Dimos Gkitsakis, Spyridon Kaloudis, Eirini Mouselli, Veronika Peralta, Patrick Marcel, Panos Vassiliadis (2024). Cube query interestingness: Novelty, relevance, peculiarity and surprise. Information Systems, Volume 123, 102381, ISSN 0306-4379.

Available at: https://doi.org/10.1016/j.is.2024.102381. (https://www.sciencedirect.com/science/article/pii/S0306437924000395)

Cube Query Interestingness: Novelty, Relevance, Peculiarity and Surprise

Dimos Gkitsakis[a], Spyridon Kaloudis[a], Eirini Mouselli[b,1], Veronika Peralta[c], Patrick Marcel[d], Panos Vassiliadis[a]

[a] *University of Ioannina, Ioannina, Hellas*
[b] *Natech S.A., Hellas*
[c] *University of Tours, Blois, France*
[d] *University of Orleans, Orleans, France*

Abstract

In this paper, we discuss methods to assess the interestingness of a query in an environment of data cubes. We assume a hierarchical multidimensional database, storing data cubes and level hierarchies. We start with a comprehensive review of related work in the fields of human behavior studies and computer science. We define the interestingness of a query as a vector of scores along different aspects, like novelty, relevance, surprise and peculiarity and complement this definition with a taxonomy of the information that can be used to assess each of these aspects of interestingness. We provide both syntactic (result-independent) and extensional (result-dependent) checks, measures and algorithms for assessing the different aspects of interestingness in a quantitative fashion. We also report our findings from a user study that we conducted, analyzing the significance of each aspect, its evolution over time and the behavior of the study's participants.

1. Introduction

How interesting is a (data cube) query? What are the fundamental characteristics that make a (data cube) query interesting for a user?

Assessing query interestingness is important for at least two common scenarios: (a) *a-priori* interestingness prediction, and, (b) *a-posteriori* interestingness evaluation.

- A-priori prediction of query interestingness occurs in the case where a recommender system is in the process of automatically generating candidate queries, in order to provide the user with an overview of the information space, as well as with suggestions on how to explore it, or how to follow up on previous query in an on-going query session.
- A-posteriori evaluation of query interestingness is relevant in the case where a large number of queries have already been issued (possibly by other users too), they are cached and readily available, and we need to pick the ones that seem the most significant either in order to recommend them to a user, or, because they highlight best the user actions and goals in the query session.

We note that both scenarios are very relevant to the recent trend in the data management community of automating the exploration of datasets (see e.g., [1, 2, 3, 4]). Being able to accurately score the interestingess of an action over the data (in particular, a query) is instrumental to automatically producing explorations, i.e., sequences of actions over the data. Doing so can be done either by predicting interestingness in an a-priori fashion, or evaluating many potentially interesting queries and picking the best in an a-posteriori fashion. The above are by no means an exhaustive enumeration of cases where the evaluation of query interestingness is important. The common thread in both cases, however,

Email addresses: `dgkits@cs.uoi.gr (ORCID 0009-0006-9559-819X)` (Dimos Gkitsakis), `kaloudis.sp@gmail.com` (Spyridon Kaloudis), `e.mouselli@natechsa.com` (Eirini Mouselli), `Veronika.Peralta@univ-tours.fr (ORCID 0000-0002-9236-9088)` (Veronika Peralta), `Patrick.Marcel@univ-orleans.fr (ORCID 0000-0003-3171-1174)` (Patrick Marcel), `pvassil@cs.uoi.gr (ORCID 0000-0003-0085-6776)` (Panos Vassiliadis)

[1]Work done with Univ. Ioannina

is that both for reasons of efficiency and computational overhead, and for reasons of cognitive load of the person who is involved in the process, it is imperative that a small subset of queries, out of a large number of candidates, are picked for further processing.

In our deliberations, we focus on data organized in cubes due to (a) their extreme relevance to the problem, as analysts explore data in query sessions via Business Intelligence tools, (b) their simplicity – as the simplest possible database setting in terms of how data are presented to the end-users, (c) their most focused setup, also due to the simplicity of the underlying schema, but also because the queries follow a pattern of filtering and grouping with very specific joins between the dimension and fact tables, and, (d) the richness of information content, due to the presence of hierarchically structured dimensions that allow manipulating, examining and understanding the data from multiple layers of abstraction. In other words, cubes are relevant to the problem, simple, allow focused query sessions to take place and demonstrate information richness. This last property is also what differentiates cube queries from regular, relational ones: the presence of a hierarchical multidimensional space allows comparisons at multiple levels of granularity that would otherwise be extremely difficult to express or detect in a relational environment. [2]

Therefore, in our work, we assume an OLAP environment, consisting of cubes, dimensions, levels, and aggregate cube queries posed in the context of user sessions. We will also assume the ability to register, extract, or simply approximate user goals, beliefs and profiles.

What is then the assessment of interestingness for cube queries? To address the question, we will first frame the assessment aspect: we regard assessment as the process where an assessor (person or software) examines specific properties of an object that is evaluated (in our case: cube queries), within a certain context (in our case, as we will demonstrate, the multidimensional space, the query history, the goals, beliefs and interests of the user), for its degree of support/fulfillment of a property (in our case: interestingness aspects) via a method that objectively quantifies the above degree of support via a numerical score or label that is interpretable via a reference scale of assessment.

Intuitively speaking, we need to establish the different *facets of interestingness* and introduce algorithms to numerically assess the objects of study (cube queries) for their "performance" with respect to these properties, in the context of a specific user (with their own characteristics) and a specific session.

A multitude of interestingness measures have been proposed in several science domains *(see e.g., [5] for pattern mining or [6] for recommender systems). As Section 2.2 will show,* the literature is huge and not homogeneously covered, with many facets being largely studied while others are little-known, and the terminology is sometimes ambiguous. In addition, many existing measures concern specific situations and usages and frequently apply only in specific contexts. *For instance, in query recommendation (see e.g., [7]), the foresight measure, defined in [8], is specific to the context of OLAP exploration of multidimensional data.* For a given use case, it is necessary to choose the more appropriate facets and measures, and possible extend and personalize them to fit the use case. Indeed, interestingness is both multi-faceted and context-dependent, and the selection and adaptation of adapted measures is a challenging problem.

In this paper, we provide a systematic taxonomy of several aspects of interestingness, and their relationship with the case of data cubes in hierarchically structured multidimensional spaces, and, we propose specific measures and algorithms for assessing the different aspects of query interestingness in a quantitative fashion.

Our taxonomy is based on fundamental aspects of assessing interestingness, specifically, (a) the knowledge of (a1) the data and schema that we have, (a2) the history of past queries and sessions, and (a3) the user beliefs, goals and interests; (b) the internal workings of the interestingness assessment algorithms on how they exploit the aforementioned input information; and, (c) what is the nature of the output of the assessment. The framework, being based on core concepts is general and applicable to several facets of interestingness, allowing the systematized introduction of algorithms according

[2] The observant reader might have already forecasted that after having successfully addressed the problem in such a setting, generalizing it to arbitrary database schemata, queries and user intention is the next step; the simile is like solving the problem in vitro in a lab, before addressing it in an industrial factory.

to these fundamental characteristics. Moreover the framework allows personalization, by taking user characteristics into account.

Coming to what we mean by the term 'interestingness' in this paper, based on the study of the related literature, both in the area of psychology, and in the area of computer science, we have concluded that *interestingness is not a single entity, but rather, a vector of scores along several facets* [9][3]. *In order to make a solid case for our taxonomic framework, out of the multitude of possible facets of interestingness that we encountered, we decided to focus on four of them that seemed to be fundamental in the bibliographic search, relevant to Exploratory Data Analysis and Business Intelligence and reasonably feasible to measure:*

- *Relevance*: the extent to which a piece of information (here: the results of the query) is related to the overall information goals and preferences of the user.
- *Surprise*: the extent to which the result of the query contradicts and revises the user's prior beliefs.
- *Novelty*: the extent to which the information presented to the users is new, and previously unknown to them.
- *Peculiarity*: the extent to which the query is different, and not in accordance with the previous queries of the session or history.

The facets that we choose to explore are fundamental, important and grounded on the respective literature both of computer science and psychology, as the survey of the related work will soon demonstrate. This, does not imply, by any means, that other facets are not present or significant. Thus, we would like to clarify here, that by no means, do we claim a completeness result, but rather, we would like to take the opportunity to assert the possibility of several more facets to interestingness. We make a related discussion when presenting roads for future work in the end of the paper.

[3]To make concepts clear, we adopt the plain term *dimension* for OLAP dimensions, the terms *facet* and *aspect* for the different aspects of interestingness (called dimensions in [9]), and the term *taxonomic dimension* for the different aspects of the problem definition.

Contributions and Roadmap. The contributions of this paper, and the way they are laid out in this document are as follows:

- In Section 2, we perform a comprehensive review of related work. We do not just survey the related work in the field of computer science, but lay the foundations of our work from the literature in the field of psychology and studies of human behavior. Thus, foundational concepts like interest, curiosity, novelty and surprise act as the starting point for our deliberations.
- In Section 3, we provide a formal framework of the data space within which we operate, along with a reference example, to be used throughout the paper.
- We define the interestingness of a query as a vector of scores along different dimensions, which we call *aspects*, like novelty, relevance, surprise and peculiarity. To assess these scores, we need metrics and algorithms. Before presenting such tools, however, in Section 4, we provide a taxonomy of what information can be exploited, or, equivalently, is needed, for each of the aspects of interestingness.
- In the context of the aforementioned taxonomy, for all the interestingness aspects, we provide both syntax-based (result-independent) and extension-based (result-dependent) measures and algorithms, structured as follows: novelty is discussed in Section 5, relevance in Section 6, peculiarity in Section 7, and, surprise in Section 8.
- We have assessed the proposed framework in terms of effectiveness and efficiency. Concerning the efficiency of the proposed metrics, a detailed experimental evaluation has already been presented in [10] and [11]. Concerning the effectiveness of the framework of interestingness aspects, in Section 9, we present the results of a user study that we conducted, analyzing the significance of each aspect, its evolution over time and the behavior of the study's participants. We demonstrate that although no particular aspect dominates the overall interest for a query, surprise and relevance seem to be more significant. Novelty seems to gain some significance later in the user delibera-

tions, whereas surprise progressively loses significance as the time progresses.

- Finally, we conclude our deliberations in the final section, with points for future work.

2. Related work

In this Section, we start by surveying the different aspects of interestingness in the field of psychology and the study of human behavior. Then, we move on to survey how computer science has attempted to address the issue.

2.1. Interestingness from the viewpoint of the study of human behavior

How can we define interestingness? In this subsection, we frame an answer to this question from the viewpoint of the study of human behavior.

Interest. To the best of our knowledge, there is no formal definition of interestingness. We define interestingness as the property of an object, event or piece of information to be of interest to an individual. Of course this delegates the definition to the task of defining interest.

Online Definitions. Online definitions of interest return "the feeling of wanting to know or learn about something or someone."[4], "the feeling of wanting to give your attention to something or of wanting to be involved with and to discover more about something"[5], "a: feeling that accompanies or causes special attention to an object or class of objects; concern; b: something that arouses such attention; c: a quality in a thing arousing interest"[6] - in other words, it appears that *interest is mostly characterized by the urge of learning more about a subject.* Our exploration of Wikipedia[7] revealed a consistent definition of interest as "Interest is a feeling or emotion that causes attention to focus on an object, event, or process. In contemporary psychology of interest, the *term is used as a general concept that may encompass other more specific psychological terms, such as curiosity and to a much lesser degree surprise*". Practically, this means that *the **interestingness of a piece of information** is the **degree** to which this piece of information ignites the emotion of **curiosity** (which in terms, means the desire to **acquire more knowledge on the issue**), or (less importantly) **surprise** (i.e., the detection -and adaptation to- a **discrepancy** between **newly acquired information** and **preexisting cognitive schemas**).*

Interest from the viewpoint of psychology. In more technical terms, psychology characterizes interest along similar aspects. In [12], interest is characterized as an emotion whose function is to motivate learning and exploration. The author explains that it is hard to structure characteristics of interest due to between-people variability (different people are interested in different things) and within-person variability as interest changes over time. As emotions come from appraisals, i.e., the way people evaluate events, *the author argues that interest comes from two appraisals: (a) the evaluation of an event's **novelty**, **complexity** and **surprise** ("Intuition and decades of research (Berlyne, 1960) show that new, complex, and unexpected events can cause interest") and (b) the evaluation of the **comprehensibility** of an event.*

In [13], the authors provide a definition of interest from the viewpoint of psychology: "We define interests as trait-like preferences for activities, contexts in which activities occur, or outcomes associated with preferred activities that motivate goal-oriented behaviors and orient individuals toward certain environments." The definition highlights two aspects of interests: (a) they are trait-like and (b) they are contextualized, because of an object or activity of interest. According to the authors, interest not only determines choices that people make, but also the success they achieve. In [14], a new theory combines two aspects. On the one hand, interest, referred to as situational interest, is defined as "momentary feelings of curiosity, fascination, and enjoyment triggered by an environment or a task" along with "cognitive evaluations of the value or importance of the environment or task". On the other hand, interests are also traits, referred to as dispositional or individual interests. The authors show how the two aspects can be combined and emphasize the contextualization of interest, i.e., the need for an object of interest in relation to an environment.

On the basis of the aforementioned aspects of interest, we further explore the related concepts of

[4] https://en.oxforddictionaries.com/definition/interest

[5] https://dictionary.cambridge.org/dictionary/english/interest

[6] https://www.merriam-webster.com/dictionary/interest

[7] https://en.wikipedia.org/wiki/Interest_(emotion)

novelty, curiosity and surprise, in order to determine more concretely what interest is all about. In the context of these deliberations, peculiarity also emerges as an important factor too.

Curiosity. Are you hungry for new information? According to [15], "curiosity may be defined as the desire to know, to see, or to experience that motivates exploratory behaviour directed towards the acquisition of new information". [15] gives a vivid presentation of how antagonizing theories on curiosity can converge to a unifying model. Specifically, the curiosity-drive theory treats curiosity as a need to acquire information in order to close a knowledge gap between information that is known and information that is unknown to them. This is inline with one of the most highly cited works in the area [16]. The optimal stimulation theory suggests that the exploration for information that takes place concerns pleasurable states of arousal. The combination of these two models into a single, "interest/deprivation" model (where curiosity stems from 'deprivation of information' or from 'interest' (towards pleasing emotions)) is also discussed. [15] also makes a connection of this unifying model to the different neural circuits of wanting and liking which are correlated but distinct and discusses the issue of indifference/boredom/lack of curiosity.

So how is curiosity related to a person's interest after all? Depending on whether we are hungry to eliminate our ignorance, or simply enjoying learning something new, the answer can differ. In the first case, when the 'wanting' of information is intense, a concrete answer to an underlying question, the 'solving of a puzzle' and, in summary, the ***closing of the knowledge gap****, are the issues that have to be addressed. In the second case,* ***novelty via new and unusual stimuli*** *('tell me something I don't know') seems to be the answer (and esp., the cure for boredom when both the 'wanting' and 'liking' motives are low).*

We refer the interested reader to [17] for a recent survey on the developments in the area of understanding curiosity; [16] albeit older gives a nice categorization of the efforts encountered up to its time (including a historic overview starting from Aristotle and St. Augustine, to Bentham, Kant, Freud and Pavlov) and also offers the information gap theory which seems to withstand criticism up to now.

Novelty. [18] discusses novelty from the viewpoint of psychology with respect to when people characterize events as novel, and how the mental processing of these events takes place. Interestingly, people are predominately correlating positive feelings to the opposite of novelty, familiarity. Yet, this does not necessarily mean that novelty is correlated with negative feelings; in fact, it appears that both our attraction/aversion to a novel event, as well as the characterization of the event itself as novel or not, depend on several other factors (predisposition being a major one).

Novelty occurs when an event (in our case: demonstrated information) does not fit existing mental categories. People are not necessarily negatively predisposed to such a situation, due to their inherent 'motive to know' (as already mentioned for curiosity, closing knowledge gaps can produce pleasant feelings). Then, people try to understand it and in order to do so, they apply a typical mental reaction: they try to relate it to events or information with which they are already familiar. Practically this means that people try first to abstract the incoming input and categorize it in larger, preexisting mental categories (practically searching for similarities with these larger categories). If this attempt fails, the focus is shifted to details and dissimilarities from more detailed mental categories, on the grounds of detailed aspects. Notably, the above process is not followed in the case of threat, where people immediately focus to the details, as typically happens when self-protective motives predominate.

Overall, novelty is strongly correlated to curiosity and occurs when a person fails to include the demonstrated information / event / object into a ***pre-existing mental category****. The processing of novel information starts from trying to align it with high-level, abstract phenomena that promote the* ***detection of commonalities****, and later, esp., if the commonalities are not there, with a drilling into the* ***details that cause dissimilarities****.*

Surprise. Surprise is the third aspect of interest that we discuss. [19] defines surprise as "A peculiar state of mind, usually of brief duration, caused by unexpected events of all kinds ... (via) ... an evolved mechanism whose function is (a) to detect discrepancies between cognitive schemas and newly acquired information, and (b) if they are detected, to instigate processes that enable the short- and long-term adaptation to them." Practically speaking, the main idea is that our beliefs about objects, events and their sequences are structured in so-called 'schemas' and whenever a significant discrepancy (above a certain threshold) is detected between the underlying belief schema and a new input (new information, in our case), the surprise mechanism elicits a surprise reaction that involves (a) analysis and evaluation of the event, (b) the possible reaction to it, and (c) the revision and adaptation of the schema, to remove the discrepancy. Therefore, *surprise occurs when our previous beliefs are disconfirmed or contradicted.* In fact, there are two types of surprise depending on what kind of belief is challenged: (a) misexpected events occur when a belief is directly challenged (e.g., I originally believed that sales in Athens are approximately 100K and they turn out to be less than 50K, which I deem as an important discrepancy), and, (b) unexpected events occur when an implied belief is challenged, due to the challenging of background or contextual beliefs (e.g., I expected to see a drop in the sales of wine, because the price had gone up, but instead consumption turned out to be steady).

Surprise is different from novelty: whereas surprise involves new information that challenges the things we already know, novelty involves new information concerning things that we did not previously know.

An important lesson coming from the study of the mechanisms of surprise is that when attempting to enrich our data exploration systems with a forecasting of what can be surprising for the user, it is important to (a) try to structure the **beliefs of the users** (practically: the values they expect to see) for the explored data in a structured schema (which can include rules, inferences, probabilities, . . . , based on factual data, the history of what they have seen before, explicitly stated assumptions that the users make, etc.), and, (b) to incorporate mechanisms of adapting this schema to new information, as it progressively demonstrates itself.

In the context of these deliberations, peculiarity, novelty and relevance also emerge as important factors too, both collaborating towards satisfying the analyst's curiosity.

Satisfying the Analyst's Curiosity. Both [16] and [17] discuss the efforts of D.E. Berlyne [20] to establish a taxonomy on curiosity. The taxonomy classifies curiosity as perceptual (typically encountered in animals) vs epistemic (mostly encountered in humans, aimed at acquiring knowledge) on the one hand, as well as specific (targeted at a particular piece of information) vs diversive (not associated with specific rewards or punishments). Diversive curiosity - which is not part of the above-mentioned aspects of interestingness – has been heavily criticized as concerns its essence as curiosity or not (see [16] for a discussion), yet it reveals a new possibility, the one of seeking information beyond a specific task, the one that [15] tries to unify into a single theory with the closing of an information gap. [17] makes an interesting observation on information tradeoff tasks: *"The optimal strategy requires adjudication between exploration (sampling to improve knowledge and, therefore, future choices) and exploitation (choosing known best options). Sampling typically gives a lower immediate payoff but can provide information that improves choices in the future, leading to greater overall performance."* In other words, the idea of sampling the information space for a broader understanding of what lies in it, might provide delayed rewarding, but overall greater performance. This view is further enhanced by the authors discussing how a longer time horizon strengthens the propensity of subjects to explore, as opposed to behaviors in the knowledge that the context will dramatically change soon, in which case subjects opt for more immediate rewards.

Thus, investing into understanding the information space in its entirety seems to be an inherent aspect of curiosity. Curiosity per se is a trait of the analyst's mind and not an aspect of query interestingness; however, to translate curiosity to query interestingness aspects on the basis of the aforementioned discussion, we employ **relevance, novelty and peculiarity**. Whereas relevance is targeting towards pursuing a specific, exploitative goal, peculiarity and novelty aim to strengthen the understanding of the broader information space: novelty in terms of information not previously known, and peculiarity in terms of information significantly different than what is already known.

2.2. Earlier proposals of interestingness measures

Various interestingness measures were proposed in the different areas of data exploration. In this subsection we discuss interestingness measures proposed for (i) pattern mining, (ii) recommendation, and, (iii) interactive exploration of multidimensional datasets.

2.2.1. Interestingness criteria for pattern mining

In [5], the authors point out that interestingness is a broad concept and identify from the literature 9 criteria to determine whether or not a pattern is interesting: conciseness, generality/coverage, reliability, peculiarity, diversity, novelty, surprisingness, utility and actionability/applicability. Specifically:

- Conciseness. A pattern is concise if it contains relatively few attribute-value pairs, while a set of patterns is concise if it contains relatively few patterns.

- Generality/Coverage. A pattern is general if it covers a relatively large subset of a dataset.

- Reliability. A pattern is reliable if the relationship described by the pattern occurs in a high percentage of applicable cases.

- Peculiarity. A pattern is peculiar if it is far away from other discovered patterns according to some distance measure.

