Are Data Science Pipelines Fuzzy Queries?

Genoveva Vargas-Solar

Are Data Science Pipelines Fuzzy Queries?

Genoveva Vargas-Solar
genoveva.vargas@imag.fr
Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG-LAFMIA, 38000 Grenoble, France

ABSTRACT

This short paper states a scientific position that proposes a new vision of data science pipelines defined as queries, namely data science queries (DSQ's). Different from classic queries, the results of DSQ's are not only data but also estimated models with associated error and performance scores. Besides, queries can have different attainable results according to the algorithms that implement them behind the scenes. A data scientist must choose the best or most adapted result according to given expectations related to a target domain. In this sense, it is possible to consider DSQ's as fuzzy queries that estimate results and choose those close to a combination of expected criteria. The paper discusses the aspects to consider for modelling a data science pipelines as fuzzy queries and possible research directions.

KEYWORDS

data science pipeline, query workflow, machine learning environment

ACM Reference Format:
Genoveva Vargas-Solar. 2020. Are Data Science Pipelines Fuzzy Queries?. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Datasets representing partial and incomplete observations of phenomena have become the backbone of scientific, analytic, and forecasting processes. It is possible to compute mathematical models to understand and predict phenomena, combining simulation techniques, artificial vision, and artificial learning with data science techniques. Therefore, datasets must go through complex and repetitive processing and analysis pipelines, namely *data science pipelines*. A data science pipeline combines data visualisation, cleaning, preparation, modelling or prediction, and assessment tasks.

This short paper states a scientific position that proposes a new vision of data science pipelines defined as queries, namely data science queries (DSQ's). Different from classic queries, the results of DSQ's are not only data but also estimated models with associated error and performance scores. Besides, queries can have different possible results according to the algorithms that implement them behind the scenes. A data scientist must choose the best or most adapted result in given expectations related to a target domain. In this sense, it is possible to consider DSQ's as fuzzy queries that estimate results and choose those close to a combination of expected criteria. The poster discusses the aspects to consider for modelling a data science pipelines as fuzzy queries and possible research directions.

DSQ's can be expressed, managed and efficiently enacted. The emergence of new architectures, for instance, the cloud, have opened new challenges for executing the tasks that compose data science pipelines. It is no longer pertinent to reason with respect to a set of computing, storage and memory resources, instead, it is necessary to design algorithms and processes considering an unlimited set of resources usable via a "pay as U go model". Instead of designing processes and algorithms considering as threshold the resources' availability, the cloud imposes to consider the economic cost of the processes vs. resources use, and the exploitation of available resources.

This short paper proposes a new vision of data science pipelines defined as fuzzy queries that can be expressed, managed and efficiently enacted. The paper defines the notion of data science query discussing the main guidelines to model such a query as a workflow. It then discusses the aspects to consider enacting a data science query.

The remainder of the paper is organised as follows. Section 2 introduces data science queries that can be used to model pipelines and how they can be associated to a degree of fuzziness. It discusses the design and enacting issues related to data science queries and approaches that can be revisited for enacting, reusing, and eventually optimising them. Section 3 gives an overview of existing querying techniques and their position with respect to data science queries. Finally, Section 4 concludes the paper and discusses future work.

2 DATA SCIENCE PIPELINES AS QUERIES

From our point of view, data science pipelines are a new type of queries that we name *data science queries*. Different from classic queries devoted to retrieving data, data science queries apply operations that give structure to datasets, compute statistics, and models. Computed partial and final results have a given accuracy and error score that must be estimated and determines the extent of the veracity of these results. Data collections have different qualities regarding their completeness (percentage of missing values) and consistency (percentage of erroneous values).

Similar to classic queries, data science queries can be modelled as dataflows. Such dataflow consists of tasks representing operations applied to datasets. These operations can be of many different types: harvesting, cleaning, features' engineering, descriptive statistics, machine learning, deep learning, results' assessment [6], visualisation and decision making. Tasks exchange partial results through a flow that transmits them through shared memory or data transmission protocols with different data structures (vectors, matrices,

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

values, etc.). A data science query consists of several dataflows integrated by a control flow. The control flow defines the execution order of dataflows. It also determines whether they should all be executed or not.

For example, consider a scenario consisting of different datasets for dealing with the process of the treatment of diabetes patients. Datasets contain clinical analysis' variables that can potentially have a role for a patient to contract diabetes: physiological (pregnancy, blood pressure) and chemical variables (insulin, BMI) and the diagnosis outcome. Doctors want to query *is it possible to predict whether a patient can contract diabetes, given previous diagnosis experiences?* based on collected datasets that associate clinical analysis variable with a diagnosis for a set of patients.

