Content-based Analytics: Moving Beyond Data Size

Dimitrios Tsoumakos
Department of Informatics
Ionian University
Corfu, Greece
dtsouma@ionio.gr

Ioannis Giannakopoulos
Computing Systems Laboratory
School of Electrical and Computer Engineering, NTUA
Athens, Greece
ggian@cslab.ece.ntua.gr

***Abstract*—Efforts on Big Data technologies have been highly directed towards the amount of data a task can access or crunch. Yet, for content-driven decision making, it is not (only) about the size, but about the "right" data: The number of available datasets (a different type of volume) can reach astronomical sizes, making a thorough evaluation of each input prohibitively expensive. The problem is exacerbated as data sources regularly exhibit varying levels of uncertainty and velocity/churn. To date, there exists no efficient method to quantify the impact of numerous available datasets over different analytics tasks and workflows.**

This visionary work puts the spotlight on data content rather than size. It proposes a novel modeling, planning and processing research bundle that assesses data quality in terms of analytics performance. The main expected outcome is to provide efficient, continuous and intelligent management and execution of content-driven data analytics. Intelligent dataset selection can achieve massive gains on both accuracy and time required to reach a desired level of performance. This work introduces the notion of utilizing dataset similarity to infer operator behavior and, consequently, be able to build scalable, operator-agnostic performance models for Big Data tasks over different domains. We present an overview of the promising results from our initial work with numerical and graph data and respective operators. We then describe a reference architecture with specific areas of research that need to be tackled in order to provide a data-centric analytics ecosystem.

I. Introduction

Undeniably, data volume has been the decisive driving force behind big-data technologies. In many cases, the effectiveness of an algorithm relies entirely on the amount of data it can access [1]. As such technologies mature and evolve, emphasis is steadily placed on areas not solely related to scale. A different type of challenge shifts attention to the actual content: Content-based analytics [2] process data from social media platforms for sense-making and knowledge generation. Similarly, data content plays a key role in the quality of the insights derived in applications such as recommendation systems [3], web advertising and marketing [4], fraud detection [5], credit analysis [6], etc.

Data quality is broadly defined as a measure of how "fit for purpose" utilized sources are in the context of existing business operations and analytics. Though data has become an increasingly strategic element for companies, data quality remains a significant challenge. Maintaining data quality has been indicated as the most significant challenge around big data initiatives. Recent research indicated an average financial cost of $15 million per year per organization due to poor data quality [7].

Data scientists have recently made a case about medium data analytics [8], [9], where the size of the data is not considered to be the critical factor. It is becoming increasingly apparent that, for data-driven decision making, it is not (only) about the size, but about the "right data" [10], [11]. In these cases, analysts increasingly need to focus on *high-impact* data, i.e., intelligence that has the best potential of driving strategic decisions.

Given an ever-increasing availability of data to be processed [12], evaluating the utility and performance tradeoffs of immense numbers of inputs (or even worse so, their combinations) is prohibitively expensive, especially given the fact that analytics workflows have evolved into increasingly long and complex series of diverse operators. Yet, the availability of immense numbers of inputs and their impact on analytics performance is not the only obstacle. The challenges a data scientist faces in her quest for the "right data" are many-fold in today's analytics landscape:

- Information sources are incomplete in nature, i.e., without information regarding one or more of relevant variables. In such cases, more or different semantic information is required for the task to achieve its objectives.
- Information sources contain fuzzy data. Regardless whether the input mechanisms, processing, or collection infrastructure is to blame, uncertainty in data values causes highly uncertain insights.
- Data items are getting outdated fast via frequent updates, so they become irrelevant for current or future analysis.
- Information sources are both plentiful and accurate, but too large to handle within certain limits. Scalability issues now relate to the inability of processing all available inputs in order to choose the best ones to maximize a user-define impact measure within a time-frame.

A. Two Sample Use-cases

In derivative pricing theory [13], analysts need to consider a multitude of Credit Default Swaps (CDS) spreads for different economic entities. These are provided as input to Value Adjustments (xVAs), entities that quantify the trade, credit, funding and financial costs during derivative transactions. The

This research has received funding from the European Unions Horizon 2020 Research and Innovation programme under Grant Agreement no.824115 (HiDALGO)

978-1-7281-7022-0/20/$31.00 ©2020 IEEE
DOI 10.1109/BigDataService49289.2020.00013

calculation of xVAs, in addition to being mathematically and computationally complex, is subject to huge liquidity risk, due to lack of available information to obtain key inputs to the models used. Current methodologies do not take into account the volatility of the CDS spreads, which is an important criterion set by the European Banking Authority. Analysts currently need to consider all CDS datasets and execute the aforementioned operators exhaustively in order to select the ones that make them more suitable for specific entities and maximize accuracy. Identifying the relationships between datasets is crucial to obtain better clustering of financial institutions having similar CDS spreads for more accurate proxy calculation. Moreover, the volatility and uncertainty of the CDS spreads must be taken into consideration: By modeling CDS data and its interrelations based on the granularity and amount of observed change, proxying estimations will closely and timely adjust in accuracy.

