E-ISSN: 2707-6644
P-ISSN: 2707-6636
IJCPDM 2020; 1(2): 44-46
Received: 22-04-2020
Accepted: 26-06-2020

YL Prathapa Reddy
Head, Department of Computer Science, Sri Vijayadurga Degree College, Kurnool, Andhra Pradesh, India

Corresponding Author:
YL Prathapa Reddy
Head, Department of Computer Science, Sri Vijayadurga Degree College, Kurnool, Andhra Pradesh, India

A possible trajectory for generating overall architecture of federated query for data repository

YL Prathapa Reddy

DOI: https://doi.org/10.33545/27076636.2020.v1.i2a.18

Abstract
As the number of databases continues to grow, data scientists need to use data from different sources to run machine learning algorithms for analysis. Data science results depend upon the quality of data been extracted. The objective of this research paper is to implement a possible trajectory for generating over all architecture of federated query for data repository which extracts data from different data sources and stores the result datasets in a common in-memory data format. It includes the activities of data source exploration, data acquisition, data preparation and result exploration processes. The outlines of such a trajectory-based model and how it can be used to categories data science projects (goal-directed, exploratory or data management) is suggest in this paper. This helps data scientists to perform their analysis and execute machine learning algorithms using different data engines without having to convert the data into their native data format and improve the performance.

Keywords: federated query for data, data exploration trajectory-based model, data repository

Introduction
The diversity of the data has increased in origin, format and modalities and so has the variety of techniques coming from machine learning, data management, visualization, causal inference and other areas. But, more importantly, compared to twenty years ago there are many more ways in which data can be monetized, through new kinds of applications, interfaces and business models. While the area of deriving value from data has grown exponentially in size and complexity, it has also become much more exploratory under the umbrella of data science. In the latter, data-driven and knowledge-driven stages interact, in contrast to the traditional data mining process, starting from precise business goals that translate into a clear data mining task, which ultimately converts “data to knowledge”. In other words, not only has the nature of the data changed but also the processes for extracting value from it. We identify new activities in data science, from data simulation to narrative exploration. We propose a general diagram containing the possible activities that can be included in a data science project. Based on examples, we distinguish particular trajectories through this space that distinguish different kinds of data science projects. We propose that these trajectories can be used as templates for data scientists when planning their data science projects and, in this way, explore new activities that could be added to or removed from their workflows. Together, they represent a new Data Science Trajectories model (DST) [1].

Federated querying involves data movement among different databases, each of them having their own data formats. Traditional Extract-Transfer-Load practice collects data from different sources and performs data cleansing process which are later loaded to the data warehouse and data marts from which the users can retrieve data for performing their tasks. Databases support serialization of data from internal native format to external one. Generally it requires exporting the data from one format to another using a common data format and later obtaining the result. This task is costly because the data needs to be serialized from the source and persisted in a disk and later it needs to be de-serialized and imported into the target database and be queried. Another approach to support federated querying for data scientists is to use a common memory oriented columnar data format which avoids using disk and can be utilized by all databases and execution engines [9]. Common data implementation protocols like JDBC are good in transferring data between databases efficiently. This paper extends these JDBC protocols which are widely used today to create a common in-memory data format which can be utilized by data scientists for execution of their machine learning algorithms from various sources efficiently and with improved performance.

Literature Survey

Within an organization data is stored in different databases since no one database can support all types of data been collected from the sources. Hence querying all the databases for required datasets is one of the essential tasks for data scientists. Ability to access data stored across organization is important for input and training machine learning algorithms. Data scientist may be interested in querying the relational database and store the queried data in an array database to synch it with the images and further store the extracted features from the images in a graph database to use it with the machine learning algorithms [6]. These type of tasks are increasingly common today and execution of federated query which has cross table joins is similar to the traditional execution of the query plan.

Federated querying is poorly supported due to lack of solutions for efficiently moving and combining the intermittent query results obtained from different databases [3]. Machine learning mathematical models are built using training data in order to make decisions without having to be programmed to execute various tasks. These algorithms are been used in various fields. With the growth of big data and technology, machine learning has become a mainstream presence for data scientists. The algorithms used depends upon the kind of problem been solved. There are two main categories of machine learning algorithms regression and classification. Regression consists of numeric data while classification involves mostly non-numeric data. Apart from these categories there is supervised learning and unsupervised learning.

