FRELS: Fast and Reliable Estimated Linguistic Summaries

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

FRELS: Fast and Reliable Estimated Linguistic Summaries

Grégory Smits[1], Pierre Nerzic[1], Marie-Jeanne Lesot[2] and Olivier Pivert[1]
[1] Univ Rennes, IRISA - UMR 6074, F-22305 Lannion, France
Email: {gregory.smits,pierre.nerzic,olivier.pivert}@irisa.fr
[2] Sorbonne Université, CNRS, Laboratoire d'Informatique de Paris 6, LIP6, F-75005 Paris, France
Email: marie-jeanne.lesot@lip6.fr

***Abstract*—The linguistic summarization of a dataset is a process whose complexity depends linearly on the size of the dataset and exponentially on the size of the fuzzy vocabulary. To efficiently summarize large datasets stored in Relational DataBases, reliable estimated cardinalities can be derived from statistics about the data distribution maintained by the RDB Management System, with no expensive data scans. This paper proposes to improve the precision of such estimated summaries while preserving their efficiency, by enriching the statistics-based approach with local scan-based corrections when needed: the proposed FRELS method provides efficient strategies both for identifying the needs and performing the corrections. Experiments conducted on real data show that FRELS remains incomparably more efficient than data-scan-based approaches to data summarization and offers a better precision than purely statistics-based approaches. The generation of estimated linguistic summaries takes a couple of seconds, even for datasets containing millions of tuples, with a reliability of more than 95%.**

I. Introduction

Quickly discovering and understanding the content of large datasets is a crucial issue that many professionals have to tackle in their daily activities. As a consequence, they need efficient strategies and tools to help them determine, at a preliminary level, if it is worth spending time to explore finely a dataset, e.g. to apply more intensive data mining techniques, possibly expensive in terms of computation and user effort, to interpret their results.

In this sense, linguistic summarization is very useful [1], as it provides insights in the considered data that are both concise and legible. These summaries are usually structured according to templates, called protoforms [19], [8], whose simplest form is Q *of the* X*'s are* S, where X denotes the data to be described, Q is a quantifier and S, called summarizer, is a conjunction of linguistic labels associated with properties of interest regarding the data. For example, such a protoform can be instanciated as *most of your data concern on-time flights covering long distances*. The quantifier Q and the properties of interest involved in the summarizer S are modeled by fuzzy sets that translate numerical and categorical values into linguistic variables [20]. Additionally, a truth degree [19] is attached to each linguistic statement, in order to quantify the adequacy between the choice of the linguistic quantifier and the numerical cardinality it describes, i.e. the quantity of data that match the considered summarizer.

A major challenge raised by linguistic data summaries is the computational cost of their extraction: the search space and the volume of data to manage require the design of highly efficient strategies. The initial approaches proposed in the literature (see e.g. [11], [7]) adopted a user-guided methodology to limit the number of data scans necessary to identify unspecified parts of the summaries. Genetic algorithms have also been applied to explore the whole search space [4]. More recent approaches propose to prune *a posteriori* the set of the extracted summaries so as to focus on the most relevant ones [10], [16] or to avoid *a priori* the generation of non-relevant summaries [9], [17], [3], [18]. Algorithmically speaking, the existing approaches all rely on a linear scan of the data and their possibly optimized [16] projection on the candidate summarizers.

Recently, a novel strategy [14], denoted by STATS in this paper, has been proposed in the case where the data are stored in a Relational DataBase (RDB) and where simple protoforms are looked for: it exploits statistics about the data distribution maintained by the RDB Management System (RDBMS) to estimate the cardinalities associated with the summarizers and thus to generate very efficiently estimated summaries. At the expense of a very limited loss of precision, the summarization process then does not depend any more on the size of the dataset and achieves a 10,000 speed up factor as compared to classical query-based approaches [14].