Security Information and Event Management (SIEM) systems [14] are commonly deployed to identify and mitigate cyberattacks. Yet, they increasingly fail to identify advanced persistent attacks because of their inability to cope with the increasing amount of available datasets utilized to train them [15]. Without such timely processing ability to reference multiple data sets for segmentation and pattern detection, SIEM systems often lack the context in which they could detect advanced threats. As a result, they cannot rely on collected logs but must be trained using low-level traffic analysis. Moreover, the highly volatile nature of collected information further exacerbates this issue: The operation of a single 24-port gigabit router may generate, under full utilization, approximately 50TB of data on a daily basis. Hence, administrators need to make a key decision: Which training datasets out of the available ones should be used to stop cyberattacks of a certain type within tight time constraints? How often should this process be repeated due to different attacks or network data collected?

In both cases, data operators have access to multiple datasets (CDS time series and SIEM training datasets for the two cases respectively). However, their outputs entirely rely on the selection of a mere subset of them, based on a set of properties that are neither known nor easily identified. For example, one cannot decide a priori which CDS datasets are more suitable for the xVA estimation for a given economic entity, even if experience or prior knowledge is available [16]. Similarly, it is hard to select the most appropriate SIEM training datasets to be used for training for intrusion detection: Stringent time constraints and the streaming nature of the data require continuous and cost-effective efforts.

B. Problem Statement

To date, there has been no effort to connect analytics performance (and in multiple dimensions – not exclusively based on execution time) with the input data and specific structural, semantic and operational dimensions of it. For a specific operator or workflow, how can one rank or tangibly characterize input datasets relative to their effect on job execution? We wish to predict the performance of an algorithm, a class of algorithms or a workflow given massive numbers of possible data inputs with diverse structure, content, uncertainty and churn. What are the key semantic, structural or operational elements of input that make it desirable to a specific analytics job? What is the best feature mix that maximizes performance and which of the available inputs have it? Ultimately, what are the best inputs, given the tasks and analyst-defined criteria (time & quality constraints)? We need a performance model that enables us to achieve both low overhead and adaptivity by identifying configurations that guarantee application performance and limit the search overhead for recurring big data analytic jobs.

The challenging goal is to develop a sound methodology and tools in order to quantify the effect of dataset(s) to an analytics operator as well as a sequence of such tasks (a workflow). The proposed work operates orthogonally to the speedups and big-data crunching abilities of modern tools and frameworks, such as Hadoop, Spark, Flink, etc. It provides a novel modeling, planning and processing layer that assesses data quality and maps it to big data analytics performance. This allows powerful and continuous evaluation, planning and execution of business intelligence that drastically accelerates current platforms' performance by adding a content-aware dimension. Utilizing the most advantageous data inputs can result in massive speedups on both accuracy (given a limit to processing time) and time to completion (given a desired accuracy level for the task). The concrete parts of our vision proposal are summarized as follows:

- A novel data modeling methodology that provides the missing link between key dataset properties and analytics quality. We make the argument that dataset interrelations can be a powerful and scalable means of estimating multiple qualitative performance metrics for real-world, popular analytics operators.
- An analytics profiling methodology that builds on the previously created data models and produces a model of a specific operator's performance over different input properties. The incurred models allow for intelligent meta-analytics such as accurate prediction of task performance, similarity, top-k and range querying over available datasets, multi-objective optimization, etc.
- Extend the methodology in order to: (i) include operators with multiple inputs and (ii) provide results for a series of big data operators (workflows).
- A volatility and veracity framework that handles streaming data and updates as well as data uncertainty. Data and analytics models should incorporate varying levels of churn and uncertainty; efficient mechanisms to update them under specific cost and quality guarantees should be provided.

II. Related Work

To facilitate dataset analysis, two complementary directions have been suggested: Data Integration and Data Exploration. Data Integration approaches (e.g., [17]–[19]) aim at presenting

a unified view of distinct datasets and focus on the systemic problem of fusing data from heterogeneous sources. The outcome of these approaches is a set of metadata that reflects information regarding their origins, versions, schema, indexing, etc., as well as a mapping that enables consolidated use by a single application. However, this outcome does not target predicting the effects that different datasets have over application or analytics task execution.

Data Exploration approaches (e.g., [20]–[23]) aim at producing dataset summaries in order to inform the users about properties of the data, such as tuples that encapsulate the most representative data patterns, dependencies between tuple fields, statistical properties and tuple summaries. The main motivation behind these techniques is to assist exploratory analysis: Quickly obtain a view of the data schema and its properties so that one can incrementally issue more complex queries or choose a representative subset of the original data according to specific criteria. Data Exploration is commonly utilized in the first analysis steps of unknown datasets and is still largely manual. It cannot be directly used for identifying a dataset under specific criteria, as these approaches do not target detailed dataset evaluation and comparison, but retain a more informative role. Furthermore, Data Exploration is mainly focused on providing information regarding a specific dataset to a user.

Data Cleaning approaches (e.g., [24], [25]) are much more intrusive and aim at cleaning datasets from erroneous tuples that distort the represented knowledge either through value rewriting or through removing tuples that seem incorrect. To date, none of the aforementioned approaches handles the problem of predicting analytics performance over multiple available data inputs and their characteristics, specifically structural and semantic dimensions – type, size, skew, accuracy, freshness, etc. The affluence in available sources and input combinations as well as the volatile and often uncertain nature of data highly intensify the problem.