The data science query first consists of tasks for exploring the content of the dataset (pipeline A). For example, the attributes it contains (task A-1), the values' distribution (task A-2), grouping tests into diabetic and non-diabetic and counting them (task A-3). To see which variables values lead to positive/negative diabetics diagnosis (first partial result). Deciding whether to continue or not depends on the balance between the number of observations concerning diabetic and healthy patients. Assume that the number of these two groups is balanced. For predicting diabetics other tasks can define the query depending on the prediction model used. Considering that the mathematical properties of the variables are adapted and that the query expected result is a yes/no prediction, a data scientist can decide to apply a logistic regression model[1]. The tasks of the pipeline B include the fragmentation of the dataset into training and test sets (task B-1). Then, a training task applying logistic regression to the training dataset (task B-2). This task leads to a prediction model as a result. A validation task uses the model to lead to a prediction intent (task B-3). Finally, the remaining tasks of the data science query are devoted to assessing the model using the test dataset to determine to what extent it can be used for predicting new coming cases. The assessment tasks (pipeline C) include computing the prediction score (task C-1) and computing the confusion matrix (task C-2), given that logistic regression has been applied, and then compute precision and recall (task C-3). With the results of tasks C-1, C-2 and C-3, a data scientist can interpret the results of the query.

Consumers rely on executing data science queries under different conditions and receiving several versions of the results with assessment measures that can let them compare the "performance" of the implementations of the same query. Thus, data science queries become interactive processes where the user can intervene to adapt and modify some steps and where all design and execution details must be logged to enable reproducibility. The final result of a data science query is the one that reduces errors and with the best performance scores.

Consumers of data science query results are data scientists and decision-makers that have different insight and quality expectations regarding this "new" type of queries. Data science queries execution must be guided by "quality constraints" associated with different elements composing data science queries. Quality depends on data freshness, meaningfulness, sample classes balance, provenance. Second, other quality properties are associated with the whole data science query. For example, they depend on privacy and analysis security, and pertinence of data processing tasks. Does its enactment include explicable meta-data? Under which conditions are its results assessed and shared? A data science query can have additional associated technical quality attributes. Particularly, its requirements in terms of resources. For example, its execution time concerning how critical results are, data volume to be processed, algorithms complexity, data processes and results' security and persistence, computing cycles. This means that data loading, in memory/cache/disk indexing, data persistence, query optimisation, concurrent access, consistency and access control, and other management functions must be revisited under weaker hypothesis. For example, regarding data consistency, completeness and cleanness, to support the enactment of data science queries.

[1]Recall that for using logistic regression, the dependent variable is binary (diabetics yes/no); observations are independent of each other; there is little or no multicollinearity among the independent variables; linearity of independent variables and log odds. These properties must be verified through exploration tasks.

2.1 Estimating results quality of Data Science queries

A data science query has no predefined set of operators and composition rules. It combines algorithms respecting I/O implicit preconditions. For a data science query, it is necessary to model both its control and dataflow to express the data dependencies of its tasks and tasks execution order. Therefore, I propose to specialise the notion of query workflow[4] to model a data science query.

A query workflow (see Figure 1) consists of activities ordered by a control flow. Data dependencies among activities (i.e., dependent, independent, concurrent) determine the control flow. These dependencies model the dataflow of the query workflow. Activities can model operation types like statistical and aggregation, projection, filtering, etc. An activity represents a task of a data science query.

It represents an external module call that can execute a data processing operation. It specifies the input data to operations, how to transmit, distribute and store data in memory or cache so that they can be processed. Every activity has associated statistics like its execution time, computing cycles, and I/O properties (data structure, size, persistence, distribution).

The results of a data science query workflow can be associated with quality scores given by performance and precision measurements and error. Scores can be adjusted to enhance performance. So given a query, there are several query workflows that can implement it. The challenge is to choose the one that best fulfil ls given requirements like performance, average precision or error, etc. The good combination or balance among all these scores can be seen as a fuzziness objective that helps to choose the most "accurate" answer to a given data science query. Playing with different fuzziness index values can let a data scientist explore different results, decide and explain why she chooses a result and not others.

2.2 Enacting a Data Science Query Workflow

When it comes to enacting a data science query, operations combined in data science queries are not systematically intended to be used together in an integrated process. Each one has different computational complexities and requires pertinent computing, storage, and memory resources and data management strategies. The enactment of data science queries becomes a multi-aspect problem.

