Massive Data Exploration using Estimated Cardinalities

Pierre Nerzic, Grégory Smits, Olivier Pivert, Marie-Jeanne Lesot

Massive Data Exploration using Estimated Cardinalities

Pierre Nerzic[1], Grégory Smits[1], Olivier Pivert[1], and Marie-Jeanne Lesot [2]
[1] IRISA - Université de Rennes 1, UMR 6074, Lannion, France
Email: {pierre.nerzic,gregory.smits,olivier.pivert}@irisa.fr
[2] Sorbonne Université CNRS, LIP6 F-75005 Paris, France
Email: marie-jeanne.lesot@lip6.fr

***Abstract*—Linguistic summaries are used in this work to provide personalized exploration functionalities on massive relational data. To ensure a fluid exploration of the data, cardinalities of the data properties described in the summaries are estimated from statistics about the data distribution. The proposed workflow also involves a vocabulary inference mechanism from these statistics and a sampling-based approach to consolidate the estimated cardinalities. The paper shows that soft computing techniques are particularly relevant to build concrete and functional business intelligence solutions.**

***Keywords*—Linguistic summarization, vocabulary inference, cardinality estimation, big data, proof-of-concept**

I. Introduction

Linguistic summarization consists in providing an end user with a set of statements that describe the properties that may be observed in the data. Statements follow a syntactic protoform: *Q X are P*, where *X* denotes the analyzed data, *P* is the property observed and *Q* a quantifier that linguistically describes the extent to which *P* covers *X*. *P* is a conjunctive combination of fuzzy modalities taken from a fuzzy vocabulary. An example of such a linguistic description of a property is: *very few flights are (such that) distance is short and arrival delay is very long*.

The linguistic summarization task has received a huge attention among the soft computing community since the seminal paper [1] that introduced it. It has been especially shown, in various applicative contexts, that linguistic summaries provide a very informative first view of the data based on which users can decide to invest in a costly data integration process or in the implementation of *ad hoc* data mining tools.

This paper describes a complete workflow based on soft computing techniques to help domain experts translate massive relational data into useful knowledge. This workflow is a three step process: 1) users are assisted in the definition of their vocabulary, 2) data are linguistically summarized using terms from the vocabulary in a very efficient way, and 3) interactive exploration functionalities are then provided on top of the linguistic summaries.

As it provides intuitive functionalities to explore and understand data, the proposed approach, called FuzViz, constitutes a Business Intelligence (BI) solution. Compared to existing commercial solutions [2], [3] and soft computing approaches to BI [4], [5], the originality is that the provided synthetic views are not computed from the data but from statistics about their distribution. Data being stored in a Relational DB Management System (RDBMS), statistics are indeed automatically maintained by the system and it has been shown in [6] that reliable summaries may be estimated, in a very efficient way, from these statistics only.

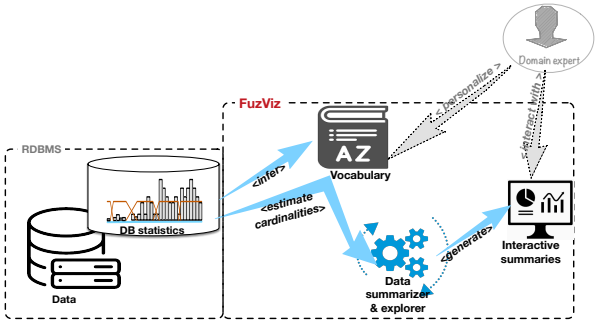

Fig. 1. Overview of FuzViz Workflow

As illustrated in Figure 1, FuzViz leverages the DB statistics to suggest a possible vocabulary that fits the data distribution and allows users to interact with linguistic summaries in a very fluid way. Whatever the size of the data, summaries are generated in less than two seconds. Contributions are:

- to provide a vocabulary elicitation mechanism from DB statistics,
- to embed estimated linguistic summaries in a data exploration tool that can manage massive relational data,
- and to consolidate the cardinalities estimated from the DB statistics using sampling techniques.

After a brief recall of the main notions FuzViz relies on in Section II, a positioning wrt. existing approaches to a subjective exploration of data is proposed in Section III. The different steps of the FuzViz workflow are then detailed in Section IV and illustrated on a real dataset in Section V.

II. Background Notions and Notations

This section first recalls basic notions about relational data and the statistics automatically maintained about their distribution. PostgreSQL is used in this work but the approach can easily be adapted to any other RDBMS. Notations used throughout the paper are also introduced.

A. Relational Data and Metadata

The main goals of an RDBMS is to store and maintain relational data and to provide efficient querying functionalities even in case of massive data. Citus[1] is e.g. a scalable distributed extension of PostgreSQL. When a query is submitted to an RDBMS, it has to determine the most efficient execution plan, as e.g. which clause (selection or join) or condition to apply first. This choice is guided by statistics maintained automatically by the system about data distribution. These statistics evolve according to modifications made on the DB using optimized *group by* queries executed on a DB sample whose size is determined by an error metric [7].

