Query-Driven Learning for Next Generation Predictive Modeling & Analytics

Fotis Savva
University of Glasgow, UK
f.savva.1@research.gla.ac.uk

ABSTRACT

As data-size is increasing exponentially, new paradigm shifts have to emerge allowing fast exploitation of data by everybody. Large-scale predictive analytics is restricted to wealthy organizations as small-scale enterprises (SMEs) struggle to compete and are inundated by the sheer monetary cost of either procuring data infrastructures or analyzing datasets over the Cloud. The aim of this work is to study mechanisms which can democratize analytics, in the sense of making them affordable, while at the same time ensuring high efficiency, scalability, and accuracy. The crux of this proposal lies in developing query-driven solutions that can be used off the Cloud thus minimizing costs. Our query-driven approach will learn and adapt on-the-fly machine learning models, based solely on query-answer interactions, which can be used for answering analytical queries. In this abstract we describe the methodology followed for the implementation and evaluation of the system designed.

ACM Reference Format:
Fotis Savva. 2019. Query-Driven Learning for Next Generation Predictive Modeling & Analytics. In *2019 International Conference on Management of Data (SIGMOD '19), June 30-July 5, 2019, Amsterdam, Netherlands.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3299869.3300101

1 INTRODUCTION

With the adoption of data-driven decision making, there is a surge of companies turning to popular Cloud providers that have developed large-scale systems able to store and process huge data quantities. However, the problem still remains in that multiple costly[1] queries are issued by multiple analysts, which often overburden a cluster. Hence, there is dire need for systems/mechanisms which allow for fast execution of analytic and predictive queries while being resource aware without additional costs.

The analytic workloads [6] often comprise a series of *aggregate* queries (`COUNT`/`MIN`/`MAX`) to which an approximate answer is often enough to move forward. As such, over the last few decades, research focuses on systems that allow Approximate Query Processing (AQP)[1, 7, 9] to facilitate the process of data analysis. Although data-driven AQP systems offer a straight-forward and rather efficient solution to the problem, they come at a cost: they require large samples and still reside in Cloud systems, which makes them costly to maintain. Our aim is to allow analysts to perform essential analytic tasks in their local environments in an efficient, light-weight and accurate manner. Our approach leverages *query-driven learning* which exploits past query workloads to learn Machine Learning (ML) models that can estimate the results of analytic queries.

2 BACKGROUND & RELATED WORK

Our work is most notably connected to prior work focusing on the benefits of using a *query-driven* approach for a number of problems ranging from query processing to database tuning [2–4, 8, 10]. In addition, our work is influenced by prior work on AQP both sampling-based [1, 7, 9] and online aggregation systems [5, 11]. Our described system is agnostic to what happens when a query is executed. The execution could be achieved by an AQP system or a (Big-)database query processing system. Our system utilizes the *query-answer pairs* to build ML models that in turn provide estimations for queries *without* using cloud resources as estimations are evaluation of functions that can be done locally. As such, what we propose is complimentary to AQP and current processing engines. The closest thing to our work is proposed by Yongjoo *et al.* [9], however their approach is to learn better estimates and refine errors using results and errors obtained by an AQP engine. Our approach drastically differs as it does not assume any system in the background and it is built by executed queries no matter where they are executed and can be used *locally* without requiring a query processing engine.

[1] https://cloud.google.com/bigquery/pricing Shows the associated costs, which increase almost exponentially with more data

SIGMOD '19, June 30-July 5, 2019, Amsterdam, Netherlands
© 2019 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-5643-5/19/06.
https://doi.org/10.1145/3299869.3300101

3 QUERY-DRIVEN ANALYTICS ESTIMATION

Representation of Queries: A challenge in this endeavor is a valid representation for queries. Such that an ML model would associate the representation with the results obtained. If we consider every analytics system as a black box, we deal with queries over sets of multidimensional points made up of a number of attributes to which a number of algebra operations are performed to return results.