- Diversity. A pattern is diverse if its elements differ significantly from each other, while a set of patterns is diverse if the patterns in the set differ significantly from each other. Diversity is a common factor for measuring the interestingness of summaries.

- Novelty. A pattern is novel to a person if he or she did not know it before and is not able to infer it from other known patterns.

- Surprisingness. A pattern is surprising (or unexpected) if it contradicts a person's existing knowledge or expectations. The difference between surprisingness and novelty is that a novel pattern is new and not contradicted by any pattern already known to the user, while a surprising pattern contradicts the user's previous knowledge or expectations.

- Utility. A pattern is of utility if its use by a person contributes to reaching a goal.

- Actionability/Applicability. A pattern is actionable (or applicable) in some domain if it enables decision making about future actions in this domain.

In [5], the authors categorize these criteria in 3 groups: i) objective measures, based only on the raw data (generality, reliability, peculiarity, diversity, conciseness), like for instance the classical support, ii) subjective measures, considering both the data and the user (surprise and novelty), like for instance the informational content [21], and iii) semantic measures, based on the semantics and explanations of the patterns (utility and actionability), like for instance measures based on user preferences [22].

According to De Bie [21], subjective interestingness is particularly well adapted for *exploratory* data mining, whose goal is to pick patterns that will result in the best updates of the user's belief state, while presenting a minimal strain on the user's resources. The data mining process consists of extracting patterns and presenting first those that are subjectively surprising, and then refining the belief. De Bie [21] introduced a formal framework for defining measures of surprise for exploratory data mining, using an information-theoretic approach. The framework consists of quantifying the interactive exchange of information between data and user, accounting for the *user's prior belief state.* Of course, in this context, one challenge is how to define and update the belief of the user. Approximating the belief that the user would attach to the result being expected is modeled as a background distribution, namely, a probability measure over the exploration results. This background distribution, which initially can e.g., be uniform over all the exploration results, is updated after each result is presented to the user.

On a final note, we would like to point out that peculiarity is of particular importance in data mining. Assume a set of objects $X = \{x_1, \dots, x_n\}$ of any kind. When is an item x peculiar? The typical answer to the question, which is pretty much the definition of outlierness, is that x is peculiar whenever it differs a lot from X - $\{x\}$. In particular, [23] provides the following definition for outliers: "*An outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism.*"

To assess the outlierness, or peculiarity of a data value, [23] suggests a few nice ideas, including a convex hull algorithm and a K-th nearest neighbor (kNN) distance algorithm. Quoting from [23]: "Because outliers are defined as data points that are far away from the "crowded regions" (or clusters) in the data, a natural and instance-specific way of defining an outlier is as follows: "*The distance-based outlier score of an object O is its* *distance to its kth nearest neighbor.* "

2.2.2. Interestingness criteria for recommendations

There is a long discussion about interestingness in the area of evaluating recommender systems [24, 25, 6]. We mention [6] as an excellent recent survey on the topic. The survey presents 4 criteria (diversity, serendipity, novelty, and coverage), in addition to the traditional accuracy, for evaluating the quality of a recommendation.

- Diversity. The average/aggregated pairwise distance between items in the recommendation list, according to some distance measure.
- Serendipity. It refers to the process of "finding valuable or pleasant things that are not looked for". It consists of two components: surprise and relevance. A common practice is to compare the generated recommendations with recommendations produced by a primitive baseline system, as the goal of a serendipitous recommender is to suggest items that are difficult to predict.
- Novelty. A novel recommended item is one that is previously unknown to the user.
- Coverage. It reflects the degree to which the generated recommendations cover the catalog of available items.

[6] defines novelty for recommender systems as "A novel recommended item is one that is previously unknown to the user" and then moves one to discuss the difference of novelty with (a) serendipity (a serendipitous item must be both novel and surprising) and (b) unexpectedness (an unexpected item does not have to be novel to the user, but only relevant and different from the user's expectations of what would be recommended to them). Recently, [26] provides an alternative perspective where structural similarity is used to recommend a new query to the user, based on the user's query session and a history of sessions over the database.

Query recommendation techniques (see e.g., [7, 8]) are usually evaluated with interestingness measures coming from the literature on recommender systems exposed above. We mention the more OLAP-specific *foresight* measure [8], that quantifies how distant is the recommendation from the current point of exploration.

2.2.3. Interestingness criteria for interactive exploration of multidimensional datasets

Started with the seminal papers by Sunita Sarawagi et al. [27], various interestingness criteria have been proposed to qualify an interesting property or pattern for a subset of the data in a dataset, often called insight, highlights, findings, discoveries, etc., typically characterized by an interestingness score [28, 29, 1, 2, 3]. Two works addressed the classification of these criteria [5, 9].

In [5], the authors also review interestingness measures for what they call summaries, i.e., aggregated cross-tabs corresponding to the result of an OLAP query, where numeric values (i.e., measures) are aggregated by several criteria (i.e., dimensions). Out of the 9 criteria defined for pattern interestingness, 4 are adapted to summaries:

- Diversity. Whether a summary is diverse is determined by two factors: the proportional distribution of classes in the population, and the number of classes.
- Conciseness and Generality. Concise summaries are easily understood and remembered, and thus more interesting than complex ones. Then, a summary is more concise if it is more general (i.e., aggregated).
- Peculiarity. A cell in a summary is peculiar if it is differs from the other cells in the summary.
- Surprisingness/Unexpectedness. A summary is surprising if it deviates from user's expectations. For example, variance can be calculated by replacing observed probabilities by expected probabilities.

According to the classification of [5], the first three criteria are objective and the last one is subjective.

In our previous work [9], we have previously identified four main aspects that differ in what is contrasted to generate interestingness: (i) *peculiarity* (P): the similarity of a cube query to a user's history is assessed (either at the level of the query expression or at the level of the query results); (ii) *novelty* (N): a cube query is contrasted to a user's exploration history; (iii) *relevance* (R): a cube query is contrasted to a user's exploration goal; and (iv) *surprise* (S): the result of a cube query is contrasted to a user's belief. We adopt this classification to review the various interestingness measures proposed.

Peculiarity. It appears that peculiarity has attracted most of the attention in the literature. The main measures defined in this aspect concern either (i) the significance, (ii) the coverage, or (iii) the coherency of the insights.

The *significance* of an insight [30, 31, 32, 33, 34, 35] allows to quantify its importance among its *peer data*. This importance is often related to the data distribution. [33] performs a preliminary ad-hoc attempt to measure significance via the difference in z-scores of the data obtained in two consecutive exploration steps. Recently, a trend is to turn insights into hypothesis testing [31, 32, 35], which has many advantages: (i) using the p-value for the insight significance, (ii) defining false discoveries (type-1 errors, e.g., visualizations supporting a non-significant insight) and false omissions (type-2 errors, e.g., visualizations not supporting a significant insight), (iii) defining credibility (e.g., percentage of visualizations supporting an insight). However, since the risk of type-1 error increases as more than one hypothesis are considered at once, a correction is needed in the statistical test to ensure that non-spurious insights are reported [31].

Discovery-driven analysis [36, 28, 27, 37] for measuring cell interestingness in the context of cube exploration is mostly based on peculiarity-related measures for individual cells. Discovery-driven analysis guides the exploration of a datacube by providing users with interestingness values for measuring the peculiarity of the cells in a data cube, according to statistical models, e.g., based on the maximum entropy principle, and leveraging the intrinsic structure of multidimensional information. From an initial user query, the system automatically calculates 3 kinds of interestingness values for each cell in the query result: (i) $SelfExp$ measures the difference between the observed and anticipated values (the latter are calculated statistically by computing the mean of subsets of attributes), (ii) $InExp$ is obtained as the maximum of $SelfExp$ over all cells that are under this cell (those that result from a drill down), and (iii) $PathExp$ is calculated as the maximum of $SelfExp$ over all cells reachable by drilling down along a given path. The DIFF, INFORM and RELAX advanced OLAP operators proposed in [36, 28, 37] use such interestingness values to recommend relevant cells for explaining drops or increases, or for recommending areas of a cube that should surprise the user, based on their history with the cube.

Klemettinen et al. [38] use skewness, as a peculiarity measure of asymmetry in data distribution, for discovering interesting paths and guiding the navigation in a data cube. Given a cuboid, the possible drill-downs are explored, measuring skewness and generating skew-based navigation rules for the more significant paths. Skewness is computed observing the underlying facts (the raw data that is aggregated), looking for outliers or substantial differences with other facts. Based on skewness, Kumar et al. [39] propose interestingness measures based on the unexpectedness of skewness in navigation rules and navigation paths.

Fabris and Freitas [40] defined interestingness measures for attribute-value pairs in a data cube: the I_1 measure reflects the difference between the observed probability of an attribute-value pair and the average probability in the summary and the I_2 measure reflects the degree of correlation among two attributes. Both measures can be seen as value-based conciseness.

Two also recent works [41, 42] are concerned with detecting the validity of insights gained by users when examining query answers. As with other works measuring peculiarity by leveraging the nature of OLAP cubes, this is again achieved by statistical tests comparing data at different levels of detail.

Measuring the *coverage* of the insight consists of quantifying how the subject of an insight represents the *entire dataset* [30, 32, 43]. In most cases, anti-monotonic conditions are checked to prune insights, like, for instance: if the subject of insight A is a superset of the subject of insight B, then the impact of A should be no less than the impact of B.

Characterizing the *coherency* of an insight compares the insight with others in the *exploration session*, to check whether a given exploratory operation is coherent at a certain point. For instance, in [2] heuristic classification rules are used to express general properties of the operations sequence (e.g., a group-by on a continuous, numerical attribute is incoherent) or on the input dataset's semantics (e.g., if the user focuses on flight delays, aggregating on the "departure-delay time" columns is preferred). Other works use distances between exploration actions to measure how coherent a sequence of actions is; for instance, in [35] a weighted Hamming distance of relational query parts is used.

Novelty. Interestingness measures of the novelty aspect are used to characterize data in terms of either being new observations or operations in terms of *favoring going further* in the exploration. In its simplest expression, novelty can simply be measured as a Boolean indicating whether some data have already been seen [33]. However, more advanced definitions exist. For instance, in [3], a diversity measure is computed as the minimal Euclidean distance between the current observation and all the previous displays obtained. In [44], curiosity is inversely proportional to the number of times a result is encountered.

Relevance. Interestingness measures of the relevance aspect are used to characterize data in terms of the user being *familiar* with them. This aspect seems to be the one that attracted less attention. In [44], a familiarity measure is defined as the concentration ratio of target objects in a set. It is implemented as a variant of the Jaccard index between objects encountered during the exploration and a given target set of familiar objects. This measure is expected to increase as the exploration of the dataset goes on, to avoid over-exploiting a set of familiar objects.

Surprise. Chanson et al. [45], propose a way to measure subjective interestingness for exploratory OLAP, inspired by De Bie's work [21]. The user belief is inferred based on the user's past interactions over a data cube, the cube schema and the other users' past activities. This belief is expressed by a probability distribution over all the query parts potentially accessible to the user. Surprise is then measured as in De Bie's work. Francia et al. [33] propose to measure surprise as the proportion of values that have not been seen frequently, presented in models (e.g., clustering) extracted from the data under observation. In a quite different setting, Sintos et al. [46] use the term surprise to refer to the extent of the incorrectness of a value in a data set – practically measuring the amount of false information of two values before and after a data cleaning procedure.

Combining interestingness measures. Many works combine various interestingness measures, often measures from different aspects. As to how they are combined, there is no consensual approach. For instance a ratio is used in [21], a weighted sum is used in [2, 33, 44], and a product is used in [30, 35]. Djedaini et al. [47, 48] use supervised classification techniques for learning two interest measures for OLAP queries: *focus*, that indicates to what extent a query is well detailed and related to other queries in an exploration, indicating that the user investigates in details precise facts and learns from this investigation [47], and *contribution*, that highlights to what extent a query is important for an exploration, contributing to its interest and quality [48].

2.3. Comparison to related work

There are several axes of comparison to related work for this paper.

What is it so important that makes cube queries special?. As already mentioned in the introduction, the presence of multidimensional spaces with dimensions that are hierarchically structured provides a very specific environment, where cubes at different levels of detail can be related, although potentially defined with different schemata or selection conditions. This facilitates the assessment of all the different aspects of interestingness at a much deeper level, as we can relate cube queries that would otherwise be unrelated.

Given the fact that there is so much previous literature in the field of data and knowledge management on interestingness, why is there a need for a new paper?. A second point that differentiates our work from the rest of the literature has to do that we follow a basic-principles approach, starting from the fundamentals of interest and its aspects in psychology, to establish the ground upon which our modeling takes place. Moreover, in Section 4 we also provide a structured taxonomy of how the analysts' goals, beliefs and interests as well as the computational environment relates to the evaluation of the different aspects of interestingness. To the best of our knowledge, this is the first time that such a structuring (also involving the multi-level hierarchical dimensions of the data space) takes place.

Comparison to our own previous work. Compared to our previous work on *cell* interestingness [9], apart from the basic aspects of interestingness, the two papers have very little to share. In [9], we deal with the problem of evaluating interestingness of individual cells rather than queries, which means we are restricted to the coordinates of the cells, rather than taking into consideration the semantics of the queries. However, a query is much more than a composition of its result cells, esp., if the interestingness of the query is to be assessed before deciding if we will execute it. To answer the reasonable question on why a recommender system might a-priori generate several candidate queries, we believe it is sufficient to mention that different queries rank differently according to different interestingness aspects: therefore, several candidate queries may qualify based on different criteria. A trade-off of performance and interestingness might also affect the recommendation of queries.

Moreover, in [11], we have presented a preliminary version of the present work as a first effort ever to explicitly handle the issue of assessing the interestingness of cubes and cube queries. The present paper extends [11] with (i) an extensive review of related work (the current section), (ii) a taxonomy of the problem's parameters, presented in Section 4, that allows us to clarify the problem and organize the algorithms assessing cube query interestingness in a principled way, (iii) several algorithms and metrics not mentioned in [11] for lack of space, and, (iv) a user study, to evaluate the effectiveness of the proposed algorithms and assess the significance and evolution over time of the assessed metrics.

A longer version of the current paper is found in [10] with more metrics and experiments.

3. Formal Background & Reference Example

In our deliberations, we assume the formal model of [49] (practically, extending [50]) for the definition of the multidimensional space, cubes and cube queries. We follow a simplified apodosis of the formalities here to allow for a concise description.

3.1. Formal Background

Multidimensional space. Data are defined in the context of a multidimensional space. The multidimensional space includes a finite set of dimensions. Dimensions provide the context for factual measurements and will be structured in terms of dimension levels, which are abstraction levels that aid in observing the data at different levels of granularity. For example, the dimension $Time$ is structured on the basis of the dimension levels Day, $Month$, $Year$, All.

A *dimension level* L includes a name and a finite set of values, $dom(L)$, as its domain. Following the traditional OLAP terminology, the values that belong to the domains of the levels are called *dimension members*, or simply *members* (e.g., the values Paris, Rome, Athens are members of the domain of level $City$, and, subsequently, of dimension $Geography$).

A *dimension* is a non-strict partial order of a finite set of *levels*, obligatorily including (a) a most detailed level at the lowest possible level of coarseness, and (b) an upper bound, which is called ALL, with a single value 'All'. We denote the partial order of dimensions with $\preceq$, i.e., $D.L_{low} \preceq D.L_{high}$ signifies that $D.L_{low}$ is at a lower level of coarseness than $D.L_{high}$ in the context of dimension D – e.g., $Geo.City \preceq Geo.Country$.

We can map the members at a lower level of coarseness to values at a higher level of coarseness via an *ancestor function* $anc_{L^l}^{L^h}()$. Given a member of a level L_l as a parameter, say v_l, the function $anc_{L^l}^{L^h}()$ returns the corresponding ancestor value, for v_l, say v_h, at the level L_h, i.e., $v_h = anc_{L^l}^{L^h}(v_l)$. The inverse of an ancestor function is not a function, but a mapping of a high level value to a set of *descendant values* at a lower level of coarseness (e.g., $Continent$ Europe is mapped to the set of all European cities at the $City$ level), and is denoted via the notation $desc_{L^h}^{L^l}()$. For example $Europe = anc_{City}^{Continent}(Athens)$. See [49] for more constraints and explanations.

Cubes. Facts are structured in cubes. A *cube* C is defined with respect to several dimensions, fixed at specific levels and also includes a number of *measures* to hold the measurable aspects of its facts. Thus the schema of a cube is a set of attributes, including a set of dimension levels (over different dimensions) and a set of measures that include factual measurements for the data stored in the cube. Thus, the schema of a cube $schema(C)$, is a tuple, say $[D_1.L_1, ..., D_n.L_n, M_1, ..., M_m]$, with the combination of the dimension levels acting as primary key and context for the measurements and a set of measures as placeholders for the (aggregate) measurements. If all the dimension levels of a cube schema are the lowest possible levels of their dimension, the cube is a *detailed cube*, typically denoted via the notation C^0 with a schema $[D_1.L_1^0, ..., D_n.L_n^0, M_1^0, ..., M_m^0]$. The results of a query q is a set of cells that we denote as $q.cells$.

Each record of a cube C under a schema $[D_1.L_1, \ldots, D_n.L_n, M_1, \ldots, M_m]$, also known as a *cell*, is a tuple $c = [l_1, \ldots, l_n, m_1, \ldots, m_m]$, such that $l_i \in dom(D_i.L_i)$ and $m_j \in dom(M_j)$. The vector $[l_1, \ldots, l_n]$ signifies the *coordinates* of a cell. Equivalently, a *cell* can be thought as a point in the multidimensional space of the cube's dimensions annotated with, or hosting, a set of measures.

A cube c includes a finite set of cells as its extension, which we denote as $q.cells$.

Queries. A cube query is a cube too, specified by (a) the detailed cube over which it is imposed, (b) a selection condition that isolates the facts that qualify for further processing, (c) the grouping levels, which determine the coarseness of the result, and (d) an aggregation over some or all measures of the cube that accompanies the grouping levels in the final result.

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

We assume (again, intentionally simplifying the model of [50]):

- Selection conditions which are conjunctions of atomic filters of the form $L = value$, or in general $L \in \{v_1, \ldots, v_k\}$. Although our theoretical framework covers the latter, as the most general case, typically, the encountered expressions in practice are of the former, special-case, format. In any case, what is important is the property that selection conditions of this form can eventually be translated to their *equivalent* selection conditions at the detailed level, via the conjunction of the detailed equivalents of the atoms of ϕ. Specifically, assuming an atom $L \in \{v_1, \ldots, v_k\}$, then $L^0 \in \{desc_L^{L^0}(v_1) \cup ... \cup desc_L^{L^0}(v_k)\}$, eventually producing an expression $L^0 \in \{v'_1, ..., v'_{k'}\}$ is its detailed equivalent, called *detailed proxy.* The reason for deriving ϕ^0 is that ϕ^0, as the conjunction of the respective atomic filters at the most detailed level, is directly applicable over C^0 and produces exactly the same subset of the multidimensional space as ϕ, albeit at a most detailed level of granularity. For example, assume $Year \in \{2018, 2019\}$, its detailed proxy is $Day \in \{2018/01/01, \ldots, 2019/12/31\}$. We assume a single atomic filter per dimension. For a dimension D that is not being explicitly filtered by any atom, one can equivalently assume a filter of the form $D.ALL = all$.

- We define a grouping level for each dimension (remember that every dimension D includes a single-valued level $D.ALL$, practically signifying the exclusion of the dimension from the grouping – i.e., we group for all the members of the dimension).

- Aggregation functions agg_i belong to the set of frequently used aggregate functions like $\{sum, max, min, count, ...\}$ with the respective well-known semantics.

The semantics of the query are:
(i) apply ϕ^0, the detailed equivalent of the selection condition over C^0 and produce a subset of the detailed cube, say q^0, known as the detailed area of the query,
(ii) map each dimension member to its ancestor value at the level specified by the grouping levels and group the tuples with the same coordinates in the same same-coordinate group,
(iii) for each same-coordinate group, apply the aggregate functions to the measures of its cells, thus producing a single value per aggregate measure.

A cube query is also a cube under the schema $[L_1, ..., L_n, M_1, ..., M_m]$, with the set of cells of the query result (denoted as $q.cells$) as its extension.

Signatures and detailed areas. We will use the term *signature* to refer to sets of coordinates that specify an area of interest in the multidimensional space. Specifically:

- The signature of a cell c, denoted as c^+, is its coordinates, that uniquely identify the area of the multidimensional space that pertains to it.

- The signature of an atomic filter $\alpha : L \in \{v_1, \ldots, v_k\}$ is the value set $\{v_1, \ldots, v_k\}$ and it is denoted as α^+.

- The signature of a selection condition of the form $\phi : \alpha_1 \wedge \cdots \wedge \alpha_n$ (assuming a single atom per dimension) is the expression $\phi^+ : \alpha_1^+ \times \cdots \times \alpha_n^+$. In other words, we compute the Cartesian product of the values of the involved atom signatures.

- The signature of a query q, q^+ is the *set* of coordinates computed as follows: (a) compute the signature, i.e., the set of coordinates pertaining to ϕ^0, the detailed equivalent of its selection condition; (b) within each of these coordinates, replace the (detailed) value of each dimension by its ancestor value at the level of the schema of the query. This guarantees that the resulting coordinates will be the coordinates of the query result.

The *detailed signatures* of the above categories are produced by replacing the respective values of their regular signatures with the expression $desc_L^{L^0}(\cdot)$, computing the respective set of descendant values and taking their union. The *detailed signature of a query* is (simply) the set of coordinates that pertain to the signature of ϕ^0.