To describe the nature of these statistics, let us consider a relation R, that may be the materialized result of a join query, composed of n numerical or categorical attributes $\{A_1, A_2, \ldots, A_n\}$. For any attribute $A_i, i = 1..n$, the RDBMS maintains, in a table of the *catalog* DB, the list of the k ($k = 100$ by default) most frequent values found in the data sample as well as their frequency. For a given value v from the domain D_i of attribute A_i, its frequency is denoted by σ_v and $\sigma_v \in [0, 1]$. In addition, if A_i is of a numerical nature, then an equi-depth histogram [8] of h buckets ($h = 100$ by default) denoted by $H_{A_i} = \{b_1^i, b_2^i, \ldots, b_h^i\}$ is also maintained to model the data distribution on less frequent values.

Despite attempts to model data distribution on cross domains using so-called multidimensional histograms [9], statistics maintained by RDBMS are unidimensional only, for the sake of compromise between maintenance efficiency, cardinality estimation precision and storage overhead.

B. Expert's Vocabulary

Properties observed in the data and described in summaries are expressed using linguistic terms taken from a subjective and contextual vocabulary. More formally, a vocabulary defined on the attributes $\{A_1, \ldots, A_n\}$ is denoted by $\mathcal{V} = \{V_1, \ldots, V_n\}$, and consists of a set of linguistic variables, associated with each attribute: V_i is a triple $\langle A_i, \{v_{i1}, \ldots, v_{i,q_i}\}, \{l_{i,1}, \ldots, l_{i,q_i}\}\rangle$ where q_i denotes the number of modalities associated with attribute A_i, $v_{i,s}$ denote their respective membership functions defined on domain D_i and $l_{i,s}$ their respective linguistic labels.

For the sake of interpretability, it is imposed that a value from an attribute definition domain may satisfy up to two modalities. These two modalities have to be adjacent when the attribute is numerical so as to form strong fuzzy partitions [10].

For instance, an attribute A_i describing a flight departure time may be associated with $q_i = 4$ modalities, in turn associated with the labels $l_{i,1}$ ='*night*', $l_{i,2}$ ='*morning*', $l_{i,3}$ ='*midday*' and $l_{i,4}$ ='*afternoon*'. Figure 3 and 2 illustrate examples of connection between data distributions and their linguistic interpretation using fuzzy variables associated with a fuzzy partition.

The vocabulary plays a crucial role in the presented approach as it provides a symbolic and subjective interface to represent and access the numerical and categorical space of data definition.

[1]https://www.citusdata.com/

III. Related Works

In addition to being operational, the data-to-knowledge process presented in this paper brings two contributions: it provides a fuzzy vocabulary inference mechanism from DB statistics, and it estimates fuzzy cardinalities from DB statistics and data samples. This section positions FuzViz functionalities wrt. existing approaches on these two topics.

A. Vocabulary Elicitation

A vocabulary materialized by means of a strong fuzzy partition [10] is considered in this work as expert knowledge about the concerned applicative context [11]. Associated with linguistic variables, modalities of the vocabulary provide an interface between the data definition space, generally numerical and categorical, and the symbolic and subjective space of human reasoning. It is essential for the expert to have a good understanding of the meaning of these modalities: providing intuitive functionalities [12] to allow users manually define and modify the vocabulary thus makes sense. Moreover, linguistic terms are used to describe data properties in a more interpretable way than their numerical/categorical description. It is thus also crucial to check that the fuzzy partitions match the data distribution or inner structure. In [13], a measure has been proposed to quantify the adequacy between the inner data structure and the one induced by a fuzzy vocabulary. Such a measure may be used to guide a cooperative vocabulary elicitation strategy [14]. Placing interpretability first, that depends on the shape of the modalities and their number within a partition, it is suggested in [15] to infer a family of possible partitions from the data. In an automatic learning context, a strategy based on tools from mathematical morphology is proposed to sketch the shape of a fuzzy term from training examples [16].

The vocabulary inference mechanism embedded in FuzViz relies on the latter technique to infer a possible partition from the unidimensional DB statistics.

B. Linguistic Summarization and Data Exploration

Since the seminal paper by R.R. Yager about linguistic summaries [1] using fuzzy subsets, a huge number of complementary contributions have been published to make this process efficient or to adapt it to different applicative contexts and data types (see [17] for a quite recent review). Linguistic summaries provide a synthetic and rough view of properties that may be observed in the data. Data summarization constitutes a perfect first step within a complete data-to-knowledge translation process. It allows to discover data distribution and to identify properties of interest, so that users can then decide whether to perform more costly and precise data mining tasks. A crucial issue is thus to generate efficiently such concise views from possibly large data. To this purpose, a novel strategy has been proposed in [6] to estimate linguistic summaries from, possibly large, data stored in a RDBMS. The relative cardinality of each candidate summarizer (i.e. the P's in statements of the form *Q X are P*) is estimated from the statistics maintained by any RDBMS.

In addition to providing a linguistic description of data properties, such summaries may also be the starting point of data exploration functionalities [18].

The FuzViz system described in this paper generates estimated linguistic summaries with an improved precision compared to [6] thanks to sampling-based consolidations, and provides richer exploration functionalities than [18]. It indeed provides the user with real time estimations of the cardinality of each vocabulary element, and involves fluid and interactive views of data properties.

IV. Interactive and Subjective Exploration of Massive Data using FuzViz

This section presents in detail the workflow followed by FuzViz to guide users during data exploration. A focus is first made on how an adequate vocabulary is suggested from DB statistics, then follows a description of the summarization and exploration functionalities.

