline

- Related Work
- Multidimensional Data Space
- <u>Interestingness</u>
  - Novelty
  - Relevance
  - <u>Peculiarity</u>
  - Surprise
- Experimental Results
- Conclusion

Peculiarity

- Peculiarity is evaluated in discriminating a particular query from its peers in the history $Q = \{q1, \ldots, qn\}$.
- Syntactic Peculiarity
    - Partial Syntactic Average Cube Peculiarity measures the peculiarity of a query q by measuring its distance to the queries of Q by pairwise checking their syntactic distance.
    - The syntactic distance of two queries is expressed by the weighted sum of structural distances between their selection conditions, their grouping levels, and their measures

$$\delta(q^a, q^b) = w^{\phi}\delta^{\phi}(q^a, q^b) + w^{L}\delta^{L}(q^a, q^b) + w^{M}\delta^{M}(q^a, q^b)$$

Syntactic Peculiarity

- In our implementation, we use average in order to measure the distance
- We measure the average distance of each query structure to the respective structure of all the queries in Q
- The total structural distance is the weighted sum of all structural distances between q and Q
- We use 0.5 as selection condition weight, 0.35 as grouping levels weight and 0.15 as measure weight

- The Syntactic Peculiarity of q is 0.74
- The query is 74% peculiar on average, with respect to its peers in the history

q2

σ:
Date.Year ∈ {1998},
Account.Region∈{Prague}

q1

σ:
Account.Region∈{Prague}

Schema:
[Account.District,
Date.Month],
[AVG(amount)]

q3

σ:
Account.Region∈{north Moravia}

q4

σ:
Date.Year ∈ {1996},
Account.Region∈{west Bohemia},
Status.Status ∈ {Contract Finished/No Problems}

q

σ:
Account.ALL∈{ALL}
Date.Year∈ {1996}

Value Peculiarity

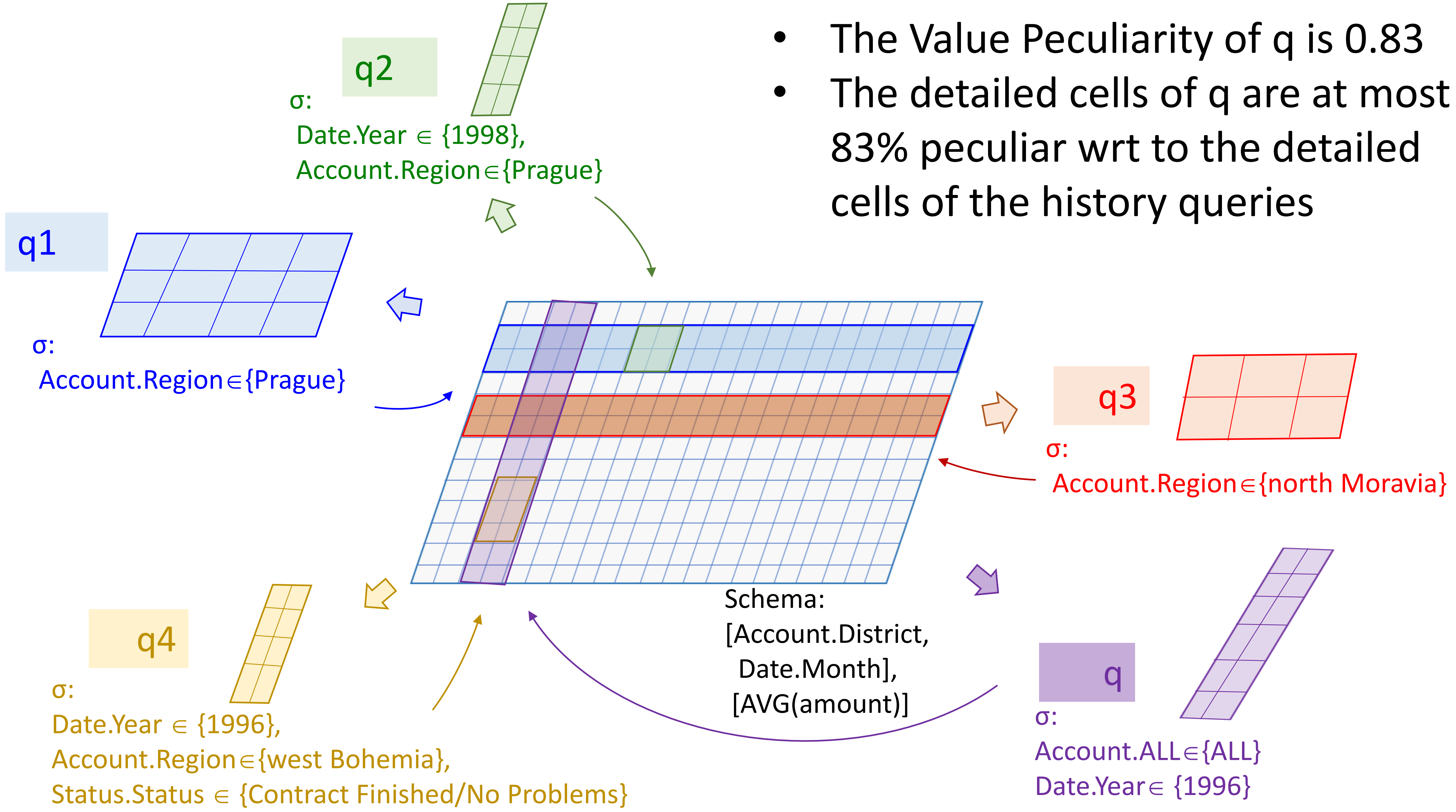
- Value Peculiarity
  - Partial Extensional Detailed Value-Based Peculiarity. We compute the Value Peculiarity as k-th element of a sorted list, which contains the Jaccard distances of the detailed area of the under-question q to the detailed areas of the queries of Q
- In our implementation, we return the 1st element of the list, the element with the maximum distance -> maximum Peculiarity