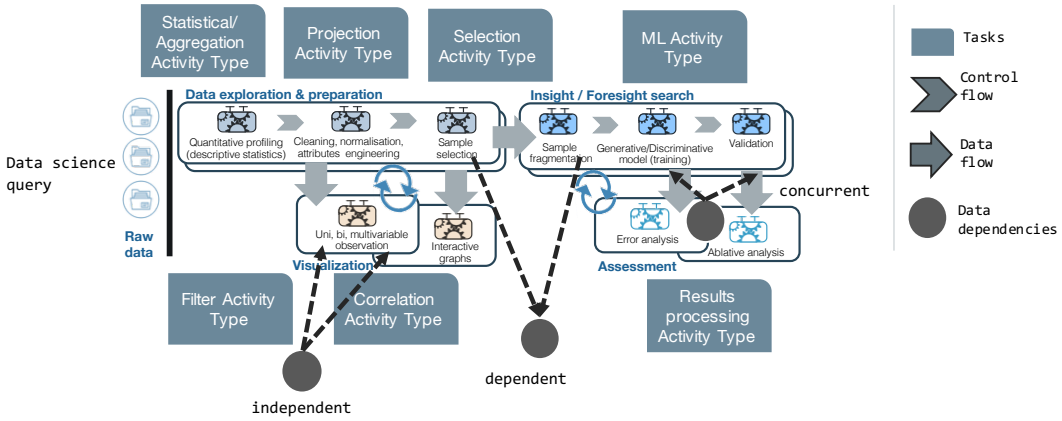

Figure 1: Data Science Query Workflow

It must deal with efficient storage systems that must provide efficient read and write operations. It must use parallel programming models to deal with the execution of greedy machine learning tasks that might require important computing and memory resources.

Data science queries enactment can be done « homemade » and then in large-scale conditions for deploying final industrial solutions. Small scale and large scale queries enactment can rely on service-oriented platforms (e.g., cloud providers). These platforms can provide elastic and "transparent" environments to execute greedy processes running on top of data hosted in heterogeneous hardware architectures. Some of these challenges have been addressed by Data Management Systems, Big Data Stacks [2, 5], Data Science and Machine Learning Environments.

Given that a data science query specifies I/O and operations features, it is necessary to provide and choose the type of underlying data management mechanisms that can support every task and the whole query. Thus, the enactment environment should use the right data structures and should prefetch, and transmit data using the right strategies to let the pipeline enactment run smoothly. This is an open issue to be included in existing ML environments [1]. For example, fragmenting, indexing or even compressing data according to main memory limits, data distribution/sharding when operations are executed in share-nothing architectures. There cannot be off-the-shelf data management solutions because every data science query is different and, because the operations it uses have their requirements.

Data Science and Machine Learning Environments provide all the necessary methods, and they are supported by enactment stacks that deal with the storage, fragmentation, indexing and distribution of the data required and produced by the tasks composing a pipeline. Yet, data scientists and machine learning engineers have to make decisions to combine these high- and low-level tools to "compose" their query and ensure that it will run at scale when used for processing datasets of different sizes.

In this sense, data science queries are programmed often as ad hoc solutions. This situation hinders the possibility of re-using some tasks or at least the strategies implemented for addressing target problems using specific methods. Besides, decision-making tools aiding to decide which are the most adapted data management strategies for every step of a data science query are still to come.

Coupling together data analytics methods and models with data management strategies and execution environments services for addressing data science query processing is a relevant challenge in the database community. Studying the problem in a general perspective will lead to its understanding and the identification of theoretical and technical issues. We propose to address novel challenges for efficiently designing and executing data science queries exploiting datasets.

3 RELATED WORK

We group querying techniques together and organise them across two families. The first one concerns classic querying in databases and information retrieval. Here the principle is that data correspond to a model of a mini-world (e.g., employees and departments of a firm, the representative terms of the content of a textual document), they are a set of facts structured according to some data model (e.g., relational). In this context, queries are stated using terms, operators (and/or/negation, relational, aggregation), and constraints. The results are collections of items that fully or partially answer queries. For example, relational query results contain all the items stored in the database that correspond with the specification of a query. In contrast, information retrieval results, provide the closest documents to the query but not necessarily all the documents that answer the query.