III. Methodology

Data-driven analytics optimization, formulated through data selection to maximize performance, establishes an orthogonal, yet equally imposing accuracy, modeling, and scalability challenge. Considerable progress has been achieved in speeding up analytics workflows that utilize huge data inputs; how to intelligently and efficiently evaluate the utility of numerous (both changing and possibly fuzzy) datasets over multiple tasks and workflows has not been thoroughly studied to date. Faced with this challenge, we propose a novel ecosystem that puts the spotlight on data content rather than on individual dataset size. Such a system would realize its goals by: (a) quantitatively modeling the interrelations among large numbers of input data; (b) utilizing these models for intelligent prediction of the quality of variable analytics tasks/workflows; (c) native management of data volatility and uncertainty; (d) powerful meta-tools that evaluate data utility and enable analysis over data inputs and their relation to analytics workflows. The natural question it aims to answer is: What is the relationship between vast input datasets and analytics performance? Based on this, how can we consistently and continuously evaluate the utility of different data in hand, at different levels of granularity?

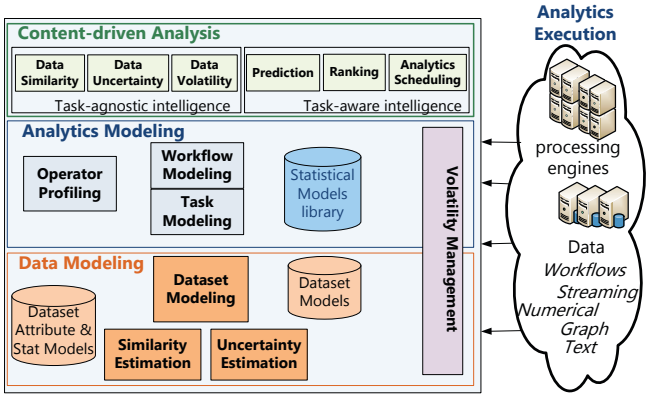

Fig. 1. Ecosystem Architecture

To realize this vision, we describe a reference system architecture of three layers which correspond to required research/engineering challenges that need to be tackled (see Figure 1):

- The Data Modeling Layer, which is the layer hosting modules that analyze input datasets of various types (domains), churn, velocity and degree of uncertainty. The analysis should strive to be *operator-agnostic* and focus on different dataset attributes and their interrelations. Data models should be scalably created, becoming a key asset to be consumed by higher-level modules.
- The Analytics Modeling Layer, which is the layer hosting the core execution and profiling functionality. Given an analytics task and the created data models, this layer handles the necessary planning, orchestration, execution and modeling tasks required in order to produce statistical models of the task's output over any available valid input. The layer also extends the methodology to tasks that accept multiple input datasets, workflows of tasks and model updates due to incoming datasets or churn.
- The Content-Driven Analysis Layer, which encapsulates the application (analyst-driven) logic. Specifically, it allows analysts to interact with the ecosystem, query and intuitively retrieve valuable information relative to analytics performance prediction, ranking and intelligent analysis.

A. Current Research Output

Assume a set of datasets $D = \{D_1, D_2, \cdots, D_N\}$ and an analytics operator F. Let us also assume that F accepts a single dataset as input and produces a scalar output value:

$$F : D \rightarrow \mathbb{R} \quad (1)$$

Each operator can be viewed as a function that projects any dataset D_i to a scalar value $F(D_i)$. The problem the work in [26], [27] addresses, is the following: *We seek for an accurate approximation of $F(D_i)$ without exhaustively executing F*

for all datasets. Typical function approximation cannot be applied in this problem because D represents an unordered set of datasets that do not belong to a *metric space* and the relationships between them are unknown. Albeit constructing a metric space for any given D is possible for a given distance function for each dataset pair in D, the quality of the approximation is heavily affected by the choice of this function. The chosen distance function must reflect the distance between two datasets D_i, D_j, both in the aforementioned metric space and the operator's output domain.

In [26], we argue that there exist some fundamental properties that can produce invaluable insights regarding an operator's outcome. Examining data interrelationships in light of a handful of fundamental statistical properties can generate a strong knowledge basis: If one quantifies the similarity between all pairs of datasets and executes an operator for only a handful of them, a first idea of F's domain would become available, as datasets with high similarity would present similar behavior. Let us generalize this idea: Given the relationship between dataset similarity and an operator's output, we seek for a projection of the datasets in D into a metric space D' (also referred to as *dataset space*) that best reflects the resemblance among them. D' can be then utilized by F as the domain space – according to Equation (1) – in order to project the original datasets into the anticipated values. Interestingly so, the relationships between datasets are *independent* of F, allowing different operators to be applied over a unique D'. For each operator, one could sample D, estimate F's values for the selected datasets $D_i \in D_s \subseteq D$ and approximate F for the rest of the datasets utilizing Machine Learning (ML) techniques. Although F is applied to some of the original datasets, i.e., $F(D_i), D_i \in Ds$ is calculated, the ML model is trained using D' as the input space and the approximated operator F' is defined as: $F' : D' \rightarrow \mathbb{R}$. Essentially, D' comprises a set of *features* that best characterize the datasets' interrelationships. Figure 2 depicts an overview of the applied methodology.