Algorithm 3: Partial Extensional Detailed Jaccard-Based (Value-based) Cube Peculiarity

Input: A new query q, the query history Q, and an integer k for picking the k-th neighbour
Output: the PartialExtensionalDetailedJaccard-BasedCubePeculiarity $valueBasedPeculiarity(q|Q)$

begin
  Let $L = \emptyset$ a list of Jaccard distances
  Compute q^0, i.e., the detailed area of interest for the query q
  forall $q_i \in Q$ **do**
    Compute q_i^0, i.e., the detailed area of interest for the query q_i
    Compute the Jaccard distance $JD_i = 1 - \frac{|q_i^0 \cap q^0|}{|q_i^0 \cup q^0|}$
    add JD_i to L
  end
  L_s = Sort L ascending into a sorted list
  return $peculiarity(q|Q) = L_s[\ k\ ]$
end

- The Value Peculiarity of q is 0.83
- The detailed cells of q are at most 83% peculiar wrt to the detailed cells of the history queries



Outline

- Related Work
- Multidimensional Data Space
- <u>Interestingness</u>
  - Novelty
  - Relevance
  - Peculiarity
  - <u>Surprise</u>
- Experimental Results
- Conclusion

Surprise

- Surprise depends on prior beliefs, evaluating how far from the prior beliefs of the analyst do the actual values lie.

- For each cell, c, we have (a) its actual value m, and, (b) an expected value m_e.

- Surprise(c) = |m – m_e|

Surprise

- Partial Extensional Value Based Surprise returns the average cell surprise for the query
- The algorithm computes the absolute distance for each cell of the query, between the actual and the expected value (if exists), sums up all the cell surprises and divides it with the number of cells that had an expected value.

Algorithm 4: Value-based surprise assessment for a single measured cube by absolute distance for expected values and averaging of cell surprise

