EDA4SUM: Guided Exploration of Data Summaries

Brit Youngmann, Sihem Amer-Yahia, Aurélien Personnaz

Utsav Pathak, Dhruv Dasadia, Dharmik Dharmesh Patel

College of Computing

Illinois Institute of Technology

30th November 2023

CS520

Data Integration, Warehousing, and Provenance

Dr. Boris Glavic

Fall 2023

Table of Contents

Abstract

The process of creating understandable and representative subsets from an input dataset is known as data summarization. Typically, it is done in a single step with the aim of obtaining the finest summary. A valuable synopsis comprises *k* uniform sets that are individually distinct yet collectively varied enough to be representational. Interpretability is addressed by uniformity, and representativity is addressed by diversity. Finding such a summary in vast, extremely different data sets is a challenging undertaking. They formulate Eda4Sum, which refers to the problem with exploratory data analysis, which aims to progressively construct linked summaries with the purpose of maximizing their cumulative usefulness, and we investigate the application of Exploratory Data Analysis (EDA) to data summarizing. Eda4Sum extends the concept of one-shot summarizing.

They suggest using one of two methods to solve it: (i) Top1Sum, which selects the most helpful summary at every stage; (ii) RLSum, which uses deep reinforcement learning to train a policy that pays an agent for discovering a fresh and varied set of uniform sets at every stage. We contrast these methods with the best EDA solutions and one-shot summarization. We conduct in-depth tests using three sizable datasets. Our findings show how effective our methods are at summarizing enormous amounts of data and how important it is to advise subject matter experts.

List of Figures

List of Tables

1. Introduction

Data Analysis has become an important component of our everyday life, almost everything, right from grocery to our music on spotify to the larger things such as stars and galaxies include analytics. This makes it very necessary to understand the concept of retrieving, collecting and summarizing data.

Data summarization is rather an art of classifying and grouping data to form small bunches of information from a larger entity of data or a large table. This helps us in getting a deeper understanding into similar, domain specific data and making faster analysis since smaller chunks consume less memory and space and are more easy to understand and manipulate (experiment with for analytical purposes).We take an input dataset and create small (k number) of uniform subsets of the input data (superset). This helps in achieving uniform as well as diverse samples to work with, ensuring not just ease of analytics but also significantly reducing bias that could otherwise occur if the data is not diverse.

The authors mention the Sloan Digital Sky Survey (SDSS) dataset in the paper, the dataset has 169 classes of galaxies (as per the Zoo Classification methodology in astrophysics). This 169 classes, with each galaxy having 7 attributes that describe its unique properties and make a uniform 7 parameters of comparison for the galaxies to be compared to each other and analyze their behavior. An isolated, one-shot SDSS report is not typical. In fact, astronomers nowadays invest a lot of time on querying the SkyServer database using SQL. They spend much of their effort reformulating searches and looking for sets of galaxies with comparable characteristics or distributions of values. In this study, the authors examine whether EDA can be used for large-scale data summaries. An array of different *k* sets of things (called itemsets) that are all uniform—that is, consisting of items that are comparable to one another—can be characterized as a summary. The summary is varied since the itemsets are distinct from one another. It's evident that people can understand uniform itemsets more easily. A lot of information may not be easily absorbed by users in a single shot. In order to identify the best, most uniform k number and varied sets, a one-shot summarizing strategy that makes use of a diversity algorithm seems like a sensible answer.

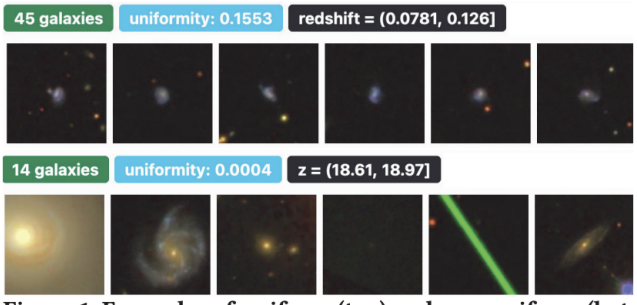


Figure 1: Examples of uniform (top) and non-uniform (bottom) galaxy itemsets.