The *detailed proxies* of expressions are the respective expressions transformed at the most detailed level for each of the involved dimensions. The *detailed proxy of a query*

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

is the query (i.e., an expression again)

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

Detailed areas are sets of cells, pertaining to an aggregate cell, or set of cells, like, e.g., the result of a query.

The *detailed area* of a cell $c :< v_1, \ldots, v_n >$ is the set of descendant cells that can be obtained by replacing each of its coordinates, say v_i, by $desc_L^{L^0}(v_i)$ and taking the Cartesian product of each such value set.

The *detailed area* of the query q is the set of cells of the result q^0, $q^0.cells$.

History. A *session S* is a list of cube queries $S = \{q_1, ..., q_n\}$ that have been recorded. We assume the knowledge of the syntactic definition of the queries, and possibly, but not obligatorily, their result cells.

A *session history of a user* is a list of sessions, resulting in a list of queries, following the order of their sessions.

The *cell history*, or simply, history, of a session history is the set of cells that belong to the queries of the session history. The history of detailed equivalents is the set of detailed equivalents of the cells of the query history.

3.2. Reference Example

In this example, we work with the loan cube from the PKDD 1999 Discovery Challenge[8]. The cube has anonymized data from Czech banks that concern loans that have been granted to customer. The dimensions of the data cube concern (a) the customer *Accounts*, with a hierarchy of levels: *Account* $\preceq$ *District* $\preceq$ *Region* $\preceq$ *ALL*, (b) the *Status* of a loan, with levels *Status* $\preceq$ *ALL*, and, (c) *Date* with a hierarchy $Day \preceq Month \preceq Year \preceq ALL$. For simplicity, we use a single measure *Amt*, referring to the amount of the loan that was granted.

Assume now that at a certain time point, four queries have been issued already, and a new one, to which we refer as q, is also submitted to the system. The desideratum is to compute the interestingness aspects of the query. Coming back to our opening remarks in the Introduction, this can occur due to several possible reasons. In a clear a-priori case, the new query q is generated by the system, and is candidate to be recommended to the user for execution. Before executing it however, and thus without any knowledge of what is included in the result, the recommender system needs to predict what it will contribute to the user's understanding on the data space. To this end, we need *syntactic* metrics and algorithms, that take only the query expression into consideration, to predict interestingness. In an a-posteriori case, the result of q has already been computed. Thus, we can use *extensional* algorithms and metrics that exploit this result and compare it to the cached results of the previous queries in order to compute its interestingness.

In Figure 1, we visually present the general setup of the problem. We assume a basic cube defined at the most primitive levels of detail: C^0: $[Account.Account, Status.Status, Date.Day, Amt]$. Then, for ease of diagrammatic depiction, we have all 5 queries of the figure defined at the schema $[Account.District, Date.Month, AvgAmt]$, practically expressed via the following formula: q_i $= <$ C^0, ϕ_i, [*Account.District*, *Status.ALL*, *Date.Month*, *AvgAmt*], [*avg*(*Amt*)] $>$, with each ϕ_i having a different expression, as depicted in the figure.

The center of Figure 1 depicts a 2D projection of the space of the basic cube C^0 along the 2 dimensions of the query schemata, *Account* and *Date* (we omit *Status* to simplify the figure). Each of the queries has (a) a detailed proxy $q_i^0 = < C^0$, ϕ_i^0, [*Account.Account*, *Status.Status*, *Date.Day*, *AvgAmt*], [*avg*(*Amt*)] $>$ and (b) a detailed area of cells, depicted as a band in the 2D projection of the multidimensional space. Some detailed areas are completely contained inside others: for example

[8] The example comes from the Discovery Challenge of PKDD 1999 `https://sorry.vse.cz/~berka/challenge/pkdd1999/berka.htm` and now can be found at `https://github.com/sabirakhtar/PKDD-99-Discovery-Challenge`

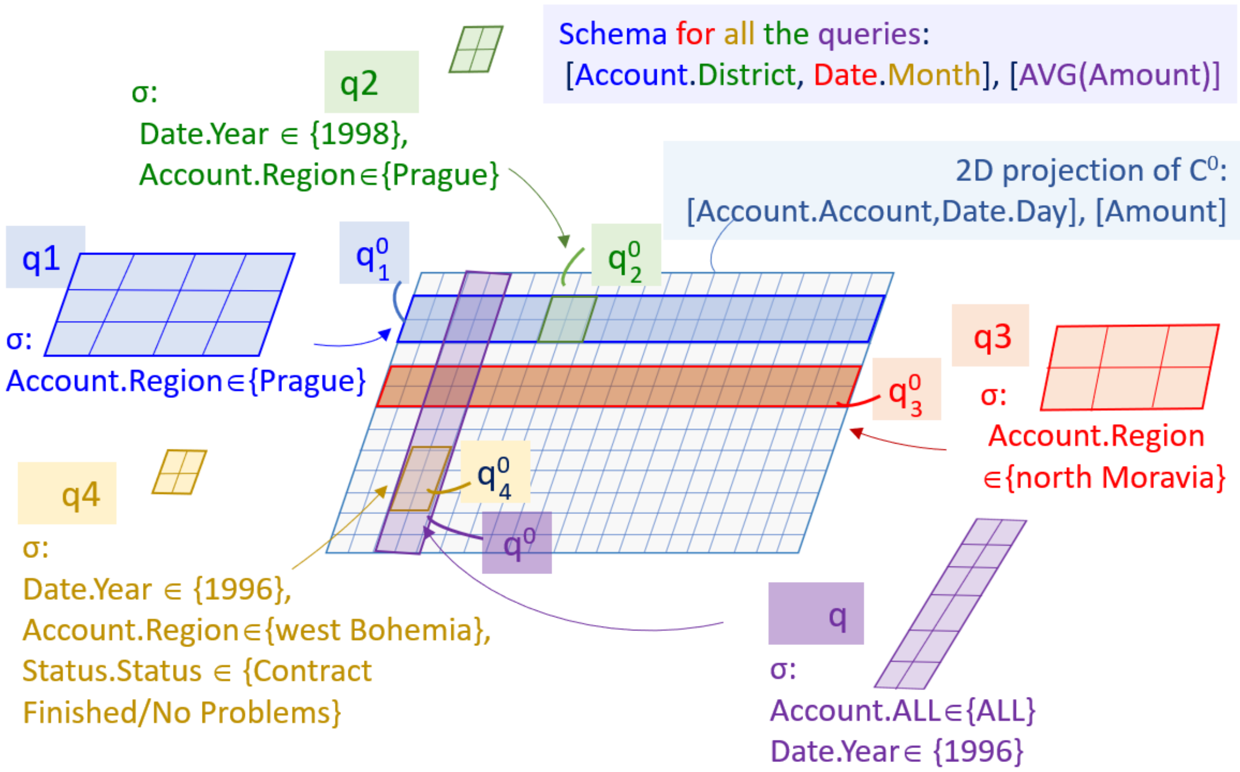

Figure 1: The setup of our reference example. For the non-colorblind readers: each query surrounding the central detailed cube comes with its own color of cell shading and font. Also, each query is accompanied with a similarly-colored rectangle over the central detailed cube that identifies its respective detailed area. For both colorblind, and non-colorblind readers: arrows depict the relationship of query definitions to the respective detailed areas.

q_2^0 is completely contained within q_1^0. The detailed area of q, q^0, has all sorts of relationships with the detailed areas of the other queries: (a) q_4^0 is completely contained within q^0, (b) q^0 has a partial overlap with q_3^0 and q_1^0, (c) there is no relationship between q^0 and q_2^0, whatsoever. There are no identical queries, either.

4. Taxonomies for the assessment of cube query interestingness

In this paper, we propose facets of interestingness and algorithms to assess them. Specifically, we define the interestingness of a cube query q, $I(q)$ as a vector of scores along four fundamental interestingness facets, which we typically refer to as *facets* – specifically, (i) novelty, (ii) relevance, (iii) surprise, and (iv) peculiarity:

$$I(q) = \langle novelty(q), relevance(q), surprise(q), peculiarity(q)\rangle$$

Before proceeding to the individual interestingness facets and the respective algorithms, we provide taxonomies that allow to mentally structure the problem and organize the algorithms accordingly. The fundamental reasons for doing so are (a) a better structuring of the problem of evaluating query interestingness in terms of the context under which it operates, (b) the possibility of placing current and future algorithms of the literature in the framework provided by our taxonomies and (c) equally, if not more importantly, the possibility of automating the process of automating the selection of algorithms for interestingness evaluation. The taxonomies that we introduce concern:

- The structuring of the *output space* practically helps with understanding what is to be expected from an algorithm as a result. To this end, in subsection 4.2 we will introduce *a taxonomy of types of returned scores.*
- The structuring of the space of the *internal workings* of each algorithm concerns modali-

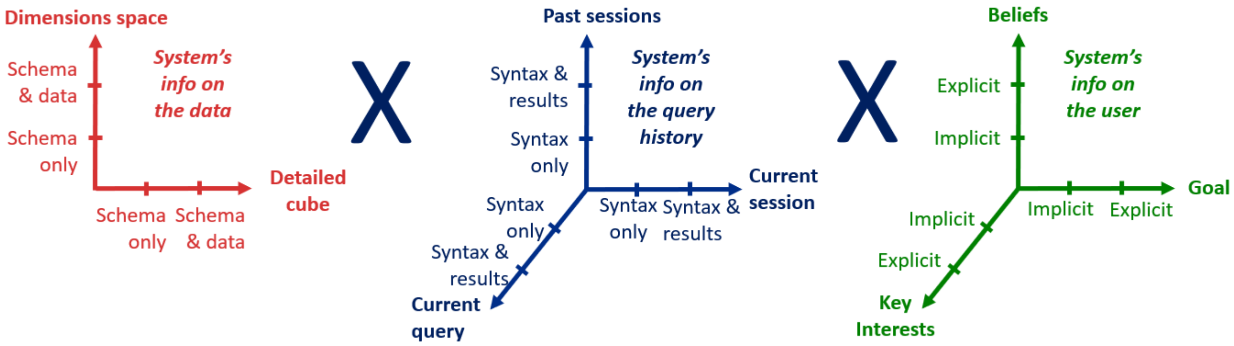

Figure 2: The 8 taxonomic dimensions that determine the different levels of information that a system must have in order to assess the interestingness of a cube.

ties that also affect the selection of algorithms for alternative contexts – in this case, not with respect to the input, but rather with respect to the inner semantics of the algorithms. Two examples can illuminate this: (a) algorithms can take advantage of the multidimensional hierarchies or not; (b) algorithms can also use only the query syntax, or the query results too. Algorithms that are not aware of hierarchies are applicable for flat relations too. Algorithms that require query results cannot operate as forecasters of candidate query interestingness. Subsection 4.3 clarifies the distinctions.

- The structuring of the space of *the input parameters* that affect/are required by an algorithm, is, however of the most vital importance. The combination of the available information creates a context for algorithm selection and execution. Depending on what information is available as input, a query interestingness assessment system can automatically decide algorithms that are applicable for the particular context under consideration. We begin our deliberations from this last taxonomy.

4.1. Taxonomy of the input information needed to assess interestingness

We identify 8 fundamental *taxonomic dimensions*[9] of necessary (equiv.: potentially available) information for our *cube query interestingness scoring system* to address the goal of automatically computing a score of interestingness for a given cube query. These 8 taxonomic dimensions are further organized in 3 major families.

For all families and taxonomic dimensions, an implicit value of our knowledge is the **no-knowledge** *value.* To avoid repetition, we will not refer to this level of knowledge again, although it is quite possible that several dimensions will be of no-knowledge value in a scoring system in practical situations.

The Data Family. The Data Family of taxonomic dimensions is concerned with what kind of information about the underlying data is available to our scoring system. Specifically, we identify two taxonomic dimensions of interest, concerning the hierarchies of the multidimensional space and the factual cubes that are available.

- *Dimension space*. The dimension space characterizes our level of knowledge/information on the dimensions of the multidimensional space within which the queries are going to be configured and posed. We identify two potential levels of knowledge (a) knowledge only of the schema (i.e, dimension & level names, hierarchical relationships) of the multidimensional space, or, (b) knowledge of both the schema and the values (aka dimension members in the OLAP literature) of the involved levels.

- *(Detailed) Cubes.* The detailed cube space characterizes our level of knowledge/information on the factual data of the multidimensional space – i.e., the detailed cubes over which the queries are going to be posed. We identify two potential levels

[9] Terminology-wise: we discriminate between plain OLAP *dimensions* for the structuring of data, and *taxonomic* or *problem dimensions* that characterize the different characteristics of the problem.

of knowledge (a) knowledge only of the schema (i.e, dimensions & levels) of the cubes' schemata, or, (b) knowledge of both the schema and the values i.e., cube cells of the involved cubes.

The Query History Family. The Query History Family of taxonomic dimensions is concerned with what kind of information about the queries being and having been issued by the user is available to our scoring system. Specifically, we identify three taxonomic dimensions of interest, concerning (a) the knowledge of the current query being posed, (b) the knowledge of the user's current session, and, (c) the knowledge about the overall history of queries of the user.

- *Current Query.* The current query being posed to the query answering system (e.g., an OLAP server) can be known by our scoring system at two levels of information: (a) syntax only, where only the query specification is available (e.g., before the query having been answered, or in order to save space or speed up computations without using the result cells), or, (b) both query specification and results are known to the scoring system.
- *Current Session.* Similarly to the current query, the current session comprises a list of queries that are known to the scoring system. Like the case of the current query we may either know only the syntax of the queries, or both the syntax and the cells of the session queries' results.
- *Past Sessions.* Similarly to the current session dimension, the "past sessions" taxonomic dimension generalizes it to include previous sessions of the user (or other users, similar to the one being assessed) too. Again, we may either know only the syntax of the queries, or both the syntax and the cells of the queries' results.

The User Profile Family. The User Profile Family of taxonomic dimensions is concerned with what kind of information about the user is available to our scoring system. Specifically, we identify three taxonomic dimensions of interest, concerning (a) the knowledge of the user Key Interests, (b) the knowledge of the user's Beliefs about the data values, and, (c) the knowledge about the current user Goals that are available to our scoring system.

- *Key User Interests.* The user's recurring interests – as close to a user profile as we can get– comprise the context for this taxonomic dimension. The Key Interests can be considered a static aspect of the user profile and we will assume they take the form of a set of Key Performance Indicators(KPIs)[51], which practically comprise a query and a labeling schema for the results of a query on the basis of expected values for them. So practically, every cell of a KPI query result is mapped to a finite set of values (e.g., bad/med/good, or a Likert scale of stars) on the basis of rules that compare it to an expected value and assign a performance score on the basis of the discrepancy of the actual vs the expected value of the cell. We discriminate two levels of knowledge the system can have on the KPI's of the user: (a) implicit, i.e., this kind of information is not explicitly specified by the user, but approximated and estimated by other information available to the system like the history of past queries, that somehow mark a range of preferences on what interests the user on a regular basis, or, (b) explicit, directly stated by the user (e.g., in this case an explicit specification of KPI's).
- *User Beliefs.* The beliefs about the data that the user has, are captured by this taxonomic dimension. In other words, assuming a query is posed, the beliefs of the user is the set of expected values for the query cells that the user expects to see. These can be (a) implicitly estimated, e.g., derived from the history of past queries by extrapolating values on the basis of similar values the user has seen in the past, or identified by some relevant KPI's carrying expected values for certain aggregate cells, or, (b) explicitly known, e.g., extracted from the history of past queries, in case a certain cell has been presented to the user in the (recent) past.
- *User Goals.* The user goals are the current information goals that the user has towards fulfilling an information need, or exploring the data space and discovering new information. The Goals are dynamic characteristics of the user, temporally local and transient (i.e., they concern an information need of the current time) and they are related closely to the intentions of the Intentional model [52] like describe, explain, analyze, etc. We discriminate

two levels of knowledge the system can have on the goals of the user: (a) implicit, i.e., this kind of information is not explicitly specified by the user, but approximated and estimated by other information available to the system, or, (b) explicit, directly stated by the user (e.g., in case he is firing intentional queries).

In the sequel, we will give explicit formulations of how these constructs can be expressed and represented internally in the system. Specifically, a belief is a probability for the value a cell's measure can take; an information goal is a specification of the subset of the multidimensional space that is of interest to the user; a KPI is a set of registered queries. We stick to fundamental core concepts to allow our framework to easily incorporate alternative ways to express these constructs in the future. We specifically define beliefs in Section 5.2, information goals in Section 6 and KPIs in Section 6.2.

How are the aspects of interestingness related to the taxonomic dimensions of the problem? Concerning the relationship of the aspects of interestingness with the taxonomic dimensions of available knowledge, we can make a few, first coarse observations:

- Peculiarity is related to the history of past queries and their results. A query can be peculiar if (a) it does not fit nicely in the set of previous queries in terms of its syntax (and thus, of the area of the multidimensional space that it covers), or (b) if its results show values quite different than the values one had seen in previous, similar queries.

- Surprise is related to the beliefs the user already has. There beliefs can be anywhere in the range of (a) concrete values of past query cells, all the way to (b) some probability distribution on the expected measure values (or labels) a given cell can have.

- Novelty is affected by the presence of the history of past queries.

- Relevance is an aspect related to the static profile (KPI's, preferences, interests) as well as to the dynamic profile (current goal) of the user. The static profile refers to what the user is typically interested in, and is an approximation of the user needs on a recurring basis, whereas the current goal is a more to the point description of the specific info need of the user at this moment in time.

In the rest of our deliberations, unless explicitly stated otherwise, the system works under the following assumptions:

- the dimensions' schema and data are both known;

- the cubes' schema is known (but not necessarily the data);

- the syntax of the current query is known (but not necessarily its results);

- the two taxonomic dimensions of the past are unioned into a single taxonomic dimension, history; no assumptions can be made for its knowledge by default;

- similarly, no assumptions are by default made for the three taxonomic dimensions of the system's information on the user.

4.2. Taxonomy of algorithms with respect to the type of output they produce

Apart from the aforementioned taxonomy used to characterize the type of information needed to be able to assess interestingness (pretty much amounting to the type of *input* information the assessment algorithms need), we can also discriminate algorithms with respect to the returned value they compute (practically, the *output* of the algorithm). We use the following terminology:

1. *Decision* vs *Enumeration* Problem: the decision problems answer a Boolean check (e.g., whether a cube is novel or not), whereas the enumeration problems report which subsets of cells are part of a solution (e.g., which part of a new cube is already covered, or novel). In all our subsequent deliberations, unless explicitly specified otherwise, we work on the enumeration problem.
2. *Full* vs *partial* Assessment: full assessment means that the checks made return a true/false answer on whether a new query is interesting or not; partial assessment means that the checks return an interestingness score (in fact: a score for a particular interestingness facet) as a real number (typically in the interval [0 . . . 1]). Naturally, a partial assessment that returns 1, also implies full interestingness.

4.3. Internal taxonomy of algorithms

Finally, the algorithms can be discriminated with respect to the way they perform the checks and potential constraints the algorithms might have (practically, characterizing the *how* of the algorithm).

1. *Syntactic* vs *Extensional* Assessment: syntactic assessment is based only on query definitions, whereas extensional also assumes the presence of the cells of the query result(s).
2. *Same-Level vs Detailed* Assessment: same-level assessment assumes that two cubes are at the same level of aggregation; detailed assessment means that the comparison of two cubes will be done at levels lower than their definition – typically, we will use the most detailed level as the common ground upon which the constituting detailed cells for two cubes can be compared.

Same-level algorithms are practically unaware of the hierarchical nature of the multidimensional space. Thus, they are also directly applicable to simple relational settings. At the same time, the hierarchically-aware algorithms –e.g., the detailed assessment algorithms discussed in this paper– can take advantage of the level hierarchy and potentially compare queries expressed at different levels, that would otherwise be incommensurable.

Syntactic vs extensional variants are characterized by two different aspects: performance and applicability in different settings. Syntactic-level algorithms taking into consideration only the syntax of a query are useful in both a priori and a posteriori evaluations of interestingness. A priori evaluation can be useful when a query is to be recommended for execution to the analyst. In this case, the system knows nothing of the result – yet, due to the query syntax it can say something on the relevance, novelty, or other facets of the query. A posteriori evaluation means that the query results have been computed, and a cell-to-cell –extensional– comparison can be performed. Clearly, the latter is much more informed than the syntactic one; on the other hand, it is clearly slower and restricted to a posteriori cases only.

4.4. Final note on the taxonomies

As a final note, we would like to highlight that the three taxonomies (input, output and internal) are orthogonal to each other. Each has its benefits and its constraints (more vs less requirements for input, more or less semantic richness in the output, more or less adaptability to different situations depending on the internal workings) – however, the three taxonomies are orthogonal to each other. We will see plenty of such combinations in the subsequent sections; to make things clear, each Section comes with a table where all the algorithms are characterized with respect to all the taxonomies.

Symbol	Explanation
D	Dimension
L	Level
q	Query
$q.cells$	Query result for query q
q^+	Signature of a query = coordinates of the query results
q^0	The equivalent query of q at the most detailed level
c	A cell. If c abides by the schema $[L_1, \ldots, L_n, M]$, it is a cell of values $c = < v_1, \ldots, v_n, m >$
c^+	Cell coordinates (dim. values), $c^+ = < v_1, \ldots, v_n >$
c^0	A cell at the most detailed level, for all its coordinates
Q	Set of queries, $Q = \{q_1, \ldots, q_n\}$
Q^0	Set of the detailed queries of the queries of Q, $Q^0 = \{q_1^0, \ldots, q_n^0\}$
q^{cov}	set of cells that are covered (found also in previous queries)
q^{nov}	set of cells that are novel (not found in previous queries)

Table 1: Table of notation

5. Novelty

Novelty assesses the amount of previously unknown information delivered to the user via a query. Due to this inherent characteristic, we need to either explicitly know, or at least estimate the prior knowledge of the data that the user has.