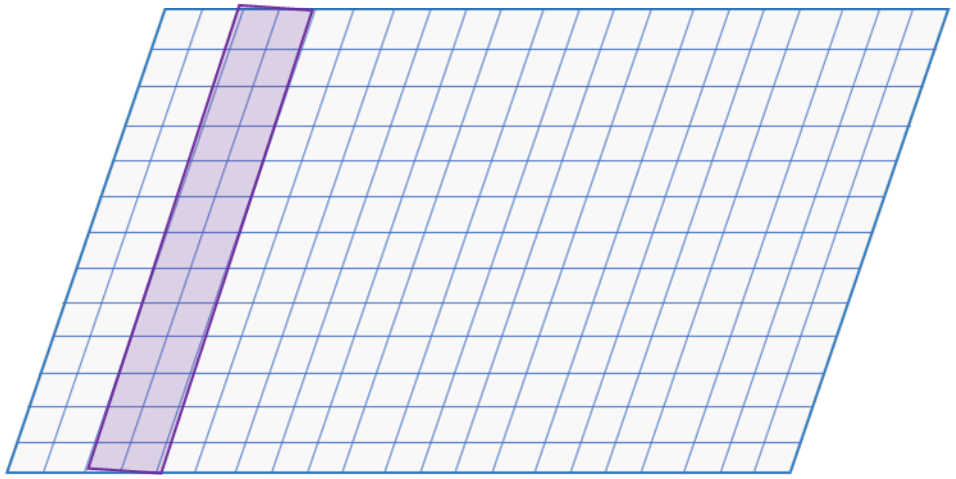
Input: A cube C including a set of cells $\{c_1, \ldots, c_n\}$ with a single measure M, a set of expected values for each cell $E = \{m_1^e, \ldots, m_n^e\}$

Output: The (average) surprise carried by the cube C

```
begin
    countOfCellsWithSurprise = 0;
    C.surprise = 0;
    forall c ∈ C do
        c.surprise = null;
        if ∃ an expected value c.m^e for c.m then
            c.surprise = |c.m − c.m^e|;
            countOfCellsWithSurprise++;
            C.surprise += c.surprise;
        end
    end
    if countOfCellsWithSurprise ≠ 0 then
        C.surprise =
          C.surprise/countOfCellsWithSurprise;
    else
        C.surprise = null;
    return C.surprise;
end
```

Surprise

- The Value Surprise for the query q of the previous example is 0.08
- The average cell surprise of q is 8%



Schema:
[Account.District,
Date.Month],
[AVG(amount)]

σ:
Account.ALL∈{ALL}
Date.Year∈ {1996}

Outline

- Related Work
- Multidimensional Data Space
- Interestingness
  - Novelty
  - Relevance
  - Peculiarity
  - Surprise
- <u>Experimental Results</u>
- Conclusion