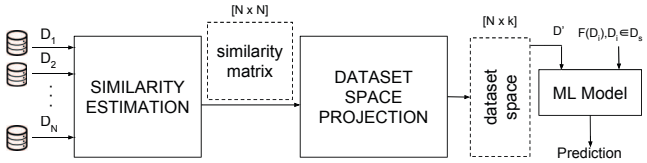

Fig. 2. Methodology workflow

The *Similarity Estimation* module quantifies the similarities between datasets $D_1, \cdots, D_N$. In [26], datasets consist of multi-dimensional tuples with numerical values, while similarity is measured over statistical distribution, dataset size and tuple order. In [27], datasets are graphs and several similarity measures (degree-distribution, graph-kernel-based, etc) are utilized. The outcome is a symmetrical $N \times N$ similarity matrix accessed by the *Dataset Space Projection* module which transforms the original similarities into a metric space. In this step, Multidimensional Scaling [28] transforms the similarity matrix into a set of points in a lower-dimensional (k-dimensional) space, with the property that the distances between the points of the space approximate the similarity represented by the original matrix. The final outcome of the process is a $N \times k$ matrix that represents the coordinates of each dataset in the dataset space. Finally, an operator F can be executed for a small subset of datasets D_s. Using the dataset coordinates and the respective operator values, a Neural Network is trained in order to approximate F for all datasets. Based on the approximated dataset scores, interesting questions can be answered: Which are the dataset(s) with the highest/lowest F values (e.g., with the highest first eigenvalue), how many dataset outputs are close to a given F value, retrieve the top-k datasets under certain criteria, etc.

Essentially, the proposed approach shifts the computational burden in the first phase of data analysis: The workflow presented in Figure 2 is executed once in an *offline* manner for all datasets and is *operator-agnostic*: The similarity estimation does not force the execution of any operator — only the relationships between the input datasets are evaluated. Whenever a new operator emerges, it is executed for a mere subset of the available datasets and its behavior is rapidly approximated with minimal computation. The overhead of the similarity estimation and Dataset Space Projection modules is amortized and the avoided computation linearly increases with the number of operators that need to be executed for the analyzed datasets. A very wide range of operators have been approximated under this approach (distance, connectivity, spectrum-based graph metrics and aggregate functions, ML operators, spectrum and time-series forecast) with good accuracy.

The *Apollo* system that implements this methodology and allows data scientists to import datasets, define and utilize custom similarity functions and execute the process over any analytics task has been described [29] and open-sourced[1]. *Apollo* is written in `Go` and utilizes `R` to model the input datasets and train different Machine Learning classifiers for predicting the operators' output. The system offers a promising proof that simple, intuitive similarity metrics (over two data formats, `csv` files and graph data represented as raw files containing graph edges) can be used for accurate, content-driven prediction. Moreover, massive speedups (more than $20\times$ in many cases) are experienced compared to exhaustively executing the operators.

As an overview of the results *Apollo* can achieve, we consider a set of 973 ego graphs from Twitter (*TW*) [30] and a set of 1442 datasets with daily household power consumption measurements (*HPO*) from a household in Denmark [31]. For TW, we consider Betweenness Centrality (bc) and PageRank (pr), two widely used node centrality measures generalized to the graph level through Freeman's method [32]. For HPO, we model the Average (avg) of each dataset and the number of clusters created after performing a DBSCAN (dbs). We present the Median Absolute Percentage Error *MdAPE*, as a measure of accuracy, and speedup results for two different

[1]https://github.com/giagiannis/data-profiler

sampling ratios (p=5%,10%) in Table I. The sampling ratio indicates the number of datasets for which the operators were executed in order to obtain the real outputs. These values were, subsequently, used by ML classifiers in order to approximate the operator's output for the rest of the datasets.

TABLE I
MODELING ERRORS AND SPEEDUPS OF THE APOLLO SYSTEM

Dataset	Operator	MdAPE (%)		Speedup ×	
		p=5%	p=10%	p=5%	p=10%
TW	bc	17.8	17.5	13.0	7.8
	pr	9.2	7.7	13.2	7.9
HPO	avg	1.3	1.2	3.93	3.4
	dbs	14.6	14.1	8.3	6.23

Finally, the process is customizable in order to accelerate data analysis and conduct less detailed dataset examination or increase modeling accuracy when higher execution time is affordable.

B. Extensions

Apollo was designed and implemented with the notion that all datasets are available from the very beginning of the data analysis process. However, this hypothesis does not hold in multiple real-world scenarios, where existing data sources may be updated (e.g., temperature sensors provide more data points with time) or totally new data sources emerge (e.g., data from more days are available in the *HPO* case above). In such cases, *Apollo* needs to re-execute the workflow of Figure 2 from scratch. Given that the number of datasets is anticipated to be constantly increasing, one can observe that the quadratic complexity of the workflow will decelerate the speedup observed due to avoiding the exhaustive operator execution. To this end, we briefly introduce two extensions added in the *Apollo* engine. The first one describes an online process that allows the introduction of new data sources without repeating the entire workflow. The second one describes an approximation to avoid the quadratic complexity calculation of all pairwise similarities between the datasets.