Naturally, a system is not in a position to actually have knowledge of the user's memory or knowledge. Knowledge can come to the user via external channels, not related to the query answering and thus, the system necessarily has "knowledge" of just a subset of the user's actual knowledge. At the same time, one should also take account of the effects of time that erases, hides or distorts the remembrance of facts encountered in the past. Although in our following deliberations we will not directly address the above problems, we will occasionally offer insights on how to handle some of them. However, when we use the term "knowledge" we simplify and approximate the situation, by assuming that the system knows what the user has seen, or what the user has explicitly stated that she believes.

Explicit knowledge is primarily attained by knowing the history of user queries (and assuming that the user remembers it). A second way to approximate what the user remembers is to exploit the registered beliefs of the user that have a low level of confidence, by making the rational assumption that since she has expressed practically uncertain beliefs about some cells, she does not know their values.

Novelty is mostly goal-independent, i.e., it is not affected by neither the current (goal) or the typical (key interests) informational needs of the user.

Overall, in terms of our taxonomic dimensions, novelty is mostly related (a) to *history*, and, (b) to *registered values for beliefs with confidence below a certain threshold*. We will examine the different alternatives in the respective subsections. To facilitate comprehension, in Figure 3 we summarize the proposed metrics in terms of our taxonomy, and, in Section 5.3 we exemplify the discussed metrics and algorithms.

5.1. Novelty assessment in the presence of a query history

First, we will assess the novelty of a cube query q assuming a query history $Q = \{q_1, \ldots, q_n\}$ exists.

5.1.1. Same-Level Assessment of Novelty

Assume that we only check q against members of Q whose schema is at the same level with Q. We also require the same detailed measures and aggregate functions to be used, otherwise the comparison is referring to essentially different measures, and also different numbers, and, therefore, novelty is guaranteed.

Full Same-Level Syntactic Assessment of Novelty. In this case, the question to be answered is: Given q and $Q = \{q_1, \ldots, q_n\}$, is there any $q_i \in Q$ such that $q = q_i$?

In this case, the solution is a trivial *syntactic* check: we iterate through the syntactic definitions of the queries of Q and check whether there is any query that is identical to q. Then, *Full Same-Level Syntactic Novelty* (FSLSN) is defined as a Boolean flag, returning false if $q \in Q$ and true otherwise The check is full, syntactic and same-level.

Partial Same-Level Syntactic Assessment of Novelty. In this case, the question to be answered is: Given q and $Q = \{q_1, \ldots, q_n\}$, can we identify which part of the results of q is already covered by the queries of Q without actually computing them?

The answer to the question is given by Algorithm ComputePartialSameLevelCubeCoverage in [49] that takes q and Q as inputs and divides the coordinates of the result of q in two sets: a set of cell coordinates that are covered by existing queries, q^{cov} and its complement, q^{nov}, a set of cell coordinates that are novel. Then, *Partial Same-Level Syntactic Novelty* (PSLSN) is the fraction of novel cells of the total population of cells of q (which is also the union of q^{nov} and q^{cov}). At the syntactic level, we only need the coordinates (signatures) of the cells, without having to compute their measures. The check is partial, syntactic and same-level.

$$PSLSN = \frac{|q^{nov^+}|}{|q^{nov^+}| \bigcup |q^{cov^+}|}$$

Partial Extensional Same-Level Assessment of Novelty. In this case, the question to be answered is the same: Given q and $Q = \{q_1, \ldots, q_n\}$, can we identify which part of the results of q are already covered by the queries of Q? However, in this case, we assume that the results of the queries are available and the check takes this into consideration.

The premise to the question is given by Theorem Same-Level-Intersection in [49] that takes two queries q_1 and q_2 with the same schema, and decides whether their selection conditions make them eligible for a check on their intersection. Then, Algorithm EnumerateSameCellsviaResultComparison in [49] returns the cells that are covered and the cells that are not.

A simple adaptation of the Algorithm ComputePartialSameLevelCubeCoverage in [49] that works with signatures, to work with cells produces the novelty of the new query q. The formula for *Partial Same-Level Extensional Cube Novelty* is the same with the one of *Partial Same-Level Syntactic*

		{Syntactic vs Extens.}x {SameLevel vs Detailed} x{Full vs Partial}	**Dimensions** *Schema only vs Schema & data*	**Basic Cube** *Schema only vs Schema & data*	**Current Query** *Syntax only vs Syntax & results*	**History** *Syntax only vs Syntax & results*	**Key Interests** *Implicit vs Explicit*	**Current Goal** *Implicit Vs Explicit*	**Beliefs** *Implicit vs Explicit*
Novelty									
5.1.1	FullSameLevelSyntacticNovelty	Full SameLevel Syntactic	---	---	Syntax only	Syntax only	---	---	---
	PartialSameLevelSyntacticNovelty	Partial SameLevel Syntactic	Schema & data	Schema only	Syntax only	Syntax only	---	---	---
	PartialSameLevelExtensionalNovelty	Partial SameLevel Extensional	Schema & data	Schema & data	Syntax &results	Syntax &results	---	---	---
5.1.2	FullDetailedSyntacticNovelty	Full Detailed Syntactic	Schema & data	Schema only	Syntax only	Syntax only	---	---	---
	PartialDetailedSyntacticNovelty	Partial Detailed Syntactic	Schema & data	Schema only	Syntax only	Syntax only	---	---	---
	PartialDetailedExtensionalNovelty	Partial Detailed Extensional	Schema & data	Schema & data	Syntax &results	Syntax &results	---	---	---
	Weighted v. of PartialDetailedExtensionalNovelty	Partial Detailed Extensional	Schema & data	Schema & data	Syntax &results	Syntax &results	---	---	---
5.2	PartialExtensionalSameLevelBeliefBasedNovelty	Partial SameLevel Extensional	Schema & data	Schema only	Syntax &results	---	---	---	Explicit
	PartialExtensionalDetailedBeliefBasedNovelty	Partial Detailed Extensional	Schema & data	Schema & data	Syntax &results	---	---	---	Explicit
	PartialExtensionalArbitraryBeliefBasedNovelty	Partial Arbitrary Extensional	Schema & data	Schema & data	Syntax &results	---	---	---	Explicit

Figure 3: List of Novelty algorithms, characterized with respect to the reference taxonomy

Cube Novelty and the difference is only in efficiency (which of the two variants is faster is open to experimental evaluation). The check is partial, extensional and same-level.

5.1.2. Detailed Assessment of Novelty

Assume now that instead of checking cubes defined at the same level, we compare cubes with respect to their constituting cells at the most detailed level.

Full Syntactic Detailed Assessment of Novelty. In this case, the question to be answered is: Given q and $Q = \{q_1, \ldots, q_n\}$, is there any $q_i \in Q$ such that $q_i^{0^+}$, the *detailed signature of* q_i (i.e., the coordinates of the most detailed cells over which q_i is computed), is a superset of q^{0^+}, *the detailed signature of* q?

The premise to the question is provided by the Theorem on Foundational Containment in [49] stating when a certain query q_i can foundationally contain a new query q (denoted $q_i \supseteq^0 q$). A simple iteration over the contents of the query set Q can reveal whether such a query exists or not. *Full Syntactic Detailed Novelty* (FSDN) determines whether a query q is novel with respect to a previous query $q_i \in Q$.

$$FSDN = \begin{cases} 0 & \text{if } \exists\ q_i \in Q \text{ s.t., } q_i \supseteq^0 q \\ 1 & \text{otherwise} \end{cases}$$

The check is full, syntactic and detailed.

Partial Detailed Extensional Assessment of Novelty. In this case, the question to be answered is: Given q and $Q = \{q_1, \ldots, q_n\}$, can we identify which part of the results of the detailed area of q^0 are already covered by the detailed areas of the queries of Q?

Algorithm 1 computes the union of the detailed areas of the queries in the query list and intersects it with the detailed area of the query under question. We remark that only the queries in the history concerning the same measures and aggregation functions than q are passed to the algorithm. The resulting *Partial Detailed Extensional Novelty* (PDEN) is the fraction of the detailed not covered (i.e., novel) cells over the entire detailed area of q.

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, all with the same aggregate functions over the same detailed measures

Output: q^{cov}: the subset of the cells of q^0 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}

Variables: Q^0 a collection of all cells belonging to q_i^0, i.e., the detailed equivalents of the queries q_i

1 **begin**
2   produce $q^0.cells$
3   produce $q_i^0.cells$ for all q_i
4   $Q^0 \leftarrow \bigcup_i q_i^0.cells$
5   $q^{cov^0} \leftarrow \emptyset$
6   $q^{nov^0} \leftarrow q^0.cells$
7   **forall** $c^0 \in q^0.cells$ **do**
8   **if** $c^0 \in Q^0$ **then**
9       $q^{nov^0} = q^{nov^0}$ - c^0
10       $q^{cov^0} = q^{cov^0} \cup c^0$
11   **end**
12   **end**
13   **return** q^{cov^0}, q^{nov^0}
14 **end**

$$PDEN = \frac{|q^{nov^0}|}{|q^{nov^0}| \bigcup |q^{cov^0}|}$$

The check is (a) partial (practically a normalized score), (b) extensional (via cells), and, (c) detailed, i.e., with respect to the detailed levels of the involved cubes.

The complexity of Algorithm 1 is mainly determined by the cost of answering the detailed queries in Lines 2 and 3 that produce the cells for q^0 and q_i^0 for both the input query and the query history. The complexity of these actions is: (a) linear with respect to the size of the query history, and, (b) linear with respect to the cube size, assuming that the cube query is linear with respect to the cube size. The rest of the algorithm, requires a linear in-memory pass of the result to populate Q^0 and a linear lookup for each cell of $q^0.cells$ to cross-check if it belongs to Q^0. Again, this cost is linear, yet, we consider it insignificant comparing it to the time needed for query answering. Therefore, the overall cost of the algorithm is linear with respect to the size of the query history and to the cube size.

Partial Detailed Syntactic Assessment of Novelty. In this case, the question to be answered is practically the same, albeit with a different means to compute the answer, specifically, signatures instead of cells: Given q and $Q = \{q_1, \ldots, q_n\}$, can we identify which part of the results of the detailed area of q^0 are already covered by the detailed areas of the queries of Q, by comparing solely the signatures of the queries? The algorithm is similar to Algorithm 1 , but uses q^{0^+} and $q_i^{0^+}$ for all q_i instead of the respective detailed cells. The resulting *Partial Detailed Syntactic Novelty* (PDSN) is the fraction of the detailed not covered (i.e., novel) cells over the entire detailed area of q.

$$PDSN = \frac{|q^{nov^{0^+}}|}{|q^{nov^{0^+}}| \bigcup |q^{cov^{0^+}}|}$$

The check is partial, syntactic and detailed.

Remark. Observe that, since the check is done at the most detailed level, the only thing we care about is that the measures and aggregate functions are the same. Selection conditions can be arbitrary. The same applies for the grouper levels: to the extent that we assess novelty with respect to the detailed cells, the grouping levels of the compared queries can be arbitrary.

Remark. It is easy to introduce a **weighted variation** of the above algorithm. Observe that the Algorithm 1 computes the union of the detailed areas of the queries with set semantics. We can produce a weighted variant if we introduce the following variations to the algorithm:

- Each cell is accompanied by a counter of its occurrences; so, every time we perform the union of Q^0 with the next $q_i^0.result$, for every detailed cell that is already part of Q^0, we increase its counter by one. Let us denote the number of occurrences of each cell c with $c.weight$.
- Given a set of cells, C, we can compute its total weight, $C.weight$, as the sum the weights of its constituent cells.
- Then, $WeightedDetailedNovelty$ (WDN) is the total weight of q^{nov^0} over the sum of the total weights of q^{nov^0} and q^{cov^0}.

$$WDN = \frac{q^{nov^0}.weight}{q^{nov^0}.weight + q^{cov^0}.weight}$$

This way, cells that are more frequently encountered count more (thus, increasing the denominator and reducing the total novelty of the new query, if it includes such cells in its result). In case q^{cov^0} is empty, novelty takes the value of 1.

Remark. Variations of the above formula on the total weight can also be devised, to normalize the weights of the cells. Also, the same theme can be applied to (a) signatures and (b) same-level checks, too.

5.2. Novelty assessment in the presence of belief statements

Assume we do not have explicit knowledge of the user history, or key interests, but we do have an estimation of probabilities for the likely values of some cells in the multidimensional space.

A belief is the existence of a probability for the possible values that a measure value of a specific cell can take.

Although beliefs can be expressed in multiple ways, the exact expression formalism is practically orthogonal to the essence of interestingness estimation. In any case, here we provide a concrete syntactic system for expressing beliefs. Assume that for certain cells, it has been possible to either deduce or explicitly have the user register probabilities per expected value for the value $m = c.M$, of a cell c and a certain measure M. So, some cells in the multidimensional space are annotated with a set of *cell expected-value statements*, which are statements of the form

$$p(M \in [l_i \ldots u_i]|c) = p_i, p_i \in [0..1], [l_i \ldots u_i] \text{ is a range of values of } dom(M)$$

or of the form

$$p(M \in \mathbf{s_i}|c) = p_i, p_i \in [0..1], \mathbf{s_i} \text{ is a discrete finite set of values of } dom(M)$$

For uniformity of notation, to express beliefs, we will use the syntactic form

$$p(M \in m_i|c) = p_i$$

to denote either a range or a finite set of values for the value-set of the expressed belief. The distinction makes no difference for the evaluation of novelty.

We refer to the set of statements of the above form for a cell c, the *probable active domain* of c, or $dom^{pa}(c)$. A *well-formed probable active domain* of a cell has the property that all its statements' probabilities sum up to 1. However, requiring well-formed probable active domains is too restrictive, in the sense that maybe some probabilities are unknown, or hard to evaluate; thus, we do not require it as a necessary property for the sequel.

We call a cell c to be $\Pi - known$ if, within the statements of the probable active domain of c, there exists a probability p_i which is equal or higher to a threshold Π. Otherwise, if all the probabilities of c are below Π the cell is called $\Pi - unknown$.

The intuition behind this treatment lies on the observation that if a user has a set of beliefs about the behavior of a cell, with a high amount of certainty (i.e., the probability is above a certain threshold), then we cannot consider the cell to be "unknown" to the user. The result of a query might be surprising, if it is far from the expected value, but the existence of this area of the multidimensional space is not novel to the user.

To give a practical example, assume the following user beliefs

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

assuming all other dimensions set to ALL. For a particular cell therefore, concerning the sales in Athens for 2020, we have a probability distribution for the range of its values. Let's also assume that we have agreed that if a user has a belief higher or equal to 50% for a cell's measure, then she "knows" the cell; this means setting a value of $\Pi = 50\%$. Given the above belief set, and the existence of a belief with probability 70% (i.e., higher than Π), we can say that this particular cell is indeed 50%-known, and thus consider it not novel.

Let B a set of beliefs expressed as cell expected-value statements for a set of cells C^B. Assume now a query q, and its resulting cells $C = q.cells = \{c_1, \ldots, c_n\}$. Assume also a threshold Π. Then, the $\Pi -$ *direct novelty* of q is the percentage of cells of $q.cells$ that are $\Pi - unknown$. We can distinguish three cases for computing belief-based novelty, depending on the level that the cells of C^B have been defined.

5.2.1. Same-Level Belief-Based Novelty

In this case, the set of beliefs B is expressed over a set of cells C^B at the same aggregation level as q. Therefore, we can immediately compare the cells of the query to the cells of the belief-set. Algorithm 2 performs the computation of novelty.

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

Input: A query q and its result; a set of beliefs B over a set of cells C^B at the same aggregation level as q; a threshold Π for deciding if a cell is eligible for being novel

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

Variables: $C^\star$ is a set of cells, $C^\star \subseteq C^B$ for which there exists a known belief with a probability above or equal to Π, i.e., $\{c \mid c \in C^B, \exists\, p(M \in m|c) \in B, p(M \in m|c) \geq \Pi\}$

begin
  $q^{cov} \leftarrow \emptyset$
  $q^{nov} \leftarrow q.cells$
  compute $C^\star$
  forall $c \in q.cells$ **do**
    if $c^+ \ \in \ {C^\star}^+$ **then**
      $q^{nov} = q^{nov}$ - c; $q^{cov} = q^{cov} \cup c$
    end
  end
  return q^{cov}, q^{nov}
end

The algorithm, starts by assuming that all cells are novel and none is covered. The first action of the algorithm is to isolate the $\Pi - known$ cells, based on the input set of beliefs, into a set $C^\star$. Then, for each cell of the query, it checks whether its signature fits with the signature of any of the cells that belong to $C^\star$, and if it does, then it considers the cell to known, adds it to the set of covered cells and removes it from the set of novel cells. The reason for using signatures here, is that the beliefs are expressed with respect to signatures and probabilities for the value range of the measure. Thus, the cell's measure should not be used for assessing the presence of the cell in $C^\star$ (i.e., checking for identity/equality of the measure is not within the spirit of using the beliefs in the first place).

Then, we can compute the *Partial Extensional Same-Level Belief-Based Novelty* (PESLBBN) of the query q as usual:

$$PESLBBN = \frac{|q^{nov}|}{|q^{nov}| \bigcup |q^{cov}|}$$

The check is (a) partial (practically a normalized score), (b) extensional (via cells), and, (c) same-level, i.e., with respect to the actual cells of the involved query. The *Syntactic* version of the algorithm (as contrasted to the *Extensional* one) is quite similar, albeit with the difference that no cells in the query result are needed and all sets and comparisons are performed with respect to the signatures of the queries. The complexity is linear to the result size (assuming the set C^B is fixed) and linear to the size of the set C^B assuming the query result size is fixed. In the case that only the query expression is given as input to the algorithm, and the query result has to be computed, the cost is dominated by the computation of $q.cells$, which is linear to the data cube size.

5.2.2. Detailed Belief-Based Novelty

Another (rather extreme) case, assumes that the set of beliefs B is expressed over a set of cells C^B at the most detailed aggregation level. Then, we can compare the cells of q with the cells of C^B by converting them to their detailed equivalents. The algorithm used, is similar to Algorithm 2, but used $q^0.cells$, q^{cov^0} and q^{nov^0}, instead of $q.cells$, q^{cov} and q^{nov}, respectively. Then, we can compute the *Partial Detailed Extensional Belief-Based Novelty* (PDEBBN) of the query q as usual:

$$PDEBBN = \frac{|q^{nov^0}|}{|q^{nov^0}| \bigcup |q^{cov^0}|}$$

The check is (a) partial (practically a normalized score), (b) extensional (via cells), and, (c) detailed, i.e., with respect to the most detailed cells of the data space. The complexity analysis, as well as the discussion of the *Syntactic* variant are homologous to the ones of subsection 5.2.1.

5.2.3. Arbitrary-Level Belief-Based Novelty

In this case, the cells of C^B are defined at arbitrary levels of aggregation. Thus, it is not straightforward to compute novelty. For the cells of C^B that are defined at higher levels of aggregation compared to the ones of q, even for a single dimension

Algorithm 3: Partial Extensional Arbitrary Belief-Based Enumeration Of Covered Cells

Input: A query q and its result; a set of beliefs B over a set of cells C^B at arbitrary aggregation levels; a threshold Π for deciding if a cell is eligible for being novel

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

Variables: $C^\star$ is a set of cells, $C^\star \subseteq C^B$ for which there exists a known belief with a probability above or equal to Π, and, all their levels are lower or equal to the respective ones of q, i.e., $\{c \mid c \in C^B, \exists\, p(M \in m|c) \in B, p(M \in m|c) \geq \Pi \wedge \forall\ dimension\ D,\ c.D.L^c \preceq q.D.L^q\}$

```
begin
    q^cov ← ∅
    q^nov ← q.cells
    compute C*
    forall c ∈ q.cells do
        if c is fully covered by cells of C* then
            q^nov = q^nov - c; q^cov = q^cov ∪ c
        end
    end
    return q^cov, q^nov
end
```

(i.e., even if there is a single dimension D for which the cell of C^B is at a higher level than the aggregation level of q), it is clear that we cannot use them for assessing novelty, as they express a coarser computation than the one of the query. Assume that we disqualify these cells and stick to the ones that have their levels at a lower or equal level with respect to the levels of q. Again, comparison is not straightforward; converting all to the detailed equivalents is not usable, as the knowledge of an aggregate value does not imply the knowledge of its detailed equivalents. Thus, we need to resort to even stricter measures.

Of course, Algorithm 3 requires the definition of *full coverage* of a higher-level cell by a set of more detailed cells.

A cell c, defined at a set of levels L^c, is *fully covered* by a set of cells C, all of which are defined at lower or equal levels that the ones of L^c if:

$c^{0^+} \subseteq \bigcup_i c_i^{\star^{0^+}}$, for all $c^\star \in C$

i.e., the cells of its detailed area are a subset of the detailed cells that correspond to the members of C. For all practical purposes, this means that one can compute c from the more detailed levels of C – thus, "knowing" it. The check requires a full scan of C for each cell of c and the determination of coverage.

Remark. Note that non-probabilistic statements can be treated as having a single cell-expected value statement per cell with probability 1. Also, other variants (e.g., weighted) are eligible.