A. Vocabulary Inference

1) Overview: To help the user define his/her exploration vocabulary, FuzViz integrates three strategies to suggest an initial discretization of an attribute domain that can then be manually adjusted. The first one simply performs an equi-width discretization of a numerical domain according to a user-given number of expected modalities. Whereas this first strategy does not depend on the data distribution, the second one generates an equi-depth strategy that builds a partition of q modalities, where q is given by the user, such that each of the modalities covers the same amount of data. It has been shown in [15] that, to remain interpretable, a partition should contain around four modalities, so by default $q = 4$.

The third strategy, described in the next subsection, does not require that the user provides the number of expected modalities. This so-called adjusted strategy indeed analyzes the histogram reconstructed from the DB statistics to identify subsets of high coverage that are worth being linguistically described by dedicated modalities.

Whatever the employed vocabulary inference strategy, users then have to define a linguistic label for each built modality. This can be manually done using FuzViz's interface. They also can adjust the boundaries of any modality at will.

2) Proposed Approach: Let $H_A = \{b_1, b_2, \ldots, b_h\}$ be a histogram describing the tuples distribution on domain D of attribute A. Each bucket $b_j, j = 1, ..., h$, is associated with its relative frequency denoted by σ_{b_j}. It is considered that σ_{b_j} also includes the relative frequency of each top-k frequent value that is in between the bounds of b_j. This consolidated histogram is not equi-depth. The mean frequency denoted $\hat{\sigma}_{H_A}$ is then computed as: $\hat{\sigma}_{H_A} = \frac{1}{h}\sum_{j=1}^{h} \sigma_{b_j}$.

The histogram H_A, viewed as a sequence of h buckets, is translated into a word of h symbols that can be $\triangle$ or ∇ (line 2 in Alg. 1): $\triangle$ indicates a bucket of high frequency, as compared to the mean frequency and ∇ a bucket of low cardinality. More formally, each bucket is translated according to the following rule:

$$b = \begin{cases} \triangle & \text{if } \sigma_b \geq \hat{\sigma}_{H_A} \\ \nabla & \text{otherwise.} \end{cases} \quad (1)$$

Data: $H_A : \{b_1, b_2, \ldots, b_h\}$; δ;
Result: Fuzzy partition

1 $\hat{\sigma}_{H_A} \leftarrow \frac{1}{h}\sum_{j=1}^{h} \sigma_{b_j}$;
2 $word \leftarrow rewrite(H_A, \hat{\sigma}_{H_A})$;
3 $cores \leftarrow []$;
4 **for** $idx \leftarrow 1..|word|$ **do**
5 **if** $word[idx] = \triangle$ **then**
6 $[A, B] \leftarrow$ $[\arg min_{i=1,\ldots,idx}(i), \arg max_{j=idx,\ldots,h}(j)]$ st. $\frac{|\{\nabla \in word[i,j]\}|}{j-i} \leq \delta$;
7 $cores.append([A, B])$;
8 $cores.removeSubsetsOf([A, B])$;
9 **end**
10 **end**
11 $mbf \leftarrow []$;
12 **foreach** $core \in cores$ **do**
13 $[a, A, B, b] = buildTransition(core)$;
14 $mbf.append(\mu_{[a,A,B,b]})$;
15 **end**
16 **return** $\langle A, mbf \rangle$;

Algorithm 1: Partition inference from an equi-width histogram describing data distribution

In the spirit of the approach introduced in [16], the principle of Algorithm 1 is to identify sequences containing a large majority of $\triangle$s that then form the cores of the fuzzy modalities (line 6 in Alg. 1). Starting from each $\triangle$, the largest intervals containing a proportion of ∇ symbols smaller than δ are identified, a threshold is empirically set by default to 0.25. Gradual transitions between adjacent cores are then built to satisfy the structural constraints of a strong fuzzy partition (line 13 in Alg. 1).

Once the histogram rewritten into a word containing the symbols $\{\triangle, \nabla\}$ only (line 2 in Alg.1), the function *removeSubsetsOf* (line 8 in Alg. 1) is used to keep the largest found intervals, removing all of their subsets. The function *buildTransition* (line 13 in Alg. 1) returns the four bounds of the trapezoidal fuzzy subset built around the core given as parameter, it sets a to the core right bound of the previous modality, and b to the core left bound of the following modality.

Figure 3 shows that the proposed approach generates a partition that fits the data distribution.

B. Fuzzy Cardinality Estimation

For each term of the user's vocabulary, FuzViz estimates and shows its relative coverage of the analyzed tuples using the DB statistics only. This section describes this estimation process for atomic and conjunctive properties successively.

1) Atomic Properties: The estimation of the relative coverage of the analyzed tuples by atomic properties consists in confronting the fuzzy subset with the histograms and the lists of frequent values maintained by the RDBMS. It is computed using the Choquet-based approach introduced in [6]. One of the advantages of this approach is that it can be applied on both numerical and categorical attributes. Whatever the size