1) Accommodating new data sources: The problem of projecting new datasets in the dataset space, also referred to as the *Online Indexing* problem, is the following: Given a set of datasets $D_1, \cdots, D_N$ along with their coordinates $p_1, \cdots, p_N$ respectively in a k-dimensional space, find the coordinates p_{N+1} of a new dataset D_{N+1}. Note that the similarities between D_{N+1} and $D_1, \cdots, D_N$ are unknown, but easily computable. Assuming that $d_1, \cdots, d_N$ are the distances between D_{N+1} and $D_1, \cdots, D_N$ respectively, we seek for a vector $p_{N+1} = (x_1, x_2, \cdots, x_k)$ that minimizes the *distortion* of the dataset space, i.e., the difference between the pairwise distances as measured by the similarity matrix and the dataset space. Note that this time, the coordinates of $D_1, \cdots, D_N$ are fixed and only the coordinates of D_{N+1} need to be updated. Given that, the problem reduces to a typical optimization problem with the objective of finding the vector p_{N+1} that minimizes a *distorion* function. Since the problem space is not convex (as more than one local minima may exist), Simulated Annealing (SA) is employed. If the execution time of the Online Indexing process needs to be minimized, one can estimate the distances between D_{N+1} and a subset of $D_1, \cdots, D_N$. This option reduces both the time needed to measure the similarity between the datasets and the number of steps needed by SA in order to converge, because it essentially reduces the constraints of the objective function.

In order to evaluate the performance of this extension, we consider the following experiment: Based on *HPO* data from 2008 (366 datasets), we construct the dataset space using the Distribution Similarity metric. We then "insert" datasets for the next 3 months of 2009, i.e., introduce 90 new datasets. For each new dataset, the similarity with m of the existing datasets is measured and SA is executed to identify the best coordinates for the new entries. Parameter m is expressed as a portion of the already calculated datasets. Using the Sammon Stress (E_s) [33] in order to quantify the space distortion and the Normalized Mean Squared Error, in order to quantify the modeling accuracy (trained with a sampling rate of 16%), we compare the cases where the new datasets are Online Indexed for varying m against the case where Dataset Space Projection is executed from scratch for the old and the new datasets. Figure 3 provides our findings expressed in relative terms, i.e., both E_s and $NRMSE$ are normalized with the respective values for the case where the workflow is executed from scratch.

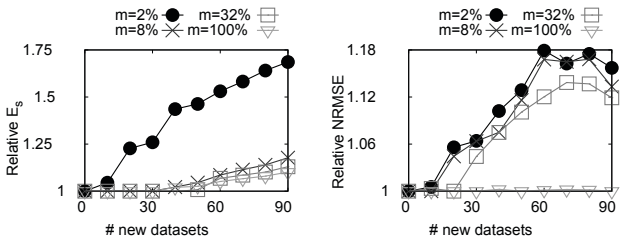

Fig. 3. Performance of "online" accommodation new data sources

When a small number of datasets is introduced (i.e., 10), our optimization achieves both minimal E_s values and minimal modeling error. This renders our approach most suitable for cases where the insertion of only a few datasets is required. When the number of new datasets increases, E_s rapidly increases for two reasons: First, the new datasets have a stronger impact and the introduced errors propagate to the new entries. Second, while new datasets arrive, the dynamics of the space change. This means the dimensionality of the space would differ if all datasets were available from the beginning. Comparing against extremely few datasets generates higher errors, hence, the rapid E_s increase for $m = 2\%$.

The above behavior is also observed for the modeling error, especially when m receives low values. In this case, the coordinates of the new datasets become increasingly inaccurate and this severely impairs accuracy. Interestingly, when $m = 100\%$, the modeling accuracy follows the accuracy achieved when the workflow is executed from scratch, even when 90 new datasets are inserted. However, even with a

considerable $m = 32\%$, NRMSE increases after 30 new datasets are introduced. In conclusion, this optimization is capable of dynamically introducing new datasets by executing a marginal number of similarity comparisons (2%) with a tolerable modeling error increase of 6%, provided that the number of new datasets does not exceed 10% of the existing ones. When this percentage increases, one should first use an increasing number of datasets for the comparisons; after a certain point, execution of the workflow from scratch is suggested.

2) Approximate Similarity Matrices: The methodology discussed so far entails the calculation of a squared matrix of size N^2, N being the number of datasets, and the complexity equals $\mathcal{O}(N^2 x)$, in which x represents the complexity of the employed similarity metric. For an increasing number of datasets, a quadratic complexity becomes prohibitive, as the computational effort required grows rapidly. A way of tackling this challenge is to avoid the calculation of similarities for all the distinct dataset pairs. However, this could lead to information loss, since the non-computed similarities should be replaced by values that approximate them, else this "approximate" similarity matrix may distort the dynamics of the space.

Fig. 4. Dataset distances

In order to provide a solution, assume the datasets depicted in Figure 4 projected to a 2-d dataset space, in which the thin lines represent the distances among them. The distances between the datasets from the left and the right sides are much larger when compared to the distances of the datasets of the same side, e.g., $d_{2,3} < d_{1,2}$. Furthermore, assuming that only $d_{1,2}$ is known, one can say that $|d_{1,2} - d_{2,3}| \leq d_{1,3} \leq |d_{1,2} + d_{2,3}|$ and if $d_{2,3} \ll d_{1,2}$ then $d_{1,3} \approx d_{1,2}$. In other words, in this example one only needs to calculate one of the "large" distances in order to avoid high approximation error. This interesting observation highlights the necessity of prioritizing for large distances when considering which of them should be evaluated. When such a "backbone" of distances is calculated, e.g., the set of thick lines of Figure 4, one can easily estimate the distances between the unknown pairs, providing the distances between the closest – to them – known datasets. Such a "backbone" of nodes can easily be built: We pick a random dataset and calculate its similarity against the rest of the datasets. We, then, pick the most dissimilar one. We continue this process, always selecting the most dissimilar datasets to the ones that have been seen so far, until we reach the desired number of examined datasets.