The problem simplifies to a one-shot summary when the pipeline's fixed length equals 1. Through a reduction to the Heaviest Path issue, they demonstrate the NP-hardness of the Eda4Sum issue. As a result, scalable and effective methods are required to solve it. We use the SWAP technique to bootstrap a pipeline for summarization. Therefore, EDA4Sum simplifies to SWAP for a single shot summary. To create the first step, alternative one-shot summarizing techniques might be applied. Two modifications of current algorithms, Top1Sum and RLSum, are what the authors suggest as next stages. The method generates a fresh summary at each step by selecting one of the itemsets that the preceding step returned and deciding which operator to use on that itemset.
The greedy Top1Sum algorithm selects the greatest utility summary to return at each iteration. They explore the suitability of Deep Reinforcement Learning for summarization, building on previous EDA research. To mimic an operation that can do an end to end summarization strategy as a series of EDA operators that produce the most reward, they create RLSum, a Deep Reinforcement Learning adaption.

2. Previous Works

a) One Shot Data Summarization

Summarizing data in a single go: Many different methods have been suggested to summarize data. The authors use the minimum described length as a technique to provide *k* different groups with shared attributes in the data, ways that detect extreme aggregates, and methods that summarize all aggregates are notable examples. Some approaches treat data summarizing as a one-time effort, which is not the case with their work. Systems that provide an exhaustive summary usually compromise on information loss vs summary size. The most uniform and different components are useful, as was noted in the Introduction, when dealing with large amounts of data.

Therefore, in contrast to other studies, they do not quantify the issues that occur due to information loss because their objective is not to summarize the complete input. As a result, they describe summarizing as the process of identifying the data's most homogeneous and varied subgroups. To address this definition, using diversity algorithms like QAGView, MMR, SWAP, and GMM makes sense. There is a conflict, nevertheless, between addressing diversity in data and showcasing *k* sets, as was covered in the Introduction.

b) Diversification of Results

In database, search engine, and recommender system query responding, result diversity has been thoroughly investigated. The goal of this task is to deliver *k* solutions that balance utility and variety. Utility is frequently sacrificed in the sake of diversity. Pairwise similarities are a standard method of evaluating variety, which they also used in the investigation. The primary distinction from earlier research is that they take into consideration novelty within itemsets chosen in earlier phases, and that the novelty score may fluctuate during the summarizing process.

c) Data Exploration via Multi-step Process

The purpose of the multi-step data exploration method is to draw conclusions from the data. It has been extensively researched to assist users in carrying out data investigation. Several works suggested ideas for the future steps. They created new operators that allow users to interactively explore data and find interesting collections of tuples. Their objective is to summarize large datasets by identifying very consistent and diversified itemsets, in contrast to this line of study which aims to extract general insights. Table 1 further illustrates that, in contrast to earlier research, which made data-driven suggestions, making the further recommendations as operation-driven. This enables the creation of pipelines that take use of semantic connections between data areas while maintaining the user's flow of thinking.

d) Exploration using Machine Learning

Recent research proposed utilizing reinforcement learning to automate data exploration. EDA4Sum uses a similar strategy to provide users direction without requiring training data. Our RLSum technique follows the logic of the approach described in, which helps users locate interesting objects in big datasets. In this approach, interest and familiarity with the facts drive the process. The lack of an extrinsic incentive in RLSum eliminates the requirement for labeled data and previous knowledge. Furthermore, homogeneity, variation, and innovation are the driving forces behind RLSum's iterative summarizing process.

3. Data Model and Summaries

They examine a collection of objects D that are characterized by a collection of (numerical or categorical) ordinal characteristics A. They utilize SDSS to demonstrate the data model without sacrificing generality. It is expected that values of numerical attributes are divided into a set number of bins. The values of d for each attribute $a \in A$ with the vector representation for each item $d \in D$, and are denoted as vd. So, they use the concept of an itemset later on, which is defined as a group of items. With the advantage of quickly expressing the itemset's content, those characteristics constitute the itemset description. D is the collection of all itemsets that were made with D.

Itemsets may overlap, as they noted. Figure 1 provides instances of galaxy itemsets and their descriptions as an example. They use a vector vi to represent each itemset i, which computes the sum of the values of the items in i for every attribute in A. Each vector entry's value is determined by taking the average values of the related attributes in the itemset. One might utilize alternative aggregations, such as the median for ordinal attributes.

Table 1: Positioning of EDA4Sum with respect to Data Exploration and Result Summarization and Explanation.