This paper aims at improving the precision of such estimated-cardinality-based summaries while preserving the efficiency of the method. The proposed approach, named FRELS for Fast and Reliable Estimated Linguistic Summaries, promotes a compromise between the efficiency of the process and the precision of the generated statements, as graphically illustrated in Figure 1. It enriches the statistics-based approach with local scan-based corrections when needed, providing efficient strategies both for identifying the needs and performing the corrections. More precisely, in the case of simple protoforms involving only numerical attributes, FRELS only uses statistics maintained by the RDBMS to determine lower and upper bounds of the estimated cardinalities. It can then check the relevance of the locally uniform distribution assumption STATS is based on (see details in Section II-B). When this uncertainty is considered strong, FRELS provides an efficient local scan strategy to compute the involved cardinalities.

In addition to this combination of scan- and statistics-based

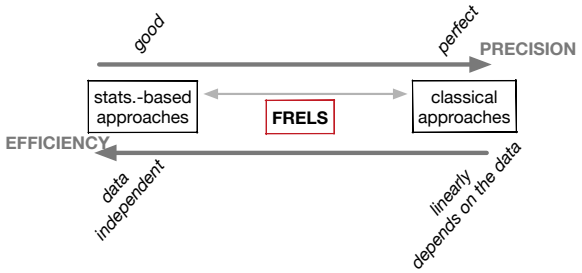

Fig. 1. Position of FRELS wrt. classical and statistics-based approaches to data summarization: a compromise between precision and efficiency.

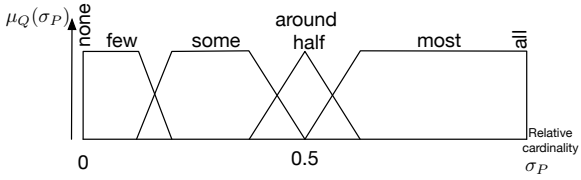

Fig. 2. Example of possible relative quantifiers.

approaches, FRELS also proposes an optimized implementation of STATS, that exploits more efficiently meta-information about the data distribution made available by the RDBMS.

The paper is structured as follows: Section II briefly provides some reminder about linguistic summaries and the STATS use of the RDB statistics to cardinality estimation. Section III describes the proposed FRELS method used to quantify how reliable an estimated cardinality is and how to make the most of such a confidence assessment to improve the reliability of the generated summary through a targeted access to the data. It also presents the proposed optimized exploitation of the RDBMS available statistics. Section IV presents some experimental results on real data and shows that FRELS remains more efficient than classical approaches on the one hand, and offers a better precision than the purely statistics-based approach on the other hand. Section V concludes and draws perspectives for future work.

II. Reminder about Linguistic Summaries and DB Statistics

This section provides a brief reminder about the type of linguistic summaries considered in this paper and the STATS extraction approach that is based on cardinalities estimated from the statistics maintained by any RDBMS [14], on which the FRELS approach proposed in this paper also relies.

A. Simple Linguistic Summaries for Numerical Data

The soft computing community has a long history in data summarization, see e.g. a recent overview in [1], both regarding the type of data to be summarized and the type of summaries to be extracted. This section focuses on the case of simple summaries of numerical data: the dataset, denoted by R, contains the description of m tuples, $R = \{t_1, t_2, \ldots, t_m\}$ wrt. n numerical attributes $\{A_1, A_2, \ldots, A_n\}$.

An associated vocabulary, denoted by $\mathcal{V} = \{V_1, \ldots, V_n\}$, consists of a set of linguistic variables: V_j is a triple $\langle A_j, \{v_{j1}, \ldots, v_{jq_j}\}, \{l_{j1}, \ldots, l_{jq_j}\}\rangle$ where q_j denotes the number of modalities associated with attribute A_j, the v_j's denote their respective membership functions and the l_j's their respective linguistic labels (generally adjectives from the natural language). It is assumed that the linguistic variables associated with an attribute define strong fuzzy partitions [13].

Relative quantifiers linguistically describe scalar relative cardinalities. They are also defined by fuzzy modalities, for instance taken from the classic partition illustrated in Figure 2.

