Describing and Assessing Cubes Through Intentional Analytics

Matteo Francia
DISI - University of Bologna
Bologna, Italy
m.francia@unibo.it

Matteo Golfarelli
DISI - University of Bologna
Bologna, Italy
matteo.golfarelli@unibo.it

Stefano Rizzi
DISI - University of Bologna
Bologna, Italy
stefano.rizzi@unibo.it

ABSTRACT

The Intentional Analytics Model (IAM) has been envisioned as a way to tightly couple OLAP and analytics by (i) letting users explore multidimensional cubes stating their intentions, and (ii) returning multidimensional data coupled with knowledge insights in the form of annotations of subsets of data. Goal of this demonstration is to showcase the IAM approach using a notebook where the user can create a data exploration session by writing describe and assess statements, whose results are displayed by combining tabular data and charts so as to bring the highlights discovered to the user's attention. The demonstration plan will show the effectiveness of the IAM approach in supporting data exploration and analysis and its added value as compared to a traditional OLAP session by proposing two scenarios with guided interaction and letting users run custom sessions.

1 INTRODUCTION

In the context of *exploratory data analysis*, it has recently become evident that the OLAP paradigm alone is no longer sufficient to keep the pace with the increasing needs of new-generation decision makers when exploring multidimensional data cubes. Indeed, the enormous success of machine learning techniques has consistently shifted the interest of users towards more sophisticated analytical applications [12]. In this direction, the *Intentional Analytics Model* (IAM) has been envisioned as a way to tightly couple OLAP and analytics [16]. The IAM approach relies on two major cornerstones: (i) the user explores a multidimensional cube by expressing her analysis *intentions* rather than by explicitly stating what data she needs, and (ii) in return she receives an *enhanced cube*, i.e., multidimensional data coupled with knowledge insights in the form of annotations of subsets of data. As to (i), five intentional operators were envisioned, namely, describe, assess, explain, predict, and suggest. As to (ii), the insights are ranked based on their estimated relevance for the user, and the one deemed most relevant is shown.

In our previous work, we have focused on the first two operators: describe, whose goal is to describe one or more cube measures, possibly focused on one or more level members [5], and assess, which evaluates the performance of a cube measure with reference to some benchmark [3]. For both operators we have proposed a syntax, a semantics, and some optimization strategies.

Contributions and Outline. Goal of this demonstration is to showcase the IAM approach using a notebook where the user can create a data exploration session by writing describe and assess statements, whose results are displayed by combining tabular data and charts so as to bring the highlights discovered to the user's attention. The underlying DBMS is Oracle, coupled with a custom multidimensional engine. The demonstration is based on a cube storing COVID-19 data and hinges on two scenarios: in the first one, the user will get an overview of the epidemic trend in Europe; in the second one, (s)he will focus on the deaths in Europe.

The paper outline is as follows. After introducing the two operators with their syntax in Section 2, in Section 3 we present the implementation that supports the demo, while in 4 we describe the demonstration experience in terms of user scenarios.

Related Work. The idea of coupling data and analytical models was born in the 90's with inductive databases, where data were coupled with patterns meant as generalizations of the data [13], and is at the core of the IAM approach [16] on which this paper relies. Specifically, coupling the OLAP paradigm and data mining to create an approach where concise patterns are extracted from multidimensional data for user's evaluation, was the goal of some approaches commonly labeled as OLAM [9]. In this direction, several additional operators have been proposed over the years to complement the fundamental ones of OLAP (e.g., [6, 7, 14, 15]),

The IAM approach can be regarded as OLAM since, like the approaches mentioned above, it relies on mining techniques to enhance the cube resulting from an OLAP query. Its novelty over the previous approaches lies in (i) the adoption of a declarative syntax to hide the complexity of query specification; (ii) the use of multiple mining techniques, rather than a single one like in previous approaches, to give users a wider variety of insights; and (iii) the automatic selection of the most relevant insight.