Related Work		Pipeline		Recommendation		Guidance	
		One-Shot	**Multi-Step**	Data-Driven	**Operation-Driven**	Hands-Free	**Connected**
EDA	[7–9, 44]		✓	✓		✓	✓
	[37, 38]		✓		✓	✓	✓
	[6, 27, 34, 46, 53]		✓	✓			✓
	[32]	✓		✓			
	[15]	✓		✓		✓	
Summarization and Explanation	[5, 20, 41, 50, 51]	✓		✓			✓
	[54]	✓		✓		✓	
	[29]	✓		✓		✓	✓
EDA for Summarization	EDA4Sum	✓	✓	✓	✓	✓	✓

An overview If k is a system parameter, then $I \subseteq D$ is a collection of k that belongs to D. It seems sensible that an effective summary would include item sets with comparable items (uniformity) and sets with items that differ from one another pairwise (diversity). Since their goal was to produce multi-step summaries, one crucial thing to consider was how much the current step's summary exhibits novelty (new itemsets) in comparison to the summaries of earlier stages. In order to determine the usefulness of a summary, they defined the concepts of uniformity, variety, and innovation.

The degree of similarity between items in a summary throughout all of its itemsets is measured by its uniformity. First, we define what an itemset's homogeneity is. Let v(x) be a variance measure, and let y represent variance of items y with respect to an attribute y.

$$uni(i) := \frac{|A|}{\sum_{a \in A} var_a(i)}$$

$$Uni(I) = \min_{i \in I}(uni(i))$$

$$Div(I) := \min_{i,i' \in I, i <> i'} vectorDist(v_i, v_{i'})$$

$$utility(I) = \alpha \cdot Uni(I) + \beta \cdot Div(I) + \gamma \cdot Nov(I, SEEN)$$

4. Application of EDA for Summarization of Data

During the offline stage, they initialized a set-based model and preprocessed the data. Every attribute undergoes equi-depth binning, and mining methods such as LCM are utilized to produce itemsets that may overlap. Modified Reinforcement, learning models are trained. They let users create summarizing pipelines online by choosing one of the following modes: manual in which the user enters the following itemset, operation, and matching characteristics to be applied to the selected itemset after the system provides a summary at each stage.

Partial Guidance: In this approach, the user can furnish only a portion of the necessary information for the subsequent step, with the system displaying a summary at each stage.

Under Full Guidance, a *t*-size summary pipeline is displayed by the system. Both partial and full advice depend on a summarization pipeline being run.

The SWAP algorithm, which determines k number of datasets that have a common ground and ensure diversity, is executed first in the pipeline execution process. As a result, EDA4Sum functions precisely like the SWAP method if the pipeline length is 1 for summarization. One of the RLSum or Top1Sum is executed in the following phases. To produce a new summary, the algorithm selects one from the many available itemsets that SWAP or the prior operator returned, then decides which operator to run on that itemset. The Top1Sum method is a straightforward greedy algorithm that selects an operator to apply at each stage, producing the summary with the maximum utility. RLSum leverages deep reinforcement learning methodology to automatically produce summarization sessions. This cuts down on runtime computation time using this method.

Now the models are already pre-trained, and selecting the optimal predicted action requires negligible inference time. The authors compare the Top1Sum and RLSum findings in their experimental investigation.

5. Algorithms

a) Top1Sum

At each stage of the summarizing pipeline, the Top1Sum algorithm uses local optimization to identify the operation that yields the highest utility summary. Assuming that the itemset is being viewed currently by the user, Top1Sum automatically looks at every potential next step at each step, i.e., every (itemset, *expore*()), attributes) combination, and performs the step that produces the summary with the highest utility. Formally, at each stage, Top1Sum selects the summary I such that the operator used on the itemset $i \in I$ that yields the maximum utility across all operators and input itemsets.

For the Eda4Sum Problem, Top1Sum offers no theoretical assurances. However, as demonstrated by their experimental investigation, Top1Sum performs well in real-world scenarios and can produce good results as pipelines. They saw that even when they had the vectors, the primary limitation of Top1Sum is its execution times, which are quite sluggish. To expedite calculation, the next-step summaries' utility computation might be parallelized.

b) RLSum

Without a model they can solve the issue of determining a pipeline, or policy, that maximizes the discounted cumulative reward by using reinforcement learning. Policy gradient techniques combined with a learnt value function are known as actor-critic approaches. Learning agents through the reward function, each learning episode comprises action probabilities and values that are updated on a regular basis. Using the current estimated benefit of doing that action as a basis, the policy (the actor) modifies action probabilities; value function changes this advantage depending on returns.

$$\pi^* = argmax_\pi \mathbb{E}\left[\sum_{t=1}^{|\pi|} \gamma^i R(s_t, e_t, s_{t+1})\right]$$

6. Implementation and Results

The standard deviation metric is how we calculate utility. In the event that some qualities are categorical, we might employ alternative deviation metrics, such entropy, without compromising our solution. As the vector distance measure, we employ the Manhattan distance formula to quantify variety. With relatively slight adjustments, other vector distance measures might be employed.