Simple summaries are linguistic statements of the form *Q of the data from R are S*, where Q is a fuzzy quantifier, R is the dataset to be summarized and S, the summarizer, is a conjunctive combination of terms taken from the vocabulary $\mathcal{V}$. All the possible conjunctive summarizers form a lattice by means of an inclusion operator on the sets of terms they involve. Summarizers containing one term are called atomic.

The relevance of a candidate summary for the considered data is then measured by a truth degree that quantifies its validity with respect to R. It is defined as [19]:

$$\tau(Q\ R \text{ are } S) = \mu_Q\left(\sigma_S(R)\right), \tag{1}$$

where $\sigma_S(R)$, also denoted by σ_S in the following, is the cardinality of the summarizer S: $\sigma_S(R) = \frac{\sum_{r\in R}\mu_S(r)}{|R|}$.

The summarization process then consists in projecting the data onto the lattice of possible summarizers, starting with atomic ones, and quantifying the extent to which each summarizer covers the data. The relative cardinality attached to each summarizer is then linguistically described by an appropriate relative fuzzy quantifier.

B. Extraction Based on DB Statistics

In the case where the data are stored in a Relational DataBase, the STATS method [14] efficiently estimates the cardinality σ_S only from the statistics maintained by any RDBMS about the data distribution, as recalled below.

1) DB Statistics for Numerical Attributes: For each attribute, an RDBMS maintains a set of metadata tables that describe the data distribution; they are mainly used to determine the most efficient query execution plan. These statistics are automatically built by the RDBMS that scans a sample of the data whose size is determined by an error metric [2]. Optimized *group-by* queries performed on these samples are used to update the statistics when several modifications of the data have been done.

More precisely, for a numerical attribute A, the data distribution is described by an equi-depth histogram of k buckets denoted by $H_A = \{h_1, h_2, \ldots, h_k\}$ [6]. For each bucket, say h, the RDBMS additionally stores and maintains the *selectivity degree*, denoted by $\sigma_h \in [0, 1]$, that corresponds to an estimation of the proportion of tuples whose values fall in h. As the histograms are equi-depth ones, the selectivity degrees associated with the different buckets of a given attribute should all be equal; in practice, the RDBMS tries to maintain them as

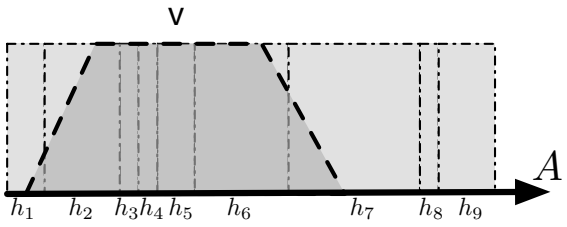

Fig. 3. Histogram-based cardinality estimation of an atomic summarizer

close as possible. It must be underlined that they may contain some imprecision, as they are based on sample estimation.

Beside these histograms, in order to give a more complete view on the data distribution, the metadata tables stored and maintained by the RDBMS also contain, for each attribute, a list of the most frequent values for each attribute.

2) Exploitation for Atomic Summarizers: As recalled above, the extraction of linguistic summaries requires to compute the cardinality σ_S of any summarizer S. In the case of an atomic summarizer, i.e. a summarizer that contains a single term, say $S = v$ associated with attribute A, σ_S is denoted by σ_v. The principle of the computation is to determine the extent to which each bucket of the histogram associated with A contributes to the computation of σ_v, and then to sum up these contributions.

In order to exploit the sole statistics provided by the RDBMS, the STATS approach [14] i) estimates the individual bucket contribution assuming that the tuples are uniformly distributed within each histogram, and then ii) sums up their relative cardinalities weighted by their respective contributions. For example, in the case represented in Figure 3, the cardinality of the bucket h_3 should be fully taken into account, whereas that of h_2 should only contribute with an approximate weight of 2/3.