5.3. Reference Example Revisited

Coming back to the reference example of Section 3.2, we can start with examples on the novelty of query q that is assessed over the presence of a query history $Q = \{q_1, \ldots, q_4\}$.

- Concerning the *Full Syntactic Same-Level Assessment of Novelty* (FSLSN), it takes the value of 1, as we can observe that no query in the query history has an identical definition with q.

- The same would apply for the *Partial Syntactic Same-Level Assessment of Novelty* (PSLSN). For the test to apply, we would require the existence of queries with compatible selection conditions to the ones of q in order for the resulting coordinates to be comparable. However, in contrast to the queries of Q, q has no selection filters, therefore, its syntactic novelty is also 1. Similarly for the extensional variant of the same metric.

- Concerning the *Full Syntactic Detailed Novelty* (FSDN), to the extent that there is no query that encompasses the entire q, the novelty is 1.

- Concerning the *Partial Detailed Extensional Novelty* (PDEN), if we run the algorithm, we need to (a) take the union of the detailed areas of the queries of Q, say Q^0, and (b) intersect it with the detailed area of the q, q^0. In practice, we detect that 70% of the cells of q^0 do not belong to Q^0, thus PDEN = 0.7.

Coming now to Belief-based queries, we will also use the query of the reference example, evaluated over a set of prior beliefs. For simplicity, let us assume that the system has registered a single belief statement for the analyst as follows:

$p(AvgLoanAmt \geq 25K \mid$ *district=Prague, year=1996*$)$ $= 70\%$

The query q has 107 tuples as a result and 117 tuples as a detailed result, at the most detailed level. The tuples that pertain to the combination $< Prague, 1996 >$ in the query result $q.cells$ are 7. The detailed tuples in $q^0.cells$ are 11. Observe that the belief statement is expressed with probability 70%. Then:

- Assuming a level of confidence $\Pi = 60\%$, i.e., lower than the 70% confidence of the belief statement for Prague, all the tuples of the query result that pertain to Prague in 1996 are considered not novel. Then the *Partial Extensional Same-Level Belief-Based Novelty (PESLBBN)* is 1 -$\frac{7}{107}$ = 93%. Had we used a level of confidence $\Pi = 80\%$, no tuple for Prague in 1996 would be considered as novel, and thus PESLBBN would be 100%.

- In full symmetry to the above, with a level of confidence $\Pi = 60\%$, the *Partial Detailed Extensional Belief-Based Novelty (PDEBBN)* of the query would be $1 - \frac{11}{117} = 91\%$, whereas PDEBBN would be 100% for a confidence level $\Pi = 80\%$, where no detailed tuple would qualify for novel.

6. Relevance

Relevance is a facet that pertains to retaining focus towards a specific information goal (or a set of them). The facet of relevance ensures that the data exploration does not wander around areas of the multidimensional space that are not of interest to the current information acquisition goal.

This is particularly the case with business intelligence scenarios, where the need to satisfy an informational gap (either on an ad-hoc or a recurring basis) is the main driver for accessing the database for data. This does not mean that the queries are pre-fixed, however: the quest for an information goal is very often "open" and an exploration of a certain sub-space of the data, possibly viewed from different angles and at different levels of granularity. In [9] we have named this exploration a "walk" in the multidimensional space.

As the above discussion demonstrates, a foundation for the assessment of the relevance of a query to an exploratory session or a recommendation to the use is the existence of an informational goal. The goal can be an ad-hoc goal for information, or a recurring one, based on a profile of data that have to be collected to answer recurring questions of the analyst. Specifically, we can discriminate between several cases: (a) the case where the goal is *explicitly* stated, or, (b) the case where the goal is *implicit* and has to be inferred from collateral profile information.

In the former case of explicitly stated goals, we will assume that the analyst specifies an area of the information space via a selection condition (again, the way this is extracted is orthogonal: it can be explicitly requested, it can be inferred from a natural-language expression, it can be part of a query or a KPI, etc).

In the latter case of implicit goals, the user has not provided any such information, and the system has to infer the intended goal from other means – examples include the history of past actions of the analyst, or possibly a profile, or a set of registered KPIs for the analyst.

In any case, independently of how to the system gets to register user goals, and in the interest of providing a concrete definition, in the rest of our deliberations, we will define *a goal as a selection condition ϕ_G over the multidimensional space.* We remind the reader that selection conditions are conjunctions of atoms of the form $L \in \{v_1, \ldots, v_k\}$ (see Section 3).

We examine the different alternatives in the respective subsections. To facilitate comprehension, in Figure 4 we summarize the proposed metrics in terms of our taxonomy, and, in Subsection 6.3 we exemplify the discussed metrics and algorithms.

6.1. *Relevance assessment in the presence of an explicit user goal via a selection predicate*

Assuming, then, that the goal is precisely or approximately specified, the essence of relevance estimation answers the question *how relevant is the query to a user's goal?* The main idea here is we formalize the user's declaration (via an explicit statement) that a specific area of the multidimensional space is of interest to him via a simple selection condition ϕ_G that characterizes the user interest.

Algorithm 4: Goal-Based Syntactic Enumeration Of Covered Detailed Cells

Input: A query q and, ϕ_G, a selection condition characterizing an area of the multidimensional space

Output: A pair: (a) $q^{rel^{0^+}}$, the subset of the coordinates of q^0 that are also part of the space the detailed proxy of ϕ_G covers, and, (b) its complement $q^{irr^{0^+}}$ of irrelevant cells

begin
    produce q^{0^+} and $\phi_G^{0^+}$
    $q^{rel^{0^+}} \leftarrow \emptyset$
    $q^{irr^{0^+}} \leftarrow q^{0^+}$
    forall $c^{0^+} \in q^{0^+}$ **do**
        if $c^{0^+} \in \phi_G^{0^+}$ **then**
            $q^{irr^{0^+}} = q^{irr^{0^+}}$ - c^{0^+}
            $q^{rel^{0^+}} = q^{rel^{0^+}} \cup c^{0^+}$
        end
    end
    return $q^{rel^{0^+}}$, $q^{irr^{0^+}}$
end

Algorithm 4 computes the subset of the multidimensional space at the most detailed level, i.e., the detailed *signature*, that pertains to the user goal ϕ_G. Then, it also computes the detailed signature of the query q. The algorithm splits the coordinates of the detailed signature of the query in two subsets (a) the ones *relevant to* (or *covered by*) the detailed signature of the user goal, and (b) the *irrelevant*, non-covered ones, represented by the sets $q^{rel^{0^+}}$ and $q^{irr^{0^+}}$, respectively.

Then, the *Goal-Based Detailed Syntactic Relevance* (GBDSR) of a query is the fraction of its detailed space that overlaps with the user's goal.

$$GBDSR = \frac{|q^{rel^{0^+}}|}{|q^{irr^{0^+}}| \bigcup |q^{rel^{0^+}}|}$$

The check is (a) partial (practically a normalized

		{Syntactic vs Extens.}x {SameLevel vs Detailed} x{Full vs Partial}	**Dimensions** *Schema only vs Schema & data*	**Basic Cube** *Schema only vs Schema & data*	**Current Query** *Syntax only vs Syntax & results*	**History** *Syntax only vs Syntax & results*	**Key Interests** *Implicit vs Explicit*	**Current Goal** *Implicit Vs Explicit*	**Beliefs** *Implicit vs Explicit*
Relevance									
6.1	GoalBasedDetailedSyntacticRelevance	Partial Detailed Syntactic	Schema & data	Schema only	Syntax only	---	---	Explicit	---
6.2	FullSameLevelSyntacticRelevance	Full SameLevel Syntactic	---	---	Syntax only	Syntax only	Explicit	Implicit	---
	PartialSameLevelSyntacticRelevance	Partial SameLevel Syntactic	Schema & data	Schema only	Syntax only	Syntax only	Explicit	Implicit	---
	PartialSameLevelExtensionalRelevance	Partial SameLevel Extensional	Schema & data	Schema only	Syntax &results	---	Explicit	Implicit	---
	FullDetailedSyntacticRelevance	Full Detailed Syntactic	Schema & data	Schema only	Syntax only	Syntax only	Explicit	Implicit	---
	PartialDetailedSyntacticRelevance	Partial Detailed Syntactic	Schema & data	Schema only	Syntax only	Syntax only	Explicit	Implicit	---
	PartialDetailedExtensionalRelevance	Partial Detailed Extensional	Schema & data	Schema & data	Syntax &results	Syntax &results	Explicit	Implicit	---

Figure 4: List of Relevance algorithms, characterized with respect to the reference taxonomy

score), (b) syntactical (without using the cells of the query results), and, (c) detailed, i.e., with respect to the detailed levels of the involved cubes. Assuming a fixed goal, and thus a fixed set of signatures for the goal, the complexity of the algorithm is linear with respect to the query result size. Also, to the extent that the test is syntactic, the data size is irrelevant.

Remark. An extension to a set of multiple goal statements $\Phi = \{\phi_1, \ldots, \phi_k\}$ is also possible. The union of the detailed signatures of the goals can provide the equivalent of $\phi_G^{0^+}$ for such an extension. Again, weighted variants can be part of the score evaluation.

6.2. Relevance assessment implicitly, in the absence of an explicit user goal

Assume that an explicit goal to study a certain subset of the multidimensional space is not available, but instead, the system has access to a set of KPIs. KPIs are explicit expressions of time-invariant interests (rather than a current user goal), so, even if they do not explicate exactly what the user wants to achieve *now*, they act as reference points of relevance for the user's interest.

In the rest of our deliberations, we assume that KPIs are expressed as a set of annotated queries $Q = \{q_1, \ldots, q_n\}$*, which we call beacon queries, that implicitly approximate the user interest.*

As a side-note, observe that, in extremis, one could even resort to the user's history for indications of relevance. We emphasize that past queries are last-resort, coarse manifestations of relevance, as they are only in the past and not necessarily linked to what the user explores now, or, they could be erroneous, or playful, or eventually irrelevant, etc. However, despite all these valid reservations, it could be the case that this is the only thing that the system knows about the user's idea of what is relevant.

Intuition. What we want to assess is how much a new query q overlaps with the set Q of beacon queries. Observe that all the methods that we define assess the overlap of levels and coordinates between q and the queries of Q; *measures and aggregate functions are not involved in the assessment of relevance, as the idea is to "highlight" the subset of the multidimensional space that seems relevant to the user.*

In the rest of this subsection, we simplify the discussion by avoiding aging factors and possible weights of the different queries and considering a single input for the interestingness assessment algorithm: a set of beacon queries which we (approximately) deem to be relevant. We will also use the notion of *coverage*, already discussed for novelty, aiming towards finding the overlap of the area covered by the beacon set and the area pertaining the current query.

The special case where all queries are defined at the same level. Assuming all cubes of Q and q are at the same level, we can assess relevance via (a) a full syntactic check returning true/false, as *Full Syntactic Same-Level Relevance* (FSSLR):

$$FSSLR = \begin{cases} 1 & \text{if a } q_i \equiv q \text{ exists} \\ 0 & \text{otherwise} \end{cases}$$

and (b) a partial check returning a *Partial Syntactic Same-Level Relevance* (PSSLR) score

$$PSLSR = \frac{|q^{cov^+}|}{|q^{nov^+}| \bigcup |q^{cov^+}|} = \frac{|q^{cov^+}|}{|q^+|}$$

It is very important to stress that the same-level relevance can only be applied in the case where all the cubes are at the same level of abstraction. Overall, the idea is that the beacon-set provides a homogeneous space for query evaluation at the same level, and thus, we can compute relevance without having to resort to the detailed space.

The *Extensional* counterpart of relevance (e.g., $PartialSameLevelExtensionalRelevance$) is defined equivalently, with cells of the query result instead of signatures.

Now, we are ready to move on to the fundamental definitions of relevance that are based on the detailed level.

Foundations of history-based relevance assessment. The most fundamental definition for relevance comes from the space of detailed cells, as *Full Detailed Syntactic Relevance* (FDSR).

$$FDSR = 1 - FullDetailedSyntacticNovelty$$

The most fundamental assessment method of all is to compare the union of the detailed signatures of the queries of Q with the signature of q. The amount of overlap signifies the relevance of the new query.

To characterize the cells of the result of q (in fact: their coordinates) as previously covered vs novel, we can simply refer to Algorithm 1 this time passing all the history as argument, i.e., without the requirement of same measures and aggregate functions. Equivalently, we can use (a) the detailed proxy of q, q^0 and (b) the detailed equivalents of the queries of Q, q_i^0, and pass them as input to the algorithm ComputePartialImmmediateCubeCoverage of [49]. Observe, that when working at the detailed level, coordinates and cells are practically of the same cost, esp., since measures are not taken into consideration. Then, the sets $q^{cov^{0^+}}$ and $q^{nov^{0^+}}$ (respectively, q^{cov^0} and q^{nov^0}) are produced. Based on these sets, we can compute *Partial Detailed Extensional Relevance* (PDER)

$$PDER = \frac{|q^{cov^0}|}{|q^{nov^0}| \bigcup |q^{cov^0}|} = \frac{|q^{cov^0}|}{|q^0|}$$

Partial Detailed Syntactic Relevance (PDSR) is computed via signatures, respectively. The complexity of computing all these formulas is practically the same with the one of Algorithm 1, and therefore, linear with respect to query history and fact table size.

Remark. Interestingly, when the assessment is history-based, relevance is practically complementary to novelty. For all variants of syntactic vs extensional, partial vs full, same-level vs detailed, when the assessment is history-based, the following formula holds: $relevance + novelty = 1$. We can only emphasize that this is an approximation applicable only to the history-based metrics that we have introduced here, and by no means do we insinuate that being relevant precludes being novel. Quite the opposite: in an exploratory phase, when a concrete goal starts to shape in the mind of the analyst, the early queries are both relevant and novel. But this, pertains to the case where the analyst has a concrete goal, against which relevance is assessed.

6.3. Reference Example Revisited

Coming back to the reference example of Section 3.2, we can comment on the relevance of the query q that is assessed over the presence of an explicit user goal, say $\phi_G : district =' Brno - mest'$. We can assess the *Goal-Based Detailed Syntactic Relevance (GBDSR)*, as follows. The signature of the query q would involve the Cartesian product of all 57 districts of the cube over the 12 months of a single year, 1996. The user goal involves only one district. Thus, GBDSR $= \frac{1}{57} = 0.018$.

Assuming no gal was ever stated, we can still comment on the relevance of the query q that is assessed over the presence of a query history $Q = \{q_1, \ldots, q_4\}$. Basically, the explanations given for the case of novelty, in Section 5.3, also cover the discussion for relevance.

Given that the syntactic checks give a novelty of 1, then, as one would expect, syntactic relevance takes a value of zero. So, *Full Syntactic Same-Level Assessment of Relevance* (FSSLR), *Partial Syntactic Same-Level Assessment of Relevance* (PSSLR), and *Full Syntactic Detailed Relevance* (FSDR) are all zero. Concerning the *Partial Detailed Extensional Novelty* (PDER), however, it takes the value of 0.3, to the extent that it is the complement of its Novelty counterpart that took the value of 0.7.

7. Peculiarity of a query

How peculiar is a query? To understand peculiarity we must understand that its essence lies in discriminating a particular object (in our case: a query) from its peers (in our case: a session, history, or just collection of other queries, to be used as the context for the assessment of peculiarity). Beliefs, Key Interests and Goals are not explicitly treated here. However, peculiarity can be evaluated on the grounds of whichever entities can be implicitly represented by queries; to the extent that at least Key Interests can be expressed as queries, peculiarity can be implicitly related to them.

Therefore, in the rest of our deliberations, we assume that every query q is going to be assessed against a collection of queries $Q = \{q_1, \ldots, q_n\}$. This generic setup can cover two alternative situations: (a) a set of KPIs, each expressed via a query, collectively describing a set of static key interests of the user, and, (b) a set of queries in the history (be it the current session, or the history of previous sessions).

We examine the different alternatives in the respective subsections. To facilitate comprehension, in Figure 5 we summarize the proposed metrics in terms of our taxonomy, and, in Section 7.3 we exemplify the discussed metrics and algorithms.

7.1. Syntactic Peculiarity

Assume the query q and a collection of queries $Q = \{q_1, \ldots, q_n\}$. How different is q from the collection Q?

Fundamentally, the question boils down to answering the assessment of the distance of two queries. To support our discussion in the sequel we assume two queries *over the same data set* in a multidimensional space of n dimensions.

$$q^a = \mathbf{DS}^0,\ \phi^a,\ [L_1^a, \ldots, L_n^a, M_1^a, \ldots, M_{m^a}^a], \\ [agg_1^a(M_1^{a^0}), \ldots, agg_m^a(M_{m^a}^{a^0})]$$

and

$$q^b = \mathbf{DS}^0,\ \phi^b,\ [L_1^b, \ldots, L_n^b, M_1^b, \ldots, M_{m^b}^b], \\ [agg_1^b(M_1^{b^0}), \ldots, agg_m^b(M_{m^b}^{b^0})]$$

To solve the problem of computing the distance of two queries, we use the syntactic formula from [49], which, in turn, is based on results from (see [53], [54], [55]).

The syntactic distance of the two queries is expressed by the weighted sum of structural distances between their selection conditions, their grouping levels, and the measures they employ, as:

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

such that the sum of the weights w^i adds up to 1. We follow [55] and recommend the following weights: w^ϕ: 0.5, w^L: 0.35, w^M: 0.15.[10]

Given, then, the [49] method for computing distance of two queries $\delta(q^a, q^b)$, the computation of the distance of a new query q to a pre-existing collection of queries Q can be computed via several possible methods, out of which we highlight a couple of prominent ones:

1. A simple statistic over the distances of the query to the set members, $\delta(q, Q) = \gamma(\delta(q, q_i)), q_i \in Q, \gamma \in \{min, max, average, median\}$.
2. k-nn distance of the query to the set, $\delta(q, Q) = k$-th smallest $\delta(q, q_i), q_i \in Q$. Practically, this entails ranking all the distances of q to the elements of Q in ascending order and take the k-th one.

[10] For the particularities of the different components of the formula, we refer the interested reader to [49], Sec. "Query Distance".

The check is (a) partial (practically a normalized score), (b) syntactical (without using the cells of the query results), (c) depending upon the statistic or function that determines the final value of the metric, and, (d) indifferent to the schema levels of the involved cubes. We can define a *Partial Syntactic Cube Peculiarity* based on which method we pick for the determination of the final value, e.g., *Partial Syntactic Average Cube Peculiarity* uses the average query distance to determine the peculiarity of the measured query. To the extent that we refer to syntactic checks, data size is irrelevant for the complexity of the algorithm. However, the algorithm requires a linear pass from all the queries of the history and a pairwise computation of distance at its first phase, as well as the determination of the final peculiarity (again requiring at most a linear past of all distances): therefore, the complexity is linear with respect to the size of the collection Q.

7.2. Value-based Peculiarity

When we address the issue of value-based peculiarity assessment, we base the result of the assessment on the actual values of the cells of the result of the query. Then, we treat each query as a set of cells (each cell primarily identified by its coordinates).

The general setup of value-based peculiarity. The general setup of the value-based query peculiarity problem is as follows. Assume a set of background queries $Q = \{q_1, \ldots, q_n\}$ (either due to the history of a session, or, due to the existence of a set of KPI's). Assume also a new query q that is also submitted to the system.

Algorithm 5 provides the generic recipe for computing query peculiarity. Depending on the setup of individual design choices, we can have several configurations of the algorithm.

The combination of the query distance function δ_q and the aggregate function f_p^{agg} can determine the peculiarity of the query. As we will present in the sequel, the two prominent methods for assessing the cube query distance δ_q are the Hausdorff and

		{Syntactic vs Extens.}x {SameLevel vs Detailed} x{Full vs Partial}	Dimensions Schema only vs Schema & data	Basic Cube Schema only vs Schema & data	Current Query Syntax only vs Syntax & results	History Syntax only vs Syntax & results	Key Interests Implicit vs Explicit	Current Goal Implicit Vs Explicit	Beliefs Implicit vs Explicit
Peculiarity									
7.2	Partial Syntactic Average Peculiarity	Partial Syntactic LevelIndiff.	Schema & data	---	Syntax only	Syntax only	Implicit	---	---
7.2.1	Partial Extensional Value-Based Peculiarity	Partial Syntactic LevelIndiff.	Schema & data	---	Syntax &results	Syntax &results	Implicit	---	---
7.2.2	PartialExtensionalDetailedJaccard-BasedPeculiarity	Partial Detailed Extensional	Schema & data	Schema & data	Syntax &results	Syntax &results	Implicit	---	---

Figure 5: List of Peculiarity algorithms, characterized with respect to the reference taxonomy

Algorithm 5: The general setup of value-based query peculiarity assessment

Input: A set of background queries $Q = \{q_1, \ldots, q_n\}$; a new query q to be assessed over Q for its peculiarity; a distance function δ_q for computing the distance of two queries, an aggregate function to compute the query peculiarity f_p^{agg}

Output: The value-based peculiarity of the query q

Variables: V: the bag of distance values of q over members of Q

begin
  $V = \emptyset$
  forall $q_i \in Q$ **do**
    $V = V \bigcup \delta_q(q_i, q)$
  end
  $q.peculiarity = f_p^{agg}(V)$
  return $q.peculiarity$
end

the Closest Relative methods, whereas the f_p^{agg} aggregate function can be serviced by any aggregate function like min, max, k-NN, etc.