A summary's uniqueness, diversity, and homogeneity are all trade-offs. It could be more expensive to find a very consistent and varied summary than to return a unique one. The preferences of the user may alter at various stages of the summarizing process. Assume that the user has viewed several itemsets in earlier phases. In this instance, returning a consistent and varied summary is more crucial than a new one. We experimented with two changing weight schemes—increasing novelty and decreasing novelty—to capture this. The length of the pipeline, count of itemsets, and the number of observed itemsets are the functions that determine the novelty weight for these systems. There will be comparisons between this weighing system and others, such fixed-value weights (like balanced weights).

Table 2: Examined Datasets.

Dataset	|items|	|atts|	|itemsets|	|ground truth itemsets|
SDSS	2.6M	7	348, 857	169
SPOTIFY	232, 725	11	2, 204, 806	27
FOOD	11, 762	11	226, 381	22

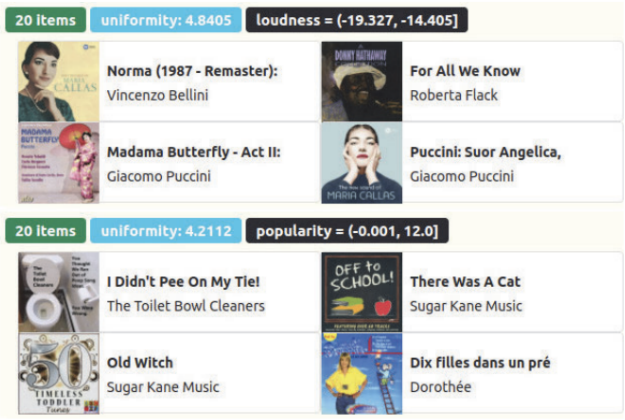
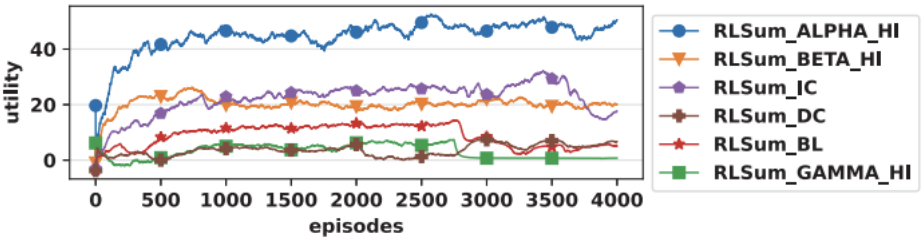


Figure 2: Example of song itemsets.



SDSS

Figure 3: Cumulated utility during training.

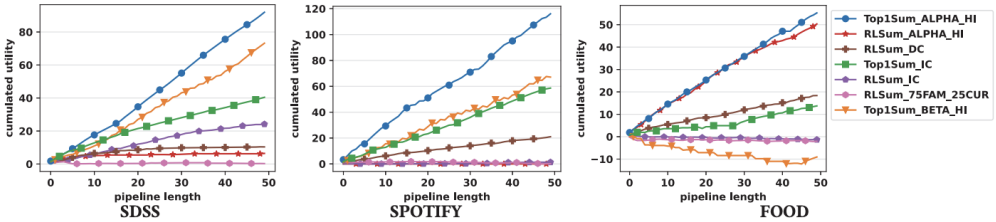


Figure 4: Cumulated utility as a function of pipeline length.

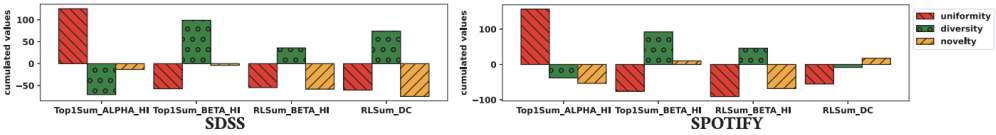


Figure 5: Cumulated uniformity, diversity and novelty during pipeline execution.

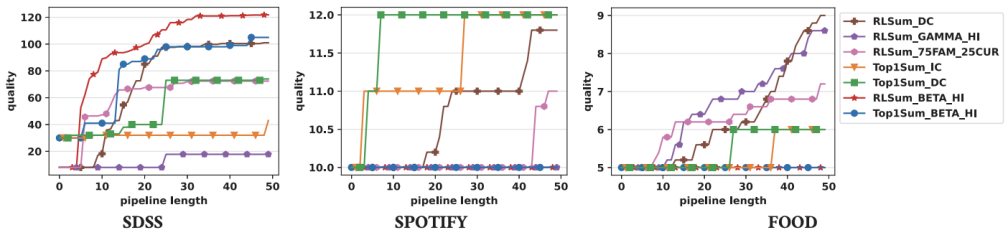


Figure 6: Cumulated relevance to a ground-truth as a function of pipeline length.