Formally, the contribution of a bucket h defined as the interval $[h^-, h^+]$ to the cardinality of v is defined as

$$c(h) = \int_{h^-}^{h^+} \mu_v(x)\, p_h(x)\, dx \tag{2}$$

The uniform distribution assumption implies that, for any x, $p_h(x) = \frac{\sigma_h}{h^+ - h^-}$. Hence, denoting by $H = \{h_1, h_2, \ldots, h_k\}$ the histogram associated with A, σ_v is estimated as follows:

$$\sigma_v = \sum_{h \in H} \mathcal{I}_v(h) \times \sigma_h \tag{3}$$

$$\text{with } \mathcal{I}_v(h) = \frac{\int_{h^-}^{h^+} \mu_v(x) dx}{h^+ - h^-} \tag{4}$$

In the STATS method [14], the individual bucket contributions defined in Equation 4 are estimated based on an alpha-cut computation, so as to have a unified approach for both numerical and categorical attributes.

3) Exploitation for Conjunctive Summarizers: Estimating the cardinality of a conjunctive summarizer, e.g. defined as $S = v_1 \wedge v_2 \wedge \ldots \wedge v_f$, comes down to combining the individual selectivity degrees of the different conjuncts, i.e. the $\sigma_{v_i}, i = 1..f$, each σ_{v_i} being computed using Equation (4). An RDBMS indeed only maintains statistics about tuple distribution on each attribute individually, but not on the Cartesian products of their domains.

As the STATS method [14] only exploits the available statistics in an RDBMS, it relies on the assumption that the tuples belonging to a fuzzy subset v_i, $i = 1..f$ are uniformly distributed on the other conjuncts, i.e. the v_j, $j = 1..f, j \neq i$, and it estimates

$$\sigma_{v_1 \wedge \ldots \wedge v_f} = \prod_{i=1..f} \sigma_{v_i} \tag{5}$$

4) Global STATS Approach: The STATS method [14] thus consists in estimating the cardinalities of candidate summarizers S using the previous formula or the alpha-cut-based estimation for atomic ones. The derived estimated related cardinalities are then described by a relative fuzzy quantifier Q such that $\mu_Q(\sigma_S) \geq 0.5$. In the (rare) cases where two adjacent quantifiers satisfy this condition, the left one is chosen.

It has been shown [14] that, compared to classical query-based approaches, the STATS method constitutes a highly efficient approach whose complexity is independent on the number of tuples to process, at the sole expense of some imprecision in the choice of the quantifier involved in the output summaries.

III. The FRELS Approach

This section describes the approach proposed to improve the precision of the estimated cardinalities for atomic summarizers, and therefore the precision of the output summaries, with a little increase in the computational cost. After giving an overview of FRELS, it successively describes the measure that quantifies the uncertainty associated with the estimated cardinalities, and the potential consolidation step. It also presents the efficient implementation FRELS relies on.

A. Overview

The possibly erroneous estimated cardinalities of the STATS approach are due to the hypothesis of a uniform distribution of the data within the histogram buckets and of attribute independence. FRELS proposes a way to quantify the impact of the uniformity assumption, by determining lower and upper bounds of the estimated cardinalities when considering other, extreme, data distributions.

When this impact can be considered strong, leading to an important uncertainty, FRELS consolidates the estimation, performing a local scan of the involved data based on an efficient query, as described below, thus combining the statistics-based approach with a scan-based one, but efficiently limiting the latter to the unavoidable cases. The aim is to limit the propagation of this uncertainty to the conjunctive summarizers as much as possible.

As detailed in the following, FRELS depends on a single parameter that controls the tolerated uncertainty attached to the estimation.

B. Bounds of the Estimated Fuzzy Cardinalities

1) Case of Atomic Summarizers: As discussed above, the hypothesis of uniform data distribution within buckets leads to imprecise partial cardinalities. We thus propose to compute bounds of the estimated values, by considering two extreme distributions, that respectively correspond to the case where all tuples are located at each bound of the considered bucket, h^- or h^+: the first one for instance corresponds to the case where $p_h(x) = 0$ for all x except $x = h^-$ and $p_h(h^-) = \sigma_h$, leading to $c(h) = \sigma_h \times \mu_v(h^-)$. Similarly, the second case leads to $c(h) = \sigma_h \times \mu_v(h^+)$. Taking into account whether μ_v is increasing or decreasing on the considered bucket, the contribution $\mathcal{I}_v(h)$ defined in Equation (4) can be bounded by

$$h_v^- = \min(\mu_v(h^-), \mu_v(h^+)), \quad (6)$$

$$h_v^+ = \max(\mu_v(h^-), \mu_v(h^+)). \quad (7)$$

One may observe that, in the case of a bucket h corresponding to an interval included in the core of v, the lower and upper estimated cardinalities are the same: $h_v^- = h_v^+ = \mathcal{I}_v(h) = 1$: whatever the type of data distribution within such a bucket, it has the same contribution to the final fuzzy set cardinality.