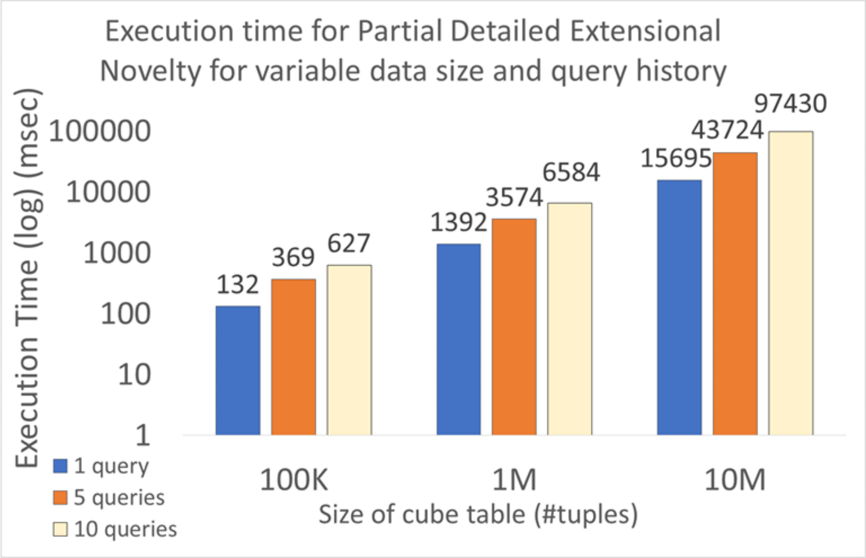
Experimental Results

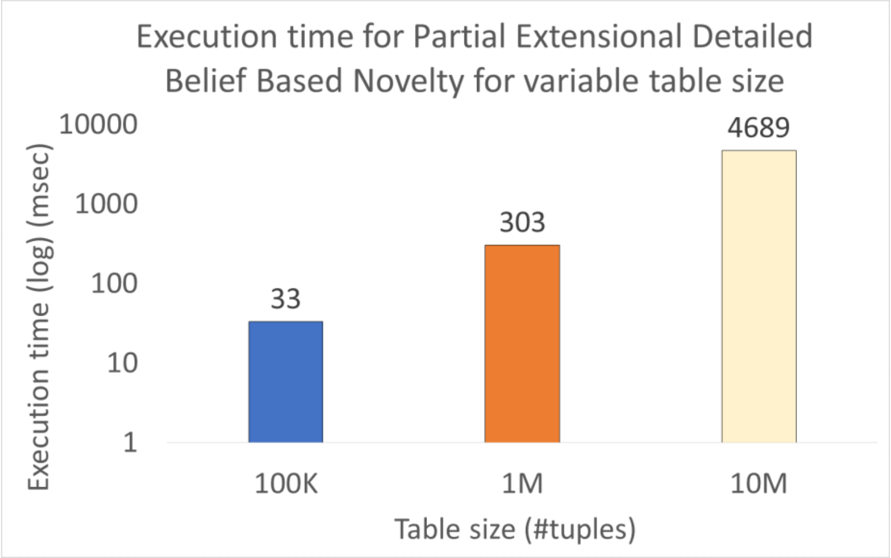
- For algorithms that use history, we assess their performance by increasing
  - The fact table size (100K, 1M, 10M tuples)
  - The history size (1, 5, 10 queries)
- We assess the performance of algorithms that use beliefs by increasing the result size of the executed query
- Performed on the Loan cube of the pkdd99_star database

All the code for the algorithms is available at our Delian Cubes Engine Github:

https://github.com/DAINTINESS-Group/DelianCubeEngine

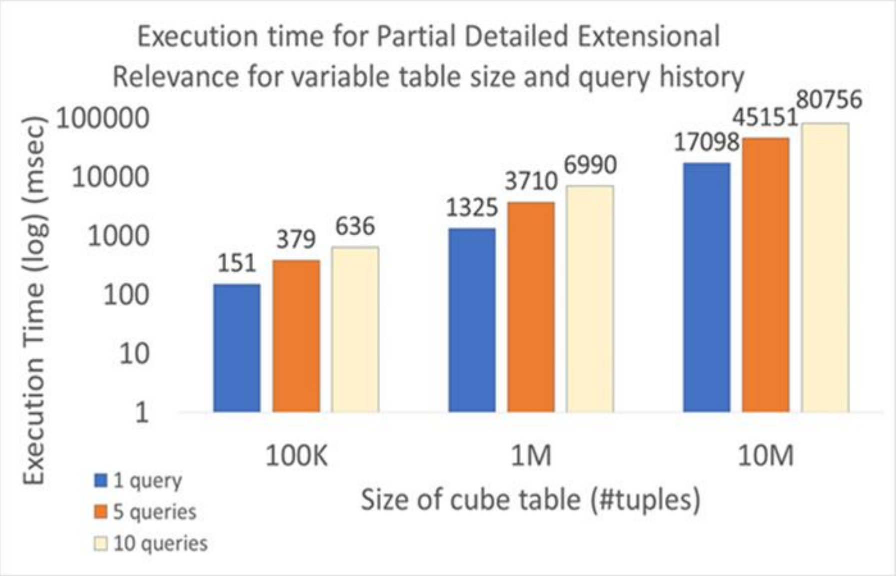
Experiments for Novelty Algorithms

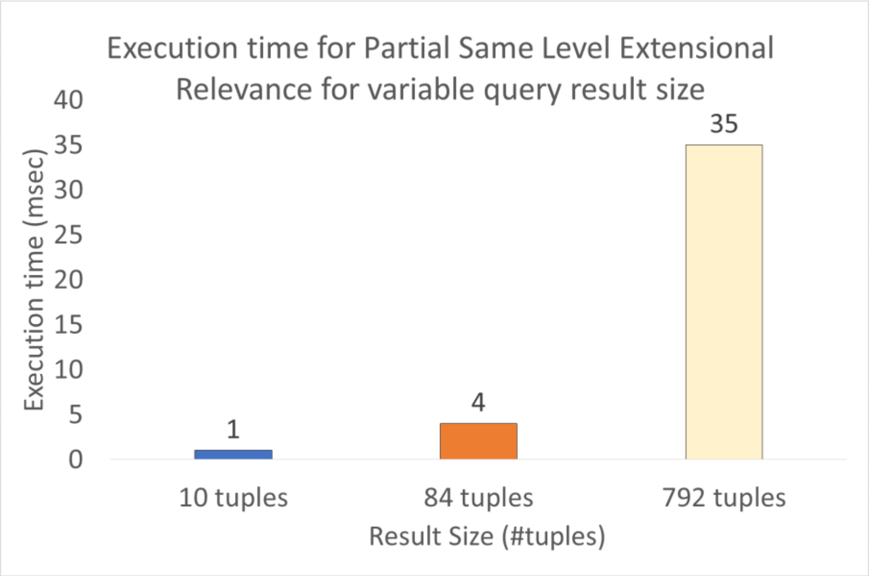




- Detailed Novelty's execution time is linear with respect to the table size and the query history size
- Belief Based Novelty's execution time is linear with respect to the table size
- Belief Based Novelty algorithm is faster because it does not calculate the detailed areas of all the history queries

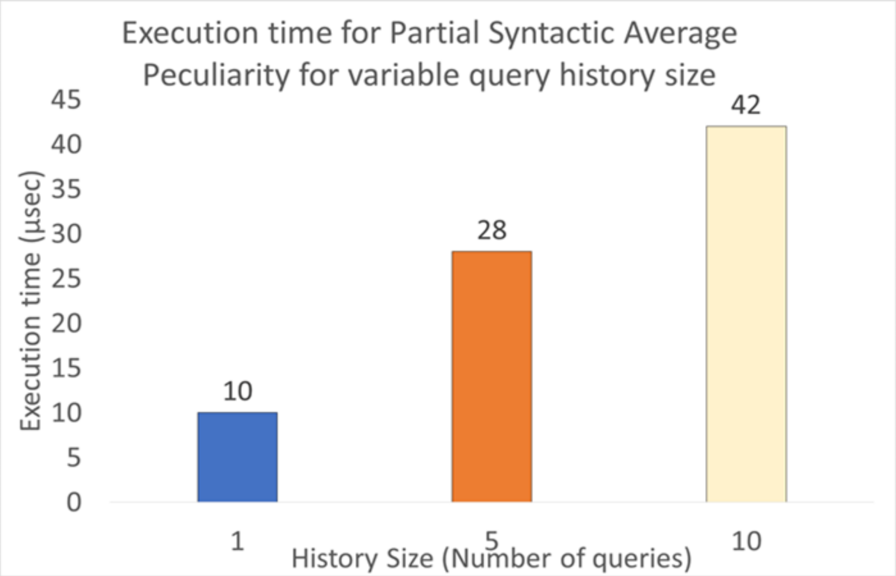
Experiments for Relevance Algorithms

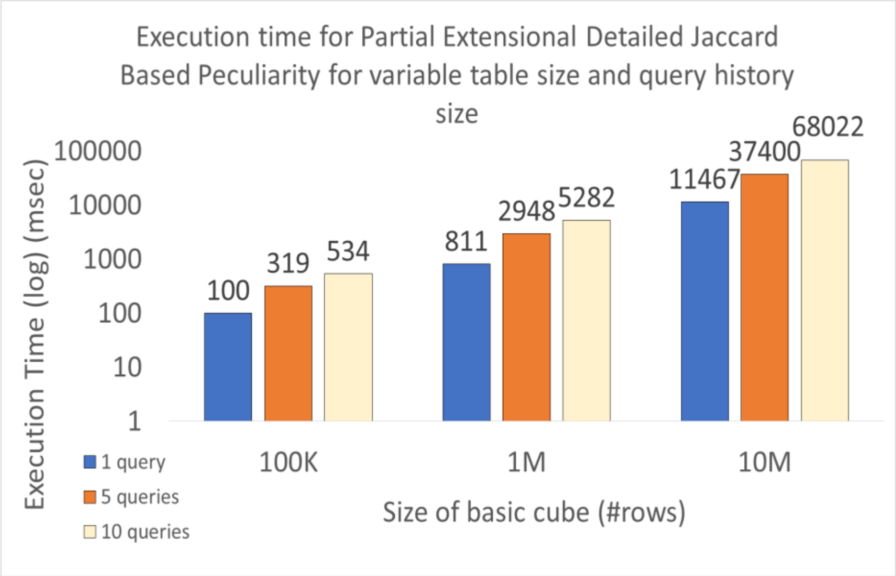




- Detailed Relevance's execution time is linear with respect to table size and query history size
- Partial Same Level Extensional Relevance's execution time is linear with respect to the result size
- Partial Same Level Extensional Cube Relevance is much faster because it does not calculate detailed areas at all