We finally mention that, though some tools (e.g., Spotfire and Tableau) integrate OLAP and analytics capabilities in the same environment, none of them allows users to formulate queries at a higher level of abstraction than OLAP (as done in the IAM using intentions), nor they support the automated *out-of-the-box* enrichment of cubes with insights obtained by analytics (as done in the IAM through enhanced cubes).

2 THE IAM APPROACH

The IAM approach is sketched in Figure 1 and operates as follows:

(1) The user expresses an intention on a cube C_0 by writing either a describe or an assess statement; both types of statement express an aggregation and, possibly, a selection clause over C_0.
(2) The cube C derived from C_0 by applying the aggregation and selection expressed in the statement is retrieved.
(3) A set of models are computed over C; a *model* is a concise, information-rich knowledge artifact that gives an insight on the cube cells. The possible model types range from simple functions such as top-k to more elaborate techniques such as clustering, outliers, etc. A model is made of a set of components (e.g., a clustering model is made of a set of clusters).

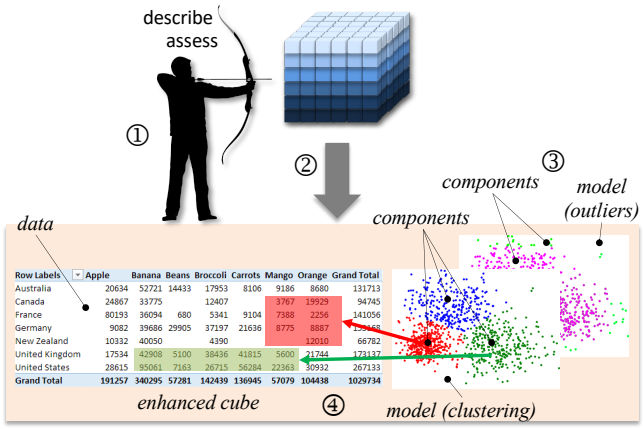

Figure 1: The IAM approach: the user expresses an intention and receives in return an enhanced cube

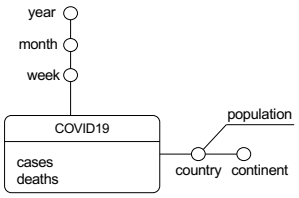

Figure 2: The COVID19 cube

(4) A measure of relevance is computed for each component and model. An enhanced cube including the data of C and the most relevant model is then displayed.

The IAM approach follows the direction of so-called *augmented analytics*, which employs enabling technologies —such as machine learning and AI— to assist users with data preparation and insight generation, aimed at automating many aspects of data science. This lets skilled users run their analysis process more efficiently and effectively on the one hand; on the other, it widens the user-base to people non skilled in model development and deployment. In fact, with IAM, having different models automatically computed and evaluated in terms of their relevance relieves skilled users from the time-wasting effort of trying different possibilities. Additionally, even users who just have basic OLAP skills and no knowledge of programming languages (e.g., Python) are enabled to run complex analytical models on their data.

Working example. Both the paper and the demonstration will use the COVID19 cube, whose conceptual model is depicted in Figure 2 using the DFM notation [8]. The cube stores the COVID-19 data by week and country over 4 years, and features two measures, cases and deaths[1].

2.1 The describe operator

The describe operator provides an answer to the user asking "show me my business" by describing one or more cube measures, possibly focused on one or more level members, at some given granularity. The resulting cube is enhanced by showing either the top/bottom-k cells, the skyline, the outliers, or clusters of cells.

[1]https://www.ecdc.europa.eu/en/publications-data/data-national-14-day-notification-rate-covid-19

Let C_0 be a cube, M be the set of its measures, L be the set of its levels. The syntax for describe is

$$\text{with } C_0 \text{ describe } m_1, \ldots, m_z \text{ [for } P \text{] by } l_1, \ldots, l_n$$

where $m_1, \ldots, m_z \in M$; P is an optional set of selection predicates, each of type $l = u$ (where $l \in L$ and u is a member of l); and $l_1, \ldots, l_n$ (with $l_i \in L$) is a group-by set of C_0. The cube C is derived from C_0 by applying the conjunction of the predicates in P and aggregating by $l_1, \ldots, l_n$. The description models computed over C are top-k, bottom-k, skyline (only if $z \geq 2$), outliers, and clustering. Each of the first four models returns two components (e.g., outlier and not outlier), while the last one returns one component for each cluster. The *size* of each model (e.g., the value of k for top-k and the number of clusters) is automatically determined as explained in [5].