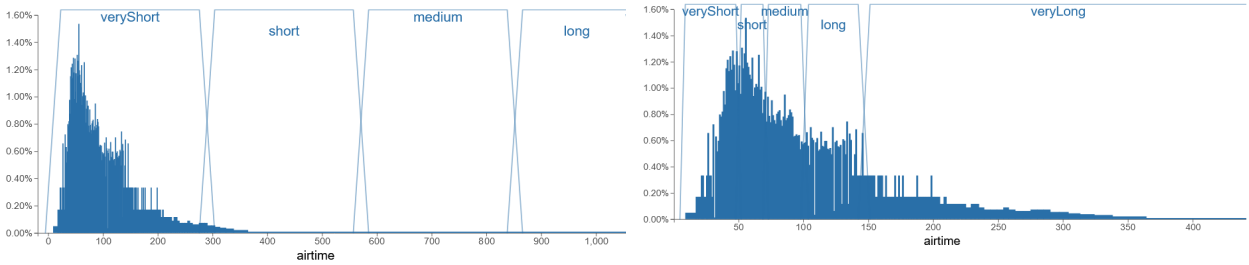

Fig. 2. Equi-width partition (left), equi-depth partition (right)

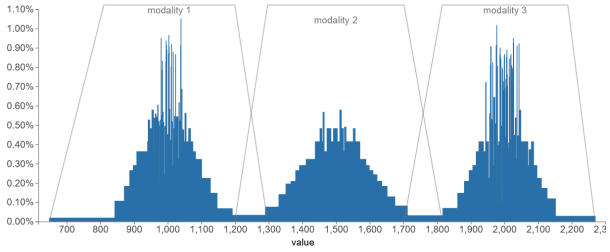

Fig. 3. Adjusted partition according to data distribution

of the database, the cardinality of a term is estimated in only a few milliseconds with a high precision (Section V). The user may ask for the actual cardinality of a term, but this is done with a count query, whose execution time generally linearly depends on the size of the DB.

2) Case of Conjunctive Properties: This section describes the proposed method implemented in FuzViz to estimate the cardinality of a conjunctive property with a good precision. An RDBMS only maintains statistics about data distributions on the different attributes individually. To estimate the cardinality of a conjunction of terms, several strategies can be used. The first one is to consider the attributes as independent. This is used by most RDBMS query planners. Under this hypothesis, the relative cardinality of a conjunction is the product of the cardinalities of the individual terms. Few other RDBMS assume that the attributes are fully dependent, in this case the cardinality of a conjunction is the smallest cardinality of the terms. However, the RDBMS planner rather aims at having an upper bound of the amount of tuples to process, than computing the most precise cardinality as in the case for summarizing data.

The conjunctions considered in this work involve at most one modality per attribute. For instance, on a database describing commercial flights, one can build the conjunction of the modality "long" for attribute "AirTime" and the modality "short" for attribute "Distance", that will be denoted by "AirTime.long $\wedge$ Distance.short" in the following. The question is about dependencies between modalities, not only attributes. The relative cardinality σ_P of a conjunction of modalities $P = m_1 \wedge m_2 \wedge ...$, relies on the number of tuples that satisfy all the modalities simultaneously. Let us recall the following possible cases:

- dependent modalities: the modalities are satified together by the same tuples in the database, for instance "Distance.long $\wedge$ AirTime.long". In this case, the relative cardinality of the conjunction P is $\sigma_P = \min_{m \in P} \sigma_m$, denoted by σ_{min} in the following.
- independent modalities: there is no link between the modalities, for instance "Distance.long $\wedge$ Month.spring". The cardinality of the conjunction is $\sigma_P = \prod_{m \in P} \sigma_m$, denoted by σ_{prod}.
- incompatible modalities: no tuple satisfies all the modalities simultaneously, for instance "Distance.long $\wedge$ AirTime.short". In this case $\sigma_P = 0$.

Note that $0 \leq \sigma_{prod} \leq \sigma_{min} \leq 1$.

In the absence of any appropriate DB statistics, and also because the user can revise the modalities at any time, FuzViz implements a simple heuristic method to estimate the relative cardinalities of conjunctions with a small processing time and a precision better than with the independence assumption σ_{prod}. A sample of the database is extracted in order to compute a *score of dependency* of the modalities. As described below, this score qualifies the dependency that is observed between the modalities of a conjunction on this sample, in the form of a real number in the interval $[-1, +1]$. The fact is that with real data, the dependencies are not purely Boolean. There are many exceptions that are well captured by this score having intermediate values.

Actually, this score spans on two intervals. From 0 to $+1$, it indicates that the modalities are somewhat dependent; the relative cardinality of their conjunction is estimated as being between σ_{prod} and σ_{min}. From -1 to 0, the modalities are either incompatible or independent; the relative cardinality of the conjunction is estimated as being between 0 and σ_{prod}. We propose to estimate the cardinality of the conjunction σ_P as a linear interpolation between the two extreme cases, in their respective intervals, using the score of dependency as weight, as defined in Equation (2) below. This is almost as fast to compute as σ_{prod}, once the score of dependency is known.

$$\sigma_P = \begin{cases} (1 - score_P)\sigma_{prod} + score_P \sigma_{min} & \text{if } score_P \geq 0 \\ (1 + score_P)\sigma_{prod} & \text{otherwise.} \end{cases} \quad (2)$$

Here are some explanations on how the score of dependency is computed.