In order to evaluate this optimization, we design the following experiment: based on the *HPO* datasets, we construct similarity matrices based solely on the dataset distribution property, estimate an approximate similarity matrix, in which only t datasets are fully calculated (expressed as a percentage of the total number of datasets). In the left plot of Figure 5, we provide the relative construction time of the similarity matrix for varying t values, i.e., the ratio of the time needed to construct the matrix for the $t = 100\%$ case, divided by the time for each t value.

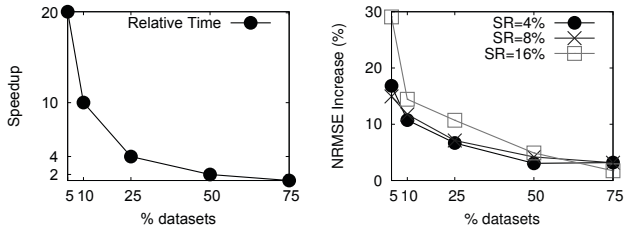

Fig. 5. Approximate Similarity Matrix Evaluation

Our optimization linearly reduces the computation time. In order to evaluate the impact to the modeling accuracy, dataset spaces are constructed based on the similarity matrices and SVM models are trained using the respective spaces for the *avg* operator. In the right plot of Figure 5 we provide the NRMSE increase, i.e., $\frac{NRMSE_t - NRMSE_{100\%}}{NRMSE_t}$, of each model for varying t values for three sampling rates. The drop in accuracy becomes increasingly important when higher sampling rates are employed. However, the error degrades quickly with increasing t values, and even when $t = 10\%$, the introduced error does not exceed 14% compared to the full similarity matrix but, simultaneously, the construction is accelerated by a factor of $10\times$. Therefore, our optimization is able to significantly speedup the construction, introducing a relatively small increase in error.

IV. Research Areas

A significant amount of research questions and development activities lie ahead in order for a content-driven ecosystem to be functional and extensible. Relative to the architecture depicted in Figure 1, we now discuss important areas/topics where significant research is required for the realization of this environment.

A. Data Modeling

The goal of data modeling is to capture *structural*, *semantic* and *operational* information pertaining to datasets that organizations and analysts alike utilize as possible inputs to their analytics tasks. In the former group, properties such as structure type (tabular, free-text, graph, hierarchical, etc), size, location (local disk, file system, remote access, etc), distribution, skew, order are investigated. In the semantic category, various properties not covered by purely formatting aspects are considered. Examples include (but are not limited to) data freshness, density/distribution of specific feature(s) or values, etc. Lastly, operational features relate to the degree of fuzziness and change that is observed on the underlying sources. As such, the level of data uncertainty, the speed of incoming data streams, the amount of changes (be they additions of new data points or updates of existing ones) are all properties to be considered.

The solution should span the whole range of dataset type and task domains. As such, the Similarity Estimation module

plans on identifying, for each captured dataset feature, suitable and low-cost similarity functions. These functions take two distinct datasets as input and produce a scalar value (or vector) that represents the (dis)similarity between them based on a specific set of attributes. The goal of the Similarity Estimation Module is to define a process to compute, store and evaluate a measure of similarity between two datasets based on specific, relevant features as defined in the Attributes and Statistical Models Database. Special focus will be given to defining a) efficient similarity metrics, as their complexity plays a huge role in the scalability of the methodology, and b) accurate similarity assessment as this relates to the analytics workflow it pertains to.

Moreover, dataset veracity and churn must be taken into consideration when creating models that estimate their inter-relations. Data veracity accounts for the degree of uncertainty in the content of the generated data. Uncertainty characteristics of each dataset must be evaluated and taken into consideration during the similarity estimation. Uncertainty on data values (missing values, imprecise measurements, etc.) may be represented by intervals where data lies, or by probability distributions over such intervals if such information is available or can be inferred. Uncertainty on entire facts, e.g., results of an analytics task, may, in turn, be represented by discrete choices between alternatives, or by probability distributions over these alternatives. It could also be of interest to additionally record provenance information about data items, to provide traceability.

Data churn relates to data velocity and volatility. The system should be able to manage both new (unseen) datasets and incremental updates to existing ones, efficiently supporting real-time, streaming data and analytics modeling optimization. While incremental changes are usually small (compared to the original dataset), their relative frequency and nature can cause significant digression in the previously computed similarity models. The same is true for the streaming data case, where new datasets are continuously created and should be thus considered for optimized, content-based analytics. As such, the Volatility Management module should decide on cost-performance tradeoffs between full re-computation cycles and approximation algorithms that that provide low-cost estimates of the new dataset inter-relations.

The produced models will be stored in the Dataset Models library. A unique feature that characterizes this approach is the fact that these models are completely *task-agnostic*: The models represent the relative positions of the datasets to each other, based on standard, intuitive statistical and well-established features. As such, they can be readily used by a variety of analytics tasks and workflows in the Analytics Modeling layer.