To evaluate the relevance of a description model, a measure based on the *interestingness* of its components —expressed in terms of how novel, peculiar, and surprising they are expected to be— is used [5]. Intuitively, given a component c, its novelty is higher if c corresponds to a larger number of previously-unseen cells; the peculiarity of c is related to the measure deviation between the cells of c and the corresponding ones in the cube resulting from the previous intention; finally, c is surprising if it includes cells that have not been seen frequently.

The enhanced cube obtained from a describe statement includes C and the model having maximum interestingness. It is visualized by coupling a *table area*, which shows the cube cells using a pivot table, a *chart area*, which represents the cube cells through a suitable chart, and a *component area*, which shows the model components and their properties. A color code is used to emphasize, in the three areas, the cells belonging to each component.

Example 2.1. The statement

```
with COVID19 describe cases
for continent = 'Europe' by country, month
```

returns a (bidimensional) cube C showing the monthly cases for each European country. As shown in Figure 3, the model with maximum interestingness in this case is clustering; the chart chosen is a bubble chart. In the upper part of the screenshot, the intention written by the user and its system-generated expansion listing the models considered (via the using clause). Below, clockwise, the chart, the component area, and the table area. □

In [5] we measured the saving in user's effort when writing a describe intention over the one necessary to obtain the same result using plain SQL and Python. To this end we adopted the simple metric proposed by [10], where the ASCII character length is used as an approximation for the effort it takes to craft a query. We tried different intentions with increasing complexities; it turned out that the total formulation effort using SQL+Python is almost two orders of magnitude larger than using describe intentions. For instance, for the intention in Example 2.1, the effort for writing the intention is 70 characters, while the one for writing the necessary OLAP query and Python code would be 286 and 5200 characters, respectively.

2.2 The assess operator

The assess operator aims at comparing the behavior of the phenomenon represented by C_0 to a benchmark and judging, through a *labeling*, the outcome of the comparison. The resulting cube is enhanced by showing the labeling deemed the most relevant.

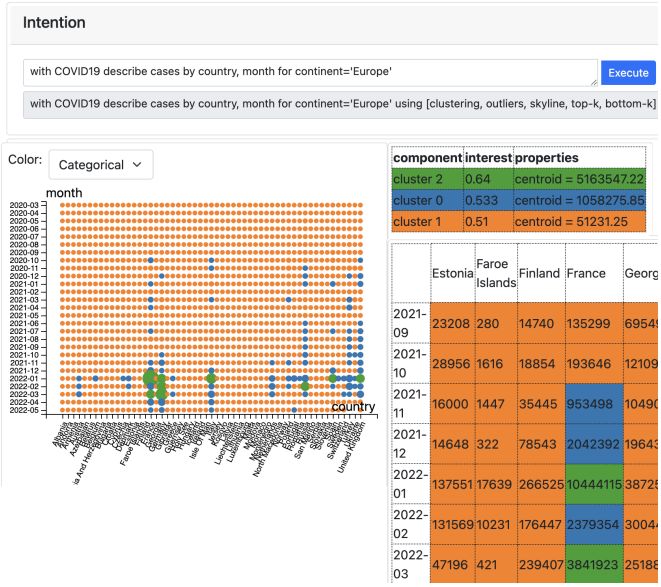

Figure 3: A description of the monthly cases for each European country via the describe operator

The syntax for assess is

$$\text{with } C_0 \text{ assess } m \text{ [for } P \text{] by } l_1, \ldots, l_n$$

where $m \in M$; P is an optional set of selection predicates, each of type $l = u$; and $l_1, \ldots, l_n$ (with $l_i \in L$) is a group-by set of C_0. Like for describe, cube C is obtained from C_0 by applying the conjunction of the predicates in P and aggregating by $l_1, \ldots, l_n$. Each assessment model includes (i) a benchmark, that represents the expected or desirable performance of m; (ii) a function to be used for comparing m to the benchmark measure; and (iii) the labeling of each cell in C based on the result of the comparison [3]. The benchmarks we consider are of two types:

- *Sibling benchmarks* compare the values of m in each cell of a slice of C with its values in another slice of C related to a sibling member (e.g., assess the monthly COVID-19 cases in Italy with reference to those in France).
- *Parent benchmarks* compare the values of m in each cell of C against the one taken in a parent (aggregated) cell (e.g., assess the monthly cases in Italy with reference to those in Europe).

As to comparison, three functions are considered, namely, difference, relative difference, and ratio; we call R the set of these functions. Finally, the labeling functions we consider are of two types:

- *Functions based on explicit ranges.* Specifically, we defined functions with 2, 3, and 5 labels that can operate on values resulting from either a difference (centered on 0 and working with absolute values), or a relative difference (centered on 0 and working with percentage values), or a ratio (centered on 1).
- *Functions based on the overall value distribution.* Specifically, we consider an equi-depth binning function (quartiles, can be coupled with any comparison function) and two equi-width binning functions (with 3 and 5 labels, can be coupled with both difference and relative difference).

We call T the set of these functions.

To determine the most relevant assessment model, i.e., one among all possible combinations of a benchmark, a comparison function, and a labeling function, a different approach than the one followed for description models must be followed. The reason for this is twofold. First of all, given the large number of alternatives (mainly related to the possibility of having hundreds of candidate benchmarks), we had to adopt a greedy approach, which means first selecting one benchmark B, then one comparison function r for B, and finally one labeling function λ for r. Secondly, while novelty, peculiarity, and surprise well capture the salient aspects of an analysis aimed at *describing* data, they cannot be used for assessment since they do not reflect the interest of a *judgment*. Thus, the approach we follow to select one assessment model is based on its *representativeness*, and it can be summarized as follows [4]:

(1) The set of candidate benchmarks S is determined. Specifically: for each level l_i in the by clause, S includes a parent benchmark on the level l_i' that aggregates l_i; if there is a predicate $(l_i = u) \in P$, then S also includes a sibling benchmark for each other member of l_i.
(2) Cube C is retrieved and joined with the corresponding parent and sibling benchmarks.
(3) One representative benchmark B is selected as the centroid of S obtained via the k-medoid algorithm [11] applied to the distance of measure values with $k = 1$.
(4) One representative function r is selected as the centroid of R obtained via the k-medoid algorithm applied to the the distance between the meta-features describing the result of the comparison (namely, the mean, variance, and skewness of each comparison) with $k = 1$.
(5) One representative labeling function λ is selected as the centroid of T obtained via the k-medoid algorithm applied to the Kendall's Tau distance [1] between the labelings, with $k = 1$.

The enhanced cube obtained from an assess statement is visualized like for describe; here, different color schemes can be selected for labels to convey the desired semantics (e.g., red and green for bad and good performance, respectively).

Example 2.2. The statement

with COVID19 assess deaths
for month = '2021-12' and continent = 'Europe' by country

returns a cube showing the number of deaths on Dec. 2021 for each European country. The system-generated expansion (Figure 4) compares the deaths in each country with the European average (as specified by the against clause) via a relative difference (using clause); the labeling scheme selected (labels clause) uses five ranges: *quite lower*, $(-\infty, -0.5]$; *lower*, $(-0.5, -0.1]$; *same*, $(-0.1, 0.1)$; *higher*, $[0.1, 0.5)$; *quite higher*, $[0.5, +\infty)$. □

Like for describe, in [3] we evaluated the saving in user's effort when writing an assess statement over the one necessary to obtain the same result using plain SQL and Python. It turned out that the total formulation effort using SQL+Python is, for intentions with different complexities, always two orders of magnitude larger than using assess statements. For instance, for the intention in Example 2.2, the effort for writing the intention is 84 characters, while the one for writing the necessary OLAP queries (including those for computing the candidate benchmarks) and Python code would be 940 and 11200 characters, respectively.