Definition 3.1. Consider *Selection-Projection-Aggregate* (SPA) queries, in which a single aggregate is the result of a query; that is made up of a single fact relation and multiple predicates. This is one of the most common queries used when requiring to extract analytics/descriptive statistics from a fact table. The list of predicates can be formalised as a $d-$dimensional vector, $\mathbf{m} \in \mathbb{R}^d$, in which the filtering parameter values are stored. The answer to that query (being the result of aggregate operations) is $y \in \mathbb{R}$. Therefore, a query-answer pair is a vector $\mathbf{q} = (\mathbf{m}, y)$.

Definition 3.2. (Learning) Once a vectorized representation for queries & answers is constructed, we adopt ML algorithms to learn how the query parameter values affect the result. Given a number of queries already executed and stored in log files, we obtain the set $C = \{(\mathbf{m}_i, y_i)\}_{i=1}^{n}$. The goal of any ML algorithm is to minimize the expected loss between the *true* result of y and an estimated one $\hat{y}$, i.e., to approximate the distribution funtion $p(y|\mathbf{m})$.

Estimation: Given this growing set C, our strategy is to sequentially develop *local* ML models that efficiently predict the associated outputs given unseen queries. Such knowledge extraction is achieved by on-line partitioning the query vectors $\{\mathbf{q}_1, \ldots, \mathbf{q}_n\}$ from the set C into disjoint clusters that represent the *statistical query patterns* of the analysts. Fundamentally, within each cluster, the queries are much more *similar* than the queries in other clusters. The main objective of query clustering algorithms is to minimize the *distance* of all queries to a corresponding cluster *representative*, which is essentially a *pseudo*-query representing the analysts query access patterns and best represents each cluster; it is often the *mean*-vector (centroid) of all queries associated with that cluster. The $K \ll n$ query representatives $\mathcal{W} = \{\mathbf{w}_1, \ldots, \mathbf{w}_K\}$ are estimated to optimally represent the queries in C such that they incrementally minimize the expected quantization error $\mathbb{E}[\min_k \|\mathbf{q} - \mathbf{w}_k\|^2]$. Each query $\mathbf{q}$ becomes associated with the closest representative $\mathbf{w}^* = \arg\min_{\mathbf{w} \in \mathcal{W}} \|\mathbf{q} - \mathbf{w}\|$. Based on this query-space partitioning of C, we produce K query-disjoint subsets $C = C_1 \cup \ldots C_K$, such that $C_k \cap C_l \equiv \emptyset$ for $k \neq l$ and $C_k = \{\mathbf{q}|\mathbf{w}_k = \arg\min_{\mathbf{w} \in \mathcal{W}} \|\mathbf{q} - \mathbf{w}\|^2\}$. A local predictive ML model is then trained over each query-disjoint subset of query-answer pairs $C_k, k \in [K]$.

After having partitioned the query space into disjoint clusters $C_1, \ldots, C_K$, we therein train K different supervised learning models, $\mathcal{M} = \{\hat{g}_1, \ldots, \hat{g}_K\}$ that associate query vectors $\mathbf{q}$ belonging to cluster $C_k \subset C$ with their corresponding response outputs y. That is, each *local* model $\hat{g}_k$ is trained from the query-response pairs $(\mathbf{m}, y) \in C_k$ from those queries $\mathbf{q}$ which belong to $C_k \subset C$ such that $\mathbf{w}_k$ is the closest query-representative of those queries. After training local models, the analysts can use these models for predicting the answers of new incoming queries *without* communicating with the Cloud and *without* executing them on data or samples.

This partition-and-local-learning methodology effectively assumes an *ensemble* learning strategy for our system. By adopting such an ensemble learning strategy, the various models have *voting rights*, meaning that each one of them contributes to the final query answer prediction for a given query by casting a vote. However, our strategy is to revoke voting rights for models, which do not represent the query sub-space that a given query belongs to. Hence, we assign to the most representative model an *authoritative* power over the rest of the models and thus returning to the user its prediction with the highest confidence. Formally, the proposed ensemble prediction is $\hat{y} = \sum_{k=1}^{K} \mathbb{I}_k \hat{g}_k(\mathbf{q})$ with $\mathbb{I}_k$ being an indicator function which evaluates to 1 if $\mathbf{w}_k = \arg\min_{i \in [K]} \|\mathbf{q} - \mathbf{w}_i\|$.