B. Analytics Modeling

The Analytics Task Modeler module performs data to single-task performance modeling: Utilizing the dataset models from the previous layer, a ML model (such as a neural network) can be trained to approximate a specific operator for all desirable datasets. The process and is respective cost must be thoroughly examined to ensure the method's generality as well as its performance over different analytics tasks and domains. Learned models for individual operators are stored and updated whenever new profiling loops are executed in the Statistical Models Library.

The Analytics Modeling layer bases all its functionality on this single-task modeling framework. The Scheduler is the central component that orchestrates user-defined analytics optimization: Given an analytics task or arbitrary combinations of them (i.e., workflows), user-defined optimization goals (comprising rules that bind desirable workflow performance and cost over multiple metrics) and multiple permissible data inputs, the scheduler is responsible for orchestrating iterative/concurrent invocations of the Analytics Task Modeler with the necessary parameters in order to achieve end-to-end modeling under the user-defined policy. The later will be defined both in terms of cost (time, resources) and performance (qualitative metrics measurable through the analytics tasks in hand). The scheduler requires a planning component that selects the appropriate data-task model combinations for execution by the Analytics Task Modeler such that single or multiple criteria are met.

This is ever so important in the following two cases: i) tasks that receive multiple inputs and ii) workflow-based modeling. In the case of operators that take more than one input dataset, the characteristics of all inputs as well as their inter-dependencies play a big role in the similarity and performance estimation of the operator. In the case of operator workflows, our goal is to combine individual task models in such a way that the final workflow output can be accurately predicted given the input set characteristics. In this case, the effect that different workflow graphs and operator types from different domains have over the model synthesis process must be examined. The top-k plans should be considered utilizing a modified Dynamic Programming or heuristic planner (as shown in [34], [35]). This functionality is handled by the Workflow Planner module.

The Volatility Management module extends inside this layer. Indeed, incremental updates or new datasets not only affect the operator-agnostic data models but have a cascading effect as they are used for task modeling. The Adaptivity Engine encapsulates the effort to support dynamic dataset insertions and updates to previously constructed dataset models. In mathematical terms, we need to map an arbitrary dataset D_{NEW} to the existing space. In this task we must explore incremental, scalable solutions such as estimating D_{NEW}'s coordinates based on a finite set of distances with the "old" datasets and radical re-computation of the model from scratch. The cost/accuracy tradeoffs these solutions induce will be studied and evaluated. The same holds for cases of data churn, where streaming updates inside a single dataset may significantly alter its computed data model. Scalable and accurate monitoring mechanisms that identify the amount of change will be required (e.g., [36], [37]).

V. Conclusions

In this work, we described a content-centric system in order to boost data analytics performance. Unlike current and previous approaches, our proposal takes a domain-independent, holistic and continuous approach to data modeling. Primarily, it sheds light into the problem of immense numbers of different datasets, rather than the volume of a single one. As such, it recognizes the need to create data models that measurably map dataset interrelations. Secondly, it analyzes these data interrelations based on principled statistical and semantic axes taking into consideration: Data type, data veracity and data churn. The resulting models are task-agnostic, in the sense that they represent multiple and updateable dimensions of similarity between datasets irrespective of the use that analysts intend them to have. As such, for a given set of available data, a data scientist will be able to manage multiple models. Each model will represent dataset interrelationships over a specific data feature. Analysts can arbitrarily fuse data models, creating richer semantic data dimensions. Lastly, our proposal offers powerful meta-analysis tools over the managed models in real time. In practice, this approach has already proved capable of achieving very accurate performance models, while it can gracefully degrade its efficiency over customizable gains in execution cost.

References

[1] S. Madden, "From databases to big data," *IEEE Internet Computing*, 2012.

[2] C. C. Aggarwal, "An introduction to social network data analytics," in *Social network data analytics*. Springer, 2011, pp. 1–15.

[3] Y. Song, A. M. Elkahky, and X. He, "Multi-rate deep learning for temporal recommendation," in *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '16, 2016.

[4] T. L. Tuten and M. R. Solomon, *Social media marketing*. Sage, 2017.

[5] J. O. Awoyemi, A. O. Adetunmbi, and S. A. Oluwadare, "Credit card fraud detection using machine learning techniques: A comparative analysis," in *2017 International Conference on Computing Networking and Informatics (ICCNI)*, 2017.

[6] X. Xu, C. Zhou, and Z. Wang, "Credit scoring algorithm based on link analysis ranking with support vector machine," *Expert Systems with Applications*, vol. 36, no. 2, pp. 2625–2632, 2009.

[7] S. Moore, "How to Create a Business Case for Data Quality Improvement," https://www.gartner.com/smarterwithgartner/how-to-create-a-business- case-for-data-quality-improvement, Jun. 2018.

[8] "Medium Data is the New Sweet Spot." https://goo.gl/mnxnEx, 2017.

[9] "The Big Problem Is Medium Data," http://goo.gl/5nYrrz, 2014.

[10] R. A. Baeza-Yates, "Big data or right data?" in *Proceedings of the 7th Alberto Mendelzon International Workshop on Foundations of Data Management*, 2013.

[11] M. Lindstrom, *Small Data: The Tiny Clues That Uncover Huge Trends*. St. Martin's Press, 2016.