Based on the individual contributions, the lower and upper estimations of the fuzzy cardinality of v are then computed as:

$$\sigma_v^- = \sum_{h \in H} h_v^- \times \sigma_h, \quad \text{and } \sigma_v^+ = \sum_{h \in H} h_v^+ \times \sigma_h. \quad (8)$$

Straightforwardly, one has $\sigma_v^- \leq \sigma_v \leq \sigma_v^+$.

Notice that the computation of these estimations has a very low cost: due to the previous observation concerning the case of buckets included in the core of v, the lower and upper bounds h_v^- and h_v^+ must be computed only for buckets intersecting with the area of gradual transition between full membership and full non-membership of the fuzzy set. In Figure 3, these buckets define the set $H_v^* = \{h_1, h_2, h_6, h_7\}$. For each such bucket h, the lower and upper bounds are computed and combined with their respective selectivity degrees and that of buckets included in the core providing the estimations σ_v^- and σ_v^+ at low cost.

2) Conjunctive Summarizers: To estimate the cardinality of a conjunctive summarizer, the FRELS method considers the same strong hypothesis as the STATS approach (see section II-B3), according to which the subset of tuples satisfying one of the conjuncts is uniformly distributed on the domains involved in the other conjuncts.

Indeed, for the sake of efficiency, RDBMSs do not compute multidimensional histograms, even though many research works have been devoted to this question in the literature [15]: they appear to be too expensive.

Keeping in line with the desire to favour efficiency and to exploit solely the statistics provided by the RDBMS, for a conjunctive summarizer $S = v_1 \wedge \ldots \wedge v_f$, we define the lower and upper bounds of its estimated cardinality as

$$\sigma^-_{v_1 \wedge \ldots \wedge v_f} = \prod_{i=1..f} \sigma^-_{v_i} \quad \text{and} \quad \sigma^+_{v_1 \wedge \ldots \wedge v_f} = \prod_{i=1..f} \sigma^+_{v_i}. \quad (9)$$

It is straightforward to show that σ^- and σ^+ are the bounds of the estimated cardinality as they correspond to the extreme opposite cases of a uniform distribution.

C. Confidence Consolidation using Sample-Based Queries

1) Consolidation Triggering: For a given atomic summarizer, say v, in the case where the bounds show that the uncertainty is large, FRELS performs a consolidation step to get a more precise estimation. This consolidation can help reduce the uncertainty propagation when turning to conjunctive summarizers that contain v.

FRELS relies on the following criterion to determine whether a consolidation step should be triggered for v, to simply measure the uncertainty attached to its estimation:

$$\sigma_v^+ - \sigma_v^- > \eta \quad (10)$$

where η is a user-defined parameter. A low value increases the precision of the results, at the expense of a cost increase, as it triggers more consolidation steps than a greater value.

It is worth noticing that this triggering criterion does not depend on the buckets corresponding to intervals included in the core of v: the latter have the same lower and upper bounds, as underlined in the previous section. Although they may also contain uncertainty, due to the sample-based strategy used by RDBMS to estimate their associated selectivity degrees (see Section II-B), it is not possible to assess this uncertainty. The triggering criterion therefore only depends on the uncertainty in the transition areas of the summarizer and their slopes.

2) Consolidation Process: In the cases where consolidation is triggered, the tuple distribution within the buckets must be better measured, so as to obtain a more precise estimation of the cardinality for atomic summarizers.