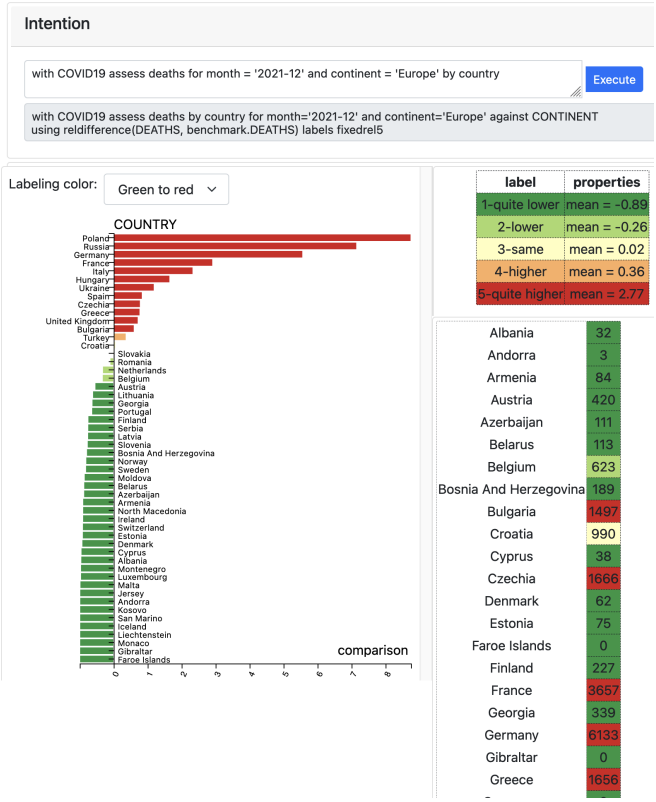

Figure 4: An assessment of the number of deaths on Dec. 2021 for each European country via the assess operator

3 IMPLEMENTATION

The prototype used for the demo relies on the multidimensional engine described by [2], which in turn relies on the Oracle 11g DBMS to execute queries on a star schema based on multidimensional metadata (in principle, the prototype could work on top of any other multidimensional engine). The mining models are imported from the Scikit-Learn Python library. Finally, the web-based interface is implemented in JavaScript and exploits the D3 library for chart visualization; it can be accessed at http://big.csr.unibo.it/projects/iam-demo/web/.

The interface hinges on a notebook where the user can create a data exploration session by writing describe and assess statements. Each statement is first written by the user through the syntax of Section 2, then it is expanded by the system by introducing additional clauses as shown in Examples 2.1 and 2.2. This expansion is particularly useful for assess as it lets users understand which benchmark, comparison function, and labeling function have been selected by the system. Besides, for both describe and assess, it allows users to override the system selections and select a different model by editing the using, against, and labels clauses.

4 DEMONSTRATION PLAN

The demonstration will aim at showing the effectiveness of the IAM approach in supporting data exploration and analysis. It will include two scenarios with guided interaction, aimed at motivating the approach and letting users familiarize with the syntax; after that, users will have a chance to run custom sessions.

In the first scenario, the users will play the role of an analyst who wishes to get an overview of the epidemic trend in Europe. They will start by describing the monthly trend of cases in Europe (with COVID19 describe cases by continent, month for continent='Europe'), then they will drill-down to countries (with COVID19 describe cases by country, month for continent='Europe', as in Example 2.1). Since the intention highlights as most interesting the cluster including the months from January to March 2022 (green bubbles in Figure 3), the users will finally focus on these months to assess the infections on a country basis (e.g., with COVID19 assess cases for month = '2022-01' and continent = 'Europe' by country).

In the second scenario, the users will investigate the deaths in Europe. After comparing the European deaths with those of the other continents (with COVID19 assess deaths by continent), they will drill down to monthly deaths in Europe (with COVID19 assess deaths by month for continent='Europe'). Finally, they will focus on the month showing the highest number of deaths and drill down to countries (with COVID19 assess deaths for month = '2021-12' and continent = 'Europe' by country, as in Example 2.2).