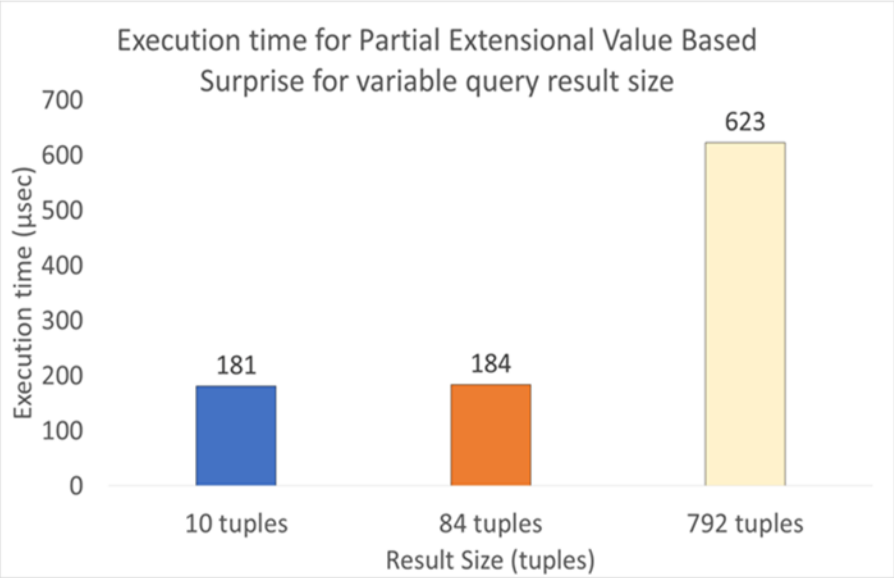
Experiments for Peculiarity Algorithms





- Syntactic Peculiarity's execution time is linear with respect to history size
- Value Peculiarity's execution time is linear with respect to the table size and to the history size
- Syntactic Peculiarity is faster because it simply performs syntactic comparison and does not calculate detailed areas

Experiments for Surprise Algorithm



- The theoretical linear increase with respect to the result size is not exactly achieved.
- We relate the variation of the execution time to the probability of hitting an expected value when the result size of the query is larger, which results in extra time for computing the surprise.

Outline

- Related Work
- Multidimensional Data Space
- Interestingness
  - Novelty
  - Relevance
  - Peculiarity
  - Surprise
- Experimental Results
- <u>Conclusion</u>

Conclusion

- We have addressed the problem of assessing the interestingness of a cube query in the context of a hierarchical, multidimensional database
- We discussed 4 interestingness dimensions, Novelty, Relevance, Surprise and Peculiarity, and we have proposed specific algorithms for their assessment.
- We discriminate signature-based algorithms, before the query is executed (a-priori Interestingness) and result-based algorithms, after the query execution (a-posteriori Interestingness)
- Future work can include more algorithms towards the solution of the problem. Moreover, beyond our four interestingness dimensions, another notable dimension concerns the expression aspect, in which data are assessed for their fitness to the medium that is used to express them

Thank you!

All the code is available at our Delian Cubes Github:

https://github.com/DAINTINESS-Group/DelianCubeEngine

WE ALSO HAVE A LONG VERSION:

Dimos Gkitsakis, Spyridon Kaloudis, Eirini Mouselli, Veronika Peralta, Patrick Marcel, Panos Vassiliadis. Cube Interestingness: Novelty, Relevance, Peculiarity and Surprise

https://arxiv.org/abs/2212.03294

European Union
European Regional Development Fund
HELLENIC REPUBLIC
MINISTRY OF DEVELOPMENT AND INVESTMENTS
SPECIAL SECRETARIAT FOR ERDF & CF PROGRAMMES
MANAGING AUTHORITY OF EPAnEK
EPAnEK 2014-2020
OPERATIONAL PROGRAMME
COMPETITIVENESS
ENTREPRENEURSHIP
INNOVATION
ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη
Partnership Agreement 2014 - 2020
Co-financed by Greece and the European Union