As discussed above, even if the triggering criterion only depends on the uncertainty in the transition areas of the summarizer, the uncertainty itself also depends on the cores: if only the transition areas are queried to get their exact cardinality, there will remain some uncertainty from the core. Actually, in some cases where uncertainties in the core and in the transitions compensate each other, querying the transition areas only can lead to a consolidation much worse than the estimations. Therefore it is mandatory to query all data in the support of the summarizer.

Experiments not detailed here show that the most efficient way to do so is also the most precise one. It consists in a query on the whole support of the summarizer, that takes the following form: for an atomic summarizer v with trapezoidal membership function, defined on attribute A, with support bounds $minS$ and $maxS$ and core bounds $minC$ and $maxC$:

```
SELECT sum(CASE
    WHEN minS <= attr AND attr < minC THEN (attr-minS)/(minC-minS)
    WHEN minC <= attr AND attr <= maxC THEN 1.0
    WHEN maxC < attr AND attr <= maxS THEN (attr-maxC)/(maxC-maxS)
    ELSE 0.0
END) as mu
FROM R WHERE minS <= attr AND attr <= maxS;
```

where R is the considered table. The result of the query must

be divided by the total number of tuples in the database, but this number is the same for all consolidations, so it can be queried once. The previous query is given for the case of trapezoidal modalities, which is the most common one. However, other types (right of left shoulders) can be queried as well.

It must be underlined that, for the efficiency of the consolidation step, a single query is executed to consolidate all the modalities identified by FRELS. A single scan of the data is thus needed in the worth case, i.e. when no indexes are defined on the concerned attributes.

D. FRELS Implementation

Beyond the previous combination of the statistics-based approach with local scans of the database, FRELS also differs from the STATS method, which is a purely statistical one [14] in the way it exploits the available metadata offered by any RDBMS. In particular, it relies on a more efficient use of the most frequent values associated with any numerical attribute.

IV. Experimental results

This section presents the experiments run to study the behavior of FRELS and to confirm that it improves the precision of the estimated cardinalities without altering too much the performances of a pure statistics-based approach.

A. Experimental Protocol

The experiments are based on a large dataset describing flights in the US from 1987 to 2008 [12], which contains 7 million tuples described on 13 numerical attributes, as e.g. the flight length or its delay on arrival. A fuzzy vocabulary, identical to the one used in [14], is defined on these attributes, with a total of 60 modalities. These data are stored in a PostgreSQL server[1]. Several sub-bases of varying sizes are extracted from the data set.

The experiments compare 3 approaches: SCAN, a classic scan-based one, STATS, the pure statistic-based approach described in [14], and the proposed FRELS method, whose parameter is empirically set to $\eta = 0.01$.

B. Precision Improvement

We first compare the estimated cardinalities generated by STATS and FRELS with respect to the real cardinalities, computed by SCAN, on 10 different data subsets of 300.000 flights extracted from the whole data.

The results show that, on average over the 10 data subsets, 13 of the 60 modalities require a cardinality consolidation. To measure the gain provided by the consolidation step, we consider as criterion the error rate $|\widehat{\sigma_S} - \sigma_S|/\sigma_S$, where $\widehat{\sigma_S}$ is the cardinality estimated by FRELS or STATS, and σ_S is the actual one, for any summarizer S.

Figure 4 depicts the mean error rate of FRELS and STATS over all summarizers S of different sizes (up to 4) that contain at least one consolidated modality. Only conjuncts of size up to 4 are considered, due to the computational time needed by SCAN to give the exact reference value σ_S in this case (around 40 minutes for the 300.000 tuples).

[1]Postgresql 9.4 running on a 3,1 GHz Intel Core i7 with 16GB of RAM

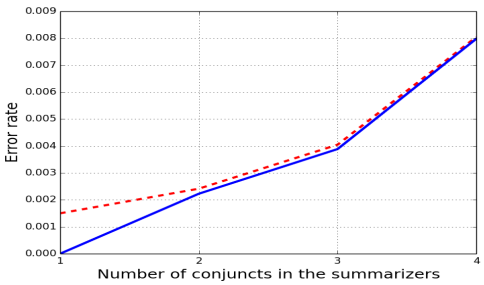

Fig. 4. Decreased error rates for FRELS (plain line) as compared to STATS (dashed line) on average over summarizers of different sizes.