The check is (a) partial (to the extent that δ_q returns a score), (b) extensional (using the cells of the query results), (c) depending upon the aggregate function that determines the final value of the metric, and, (d) indifferent to the schema levels of the involved cubes. We can define a *Partial Extensional Value-Based Peculiarity* based on which method we pick for the determination of the final value, e.g., *Partial Extensional Hausdorff/ClosestRelative Average/k-NN/Minimum Peculiarity* if we use (a) the Hausdorff or the Closest Relative method for the determination of query distance, and, (b) the average/k-NN/minimum query distance to determine the peculiarity of the measured query. The algorithm requires a linear pass from all the queries of the input query set, as well as the determination of the final peculiarity (again requiring at most a linear past of all distances): therefore, the complexity is linear with respect to the size of the collection Q. To the extent that we use query results, we can assume that the size of the query results affects the execution time of the algorithm.

7.2.1. The closest relatives of Hausdorff

Cell-based Query Distance. How then, do we compute the distance of two queries? Assume we want to assess how distant are the queries q^a and q^b with

$$q^a.cells = \{c_1^a, \ldots, c_{n^a}^a\} \text{ vs. } q^b.cells = \{c_1^b, \ldots, c_{n^b}^b\}$$

Earlier works about comparing queries through their sets of cells, such as [56], have shown that the distance of these two sets of cells is not straightforward to assess. The reasons can be identified as follows:

- It is not straightforward how to map the cells of the one query to another; this is especially true if the cardinality of the two queries is not the same;

- It is possible that the two queries are defined at different levels of aggregation, which means that they are not directly comparable;
- Even if the above problems are not present, deciding a mapping from the cells of q^a to the cells of q^b is not a straightforward task.

If we want to exploit the query results, i.e., assuming the cells of the query results are available, we can reuse the results of [53] to derive query distances. The main idea of [53] was to assess alternative ways of computing the distance of two cubes on the basis of their contents. Two formulae eventually came out as the possible winners of the benchmark, specifically the *Closest Relative* and *Hausdorff* formulae. Before discussing these alternatives, however, we will introduce a cell distance formula, which is necessary for performing the respective cube distance calculations.

Cell distance. How distant are two cells? [53] performs a thorough analysis of several alternatives, out of which, the experimental assessment clearly discriminated the *Weighted Sum of Value distances based on the Least Common Ancestor (LCA)* method as the most appropriate one.

Assume two members of a dimension D, say v and v', not necessarily at the same levels. Assume also v_{LCA} is their least common ancestor (could be one of them if they are related with an $anc()$ relation). Then, the *distance of two values of the same dimension* is

$$dist(v, v') = \frac{path(v, v_{LCA}) + path(v', v_{LCA})}{2 \times path(ALL, L^0)}$$

where *path* is the number of hops (edges) in the hierarchy between the respective values(levels, respectively).

What is the distance of Athens to Canada in figure 6? The least common ancestor is the *All* value in level *ALL* and has distance (number of intervening edges) to Athens equal to 3 and distance to Canada equal to 2. The edges between L^0, i.e., *City*, and *ALL* is 3. Then, $dist(Athens, Canada)$ is $(3 + 2)/(2 \cdot 3) = 5/6$.

To simplify [53], the *distance of two cells* over the same dimensions is the weighted sum of the distances of their respective values. Given two cells $c :< v_1, \ldots, v_n >$ and $c' :< v'_1, \ldots, v'_n >$ their distance is:

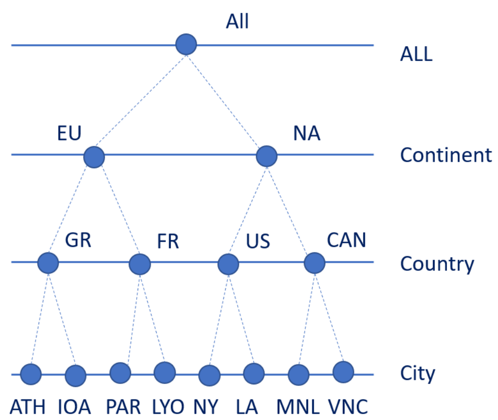

Figure 6: A sample geographical dimension

$$dist(c, c') = \frac{1}{n} \sum_{1 \le i \le n} dist(v_i, v'_i)$$

Closest Relative distance of two cube queries. The closest relative distance of two cubes [53] is based on mapping the cells of the two cubes in pairs with the minimum distance and taking their average distance. Specifically, the method to compute the closest relative distance of two cube queries q and q' includes the following steps:

1. For each cell c of query q, find the cell c' in q' with the minimum distance;
2. Add the respective distance to a bag of values B_d;
3. Once done with all cells of q, return the mean value of B_d

The intuition of the formula is very simple: we take the average distance between the cells of the two cubes as the distance of the two cubes.

Hausdorff distance of two cube queries. As mentioned in [53], the Hausdorff distance between two cube queries q and q' can be defined as:

$$H(q, q') = max(h(q, q'), h(q', q)), \quad \text{where}$$
$$h(q, q') = max_{c \in q.cells}(min_{c' \in q'.cells}(\delta(c, c'))) \quad \text{and}$$
$$\delta(c, c') \text{ is the distance of any two cells}$$

Function $h(q, q')$ is called the directed Hausdorff distance from q to q', and it is not necessarily symmetric. Practically, to compute h we have to perform the following steps: (a) for every cell $c \in q$, we find the cell $c' \in q'$ with the minimum distance (effectively pairing each cell of cube q to its closest counterpart in q'); (b) out of all these distances, we select the maximum one. Then, we do the symmetric computation for the $h(q', q)$ and we take the maximum of the two $h(\cdot)$ values.

7.2.2. Jaccard-based resolution via cell comparison at the detailed level

A possible answer to the problem is to address the issue by referring to the detailed cells that pertain to the aggregate cells that constitute the results of the compared queries. Remember that we refer to the set of cells that produce an aggregate cell as the *detailed area* of the cell; the detailed area of a set of aggregate cells is defined respectively. Let $q_1^0.cells$ be the detailed area of q_1 over C^0 and $q_2^0.cells$ be the detailed area of q_2 over C^0. Then, we can compute the Jaccard similarity of the two detailed areas. The distance of the two queries is:

$distance(q_1, q_2) = 1 - JaccardSimilarity(q_1^0.cells, q_2^0.cells)$.

Algorithm 6: 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 neighbor
Output: $pec(q|Q)$: the Partial Extensional Detailed Jaccard-Based Cube Peculiarity
Variables: L = a list of Jaccard distances

begin
  $L = \emptyset$
  Compute q^0
  forall $q_i \in Q$ **do**
    Compute q_i^0
    Compute $JD_i = 1 - \frac{|q_i^0 \bigcap q^0|}{|q_i^0 \bigcup q^0|}$
    $L = L \cup JD_i$
  end
  L_s = Sort L ascending into a sorted list
  return $pec(q|Q) = L_s[\ k\ ]$
end

The intuition of Algorithm 6 is based on the idea that the peculiarity of a query is based on how much overlap its detailed cells have with the detailed cells of the queries in the history. The check is (a) partial (practically a Jaccard distance), (b) extensional (with the use of the cells of the query results), and, (c) detailed, i.e., with respect to the detailed levels of the involved cubes.Thus, we define the *Partial Extensional Detailed Jaccard-Based Cube Peculiarity* (for short: Value-based Peculiarity) as the result of Algorithm 6.

The execution cost is dominated by the execution of the detailed queries for both the reference queries and the queries of the history Q. The complexity of the algorithm is obviously linear with respect to the history size, since there is a single detailed query q_i^0 to be executed per member of Q. Also, the in-memory check between the results of the queries is also linear with respect to the history size. At the same time, the complexity is also linear with respect to the cube size, assuming that the execution cost for all the queries linearly depends on the cube size (i.e., all the involved queries have their execution time scale linearly with the same scale factor over the cube size).

Remark. A point worth mentioning here, is that the form of peculiarity we have been discussing so far, is signature-based, i.e, defined with respect to the area of the multidimensional space it refers to. As part of future work, research might address peculiarity via a more value-based approach, where the comparison of the queries is more based on values than on signatures. The extent of the issue is vast, since we need to synthesize the combined peculiarity of a query on the basis of its cells, and, to this end, we need dedicated studies on the topic, on how users perceive derived value-based peculiarity. The extent of the problem is such that it places it out of the scope of this paper.

7.3. Reference Example Revisited

Partial Syntactic Cube Peculiarity. The distances of the new query q from the rest of the queries of the history are depicted in Table 2. Then, it is easy to pick either the *Partial Syntactic Average Cube Peculiarity* (as depicted in the Table), or any other aggregate value over the individual distances (e.g., the $k - th$ one).

Partial Extensional Detailed Jaccard-Based Cube Peculiarity. The basic ingredient for determining the Jaccard based distance is the computation of the quantity $\frac{|q_i^0 \bigcap q^0|}{|q_i^0 \bigcup q^0|}$. Assuming we take the $k = 2$ distance, the *Partial Extensional Detailed Jaccard-Based Cube Peculiarity* (for short: Value-based Peculiarity) of q is 0.94.

q vs	δ^{ϕ}	δ^{L}	δ^{M}	δ
q_1	1.0	0.0	0.0	0.5
q_2	1.0	0.0	0.0	0.5
q_3	1.0	0.0	0.0	0.5
q_4	0.67	0.0	0.0	0.33
avg				0.46

Table 2: Syntactic distances of q from the rest of the queries in the reference example. We use the following weights: w^{ϕ}: 0.5, w^{L}: 0.35, w^{M}: 0.15.

8. Surprise

Surprise is an interestingness facet that depends mainly (if not only) on prior beliefs. The main idea about assessing surprise is to evaluate how far from the beliefs of the analyst do the actual values lie. The two problems that one has to handle are: (a) *what kind of beliefs can we express, and how?*, and, (b) *assuming these beliefs have, somehow, been expressed, how can we compute surprise on their basis?*

8.1. Expressing beliefs

We can express beliefs in a variety of ways: specific values, expected intervals, probabilities; we can even label results and give probabilities for the labels, too. Is it necessary, however, for the analysts to express beliefs manually? In the case of KPIs that label performance, this is explicitly done. In the general case, all analysts work with some form of predictions that are automatically derived via methods in the spectrum from a simple regression over past values to elaborate statistical models that economists use.

8.2. Computing surprise: the overall setup

We start with the second problem and assume that for certain cells in the multidimensional space, we can register or compute their expected values for specific measures (several alternatives are discussed in the rest of this section). So, for such a cell, for each of these measures, we have (a) the actual value m, and, (b) the expected value m^e.

q vs	$\lvert q_i^0 \bigcap q^0 \rvert$	$\lvert q_i^0 \bigcup q^0 \rvert$	J	JD
q_1	11	190	0.06	0.94
q_2	0	137	0.00	1.00
q_3	21	123	0.10	0.90
q_4	2	117	0.02	0.98
2-NN				0.94

Table 3: Jaccard distances of q from the rest of the queries in the reference example. We list the number of cells in the intersection and union of the detailed areas of the involved queries, their fraction J, and the Jaccard distance.

Then, the questions that we need to answer are (a) how do we assess the surprise for a specific cell over a specific measure, (b) how do we assess the surprise for a specific cell, with respect to all its measures (assuming multiple such measures exist), and, (c) how do we assess the overall surprise of a query result (which, of course, includes a set of cells)?

Let us start with a single measure for a single cell. Fundamentally, surprise is a function of how far the expected from the actual value lies. Therefore, $surprise(c.M) = (distance(m, m^e))$ – for example, $surprise(c.M) = |m - m^e|$.

Assuming now a set of measures per cell, the total surprise of a cell is an aggregate measure computed over the set of surprise values for the various measures of a cell (e.g., the number of measures indicating a non-zero amount of surprise, or maybe the maximum, or the average surprise). Formally: $surprise(c) = f^{agg}_{cell}(surprise(c.M_i))$, $f^{agg}_{cell} \in \{count, sum, mean, median, max, min, ...\}$

Finally, now that we can compute the surprise for each individual cell, we can proceed in computing the surprise for a set of cells, e.g., a query result. The surprise of a set of cells, say $C = \{c_1, \ldots, c_n\}$ is $surprise(C) = f^{agg}(surprise(c_i))$, $f^{agg} \in \{count, sum, mean, median, max, min, ...\}$

One possible concern here is what happens if there is no expected value registered for a measure of a cell. Then, there are two ways to handle the situation: (a) this particular measure value does not participate in the rest of the computation, or, (b) a mechanism for computing a derived expected value, against which we will perform the comparison (e.g., the average of the expected values, an interpolation over certain criteria, etc), is introduced. Unless explicitly mentioned otherwise, the former policy of excluding the respective measure value from any computation will be our reaction of choice.

We examine alternatives for handling surprise in the following subsections. To facilitate comprehension, in Figure 7 we summarize the proposed metrics in terms of our taxonomy, and, in Section 8.5 we exemplify the discussed metrics and algorithms.

8.3. Value-based average cell surprise

The most simple implementation of the assessment of surprise is to follow the general setup and (a) compute a simple distance of the actual and

		{Syntactic vs Extens.}x	**Dimensions**	**Basic Cube**	**Current Query**	**History**	**Key Interests**	**Current Goal**	**Beliefs**
		{SameLevel vs Detailed}	*Schema only vs*	*Schema only vs*	*Syntax only vs*	*Syntax only vs*	*Implicit vs*	*Implicit Vs*	*Implicit vs*
		x{Full vs Partial}	*Schema & data*	*Schema & data*	*Syntax & results*	*Syntax & results*	*Explicit*	*Explicit*	*Explicit*
Surprise				---					
8.3	Partial Extensional Average Value-Based Surprise	Partial Indifferent Extensional	---	---	Syntax &results	---	Explicit	---	Explicit
8.4	Probability-Based Surprise	Partial Indifferent Extensional	---	---	Syntax &results	---	Explicit	---	Explicit

Figure 7: List of Surprise algorithms, characterized with respect to the reference taxonomy (underline: implemented, bold: experimented)

the expected value per measure, and per cell, (b) aggregate the measures' surprise per cell, and (c) aggregate the different cell surprises to compute the surprise of the set of cells.

Algorithm 7 provides the general recipe for computing the surprise according to the general setup. This generic algorithm can be specialized by fixing the involved functions to specific choices. For example, to compute the *Partial Extensional Average Value-Based Surprise*, Algorithm 8 works on a single-measured cube, with absolute distance as the distance function to assess how far the actual and the expected measures are, and averaging over all cells with surprise to produce the aggregate cube surprise.

The complexity of the algorithm is linear with respect to the result size for the query, assuming a fixed set of expected values E.

Algorithm 7: The general setup of value-based surprise assessment

Input: A cube C including a set of cells $\{c_1, \dots, c_n\}$ with a set of measures $\mathbf{M}$; a set of tuples registering the expected values for each cell $E = \{e_1, \dots, e_n\}$, with each e_i being a tuple of expected measures $e_i =< m_1^e, \dots, m_m^e >$; a distance function δ_M for each measure, for computing the distance of the actual from the expected value of a cell's measure; an aggregate function to compute a cell's surprise f_{cell}^{agg}; an aggregate function to compute the cube's surprise f^{agg}

Output: The surprise carried by the cube C

Variables: The bag of surprise values for the entire cube C: $C.S$; The bag of surprise values for a specific cell c: $c.S$

begin
- The bag of surprise values for C, $C.S = \emptyset$;
- **forall** $c \in C$ **do**
  - The bag of surprise values for this cell $c.S = \emptyset$;
  - **forall** $M \in \mathbf{M}$ **do**
    - **if** $\exists$ *an expected value* $c.m_j^e$ *for* $c.m_j$*, both over measure* M **then**
      - $c.S = c.S \bigcup \delta_M(c.m_j, c, m_j^e)$;
    - **end**
  - **end**
  - $c.surprise = f_{cell}^{agg}(c.S)$;
  - $C.S = C.S \bigcup c.surprise$;
- **end**
- $C.surprise = f^{agg}(C.S)$;
- **return** $C.surprise$;

end

Algorithm 8: 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 $\{c_1, \dots, c_n\}$ with a single measure M, a set expected values for each cell $E = \{m_1^e, \dots, m_n^e\}$

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

begin
- $countOfCellsWithSurprise = 0$;
- $C.surprise = 0$;
- **forall** $c \in C$ **do**
  - $c.surprise =$ null;
  - **if** $\exists$ *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** $countOfCellsWithSurprise \neq 0$ **then**
  - $C.surprise = C.surprise / countOfCellsWithSurprise$;
- **else**
  - $C.surprise =$ null;
- **return** $C.surprise$;

end

8.4. Expressing expectancy via probabilities of expected values

Whereas in the previous section we have assumed that a specific value is available as the expected measure of a cell, in this Section we follow a different approach and register expected values by annotating the expectation for a value to appear via a probability of appearance, and then, measure surprise on the basis of this probability. We will refer to the surprise metrics that are produced by the alternatives introduced in this Section, as belonging to the category of *Partial Probability-Based Surprise*. In [10], we also discuss the case of label-based expectancy.

8.4.1. Probability of values

Assume a cell c and a certain measure M (for ease of comprehension, we simplify by using just a single measure per cube). Apart from the previously mentioned value-based evaluation, another possibility for assessing surprise is to register probabilities per expected value for the value $m = c.M$. So, we annotate each cell with a set of statements of the form:

$$p(c.M = m) = p^m,\ p^m \in [0..1]$$

In the above expression, by abuse of notation, we use the term c to refer to the coordinates of the cell c^+. In all our deliberations, $p(< expression >)$ expresses the probability of appearance of the parameter of the function $p(\cdot)$. For example, in a 2-dimensional cube over geography and time, we can have:

$$p(sales = 100 \mid city = Athens, year = 2020) = 20\%$$
$$p(sales = 80 \mid city = Athens, year = 2020) = 70\%$$
$$p(sales = 70 \mid city = Athens, year = 2020) = 10\%$$

Assuming the actual value of the measure is m, the strict surprise of the cell for a value m is the sum of the probabilities of all the other values m' that are different than m.

$$c.StrictSurprise = \sum_{m' \neq m} p(c.M = m')$$

In the example above, assuming the actual value is 70, the surprise is $20\% + 70\% = 90\%$.

The result of applying Algorithm 7 with exact probabilities for the cells' measure will be referred to as *Partial Exact Probability Surprise*.

8.4.2. Interval-based probability definition

A more realistic approach in terms of how we express the probabilities, is that instead of identifying probabilities for individual values, we can assign probabilities to intervals of values. Thus, the statements take the form:

$$p(c.M \in [low \ldots high]) = p^m,\ p^m \in [0..1]$$

For example, one could express the statement

$$p(sales \in [100..200] \mid city = Athens, year = 2020) = 20\%$$

In the above expression, and in contrast to the setting of the previous subsection, the probability is expressed for a range of measure values, rather than a single one. To facilitate the registration of such expected values, a similar trick can be done, in terms of expression, for the cell coordinates. So, instead of saying

$$p(sales = 100 \mid city = Athens, year = 2020) = 20\%$$

one could possibly say

$$p(sales = 100 \mid city = Athens, year \in [2018..2020]) = 20\%$$

or even

$$p(sales \in [100..200] \mid city = Athens, year \in [2018..2020]) = 20\%$$

It is important to note, however, that these expressions are no more than syntactic-sugar statements on how we express the fundamental statement of assigning probabilities of the form $p(M = m \mid c) = p$. Therefore, the method for computing surprise does not change, effectively. Specifically, assuming the actual value of the measure is m, the strict surprise of the cell for a value m is the sum of the probabilities of all the expressions with ranges r' that do not include m.

$$c.StrictSurprise = \sum_{r' \not\ni m} p(M \in r' | c)$$

The result of applying Algorithm 7 with probability intervals for the cells' measure will be referred to as *Partial Interval Probability Surprise*.

8.5. Reference Example Revisited

Value Based Surprise. Suppose that we have a set of expected values for the measures of loan amounts regarding the city of Olomouc, as shown in Table 4. For the computation of the Value Based Surprise of the new query q, the absolute distance of measure value of each cell of the results of q that is also found in the expected values is calculated and from all the absolute distances, an average value distance occurs. Finally, in order for the algorithm to always return a result in the scale of 0.0 - 1.0, the average value distance is normalized.

District Name	Month	Measure
Olomouc	1998-01	22512
Olomouc	1996-09	20048
Olomouc	1998-09	46666
Olomouc	1997-05	53212
Olomouc	1995-07	60005
Olomouc	1997-10	78696
Olomouc	1996-12	155616
Olomouc	1996-05	161496
Olomouc	1996-07	187104
Olomouc	1994-05	193968
Olomouc	1995-12	263355
Olomouc	1995-09	309552
Olomouc	1997-12	465506

Table 4: Expected values for the measures of loan amounts regarding the city of Olomouc, with the respected month that the loan was granted.

In Table 5, the cells of the results of q that are also found in the expected values are presented. The table also presents the absolute distance of each cell measure value to the respected one in the expected values. Finally, the table also provides the total and average absolute distance, i.e., average Value Surprise, along with the normalized Value Surprise that is returned from the algorithm.

District Name	Month	Measure	Absolute Distance
Olomouc	1996-09	29448	9400
Olomouc	1996-12	155616	0
Olomouc	1996-05	161496	0
Olomouc	1996-07	187104	0
Sum of Absolute Distances			9400
Average Absolute Distance			2350
Value Based Surprise		2350 - 0 / 9400 - 0 = 0.25	

Table 5: Results of the new query q that are also found in the expected values as shown in Table 3.

The Valued Based Surprise result as shown in the last row of Table 5, occurs as the normalized Average Absolute Distance of the values of cells of the results that are also found in the expected values. The normalized distance is calculated by deducting the minimum absolute distance of a cell (here, 0) from the average absolute distance (2350) and by dividing it to the maximum absolute distance of a cell (9400) minus the minimum absolute distance (0). The result of this calculation in our example is 0.25, as Table 5 shows.