Firstly, a scan of a small sample of the database is performed. The size of this sample is calibrated so that the computation lasts at most 15 seconds. This could be chosen by the user. For instance, one can ask for a bigger random sample,

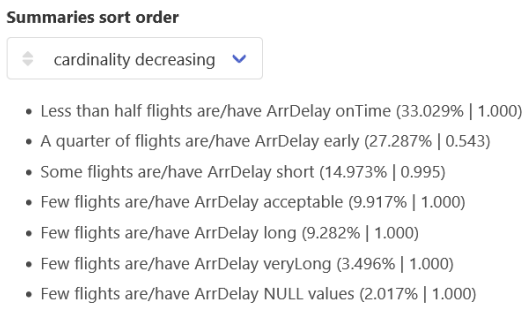

Fig. 4. Extract of the linguistic statements summarizing the properties found in the data.

then stop computations when the time limit is reached. It is necessary to ensure that the sample contains enough different representative tuples, to infer a reliable score of dependency.

Secondly, the relative cardinality of the conjunction on the sample, σ_P, and the cardinalities of the individual modalities of the conjunction, σ_{m_1}, σ_{m_2} ... are computed, from which the minimum σ_{min} and the product σ_{prod} are deduced. Finally, $score_P$ is computed with Equation (3) is the inverse of Equation (2).

$$score_P = \begin{cases} (\sigma_P - \sigma_{prod})/(\sigma_{min} - \sigma_{prod}) & \text{if } \sigma_P \geq \sigma_{prod} \\ (\sigma_P/\sigma_{prod}) - 1 & \text{otherwise.} \end{cases} \quad (3)$$

Lastly, the scores of dependencies of all modalities of the selected attributes are computed in the same data scan, thus saving a lot of time, because the individual cardinalities of the modalities are common to many conjunctions. Then the scores are cached, so that they need to be recomputed only if the user modifies the definition of any of the modalities, or adds new terms to the conjunction. In this case, the score computation process is restarted in background. The scores of dependency remains constant if the user only removes terms, or changes the order of the terms involved in the conjunction.

C. Summaries Rendering and Exploration

Once a vocabulary and quantifiers have been defined, users can ask for a summarization of the data on all or a selected subset of the attributes for which a vocabulary is available. FuzViz then generates in less than 2 seconds (see Section V) a set of linguistic statements of the form *'Q X are P'* that describe the properties that may be observed in the data. As depicted in Figure 4, filtering functions are available to focus e.g. on properties with the highest coverage or conversely on rare properties.

As shown in Figure 5, FuzViz also provides an interactive view to explore the data properties in an intuitive and fluid way. This is a zoomable sunburst diagram [19], a mix between a hierarchical diagram and a pie chart. Each concentric layer of this view represents an attribute of the analyzed data, and its portions correspond to the related fuzzy partition modalities. A dynamic tooltip appears and displays the relative cardinality when hovering on any portion. At the beginning, only the first

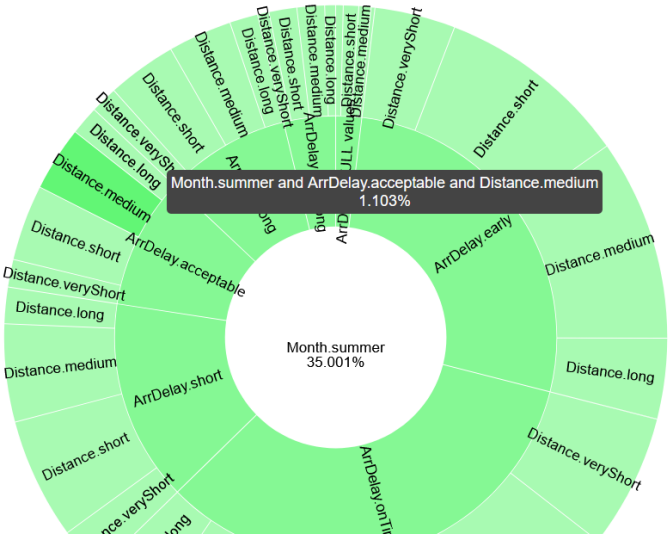

Fig. 5. Part of the sunburst view used to explore the data.

two attributes are drawn, but the user can click on any portion to zoom in and display this portion in place of the root layer. This allows to have a better view on the relative cardinalities of the modalities. Users may thus explore and discover the different conjunctive combinations of terms. Whatever the size of the underlying data, when the user changes anything in the previous steps, for instance the definition of a modality, or the order of the attributes in the conjunction, the view is almost instantly updated and the cardinality of the current conjunction of terms estimated.

V. Illustration of FuzViz Functionalities and Efficiency

FuzViz is a generic software solution to the analysis of massive data stored in a relational table, this table may be the materialized result of any join query. This section shows the main exploration functionalities provided by FuzViz and recalls how reliable cardinality estimations may be efficiently computed leveraging the statistics maintained by any RDBMS [6]. In the present illustration scenario, FuzViz is implemented as a portable web server (Python Flask) with a graphic interface (Vue.js). It runs on a Xeon 2.8GHz CPU and 32GB of RAM. The explored data are stored in a PostgreSQL 13 server.

A. Cardinality Estimation

To illustrate the approach, the *flight database* [20] is used as an example. This database contains more than 123 millions of records describing flights in the USA between 1987 and 2008. Examples of columns of the "flight" table are "Month", "DayOfWeek", "Distance", "AirTime", and so on.

The results of the cardinality estimation on individual terms are presented first, then the results on conjunctions.