Data scientists need to clean the data to be used for machine learning by excluding irrelevant attributes. This can be implemented during the data pre-processing pipeline. Some of the popular machine learning algorithms are, linear regression used for numerical data. Logistic regression and Support Vector Machines used in binary classification of data. Linear discriminant analysis used for classification in multi-category. Decision tree, KNN, Learning Vector Quantization and Naïve Bayes are used in both classification and regression models. AdaBoost and XGBoost are ensemble algorithms which buildon the previous models [10].

Many examples of data repositories can be found through platforms such as re3data10, which allows users to search among a vast number of different data repositories along the world by a simple or advance search using different characteristics. Another similar example is paperswithcode.com, a free resource for researchers and practitioners to find and follow the latest state-of-the-art machine learning-related papers and code. The company behind this (Atlas ML) has explored the way to present data regarding trending machine learning research, state-of-the-art leader boards and the code to implement it. This way users could have access in a unified and genuinely comprehensive manner to papers (fetched from several venues, repositories and open source and free license related projects) and to its code on different repositories, which can help with reviewing content from different perspectives to discover and compare research.

3. Proposed Method

Designing the logical and physical layout of the data and integrating different data sources is the data architecture. The block diagram of proposed trajectory architecture is presented in Figure (1).

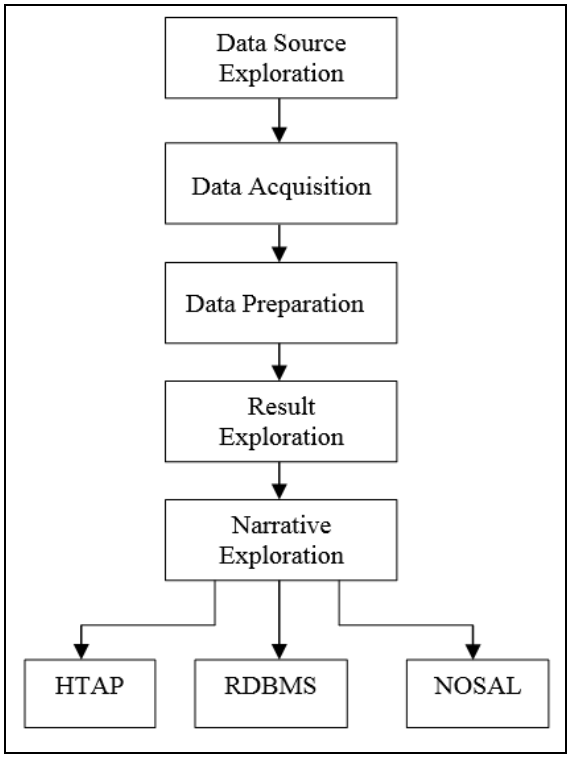

Fig 1: Proposed trajectory architecture

Big data needs to be stored and processed in different databases which are specialized for storage in different formats. There are also multiple data execution engines which process this data. Databases and execution engines use their own data formats and as the data moves between them, it is serialized and desterilized based on the source and target system data formats. This creates a performance bottleneck to data scientists for execution of machine learning algorithms.

Most of the databases support JDBC protocol for querying of data. The data retrieved from different data sources by the federated query would be stored in the same in memory format for processing. The customized federated SQL framework queries the data sources and converts the JDBC objects in to an in-memory data formats which can be used by various engines and help data scientists in their tasks. The row wise data from the data sources are converted into columnar format in-memory vectors which can be utilized by upstream data pipelines. It provides inter-process communication and there is no copy of the data during processing.

Data preparation where the data were integrated and prepared to be queried for visualization; result exploration where the visualizations were analyzed to decide which companies to offer particularized applications for; and narrative exploration where example stories were compiled in order to attract the audience to the visualization tool.

Data source exploration: discovering new and valuable sources of data;

Data acquisition: obtaining or creating relevant data, for example by installing sensors or apps;

Result exploration: relating data science results to the business goals;

Narrative exploration: extracting valuable stories (e.g., visual or textual) from the data;

Repository publishing

Repository publishing is a possible trajectory for generating a data repository that might have been taken includes the activities of data source exploration, when data comes from external sources, and data acquisition, where the required data is downloaded, scraped and explored; data preparation where data is parsed, curated and structured; data architecting, where data is annotated, stored and managed in order to provide an easy access to the users; and data release, where both the data and the automatic data extraction pipelines are shared under different licenses for public use.