Let us turn now to *Partial Probability-Based Surprise.* In Fig. 8, we provide an example based on a subset of the results of query q. Suppose we have the following three beliefs for the average monthly amount of loans given in the area of Prague in 1996:

$b1$: $p(AvgMonthlyLoanAmt \leq 45K \mid$ *district=Prague, year=1996*$) = 60\%$
$b2$: $p(AvgMonthlyLoanAmt \in [45K..85K) \mid$ *district=Prague, year=1996*$) = 30\%$
$b3$: $p(AvgMonthlyLoanAmt \geq 75K \mid$ *district=Prague, year=1996*$) = 10\%$

In Fig. 8, we depict some cells of the result of q concerning the area of Prague, for the months of 1996 that had loans. For each such cell, for each of the three beliefs, we measure the amount of surprise induced by their violation. Since the beliefs cover the entire space of values for the average monthly amount of loans, each time, one of them is satisfied, therefore inducing zero surprise, and the rest are violated. The sum of the probabilities of the violated beliefs produces the measure of surprise for the respective cell. Take for example the cell for October 1996: the value of 83120 belongs to the range of $b2$. Thus, the surprise for this cell is calculated as the sum of surprise due to b1 and b3, i.e. 70% overall.

Once all cells have their surprise computed, we can compute the *Partial Extensional Average Probability Interval Surprise* as the average value of the surprise of each cell. In this case the average such measure of surprise amounts to 77%.

The discussed mechanism is quite flexible, overall. Observe that the same mechanism works in the same spirit for exact probabilities. Moreover, the mechanism also works for probabilities that do not explicitly cover the entire space of values.

For district = Prague, month in [1996-01 .. 1996-12]
b1: p(AvgLoanAmt in [0-45K): 60%
b2: p(AvgLoanAmt in [45K-85K): 30%
b3: p(AvgLoanAmt in >=85K: 10%

district name	month	avg monthly amt loans	Surprise due to ... b1	b2	b3	Σ
Prague	1996-03	247728	60%	30%	0%	90%
Prague	1996-04	320976	60%	30%	0%	90%
Prague	1996-07	20196	0%	30%	10%	40%
Prague	1996-08	258444	60%	30%	0%	90%
Prague	1996-09	70146	60%	0%	10%	70%
Prague	1996-10	83120	60%	0%	10%	70%
Prague	1996-11	169206	60%	30%	0%	90%
			AverageSurprise:			**77%**

Figure 8: An example of interval probability surprise

9. A User Study on the interestingness facets

The proposed metrics have to be evaluated with respect to their effectiveness and their efficiency. First, in Subsection 9.1, we describe our experiments for evaluating efficiency, summarizing our main findings. The remaining of this section is devoted to detailing the evaluation of effectiveness, via a user study that we conducted in order to evaluate how do the introduced interestingness facets relate to the behavior of people working with cubes and cube queries. From the effectiveness point of view, a user study is indeed the way to assess the relevance of a proposed theory to the actual behavior of human agents. To the best of our understanding, this is the first study on cube interestingness ever conducted.

All the material of the study, along with our findings, are available via a public repository[11].

9.1. Efficiency of different algorithms

Concerning the efficiency of the proposed metrics, an experimental evaluation has already been presented in [10] and is also available with more details at [11].

[11] `https://github.com/OLAP3/2023InterestingnessUserStudy`

In a nutshell, we have evaluated the efficiency of several algorithms, via their execution time, by stress-testing them in terms of fact table size, query result size and history size, wherever applicable.We tested 7 algorithms among the ones presented in previous sections, covering all interestingness aspects and varied information in terms of the taxonomies. Specifically, we have evaluated Partial Detailed Extensional Novelty, Partial Extensional Detailed Belief-Based Novelty, Partial Same Level Extensional Cube Relevance, Partial Detailed Extensional Cube Relevance, Partial Syntactic Average Cube Peculiarity, Partial Extensional Detailed Jaccard-Based Peculiarity, and Partial Extensional Value-Based Surprise. Hereafter, we describe our main findings.

We highlight that most algorithms exhibit a linear increase of execution time w.r.t. the evaluated criteria, which agree with the complexity analysis of the algorithms. Nevertheless, there is one exception: Partial Extensional Value-Based Surprise (comparing actual values in the query results with expected values in a user model), even being quite fast, do not achieve the theoretical lineal increase with respect to the result size. We attribute the variation to the probability of hitting an expected value when the query result size is larger, which results in extra CPU time for computing surprise.

In absolute terms, of course, algorithms only based on syntactic aspects (cube schema) run much faster than those taking advantage of instances, history or user models. In particular, the computation of detailed area can be quite time consuming (almost 100 seconds for 10 million facts). Of course, same-level comparisons, or belief-based are faster than detailed comparisons, due to the sheer size of the latter.

9.2. Goal and Research Questions

The goal of the study has been to identify whether there are significant influences by particular interestingness facets, as well as patterns of behavior related to these facets, when users interact with cubes and cube queries.

To solidify this goal, our user study was based on the following research questions:

RQ1. Can we rank the interestingness facets in terms of significance to the overall interestingness of a cube query? Is there any interestingness facet that dominates the determination of the overall interestingness of a cube query?

RQ2. As a session progresses, does the significance of the interestingness facets change overtime?

RQ3. Do participants demonstrate a consistent behavior with respect to the ranking of their interestingness facets?

RQ4. Are there patterns of behavior concerning interestingness facets? Can we form clusters of users based on their preferences?

To answer these questions, we constructed and executed the experimental protocol that is detailed in the sequel.

9.3. Experimental Protocol

The user study we conducted was based on asking participants to assess how interesting a query result appeared to them, without giving them any details on how the query ranked in terms of the four interestingness facets, namely Relevance, Novelty, Peculiarity and Surprise.

Material. We created a set of cube querying sessions. For all the cube querying sessions, we have used the Adult dataset which is a census dataset that has 8 dimensions (*Age, Native Country, Education, Occupation, Marital Status, Work Class, Gender and Race*) and a single measure, *Work Hours Per Week.*

Each session was constructed as a PowerPoint presentation that was given to the participants. The presentation started with a set of slides giving a description of the dataset structure and semantics. Then, the participants were given the goal of finding out which are the categories of working people with the significantly higher and lower average working hours per week, depending on a set of data dimensions of the data set, like education, occupation, work class, age, and in the context of this task, we were giving them pre-computed queries along with their results to help them determine the answer to the task.

The following parts of the presentation given to the participants included a warm-up slide and 3 slides of 4 queries. The single warm-up slide contained query results that give a broad description of how work_hours are related to the various dimensions that we use in that specific querying session. This served as a contextualization of the participants in the data of in the data set. To make the participants pay attention to these data, we also asked them to write a short memo of what their original impression was on who works more.

Subsequently, the report contained 3 slides and, in each of these 3 slides, 4 queries were presented. The queries of each slide of the session were expressed in natural language and were presented along with their resulting tuples, without any additional information about interestingness facets or values (Fig 9).

We asked the participants to rank the queries that they faced in each slide on a scale of 1 to 4, with 1 being the most interesting and 4 being the least interesting. The ranking was based on the users' personal criteria, with respect to the specific target that we gave them, i.e., to find out which are the characteristics of people that work the most and the least in a week time period. To avoid any bias, we never referenced any of the interestingness facets to the participants. Thus, they were fully ignorant of the overall goal of the study and the underlying assessment that we were making. To make the users pay more attention to the data, we also asked them to write a short memo per slide on their rationale.

The trick, unknown to the participants was that each of the 4 queries maximized the value of an interestingness facet. *Thus, by ranking queries, the participants also ranked interestingness facets without knowing.* Practically, in each slide we had 4 queries-representatives of the interestingness facets. In simpler words, in each slide we presented to the user a highly Relevant, a highly Novel, a highly Peculiar and a highly Surprising query at

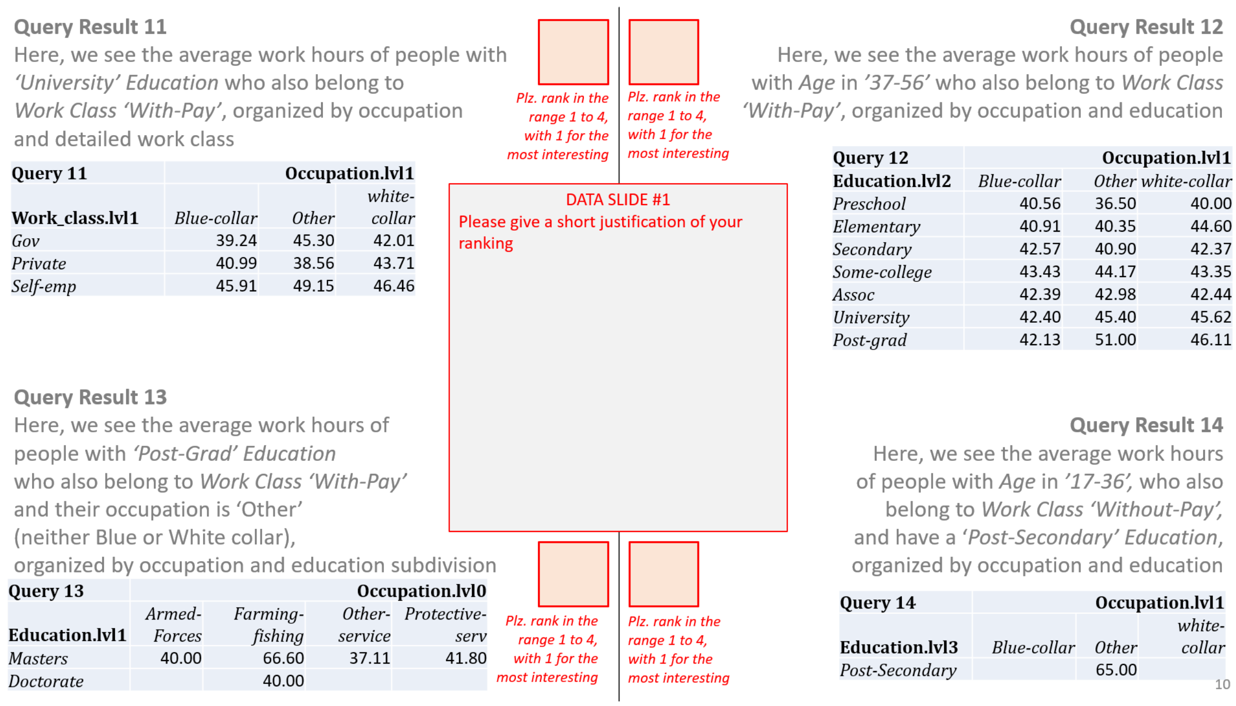

Figure 9: A slide as presented to the participants

a random order. The score of each interestingness facet for each query was the result of an algorithm that we selected to run as the interestingness-facet-representative. Specifically, for each query, we selected to run *Partial Detailed Extensional Relevance*, *Partial Detailed Extensional Novelty*, *Partial Detailed Jaccard-Based Extensional Peculiarity* and *Partial Extensional Value Based Surprise*.

The final step of the process was that once they had worked with the presentation and made their decisions and comments, the participants had to record them in a Google Form whose link was also given to them, along with the instructions and the presentation. The participants were given the fairly large time interval of an entire day to conduct the experiment. The results were collected from the Google Form's back-stage spreadsheet for further processing.

Remark. A point worth mentioning here, is that the algorithms used "under the hood" of each slide are all partial (i.e., return a score) and extensional (i.e., based on the query results). We did not require any input from the participants, other than the report ensuring that they paid attention to the warm-up queries. The evaluation of novelty, relevance and peculiarity are all history-based, i.e., based on the queries of the previous slides that the participants had seen in the slides prior to the one evaluating, at any given slide. Concerning surprise, we exploited the warm-up queries to assess surprise. Specifically, we used the fact that the warm-up queries were (intentionally) expressed at higher level of the dimension hierarchies, to extrapolate expected values for the subsequent slides; the difference between expected and actual value quantified the amount of surprise.

Population. The participants of the experiment were 25, and specifically, 7 PhD and 11 MSc students, all trained in the concepts of cubes, dimensions and business intelligence, as well as 7 undergraduate students with significantly less exposure to BI concepts. All participants were volunteers from France and Greece. Interestingly enough, in the subsequent study of the collected data, we did not observe any particular differentiation between the educational levels. The simplicity of the multidimensional model, as well as the textual description of queries have obviously made the data analysis work smooth. Therefore, we report all the results collectively, independently of the educational level.

We would also like to point out that no assumption is done about user beliefs, in particular, with respect to students background or education level. All students discovered the data set within the experiment. They all had the same set of warming queries, that allows to model both the query history and a simple representation of user beliefs.

Anti-Bias and Integrity measures. To preserve the integrity of the study, several measures were taken.

- To randomize the experiment, we grouped the participants at random in one of the two sessions that we had previously constructed.
- To forestall any technical difficulties impeding any student whatsoever, all the query expressions were presented in natural language.
- To involve participants in the data set, we asked them to construct short memos per slide, which we later checked. We found no frivolous behavior from the part of the participants.
- To avoid any ordering bias, we shuffled the position of the queries in the slides.
- By asking each user to rank 12 queries overall, we addressed the issue of volume, too.

9.4. Research Question: significance of individual interestingness facets

After collecting the participants' responses, the analysis phase begun. The first task to address was to answer the question on the significance of individual interestingness facets to the overall interestingness of an individual query.

The input to the analysis was a matrix where for every participant, for every slide and for every query in the slide, there was a rank between 1 to 4. We had instructed participants to avoid ties, and indeed we had a clean vote from this respect. To synthesize the results, we resorted in a Borda scoring of the ranks. A Borda count [57] is a simple process for synthesizing ranking preferences. The idea is that you have N candidates, and voters rank them. Then, for every rank, you give a score which is produced by the formula *score* = N+1 - *rank*. For example, with 4 candidates to be voted per slide, the query with rank 1 gets 4 points, whereas the query with rank 4, gets 1 point. Then, the scores are simply summed up per candidate.

Here, the candidates are the interestingness facets, hidden behind the queries that are voted. Once we added all the scores, the results were demonstrating a layering of preferences.

Int. Dim	Borda score
Peculiarity	151
Novelty	183
Relevance	203
Surprise	213

Table 6: Borda score for the different facets of interestingness, after composing individual rankings in our user study

The results suggest that no particular interestingness facet drives the overall interest single-handedly. However, there are differences, with Surprise and Relevance being most significant, Novelty coming third at a distance, and Peculiarity being the least significant.

Surprise came first and Relevance second, with close distance to one another. Surprise was the facet that was ranked (i) first most times than any other facet, and, (ii) last, less than any other facet. Closely following Surprise, Relevance ranked typically first or second, and rarely third or fourth. So, this practically instructs us that if recommending queries to users a-priori, or assessing them a-posteriori, surprise and relevance seem stable choices.

Int. Dim	1	2	3	4
Peculiarity	7	14	27	27
Novelty	20	16	16	23
Relevance	19	28	15	13
Surprise	29	17	17	12

Table 7: Occurrence per rank, for each of the interestingness facets (position 1 is the most appreciated, position 4 the less)

On the other hand, Novelty is practically equally distributed in all ranks (as we will see, not equally over time though). We believe that this is a result closely related to the setup of the study: users were given a specific task, as well as a contextualization warm-up, meaning that there was not a phase of exploring without any particular focus in search for interesting pieces of information. But, what we learn on the other hand, is that in these occasions, where a clear focus has been set early on, novelty is not so important as we originally expected. Finally, peculiarity went particularly low in terms of preferences. Again, we relate this to the previous discussion on novelty: digressions from the central task are not particularly appreciated once the focus has been set.

Interestingly, a statistical analysis of correlation between the measurements found a couple of interesting anti-correlations. We measured the pairwise Pearson correlation for all the four interestingness facets. Surprise is anticorrelated with Novelty, with a score of -0.62 and Relevance is anticorrelated with Peculiarity with a score of -0.50. The effect for the rest of the pairs was weaker.

9.5. Research Question: does interest change over time?

Another question we asked was if participants appreciated the interestingness facets differently as time passes. To the extent that we have a set of slides ordered over time, we assess the effect of time via the position of the respective slides. A caveat here, is of course, that the length of our sessions is short, therefore, the results should not be arbitrarily generalized. In Figure 10, we depict the average rank per slide, for each of the interestingness facets. Beware these are ranks, not scores: so, in the Figure, the higher the bar, the less appreciated a facet is.

Unsurprisingly, *Surprise and Relevance seem rather unaffected from the position of the slide, although as time passes, surprise becomes slightly less of importance. Peculiarity also seems to lose interest as time passes, especially between slides 1 and 2. What is most revealing, though, is the sharp decline of Novelty in rank over time.* At the beginning, Novelty is not that interesting, ranking top (thus, least appreciated) among all interestingness facets. From slide 2, though, Novelty starts being more appreciated by the participants. Novelty was probably considered out-of-scope at the beginning, right after contextualization had taken place, but later, it picked up in stature.

9.6. Research Question: Do participants demonstrate a consistent behavior with respect to the ranking of their interestingness facets?

The next research question concerned the existence of a constant behavior of the participants with respect to how they ranked the different interestingness facets. We will broadly use the term *consistency* to refer to the tendency of a participant to place the same rank to the same interestingness facet in different slides.

Ranking Data and Comparisons. Before proceeding with the definitions of the metrics used to quantify consistency, let us briefly summarize the available data. Remember that each participant gives 4 rankings for each slide, in the range 1 - 4, one per interestingness facet. Since there are 3 such slides, eventually each participant comes with a vector of 12 rankings.

Moreover, there are 3 comparisons to be made: (i) slides 1 and 2, (ii) slides 1 and 3, and, (iii) slides 2 and 3. This is important as we have a vector of 12 comparisons for the rankings given by the participants: 4 comparisons (one per facet) for each of the cases (i) - (iii). We call this vector the *comparison vector*.

Definitions. To address this question, we resort to two different metrics, point-based and score-based consistency. To be able to define them, we define the following metric:

Average Point-based Consistency is the total number of comparisons where the participant gave the same rank to the same interestingness facet in the two compared slides, normalized by the number of comparisons.

To define score-based consistency, we need a couple of auxiliary metrics:

Score inconsistency is the absolute difference of two rankings of the same interestingness facet in a comparison - practically the absolute value of a cell in the comparison vector.

Normalized comparison score-based inconsistency is the normalized sum of the 4 cells of the comparison vector that pertain to a comparison between two specific slides. We sum the inconsistencies for the four different measures and normalize by 8 which is the maximum amount of inconsistency for the 4 rankings within a slide. Thus, we have 3 normalized inconsistency scores, one per case (i) - (iii). *Normalized comparison score-based consistency* is defined as its complement: 1 - normalized comparison score-based inconsistency.

Then, *Average score-based consistency* is the average of the three normalized comparison score-based consistencies for cases (i) - (iii).

Intuition. Practically speaking, the two metrics handle consistency from two different points of view.

Point consistency is a “Boolean”-based metric: if the participant gave the same ranking to the same facet, it raises a true flag, otherwise a false one. Practically, we count how many times there was a coincidence of rankings. We normalize the count

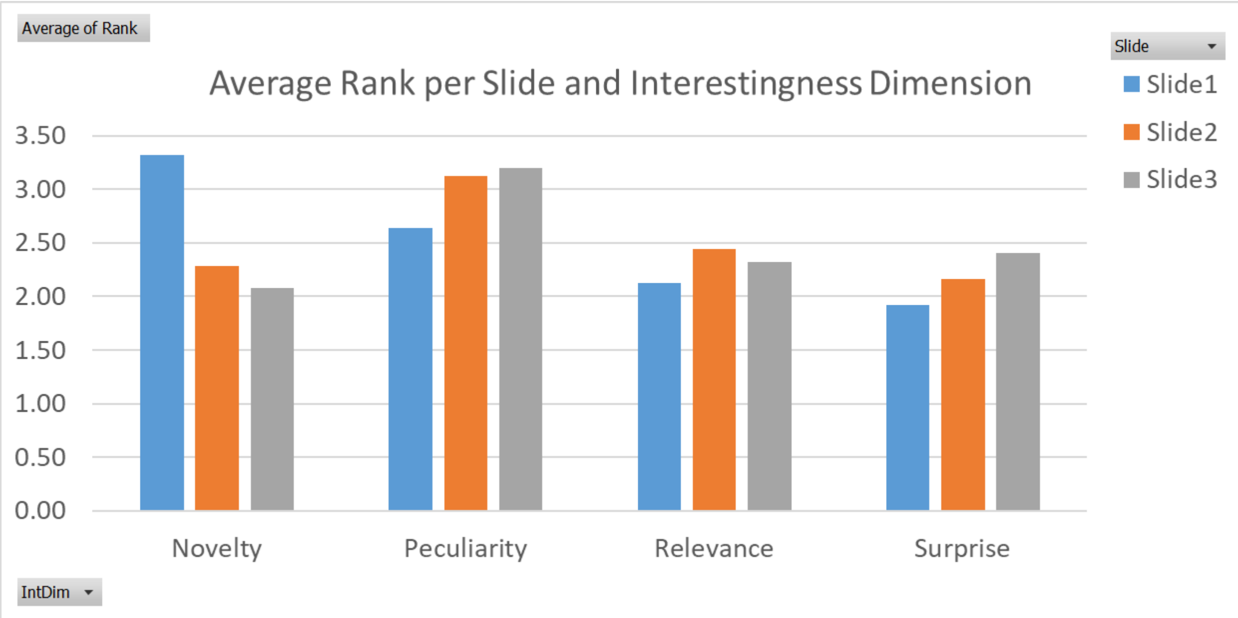

Figure 10: Average Rank of interestingness facets per Slide

of coincidence occasions by the number of comparisons (here: 12) and we get a score within 0 and 1 (1 meaning the participant gave always the same rankings).