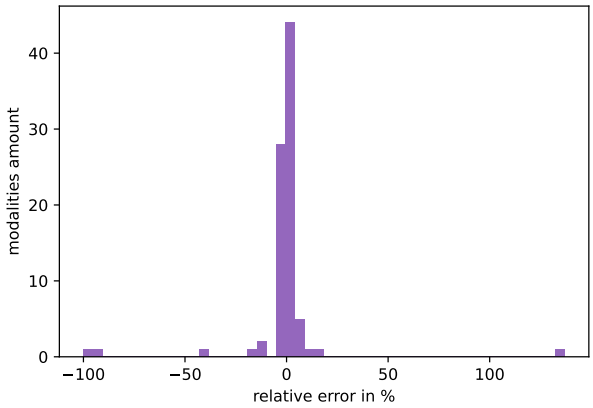

Fig. 6. Individual modalities estimation error.

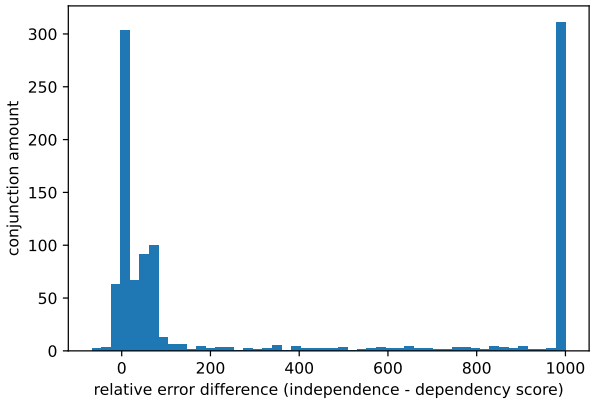

Fig. 7. Error differences on conjunctions of four dependent attributes.

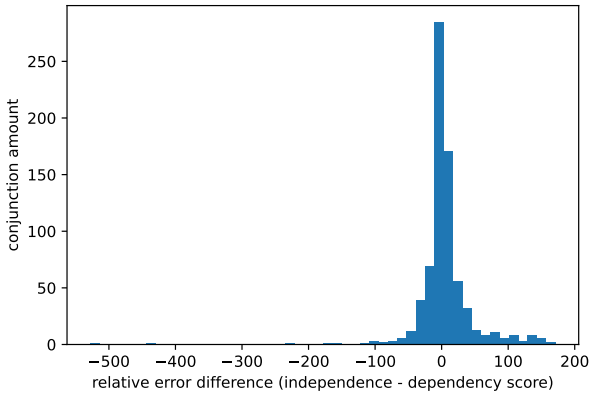

Fig. 8. Error differences on conjunctions of four independent attributes.

1) Cardinality Estimation of Individual Terms: In the experiments, the estimation of the cardinalities of individual modalities using the metadata is of good quality compared to the actual cardinalities obtained with a full scan of the database. Figure 6 shows the distribution of the errors rate (defined as $error(\sigma, \sigma_{real}) = (\sigma - \sigma_{real})/\sigma_{real}$) on the relative cardinalities of individual modalities, as histogram. Most of the modalities are well estimated, while few are very bad. Note that there is no way to detect it without a full database scan. For instance, the worse estimation is on "SecurityDelay.short", the estimated cardinality is 0.0003 while the actual is 0.000127 (136% relative error). This is because the column "SecurityDelay" contains more than 72% of null values. The presence of null values is the major concern impacting the estimation of the cardinalities as it reduces the number of values on which the dependency degrees can be computed. This classical DB issue can be solved using imputation techniques e.g., but this question is out of the scope of this paper.

2) Cardinality Estimation of Conjunctions: In each of the situations below, two methods to estimate the cardinalities on all the conjunctions that can be built on several attributes are compared, the independence assumption and the estimation based on the score of dependency. More precisely, three cardinalities are computed on every possible conjunction of few attributes: actual cardinalities σ_{actual}, estimated cardinalities under the independence hypothesis σ_{prod} and estimated cardinalities with the score of dependency σ_P. Then the relative errors between both estimations and the actual cardinalities are compared, using the difference $|error(\sigma_{prod}, \sigma_{actual})| - |error(\sigma_P, \sigma_{actual})|$. This difference is positive when the proposed method is better than the independence hypothesis.

In the next sections, the histograms of these differences along the conjunctions of all modalities of different selections of attributes are analyzed.

3) Dependent attributes: Figure 7 shows the histogram of the difference of errors in percentage, on 1049 conjunctions built on 4 attributes, "AirTime", "ArrDelay", "DayOfWeek" and "Distance". The buckets represent the amount of conjunctions that share the same error difference. The actual cardinalities on the whole dataset have been computed in more than 7 hours, where the estimation of all the dependency scores took only 13 seconds on a sample of 62061 tuples. For these conjunctions, attributes have dependencies: for example the duration (airtime) of a flight is correlated with its distance, and to a lesser extent with the delay at arrival. Most estimations provided by σ_{prod} lead to a much greater error than the proposed σ_P estimation.

Let us take an example of one of these 1049 conjunctions,

$$P = AirTime.medium \wedge ArrDelay.early \wedge DayOfWeek.beginningOfWeek \wedge Distance.medium.$$

Its relative cardinality is estimated to $\sigma_{prod} = 0.003459$ while the proposed method estimates to $\sigma_P = 0.012844$, and the actual cardinality is $\sigma_{actual} = 0.01255$. So the estimation under the independence hypothesis makes about 72% relative error, and the sample-based one only 2.3%. This is what happens for most of the conjunctions of modalities of these four attributes, as shown in Figure 7. Most of the error differences are positive. The bucket at abscissa 1000 represents all conjunctions for which the error difference exceeds 999. The case where σ_P provides a more erroneous estimation than σ_{prod} correspond to less than 60 conjunctions among the 1049 cases. Furthermore, the error difference is always very small.