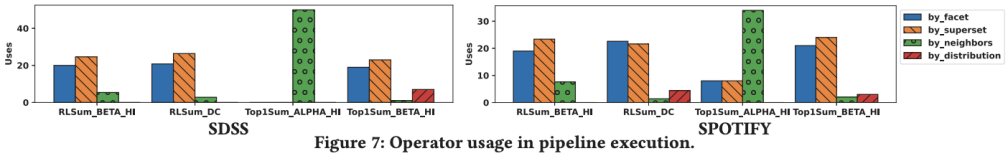


Figure 7: Operator usage in pipeline execution.

Table 3: Average pipeline execution times (in seconds).

Variant	Data size			# of attributes			# of bins		
	23K	115K	233K	3	7	11	5	10	20
Top1Sum	4.3	17.8	21.1	0.6	4.1	21.1	19.7	21.1	32.8
RLSum	0.4	0.7	1.1	0.4	0.5	1.1	1.4	1.1	0.8

Table 4: User study.

Mode	|itemsets|	utility	uni.	diversity	novelty
Manual	67	7.34	16.97	−0.5	3.26
Partial Guidance	142	2.35	−15.45	39.9	−24.77
Full Guidance	101	10.6	−59.51	68.07	−71.58

Figure 8: A relevant summary found with partial-guidance.

7. Limitations of the Analysis

A single step's running time is calculated from the moment an operation is selected until a summary is shown. Every pipeline is run under full guidance, and the average of five executions is reported. Results on SPOTIFY are shown in Table 3. The trends were similar in other datasets. Since the EDA variations' running times are identical to those of RLSum, they are excluded. Only two variants—Top1Sum_HU and RLSum_DC—are compared because running times are unaffected by weights. As anticipated, the outcomes unequivocally demonstrate that RLSum performs certain things better than Top1Sum, and that the gap between the two grows as data size, attribute count, and bin count rise.

As Top1Sum compares each itemset to every potential subsequent operator in order to get the greatest utility outcomes, the amount of itemsets that each operator returns determines how long it takes to execute. Execution times rise as the bins grow since more aspects and consequent itemsets are produced. It's interesting to note that while the number of mined itemsets decreases, RLSum's performance increases with more bins. These outcomes demonstrate that RLSum is the recommended technique for interactive summarizing. Given that RLSum requires a lengthy training period, Top1Sum is still a possible preference even if it yields the greatest utility summaries.

However, there is a massive scalability issue and issue of high memory and space consumption.

8. Conclusion

We realized that a lot less work was done on EDA using Top1Sum and even without any theoretical work or evidence, the authors experimented with it and found that it performed really well.

For RLSum, while using Deep Reinforcement Learning methods is a very good choice, it comes at the cost of high memory and space consumption and if we look at extremely large datasets, the output will not validate the cost put into the calculation and analysis.

We suggest, on our own, to look at cheaper analytical methods, if the cost is a prominent issue for such analysis.

One such example is as below, which was done by an open source contributor and posted on stack overflow, where they use VTE to encrypt and use encryption to store data and limit it to certain longer analysis groups.

```
WITH VTE AS(
    SELECT *
    FROM (VALUES('J','S',10000),
                ('P','A',15000),
                ('S','S',7500))V(Emp, Comp, Sal)),
CTE AS(
    SELECT Comp,
           Sal,
           MIN(Sal) OVER () AS MinSal
    FROM VTE)
SELECT Comp,
       Sal
FROM CTE
WHERE Sal = MinSal;
```

9. Scope of Future Work

We need a better understanding of the RLSum and not treat it as a blackbox. We also need to figure out cheaper ways to fund shortest paths to analyze to reduce cost for larger datasets.

Image data and even sound data could be analyzed using quantum modified circuits which could be of great help for images and data from platforms such as spotify.

Another addition would be NISQ modifications and quantum circuits to enable more in-depth analysis of images.

10. References

- EDA4SUM: Guided Exploration of Data Summaries, Aurélien Personnaz, Brit Youngmann, Sihem Amer-Yahia, Proc. VLDB Endow.15 (12), 3590--3593, 2022.

- Quantum ML Algorithm for Optimizing digital data using Enhanced Quantum Classifier Techniques in Learning Methods P. Mano Paul, Utsav Pathak, Siddhartha Das, Saurabh Kumar, Rakshit Govind T