Figure 4 shows that FRELS makes it possible to slightly decrease the error rate compared to STATS. It also shows that this improvement decreases when the number of conjuncts increases: FRELS makes it possible to correct, when needed, the assumption of uniform distribution within a bucket, but still applies the assumption of uniform distribution in the Cartesian product of several atomic summarizers. The latter appears to reduce the precision gain for long summarizers.

C. Efficiency Degradation

The second evaluation criterion regards efficiency: to check whether it is worth performing the proposed consolidation of uncertain cardinality estimations, we quantify the overhead, in terms of processing time, induced by FRELS on various DB sizes, from 300.000 to 7 million tuples. It is worth mentioning that the time needed to perform the consolidation of the estimated cardinalities may significantly vary if indexes are defined on the involved attributes, which is not the case in the DB used here. FRELS thus applies a sequential scan of the concerned table to perform the consolidation step. As described in Section III-C2, a single query, relying on a sequential scan of the considered table, is executed to consolidate the 13 modalities.

For various DB sizes, Figure 5 shows that the overhead (in second and log scale) induced by FRELS is negligible compared to the SCAN approach. For instance, STATS takes a constant time of 0.3 second to estimate the summary of 7 million tuples, FRELS takes 3.5 seconds whereas SCAN requires more than 11 hours.

D. Impact on the Summarization Task

The motivation for computing estimated cardinalities is to allow for efficient linguistic summarization of the considered databases. Therefore a global quality criterion of the proposed approach is the improvement of the extracted linguistic summarizers, more precisely the choice of the correct quantifier, i.e. the one the exact cardinality would lead to.

On the same DB about US flights, it has been shown in [14] that STATS leads to select the correct quantifier for 97% of the atomic summarizers and 92% of the conjunctive ones.

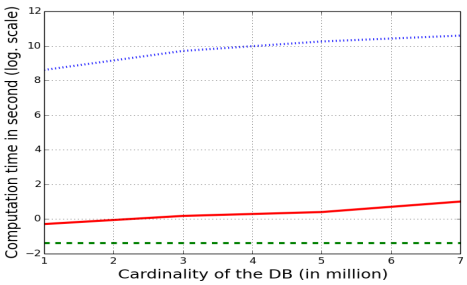

Fig. 5. Processing time overhead using FRELS (plain line) compared to STATS (dashed line), but still very efficient compared to SCAN (dotted line)

More precisely, for the latter, the estimated cardinalities lead to some inversions in the choice of the quantifier to describe rare properties, mainly between *few* and *some*.

For the *Flights* DB, the conducted experiments show that FRELS selects the correct quantifier for 100% of the atomic summarizers and 92.7% for the conjunctive ones. This improvement is consistent with the gain in precision discussed in Section IV-B, that also decreases with the size of the summarizers.

V. Conclusion and Future Works

In order to improve the computational cost of linguistic summarization of large relational data sets (as compared to exact scan-based approaches) and to improve its precision (as compared to approximate statistical approaches), this paper proposes to combine them efficiently: the proposed FRELS method enriches a statistics-based principle with local scan-based correction when needed, that relies on a single query that provides more precise knowledge about the distributions of the tuples on some subsets of the definition domains. Experiments performed on a real data set show that, as synthesized in Figure 6, FRELS improves the precision of the generated summaries, especially for atomic summarizers, while still being incomparably more efficient than a classical approach based on linear data scans.

The experimental results also show that the precision improvement only has a slight impact on the cardinalities estimated for conjunctive summarizers. However, without additional statistics it seems impossible to do better. The next step is obviously to find a way to build (and maintain in an efficient way) some statistics about attribute correlations [5], so as to better estimate the cardinality of conjunctive summarizers. Existing strategies to build and maintain multidimensional histograms (see e.g. [15]) have not been integrated into commercial RDBMSs because of their negative impact on the system's efficiency. This leaves room for the study of DB statistics dedicated to a particular task such as linguistic summarization. A similar research question arises for categorical attributes, beyond the case of the numerical ones considered in this paper: the lists of frequent values maintained by RDBMSs do not appear to provide enough knowledge for improving the strategy used by statistics-based approaches, opening directions for future research.