In both scenarios, users will be allowed to edit all the clauses to better tailor the analysis to their needs, for instance, by forcing one specific model in a describe statement or a specific benchmark in an assess statement. Besides, they will be allowed to get further information on any result by hovering on charts following the details-on-demand paradigm.

REFERENCES

[1] Ronald Fagin, Ravi Kumar, Mohammad Mahdian, D. Sivakumar, and Erik Vee. 2004. Comparing and Aggregating Rankings with Ties. In *Proc. of PODS*. Paris, France, 47–58.

[2] Matteo Francia, Enrico Gallinucci, and Matteo Golfarelli. 2020. Towards Conversational OLAP. In *Proc. of DOLAP*. Copenhagen, Denmark, 6–15.

[3] Matteo Francia, Matteo Golfarelli, Patrick Marcel, Stefano Rizzi, and Panos Vassiliadis. 2021. Assess Queries for Interactive Analysis of Data Cubes. In *Proc. of EDBT*. Nicosia, Cyprus, 121–132.

[4] Matteo Francia, Matteo Golfarelli, Patrick Marcel, Stefano Rizzi, and Panos Vassiliadis. to appear. Suggesting assess queries for interactive analysis of multidimensional data. *IEEE Trans. Knowl. Data Eng.* (to appear).

[5] Matteo Francia, Patrick Marcel, Verónika Peralta, and Stefano Rizzi. 2022. Enhancing Cubes with Models to Describe Multidimensional Data. *Inf. Syst. Frontiers* 24, 1 (2022), 31–48.

[6] Dimitrios Gkesoulis, Panos Vassiliadis, and Petros Manousis. 2015. CineCubes: Aiding data workers gain insights from OLAP queries. *Inf. Syst.* 53 (2015), 60–86.

[7] Lukasz Golab, Howard J. Karloff, Flip Korn, and Divesh Srivastava. 2010. Data Auditor: Exploring Data Quality and Semantics using Pattern Tableaux. *Proc. of VLDB Endow.* 3, 2 (2010), 1641–1644.

[8] Matteo Golfarelli and Stefano Rizzi. 2009. *Data Warehouse Design: Modern Principles and Methodologies.* McGraw-Hill, Inc., New York, NY, USA.

[9] Jiawei Han. 1997. OLAP Mining: Integration of OLAP with Data Mining. In *Proc. of Conf. on Database Semantics*. Leysin, Switzerland, 3–20.

[10] Shrainik Jain, Dominik Moritz, Daniel Halperin, Bill Howe, and Ed Lazowska. 2016. SQLShare: Results from a Multi-Year SQL-as-a-Service Experiment. In *Proc. of SIGMOD*. San Francisco, CA, USA, 281–293.

[11] Xin Jin and Jiawei Han. 2010. K-Medoids Clustering. In *Encyclopedia of Machine Learning*, Claude Sammut and Geoffrey I. Webb (Eds.). Springer US, 564–565.

[12] Ales Popovic, Ray Hackney, Rana Tassabehji, and Mauro Castelli. 2018. The impact of big data analytics on firms' high value business performance. *Inf. Syst. Frontiers* 20, 2 (2018), 209–222.

[13] Luc De Raedt. 2002. A Perspective on Inductive Databases. *SIGKDD Explorations* 4, 2 (2002), 69–77.

[14] Sunita Sarawagi. 1999. Explaining Differences in Multidimensional Aggregates. In *Proc. of VLDB*. Edinburgh, Scotland, 42–53.

[15] Bo Tang, Shi Han, Man Lung Yiu, Rui Ding, and Dongmei Zhang. 2017. Extracting Top-K Insights from Multi-dimensional Data. In *Proc. of SIGMOD*. Chicago, IL, USA, 1509–1524.

[16] Panos Vassiliadis, Patrick Marcel, and Stefano Rizzi. 2019. Beyond Roll-Up's and Drill-Down's: An Intentional Analytics Model to Reinvent OLAP. *Inf. Syst.* 85 (2019), 68–91.