4 RESULTS

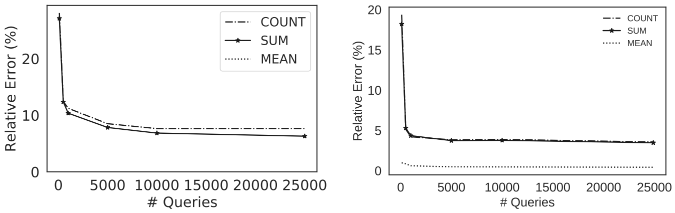

Figure 1: Accuracy over number of queries: (Left) Crimes Dataset; (right) Sensors Dataset.

Results shown in Figure 1 over *real* datasets indicate great promise for adopting such approaches in analytic query estimation. The most common aggregates can be answered accurately while using *zero* cloud resources as they are executed locally using ML. As the number of executed queries increases, the expected error decreases until reaching a plateau after $10k$ queries for the Crimes dataset and less than $5k$ for Sensors dataset. The queries used for training simulate an analysts' behavior, issuing spatial queries (restricting lat/long and extracting aggregate values for crimes reported within an area) and temporal queries for Sensors.

REFERENCES

[1] Sameer Agarwal, Barzan Mozafari, Aurojit Panda, Henry Milner, Samuel Madden, and Ion Stoica. 2013. BlinkDB: queries with bounded errors and bounded response times on very large data. In *Proceedings of the 8th ACM European Conference on Computer Systems*. ACM, 29–42.

[2] Christos Anagnostopoulos, Fotis Savva, and Peter Triantafillou. 2018. Scalable aggregation predictive analytics. *Applied Intelligence* 48, 9 (2018), 2546–2567.

[3] Christos Anagnostopoulos and Peter Triantafillou. 2017. Efficient scalable accurate regression queries in in-dbms analytics. In *Data Engineering (ICDE), 2017 IEEE 33rd International Conference on*. IEEE, 559–570.

[4] Christos Anagnostopoulos and Peter Triantafillou. 2017. Query-driven learning for predictive analytics of data subspace cardinality. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 11, 4 (2017), 47.

[5] Joseph M Hellerstein, Peter J Haas, and Helen J Wang. 1997. Online aggregation. In *Acm Sigmod Record*, Vol. 26. ACM, 171–182.

[6] Stratos Idreos, Olga Papaemmanouil, and Surajit Chaudhuri. 2015. Overview of data exploration techniques. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. ACM, 277–281.

[7] Srikanth Kandula, Anil Shanbhag, Aleksandar Vitorovic, Matthaios Olma, Robert Grandl, Surajit Chaudhuri, and Bolin Ding. 2016. Quickr: Lazily approximating complex adhoc queries in bigdata clusters. In *Proceedings of the 2016 International Conference on Management of Data*. ACM, 631–646.

[8] Lin Ma, Dana Van Aken, Ahmed Hefny, Gustavo Mezerhane, Andrew Pavlo, and Geoffrey J Gordon. 2018. Query-based Workload Forecasting for Self-Driving Database Management Systems. In *Proceedings of the 2018 International Conference on Management of Data*. ACM, 631–645.

[9] Yongjoo Park, Ahmad Shahab Tajik, Michael Cafarella, and Barzan Mozafari. 2017. Database learning: Toward a database that becomes smarter every time. In *Proceedings of the 2017 ACM International Conference on Management of Data*. ACM, 587–602.

[10] Kai-Uwe Sattler, Ingolf Geist, and Eike Schallehn. 2003. Quiet: Continuous query-driven index tuning. In *Proceedings of the 29th international conference on Very large data bases-Volume 29*. VLDB Endowment, 1129–1132.

[11] Kai Zeng, Sameer Agarwal, Ankur Dave, Michael Armbrust, and Ion Stoica. 2015. G-ola: Generalized on-line aggregation for interactive analysis on big data. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. ACM, 913–918.