[12] Kirk Bresniker, "A new era of computing is coming. how can we make sure it is sustainable?" https://www.weforum.org/agenda/2018/09/end-of-an-era-what-computing-will-look-like-after-moores-law/.

[13] J. Gregory, *Counterparty credit risk: the new challenge for global financial markets*. John Wiley & Sons, 2010, vol. 470.

[14] D. R. Miller, S. Harris, A. Harper, S. VanDyke, and C. Blask, *Security Information and Event Management (SIEM) Implementation (Network Pro Library)*. McGraw Hill, 2010.

[15] Peter Schlampp, "Spark takes on the big security threats," http://www.ibmbigdatahub.com/blog/spark-takes-big-security-threats, 2016.

[16] K. Chourdakis, E. Epperlin, M. Jeannin, and J. Mcewen, "A cross-section across cva," *Nomura. Available at Nomura: http://www. nomura. com/resources/europe/pdfs/cva-crosssection. pdf*, 2013.

[17] D. Deng, R. C. Fernandez, Z. Abedjan, S. Wang, M. Stonebraker, A. K. Elmagarmid, I. F. Ilyas, S. Madden, M. Ouzzani, and N. Tang, "The data civilizer system." in *CIDR*, 2017.

[18] J. M. Hellerstein, V. Sreekanti, J. E. Gonzalez, J. Dalton, A. Dey, Nag *et al.*, "Ground: A data context service." in *CIDR*, 2017.

[19] V. Mansinghka, P. Shafto, E. Jonas, C. Petschulat, M. Gasner, and J. B. Tenenbaum, "Crosscat: a fully bayesian nonparametric method for analyzing heterogeneous, high dimensional data," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 4760–4808, 2016.

[20] M. Singh, M. J. Cafarella, and H. V. Jagadish, "DBExplorer: Exploratory Search in Databases," in *Proceedings of the 19th International Conference on Extending Database Technology, EDBT 2016, Bordeaux, France, March 15-16, 2016, Bordeaux, France, March 15-16, 2016.*, 2016, pp. 89–100.

[21] M. Joglekar, H. Garcia-Molina, and A. Parameswaran, "Interactive data exploration with smart drill-down," in *Data Engineering (ICDE), 2016 IEEE 32nd International Conference on*. IEEE, 2016, pp. 906–917.

[22] K. Dimitriadou, O. Papaemmanouil, and Y. Diao, "AIDE: An Active Learning-Based Approach for Interactive Data Exploration," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 11, pp. 2842–2856, 2016.

[23] S. Idreos, O. Papaemmanouil, and S. Chaudhuri, "Overview of data exploration techniques," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. ACM, 2015, pp. 277–281.

[24] S. Krishnan, J. Wang, E. Wu, M. J. Franklin, and K. Goldberg, "ActiveClean: Interactive Data Cleaning for Statistical Modeling," *Proceedings of the VLDB Endowment*, vol. 9, no. 12, pp. 948–959, 2016.

[25] N. Prokoshyna, J. Szlichta, F. Chiang, R. J. Miller, and D. Srivastava, "Combining quantitative and logical data cleaning," *Proceedings of the VLDB Endowment*, vol. 9, no. 4, pp. 300–311, 2015.

[26] I. Giannakopoulos, D. Tsoumakos, and N. Koziris, "A Content-Based Approach for Modeling Analytics Operators," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, 2018.

[27] T. Bakogiannis, I. Giannakopoulos, D. Tsoumakos, and N. Koziris, "Predicting graph operator output over multiple graphs," in *Web Engineering - 19th International Conference, ICWE 2019, Daejeon, South Korea, June 11-14, 2019, Proceedings*, 2019.

[28] J. C. Gower, "Some distance properties of latent root and vector methods used in multivariate analysis," *Biometrika*, pp. 325–338, 1966.

[29] T. Bakogiannis, I. Giannakopoulos, D. Tsoumakos, and N. Koziris, "Apollo: A dataset profiling and operator modeling system," in *Proceedings of the 2019 International Conference on Management of Data, SIGMOD Conference 2019, Amsterdam, The Netherlands, June 30 - July 5, 2019*, 2019.

[30] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection," http://snap.stanford.edu/data, Jun. 2014.

[31] Lichman, "Uci machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[32] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, vol. 40, no. 1, pp. 35–41, 1977.

[33] J. W. Sammon, "A nonlinear mapping for data structure analysis," *IEEE Transactions on computers*, vol. 100, no. 5, pp. 401–409, 1969.

[34] K. Doka, N. Papailiou, V. Giannakouris, D. Tsoumakos, and N. Koziris, "Mix 'n' match multi-engine analytics," in *2016 IEEE International Conference on Big Data, BigData 2016, Washington DC, USA, December 5-8, 2016*, 2016.

[35] N. Papailiou, D. Tsoumakos, P. Karras, and N. Koziris, "Graph-aware, workload-adaptive SPARQL query caching," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, Melbourne, Victoria, Australia, May 31 - June 4, 2015*, 2015.

[36] D. Trihinas, L. F. Chiroque, G. Pallis, A. Fernandez Anta, and M. D. Dikaiakos, "Atmon: Adapting the "temporality" in large-scale dynamic networks," in *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, 2018.

[37] D. Trihinas, G. Pallis, and M. D. Dikaiakos, "Admin: Adaptive monitoring dissemination for the internet of things," in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, 2017.