Score-based consistency goes one step further, as the value of the rank is used. Assume you compare novelty in slide 1 with novelty in slide 3. If the participant gave a rank of 1 to the former and a rank of 4 to the latter, this is more inconsistent compared to the case where the participant gave 2 and 3, respectively.

Evaluation. When it comes to evaluating the consistency of individual users the situation is depicted in Figure 11.

Both scatterplots demonstrate a similar behavior of points randomly spread in a band of values. The two plots provide a different evaluation of the situation however. When we assess consistency in a strict, Boolean way, the participants find themselves spread in a band between 0.1 and 0.4 (with the exception of a single user with a consistency of exactly 1). This is an indicator that more often than not, the rankings of the same interestingness facet are different.

At the same time, the score-based consistency tells us that they are not entirely different after all: the band of points lies between 0.2 and 0.7, with 13 participants below 0.5 and 12 participants above 0.5. In other words, although they may not coincide exactly, the rankings used are not that far.

In summary: the participants did not exhibit a strong bias towards a particular ranking of the interestingness facets, although the rankings are not completely arbitrary.

9.7. Research Question: Are there any clusters of participant behavior?

Another test we applied was to attempt and cluster participants on the basis of their behavior. We employed two methods of clustering: (a) k-Means and (b) Louvain clustering on the grounds of two versions of the measurements: (i) the original 12 rankings given by each user, and, (ii) the average value of each interestingness facet per user (thus, with a vector of 4 values per user instead of 12, in an attempt to reduce dimensionality).

The results are quite indicative on the absence of clusters. All clustering methods returned low Silhouette coefficients (0.252 for the k-means clustering of the original and 0.323 for the k-means clustering of the averaged data), and their Silhouette plots indicate that clusters are not very cohesive.

9.8. Threats to validity

In this section, we discuss threats to the validity of our study.

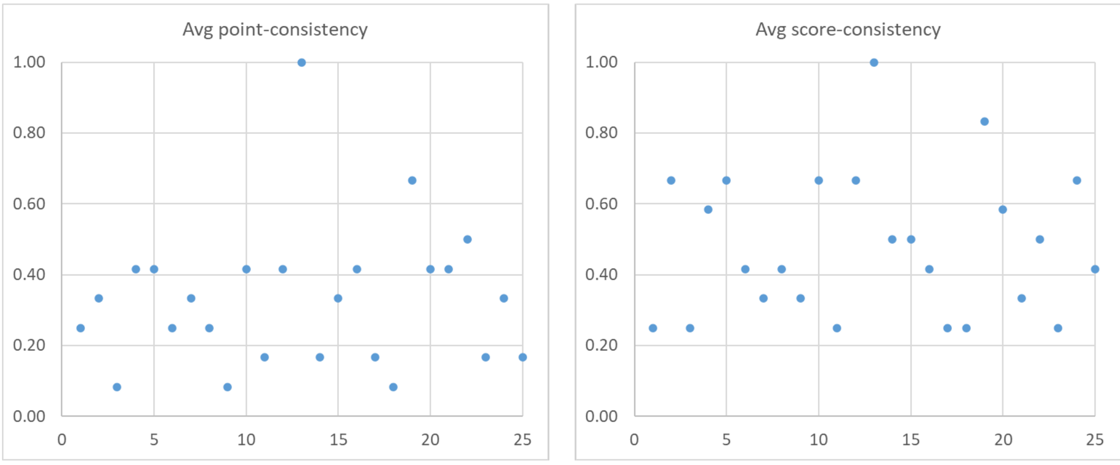

Figure 11: Consistency scatterplots: The horizontal axis refers to the participant id (thus, each point is a different participant), and the vertical axis to the average point-based and score-based consistency of the participant

Construct Validity. We have taken care to check that the answers given by the participants were valid. We had no violations of the scoring constraints and the memos returned by the users demonstrated a valid level of engagement to the study. We made sure that the true purpose of the study was not revealed to the participants. Therefore, when they ranked queries, they had no idea about interestingness facets that could affect their scoring. Based on the above, we can state that the ranking of queries in terms of overall interestingness was valid.

At the same time, a potential threat might appear from the hiding of the facets behind queries. With the exception of relevance, we made sure that each query that secretly represented a facet was either on very high values of the respective metric, or with significant difference for this metric against the others. For relevance this was not possible, as we explore a fairly well “fenced” area of the multi-dimensional space. However, the queries with high relevance were very low in all other metrics, which means that if selected, only relevance could be the reason for selecting them. Based on the above, we can claim that the rank of a query can be validly mapped to a rank of the respective interestingness facet.

Internal validity. Internal validity refers to cause-and-effect relationships. We do not measure any interventions, so we do not search for hidden variables that can override the effect of an intervention, as typical internal validity checks should do.

We attribute the significance assessment of the different facets to the scope of the study (see also external validity). Other factors that could influence the behavior of the participants towards specific facets might be applicable, although we cannot think of any. The same applies for the behavior of Novelty over time.

The effect of time has to be considered under the prism of the session length of the study. Although we see a clear behavior for the time length that we observe, it is possible that longer sessions might demonstrate different characteristics.

The lack of extreme consistency and clusters of participants is probably a good rather than a bad sign towards the integrity of the study: the population demonstrated variability in behaviors which means there is no threat of bias of some sort. Apparently, although some difference in significance exists, overall, all facets play a role. Interestingly, a potential threat to the internal validity of the study involves other facets, that we have not thought of, that might coexist with the ones we study.

External validity. What is the scope of the study? How generalizable are our findings? The existence of 25 participants is not overwhelming, but still adequate enough to allow the drawing of conclusions.

Is the use of students as participants a problem? We believe we satisfy the most important properties of [58] that supports accepting the validity of tests with students: students were trained in the context of the study, and were adequately well equipped to perform the study, and, at the same time, both our positive and negative results are novel and a starting point for further research.

Concerning the scope, we have to be clear that the study did not have an exploratory nature: the participants were given query descriptions and results around a very specific topic. Therefore, our results are restricted in the case of focused assessment of information around a specific topic and task.

10. Conclusions

In this paper, we have addressed the problem of assessing the interestingness of a cube query in the context of a hierarchical multidimensional database with cubes and level hierarchies. We have performed an extensive survey of the related work, both in the area of computer science and in the area of the study of human behavior. We have focused the discussion on 4 interestingness facets, specifically, relevance, surprise, novelty, and peculiarity. For these facets of interestingness, we have also proposed specific measures and algorithms for assessing them in a quantitative fashion. We take care to discriminate between result-based algorithms, after the query has been executed and syntax-based algorithms, before the query is executed. Finally, we have conducted a user study to determine the significance, as well as the evolution over time, of the different interestingness facets.

Future work can continue in different roads. First, although the facets of interestingness that we discuss in this paper provided a principled and well-founded setup of how interestingness can be handled, one can only expect that a deeper study – esp., of the fundamentals, in the area of human behavior– can reveal more perspectives to the essence of interestingness. Once again, we would like to clarify that we make no completeness claim on the possible facets of interestingness. Several other facets are open to exploration:

- *Conciseness* in one such facet, although defined at a meta-level: the shorter the query description and the query result are, the more easy to comprehend them. The effect of conciseness and aggregation level (which can go hand-in-hand with the size of the query result) has not been studied either.
- Another notable facet concerns the *expression* aspect, in which data are contextualized with respect to the medium used to expressed it (and not displayed) – e.g., a cube can be described by the set of cells, or by a query, or by a visualization, etc.

Moreover, even for the presented facets, it is clear that the presented algorithms are only a first attack to the problem. More algorithms and metrics are possible for the aforementioned facets. We have been particularly interested in syntactic checks in this paper, as they allow the prediction of the interestingness of a query without actually executing it. More value-based algorithms, however, are certainly possible. The role of time (but also space, and in general, dimensional context) is also worth pursuing: what is interesting now for an analyst, might be indifferent some time later. Aging, decay factors can be introduced in the assessment of interestingness when queries are compared to the history of the user, or, other users as a matter of fact.

Data quality and non-multidimensional issues have also been absent from this paper. We assume an OLAP-style structuring of the data in neat star-schema like structures with the joins, drill-downs, and value translations being straightforward. OLAP-style data also mean that the data re clean and conforming to the dimension bus. The presence of arbitrary schema structures and arbitrary values in the data can also be a part of future work in the assessment of query interestingness.

Personal profiles, crowd-wisdom and log mining can be employed to best model user beliefs. We refer the interested reader to [59, 21] for a starting point, but of course, the problem of belief estimation is a large research territory that can fit gracefully with our taxonomical framework.

The scope of our user study has not studied highly interactive user sessions. The extent that interactivity affects the assessment of interestingness is yet another unexplored territory for future research.

Acknowledgment. We would like to thank the reviewers of this paper, as well as of its predecessor [11] for fruitful comments that made the paper more clear and easy to comprehend.

References

[1] Y. Wang, Z. Sun, H. Zhang, W. Cui, K. Xu, X. Ma, D. Zhang, Datashot: Automatic generation of fact sheets from tabular data, IEEE Trans. Vis. Comput. Graph. 26 (1) (2020) 895–905.

[2] O. B. El, T. Milo, A. Somech, Automatically generating data exploration sessions using deep reinforcement learning, in: Proceedings of SIGMOD, Portland, OR, USA, 2020, pp. 1527–1537.

[3] T. Milo, A. Somech, Automating exploratory data analysis via machine learning: An overview, in: SIGMOD, 2020.

[4] B. Youngmann, S. Amer-Yahia, A. Personnaz, Guided exploration of data summaries, Proc. VLDB Endow. 15 (9) (2022) 1798–1807.

[5] L. Geng, H. J. Hamilton, Interestingness measures for data mining: A survey, ACM Comput. Surv. 38 (3) (2006) 9.

[6] M. Kaminskas, D. Bridge, Diversity, serendipity, novelty, and coverage: A survey and empirical analysis of beyond-accuracy objectives in recommender systems, TiiS 7 (1) (2017) 2:1–2:42.

[7] M. Eirinaki, S. Abraham, N. Polyzotis, N. Shaikh, QueRIE: Collaborative database exploration, IEEE Trans. Knowl. Data Eng. 26 (7) (2014) 1778–1790.

[8] J. Aligon, E. Gallinucci, M. Golfarelli, P. Marcel, S. Rizzi, A collaborative filtering approach for recommending OLAP sessions, Decision Support Systems 69 (2015) 20–30.

[9] P. Marcel, V. Peralta, P. Vassiliadis, A framework for learning cell interestingness from cube explorations, in: 23rd European Conference on the Advances in Databases and Information Systems (ADBIS 2019), Bled, Slovenia, September 8-11, 2019, Vol. 11695 of Lecture Notes in Computer Science, Springer, 2019, pp. 425–440. doi:10.1007/978-3-030-28730-6_26.

[10] D. Gkitsakis, S. Kaloudis, E. Mouselli, V. Peralta, P. Marcel, P. Vassiliadis, Cube interestingness: Novelty, relevance, peculiarity and surprise, CoRR abs/2212.03294 (2022). arXiv:2212.03294, doi:10.48550/arXiv.2212.03294.
URL `https://doi.org/10.48550/arXiv.2212.03294`

[11] D. Gkitsakis, S. Kaloudis, E. Mouselli, V. Peralta, P. Marcel, P. Vassiliadis, Assessment methods for the interestingness of cube queries, in: Proceedings of the 25th International Workshop on Design, Optimization, Languages and Analytical Processing of Big Data (DOLAP), Ioannina, Greece, March 28, 2023, 2023, pp. 13–22.

[12] P. J. Silvia, Interest: The curious emotion, Current Directions in Psychological Science 17 (1) (2008) 57–60.
URL `http://www.jstor.org/stable/20183249`

[13] J. Rounds, R. Su, The nature and power of interests, Current Directions in Psychological Science 23 (2) (2014) 98–103. doi:10.1177/0963721414522812.

[14] R. Su, G. Stoll, J. Rounds, The nature of interests: Toward a unifying theory of trait-state interest dynamics, in: C. Nye, J. Rounds (Eds.), Vocational Interests in the Workplace: Rethinking Behavior at Work, Taylor and Francis, 2019, p. 11 – 38. doi:10.4324/9781315678924-2.

[15] J. Litman, Curiosity and the pleasures of learning: Wanting and liking new information, Cognition and Emotion 19 (6) (2005) 793–814.

[16] G. Loewenstein, The psychology of curiosity: a review and reinterpretation, Psychological Bulletin 116 (1) (1994) 75–98.

[17] C. Kidd, B. Y. Hayden, The psychology and neuroscience of curiosity, Neuron 88 (2015) 449–460.

[18] J. Förster, J. Marguc, M. Gillebaart, Novelty categorization theory, Social and Personality Psychology Compass 4 (9) (2010) 736 – 755.

[19] R. Reisenzein, W.-U. Meyer, M. Niepel, Surprise, in: V. S. R. editor) (Ed.), Encyclopedia of Human Behavior, 2nd Edition, Elsevier, 2012.

[20] D.E. Berlyne, A theory of human curiosity, British Journal of Psychology 45 (3) (1954) 180 – 191.

[21] T. D. Bie, Subjective interestingness in exploratory data mining, in: Proceedings of IDA, 2013, pp. 19–31.

[22] Y. Yao, Y. Chen, X. D. Yang, A measurement-theoretic foundation of rule interestingness evaluation, in: T. Young Lin, S. Ohsuga, C.-J. Liau, X. Hu (Eds.), Foundations and Novel Approaches in Data Mining, Springer Berlin Heidelberg, 2006, pp. 41–59. doi:10.1007/11539827_3.

[23] C. C. Aggarwal, Data Mining - The Textbook, Springer, 2015.

[24] J. L. Herlocker, J. A. Konstan, L. G. Terveen, J. Riedl, Evaluating collaborative filtering recommender systems, ACM Trans. Inf. Syst. 22 (1) (2004) 5–53.

[25] A. Gunawardana, G. Shani, A survey of accuracy evaluation metrics of recommendation tasks, Journal of Machine Learning Research 10 (2009) 2935–2962.

[26] E. Y. Lai, Z. Zolaktaf, M. Milani, O. AlOmeir, J. Cao, R. Pottinger, Workload-aware query recommendation using deep learning, in: Proceedings 26th International Conference on Extending Database Technology, EDBT 2023, Ioannina, Greece, March 28-31, 2023, 2023, pp. 53–65.

[27] S. Sarawagi, R. Agrawal, N. Megiddo, Discovery-driven exploration of OLAP data cubes, in: EDBT, 1998, pp. 168–182.

[28] S. Sarawagi, User-adaptive exploration of multidimensional data, in: Proceedings of VLDB, 2000, pp. 307–316.

[29] D. Gkesoulis, P. Vassiliadis, P. Manousis, Cinecubes: Aiding data workers gain insights from OLAP queries, Inf. Syst. 53 (2015) 60–86.

[30] B. Tang, S. Han, M. L. Yiu, R. Ding, D. Zhang, Extracting top-k insights from multi-dimensional data, in: SIGMOD Conference, ACM, 2017, pp. 1509–1524.

[31] E. Zgraggen, Z. Zhao, R. C. Zeleznik, T. Kraska, Investigating the effect of the multiple comparisons problem in visual analysis, in: Proceedings of CHI, Montreal, QC, Canada, 2018, p. 479.

[32] R. Ding, S. Han, Y. Xu, H. Zhang, D. Zhang, QuickInsights: Quick and automatic discovery of insights from multi-dimensional data, in: Proceedings of SIGMOD, Amsterdam, The Netherlands, 2019, pp. 317–332. doi:10.1145/3299869.3314037.

[33] M. Francia, P. Marcel, V. Peralta, S. Rizzi, Enhancing cubes with models to describe multidimensional data, Inf. Syst. Frontiers 24 (1) (2022) 31–48.

[34] F. Abuzaid, P. Kraft, S. Suri, E. Gan, E. Xu, A. Shenoy, A. Ananthanarayan, J. Sheu, E. Meijer, X. Wu, J. F. Naughton, P. Bailis, M. Zaharia, DIFF: a relational interface for large-scale data explanation, VLDB J. 30 (1) (2021) 45–70.

[35] A. Chanson, N. Labroche, P. Marcel, S. Rizzi, V. T'kindt, Automatic generation of comparison notebooks for interactive data exploration, in: EDBT, OpenProceedings.org, 2022, pp. 2:274–2:284.

[36] S. Sarawagi, Explaining differences in multidimensional aggregates, in: Proceedings of VLDB, 1999, pp. 42–53.

[37] G. Sathe, S. Sarawagi, Intelligent rollups in multidimensional OLAP data, in: Proceedings of VLDB, 2001, pp.

531–540.

[38] M. Klemettinen, H. Mannila, H. Toivonen, Interactive exploration of interesting findings in the telecommunication network alarm sequence analyzer (tasa), Information and Software Technology 41 (9) (1999) 557 – 567.

[39] N. Kumar, A. Gangopadhyay, S. Bapna, G. Karabatis, Z. Chen, Measuring interestingness of discovered skewed patterns in data cubes, Decision Support Systems 46 (1) (2008) 429 – 439.

[40] C. C. Fabris, A. A. Freitas, Incorporating deviation-detection functionality into the OLAP paradigm, in: SBBD, 2001, pp. 274–285.

[41] Z. Zhao, L. D. Stefani, E. Zgraggen, C. Binnig, E. Upfal, T. Kraska, Controlling false discoveries during interactive data exploration, in: SIGMOD, 2017, pp. 527–540.

[42] B. Salimi, J. Gehrke, D. Suciu, Bias in OLAP queries: Detection, explanation, and removal, in: SIGMOD, 2018, pp. 1021–1035.

[43] P. Ma, R. Ding, S. Han, D. Zhang, MetaInsight: Automatic discovery of structured knowledge for exploratory data analysis, in: Proceedings of SIGMOD, 2021, pp. 1262–1274. doi:10.1145/3448016.3457267.

[44] A. Personnaz, S. Amer-Yahia, L. Berti-Équille, M. Fabricius, S. Subramanian, DORA THE EXPLORER: exploring very large data with interactive deep reinforcement learning, in: CIKM, 2021.

[45] A. Chanson, B. Crulis, K. Drushku, N. Labroche, P. Marcel, Profiling user belief in BI exploration for measuring subjective interestingness, in: DOLAP, Vol. 2324 of CEUR Workshop Proceedings, CEUR-WS.org, 2019.

[46] S. Sintos, P. K. Agarwal, J. Yang, Selecting data to clean for fact checking: Minimizing uncertainty vs. maximizing surprise, Proc. VLDB Endow. 12 (13) (2019) 2408–2421.

[47] M. Djedaini, N. Labroche, P. Marcel, V. Peralta, Detecting user focus in OLAP analyses, in: ADBIS, 2017, pp. 105–119.

[48] M. Djedaini, K. Drushku, N. Labroche, P. Marcel, V. Peralta, W. Verdeau, Automatic assessment of interactive OLAP explorations, Inf. Syst. 82 (2019) 148–163.

[49] P. Vassiliadis, A Cube Algebra with Comparative Operations: Containment, Overlap, Distance and Usability, CoRR abs/2203.09390 (2022). doi:10.48550/arXiv.2203.09390.
URL `https://doi.org/10.48550/arXiv.2203.09390`

[50] P. Vassiliadis, P. Marcel, S. Rizzi, Beyond roll-up's and drill-down's: An intentional analytics model to reinvent OLAP, Information Systems 85 (2019) 68–91. doi:10.1016/j.is.2019.03.011.

[51] A. Maté, J. Trujillo, J. Mylopoulos, Specification and derivation of key performance indicators for business analytics: A semantic approach, Data Knowl. Eng. 108 (2017) 30–49. doi:10.1016/j.datak.2016.12.004.

[52] P. Vassiliadis, P. Marcel, The road to highlights is paved with good intentions: Envisioning a paradigm shift in OLAP modeling, in: Proceedings of DOLAP, 2018.

[53] E. Baikousi, G. Rogkakos, P. Vassiliadis, Similarity measures for multidimensional data, in: Proceedings of the 27th International Conference on Data Engineering, ICDE 2011, April 11-16, 2011, Hannover, Germany, IEEE Computer Society, 2011, pp. 171–182. doi:10.1109/ICDE.2011.5767869.

[54] M. Golfarelli, E. Turricchia, A characterization of hierarchical computable distance functions for data warehouse systems, Decis. Support Syst. 62 (2014) 144–157. doi:10.1016/j.dss.2014.03.011.

[55] J. Aligon, M. Golfarelli, P. Marcel, S. Rizzi, E. Turricchia, Similarity measures for OLAP sessions, Knowl. And Inf. Syst. 39 (2) (2014) 463–489.

[56] A. Giacometti, P. Marcel, E. Negre, Recommending multidimensional queries, in: DaWaK, Vol. 5691 of Lecture Notes in Computer Science, Springer, 2009, pp. 453–466.

[57] S. El-Helaly, The Mathematics of Voting and Apportionment: An Introduction, Springer International Publishing, 2019.

[58] W. F. Tichy, Hints for reviewing empirical work in software engineering, Empir. Softw. Eng. 5 (4) (2000) 309–312.

[59] T. D. Bie, An information theoretic framework for data mining, in: Proceedings of SIGKDD, 2011, pp. 564–572.