Data publishing means curating data and making it available in a form that makes it easy for others to extract value. So data is both starting point and the product. Some amount of data value exploration has happened as part of this process, but there is not a very concrete business goal (yet) for which the data is being made available. Some data mining has happened to support the data value exploration and data understanding process, but data publishing takes the place of deployment. In this way a data repository can be created serving as a data library for storing data sets that can be used for data analysis, sharing and reporting.

4. Results

The data is loaded into in-memory data format using the JDBC protocol through a federated query from different sources. For experimentation we used MIMIC III dataset. It has over 50,000 to 2,000,000 rows of data related to patient information and medical device generated data been de-identified. The patient data was stored in H2 database and medical device data in Apache Ignite. In-memory row-wise H2 database was compared against the in memory common columnar format. It shows considerable improvement in the execution time from the table (1).

Also when the databases were accessed from the query engine directly for processing there were delays due to conversion of data into their native data formats. Using the in-memory common data format the query engines were able to process efficiently as shown in Figure (2).

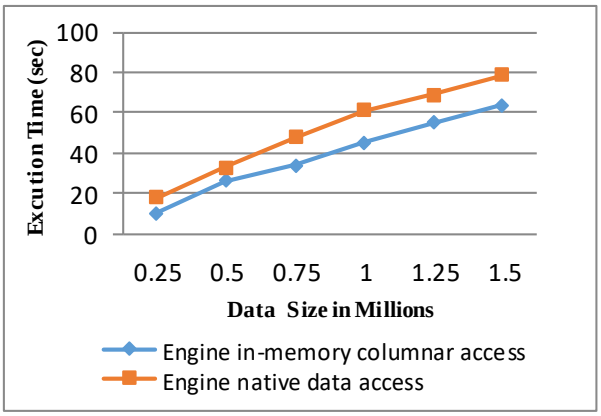

Fig 2: Comparison of Query execution time by data enginein accessing the native data and columnar data

Table 1: Comparison of data query execution time of in-memory row and columnar data format

Data size in Millions	Data Query Execution Time(Sec)	
	Column Oriented	Row Oriented
0 M	0	0
0.2500M	16	25
0.5000M	35	45
0.1000M	42	62

5. Conclusion

Since the trajectory perspective may allow a more systematic (and even automated) analysis at the process level, more flexible, less systematic, character of the new activities (exploration and data management) highlights the challenges for the automation of data science, this paper introduced a possible trajectory for generating overall architecture of federated query for data repository. It quickly retrieves the data for data scientists and is a tool that efficiently queries data between databases using a common in-memory data format. As a part of future work cloud databases can be integrated with the framework.

6. References

1. Nagashima H, Kato Y. Aprep-dm: A framework for automating the pre-processing of a sensor data analysis based on CRISPDM, in 2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops). IEEE, 2019,555-560.
2. Yang Timothy *et al*. Applied federated learning: Improving google keyboard query suggestions. arXiv preprintarXiv:1812.02903, 2018.
3. Jordan, Michael I, Tom M Mitchell. Machine learning: Trends, perspectives, and prospects. Science, 2015;349(6245):255-260.
4. Wu X, Zhu X, Wu GQ, Ding W. Data mining with bigdata, IEEE transactions on knowledge and data engineering, 2014;26(1):97-107.
5. Dharmasiri H, Goonetillake M. A federated approach on heterogeneous NoSQL data stores, Proc. Int. Conf. Adv. ICT Emerg, Regions (ICTer), 2013,234-239.
6. Flach P. Machine learning: the art and science of algorithms that make sense of data, Cambridge University Press, 2012.
7. Fujino, Takahisa, Naoki Fukuta. A SPARQL query rewriting approach on heterogeneous ontologies with mapping reliability. 2012 IIAI International Conference on Advanced Applied Informatics, IEEE, 2012.
8. Bernstein A, Provost F, Hill S. Toward intelligent assistance for a data mining process: An ontology-based approach for cost sensitive classification, IEEE Trans. on knowledge and data engineering, 2005;17(4):503-518.
9. Hellerstein, Joseph M *et al*. Adaptive query processing: Technology in evolution. IEEE Data Eng. Bull. 2000;23(2):7-18.
10. https://www.infoworld.com/article/3394399/machine-learningalgorithms-explained.html.