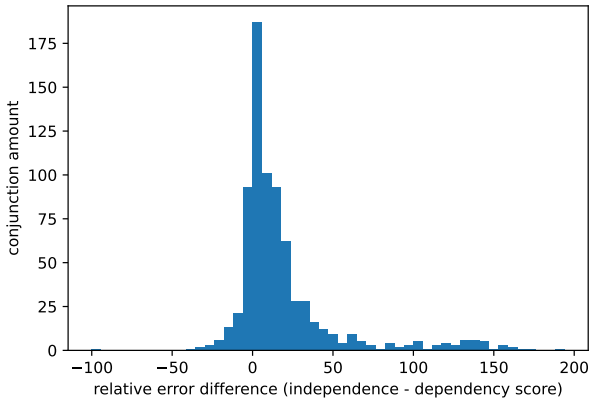

Fig. 9. Error differences on conjunctions of four independent Attributes, larger sample.

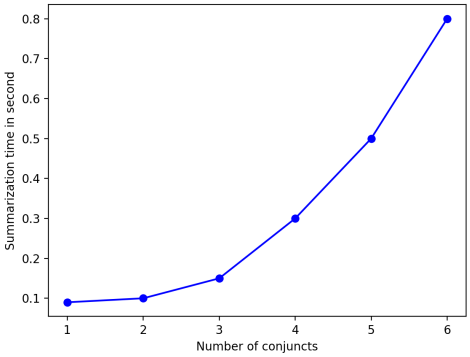

Fig. 10. Summarization time (in second) according to the size of the conjunction.

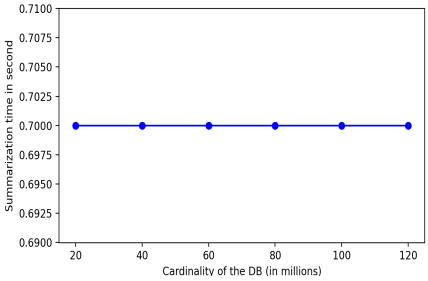

Fig. 11. Summarization time (in second) according to the number of tuples in million.

4) Independent attributes: Figure 8 shows a less favorable situation, when modalities are not at all dependent. 749 conjunctions are built on 4 attributes, "DepTime", "Distance", "Month" and "Origin". There is no real dependency between them, so the independence hypothesis would always be the best. The proposed method shows improvements for some conjunctions, but also more errors. Here is an example that leads to an erroneous estimation:

$$DepTime.night \wedge Distance.veryShort \wedge Month.summer \wedge Origin.small.$$

Its actual relative cardinality is 0.000764, estimated to $\sigma_{prod} = 0.000531$ (30% error) and $\sigma_P = 0.00142$ by the sampling-based method (85% error). It can be noticed that the cardinalities are quite small. The sampling may have missed enough representative data.

It is worth studying the impact of the size of the sample on the computed score of dependency. Figure 8 has been obtained with 0.05% of the whole database and 11 seconds of computation. If the user allows 2 minutes, FuzViz can sample 0.5% of the database and have a much better estimation of the cardinality, as shown in Figure 9

5) Discussion: The proposed method to improve the estimation of the cardinalities of conjunctions based on the computation of a score of dependency appears much better than the simple independence hypothesis in the case of dependencies between data attributes. This score of dependency is evaluated quickly, less than 15 seconds on a random sample of the data. This is enough to significantly improve the estimation in most of the cases. However, it shows no improvement in the case of independent modalities, except if the user allows more computing time to increase the size of the sample, to get more precision in the score of dependency.

B. Summarization Efficiency

The specific feature of FuzViz is to provide fluid data exploration functionalities using subjective and linguistic terms from the user's vocabulary and to estimate the relative cardinality of these terms and their conjunctive combinations using DB statistics. To estimate the initial cardinalities of each property, FuzViz implements the strategy published in [6] that leverages the RDB statistics.

Using the flight dataset described in Section V-A, the time needed to estimate the cardinalities based on which the initial view is built are shown in Figure 10 and 11. Figure 10 gives the time needed to summarize the 127 million flights wrt. to the number of attributes considered in the summaries. Figure 11 shows the evolution of the summarization time wrt. the number of tuples in the DB. The views used to explore the data are updated in less than one second whatever the size of the DB.

The summaries derive from all the conjunctions that can be built from the modalities of all the attributes in the vocabulary. A conjunction obviously never involves more than one modality from a same partition. Given a set of n attributes $a_1, a_2, ..., a_n$, each one having m_{a_i} modalities for instance, then the number of conjunctions is $\prod_{i=1}^{n} m_{a_i}$. So the time complexity is exponential wrt. the number of attributes, as visible in Figure 10. This is the reason why FuzViz explores the combinations of properties considering attributes by groups of four to remain fluid.

Figure 10 does not show the additional constant delay needed to estimate the dependencies between the modalities of the conjunction, since it can be configured by the user. In the experiments, it appears that 10 to 15 seconds are enough

to get a good precision on the score in most cases, but facing independent modalities or too many null values, this time should be increased. FuzViz updates the estimated cardinalities and views using an anytime algorithm.