In this case, the results have a notion of completeness fussiness and probabilistic approximation. In the early 90's works like [9] addressed the notion of fuzzy queries and their declarative expression. The idea of introducing fuzziness into queries is to let the statement of imprecise queries and to process them and retrieve an approximate result set. Current work has started to address again imprecise (i.e., fuzzy queries) and propose different strategies for evaluating them. A survey on this issue is proposed by [7].

The second family is more exploratory where pipelines explore and analyse the data to profile it quantitatively and with the objective of either modelling, prediction or recommendation [3, 8]. In these queries' family, the results have an associated degree of error,

and they may not only be data but also queries or data samples. Exploratory queries tackle data collections that are expanding or where the structure provides little knowledge about the data. These queries run step-by-step like pipelines and the tasks often apply statistical, probabilistic, or data mining and artificial intelligence processing functions. Methodologies are still to come to integrate data management with the execution of algorithms that are often greedy.

4 CONCLUSION AND FUTURE WORK

Data science queries design and enactment remain artisanal today. Their design does not explain and models how statistical, machine learning and artificial intelligence models and computer technologies are used and weaved to lead to efficient enactment and representative results. Data science queries design and enactment call for well-adapted platforms that can efficiently organise data, evaluate and optimise queries, and execute algorithms that require important computing and memory resources. The design of these platforms is associated with communities like high-performance computing and distributed systems. The focus on data management functions (I/O management, indexing, storage and persistence) revisited for enabling data science queries enactment, can bring complementary and collaborative research. Having intelligent and well-adapted data management solutions along the enactment of data science queries is key to ensure their efficient execution. In this sense, our work associating data management strategies for each activity of a data science query workflow and the whole workflow can be original and ambitious. It can lead to results with a considerable expected impact on "modern data science experiments" deployed on heterogeneous high-performance target architectures.

REFERENCES

[1] Daniel Abadi, Anastasia Ailamaki, David Andersen, Peter Bailis, Magdalena Balazinska, Philip Bernstein, Peter Boncz, Surajit Chaudhuri, Alvin Cheung, AnHai Doan, Luna Dong, Michael J. Franklin, Juliana Freire, Alon Halevy, Joseph M. Hellerstein, Stratos Idreos, Donald Kossmann, Tim Kraska, Sailesh Krishnamurthy, Volker Markl, Sergey Melnik, Tova Milo, C. Mohan, Thomas Neumann, Beng Chin Ooi, Fatma Ozcan, Jignesh Patel, Andrew Pavlo, Raluca Popa, Raghu Ramakrishnan, Christopher Ré, Michael Stonebraker, and Dan Suciu. 2020. The Seattle Report on Database Research. *SIGMOD Rec.* 48, 4 (Feb. 2020), 44–53. https://doi.org/10.1145/3385658.3385668

[2] Sattam Alsubaiee, Yasser Altowim, Hotham Altwaijry, Alexander Behm, Vinayak Borkar, Yingyi Bu, Michael Carey, Raman Grover, Zachary Heilbron, Young-Seok Kim, et al. 2012. ASTERIX: an open source system for" Big Data" management and analysis. *Proceedings of the VLDB Endowment* 5, 12 (2012), 1898–1901.

[3] Alec Anderson, Sebastien Dubois, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. 2017. Sample, estimate, tune: Scaling bayesian auto-tuning of data science pipelines. In *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 361–372.

[4] Vıctor Cuevas-Vicenttın, Christine Collet, Genoveva Vargas-Solar, and Noha Ibrahim. 2010. The hypatia system for processing hybrid queries. (2010).

[5] Mike Franklin. 2013. The berkeley data analytics stack: Present and future. In *2013 IEEE International Conference on Big Data*. IEEE, 2–3.

[6] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*. 4765–4774.

[7] Rachid Mama and Mustapha Machkour. 2019. A Study of Fuzzy Query Systems for Relational Databases. In *Proceedings of the 4th International Conference on Smart City Applications* (Casablanca, Morocco) *(SCA '19)*. Association for Computing Machinery, New York, NY, USA, Article 117, 5 pages. https://doi.org/10.1145/3368756.3369105

[8] Zeyuan Shang, Emanuel Zgraggen, Benedetto Buratti, Ferdinand Kossmann, Philipp Eichmann, Yeounoh Chung, Carsten Binnig, Eli Upfal, and Tim Kraska. 2019. Democratizing data science through interactive curation of ml pipelines. In *Proceedings of the 2019 International Conference on Management of Data*. 1171–1188.

[9] Yoshikane Takahashi. 1995. A fuzzy query language for relational databases. In *Fuzziness in Database Management Systems*. Springer, 365–384.