Fig. 6. SCAN, STATS and FRELS wrt. efficiency and precision

References

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

[2] S. Chaudhuri, R. Motwani, and V. Narasayya. Random sampling for histogram construction: How much is enough? In *ACM SIGMOD Record*, volume 27, pages 436–447. ACM, 1998.

[3] R. Dijkman and A. Wilbik. Linguistic summarization of event logs–a practical approach. *Information Systems*, 67:114–125, 2017.

[4] R. George and R. Srikanth. Data summarization using genetic algorithms and fuzzy logic. In F. Herrera and J.-L. Verdegay, editors, *Genetic Algorithms and Soft Computing*, pages 599–611. Physica-Verlag, 1996.

[5] M. Heimel, V. Markl, and K. Murthy. A bayesian approach to estimating the selectivity of conjunctive predicates. In *BTW*, pages 47–56. Citeseer, 2009.

[6] Y. Ioannidis. -the history of histograms (abridged). In *Proceedings 2003 VLDB Conference*, pages 19–30. Elsevier, 2003.

[7] J. Kacprzyk and S. Zadrożny. On combining intelligent querying and data mining using fuzzy logic concepts. In G. Bordogna and G. Pasi, editors, *Recent Research Issues on the Management of Fuzziness in Databases*, pages 67–81. Physica-Verlag, 2000.

[8] J. Kacprzyk and S. Zadrożny. Protoforms of linguistic data summaries: Towards more general natural-language-based data mining tools. *Soft Computing Systems*, pages 417–425, 2002.

[9] J. Kacprzyk and S. Zadrożny. Derivation of linguistic summaries is inherently difficult: Can association rule mining help? In C. Borgelt, M. A. Gil, J. M. Sousa, and M. Verleysen, editors, *Towards Advanced Data Analysis by Combining Soft Computing and Statistics*, pages 291–303. Springer, 2013.

[10] D. Pilarski. Linguistic summarization of databases with quantirius: a reduction algorithm for generated summaries. *Int. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 18(3):305–331, 2010.

[11] D. Rasmussen and R. Yager. Finding fuzzy and gradual functional dependencies with summarySQL. *Fuzzy Sets and Systems*, 106:131–142, 1999.

[12] Research and Innovative Technology Administration and Bureau of Transportation Statistics. http://stat-computing.org/dataexpo/2009/the-data.html.

[13] E. H. Ruspini. A new approach to clustering. *Information and Control*, 15(1):22 – 32, 1969.

[14] 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)*, pages 1–8. IEEE, 2018.

[15] N. Thaper, S. Guha, P. Indyk, and N. Koudas. Dynamic multidimensional histograms. In *Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, pages 428–439. ACM, 2002.

[16] A. Wilbik and R. M. Dijkman. On the generation of useful linguistic summaries of sequences. In *Proc. of the IEEE Int. World Conf. on Computational Intelligence, WCCI*, pages 555–562. IEEE, 2016.

[17] A. Wilbik and J. Kacprzyk. Towards an efficient generation of linguistic summaries of time series using a degree of focus. In *Proc. of the 28th North American Fuzzy Information Processing Society Annual Conf., NAFIPS'09*, 2009.

[18] A. Wilbik, U. Kaymak, and R. Dijkman. A method for improving the generation of linguistic summaries. In *Proc. of the Int. Conf. on Fuzzy Systems, FUZZ-IEEE'17*. IEEE, 2017.

[19] R. Yager. A new approach to the summarization of data. *Information Sciences*, 28:69–86, 1982.

[20] L. Zadeh. Fuzzy logic = computing with words. *Fuzzy Systems, IEEE Transactions on*, 4(2):103 –111, may 1996.