Estimated cardinalities are particularly adapted to the efficient generation of linguistic summaries. Due to the fact that relative estimated cardinalities are expressed using imprecise linguistic quantifiers, small errors in these estimations have a low impact on the generated statements leading to estimated summaries very close to the actual ones without paying the cost of their computation.

VI. Conclusion

In a data-to-knowledge translation process, providing end users with a concise view of the properties that may be found in the data is a crucial issue. A linguistic summary is a perfect tool to roughly describe in an interpretable way the content of a dataset. However, these summaries have to be very efficiently generated, even when massive data are analyzed, and should be interactive to allow end users dive into their data manipulating interpretable linguistic terms only. The approach, called FuzViz, described in this paper fulfils these two requirements. The first specific feature of FuzViz is to generate interactive data views based on linguistic terms taken from the end user's vocabulary. The second distinctive property is to leverage the statistics maintained by RDBMSs to suggest a vocabulary that fits the data distribution, and to estimate, in a very efficient way, the coverage of conjunctive combinations of these subjective terms. An estimated linguistic summary of the properties that may be observed in a dataset containing millions of tuples may for instance be obtained and rendered in less than one second.

To improve the precision of the estimated cardinalities, a sample-based strategy is described in this paper making it possible to better capture attribute dependencies. These contributions have been implemented and gathered all together to provide a novel "computing with words" approach to BI. The fuzzy vocabulary plays a crucial role in FuzViz. To ease the definition of an appropriate vocabulary, a strategy is provided to infer a first definition of a vocabulary that fits the data distribution described in the DB.

Future works concern the study of other techniques to better capture and represent multidimensional data distribution, using for instance variational autoencoders or random walks.

Acknowledgement: This research is part of the SEA DEFENDER project funded by the French DGA (Directorate General of Armaments).

References

[1] R. R. Yager, "A new approach to the summarization of data," *Information Sciences*, vol. 28, no. 1, pp. 69–86, 1982.

[2] SALESFORCE, "https://www.tableau.com/fr-fr/products/desktop." [Online]. Available: https://www.tableau.com/fr-fr/products/desktop

[3] Microsoft, "https://powerbi.microsoft.com/fr-fr/." [Online]. Available: https://powerbi.microsoft.com/fr-fr/

[4] R. A. E. Andrade, R. B. Pérez, A. C. Ortega, J. M. Gómez, and A. R. Valdés, *Soft Computing for Business Intelligence*. Springer, 2014.

[5] G. Smits, O. Pivert, and R. R. Yager, "A soft computing approach to agile business intelligence," in *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2016, pp. 1850–1857.

[6] G. Smits, P. Nerzic, O. Pivert, and M.-J. Lesot, "Efficient generation of reliable estimated linguistic summaries," in *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2018, pp. 1–8.

[7] S. Chaudhuri, R. Motwani, and V. Narasayya, "Random sampling for histogram construction: How much is enough?" in *ACM SIGMOD Record*, vol. 27. ACM, 1998, pp. 436–447.

[8] Y. Ioannidis, "The history of histograms (abridged)," in *Proceedings 2003 VLDB Conference*. Elsevier, 2003, pp. 19–30.

[9] N. Bruno, S. Chaudhuri, and L. Gravano, "Stholes: A multidimensional workload-aware histogram," in *Proceedings of the 2001 ACM SIGMOD international conference on Management of data*, 2001, pp. 211–222.

[10] E. H. Ruspini, "A new approach to clustering," *Information and control*, vol. 15, no. 1, pp. 22–32, 1969.

[11] S. Guillaume, B. Charnomordic, and P. Loisel, "Fuzzy partitions: a way to integrate expert knowledge into distance calculations," *Information sciences*, vol. 245, pp. 76–95, 2013.

[12] R. R. Yager, M. Z. Reformat, and N. D. To, "Drawing on the ipad to input fuzzy sets with an application to linguistic data science," *Information Sciences*, vol. 479, pp. 277–291, 2019.

[13] M.-J. Lesot, G. Smits, and O. Pivert, "Adequacy of a user-defined vocabulary to the data structure," in *2013 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2013, pp. 1–8.

[14] G. Smits, O. Pivert, and M.-J. Lesot, "Vocabulary elicitation for informative descriptions of classes," in *2017 Joint 17th World Congress of International Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems (IFSA-SCIS)*. IEEE, 2017, pp. 1–8.

[15] S. Guillaume and B. Charnomordic, "Generating an interpretable family of fuzzy partitions from data," *IEEE transactions on fuzzy systems*, vol. 12, no. 3, pp. 324–335, 2004.

[16] C. Marsala, "Fuzzy partition inference over a set of numerical values," in *Proc. of the IEEE Int. Conf. on Fuzzy Systems*. Citeseer, 1995, pp. 1512–1517.

[17] F. E. Boran, D. Akay, and R. R. Yager, "An overview of methods for linguistic summarization with fuzzy sets," *Expert Systems with Applications*, vol. 61, pp. 356–377, 2016.

[18] G. Smits, R. R. Yager, and O. Pivert, "Interactive data exploration on top of linguistic summaries," in *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2017, pp. 1–8.

[19] M. Bostock, "https://observablehq.com/@d3/zoomable-sunburst."

[20] A. Community, "https://community.amstat.org/jointscsg-section/dataexpo/dataexpo2009."