Farm Your ML-based Query Optimizer's Food! – Human-Guided Training Data Generation –

Robin van de Water Francesco Ventura Zoi Kaudi
Jorge-Arnulfo Quiané-Ruiz Volker Markl
TU Berlin & DFKI GmbH

The need for ML-based query optimization. The value that data and AI technologies offer today mandates open environments, where data assets, such as datasets, algorithms, and machine learning (ML) models, are unified under the same ecosystem [3]. Agora[1] is an ecosystem that aims at (i) offering assets to a broader audience via a set of marketplaces and (ii) providing an open execution environment that allows users to run their (composed) assets. In such an open environment, there is a need for heterogeneous task execution, i.e., a task can be executed by combining multiple processing systems. The decision on the set of systems is taken based on query optimization, similarly to the cross-platform systems [1]. However, cost-based optimization in such environments is proven to suffer from the fine-tuning effort required to produce efficient query execution plans [2]. Thus, replacing cost models with ML models that will estimate the runtime of query plans is a natural solution to ML-based query optimization in general.

The hurdle of training data acquisition. A fundamental requirement for most learning-based solutions is the availability of valuable data to train ML models on. Although unsupervised methods exist, many of the proposed techniques are based on supervised learning models. The effectiveness of such models depends on the quantity and quality of training data as well as the availability of valuable ground-truth labels. These requirements quickly become a road blocker in the context of query optimization: First, collecting a large number of real query plans with labels (e.g., execution time) requires developing thousands of plans that are not only optimal. Even if logs are available, the plans in the logs are the ones the optimizer chose to execute and, thus, most of them are (near-)optimal. Second, after gathering all these plans, one has to execute them to get their label. The latter is a very time-consuming task, as it leads to the execution not only of a large number of queries but also the execution of sub-optimal ones. For example, collecting labels for only 500 OLAP plans with input data of about 1TB in our four-quadcore-nodes cluster takes almost 10 days. This is problematic because learning a model typically requires several thousands plans. Extrapolating our previous experiment to 10, 000 plans would require more than 6 months!

DataFarm. We thus have built DataFarm, a novel framework for generating training data for learning-based query optimizers [4]. DataFarm follows a data-driven white-box approach. It augments an initial (typically small) query workload and attaches labels (runtime or cardinality estimation) with uncertainty values to each generated query. Figure 1 shows an overview of DataFarm. The abstract plan generator learns patterns from the input query workload as Markov Chains and generates new heterogeneous abstract plans exploiting real operators' distributions. The plan instantiator then creates an augmented set of realistic plans by instantiating different variants for each previously generated abstract plan, e.g., by setting different selection predicates. It exploits the user's input data to tailor the newly generated workload to real use cases, increasing its reliability. The label forecaster uses an active learning approach to label the generated query workload efficiently: It executes some of the generated plans and forecasts the labels of the rest with an interpretable ML model. It characterizes plans at the operator-level with interpretable features, actively improving forecasting performance by executing the smallest number possible of plans. It also provides the uncertainty for each forecasted label. Downstream operations can then leverage these uncertainty values to improve their output, e.g., by using them as a noise indicator.

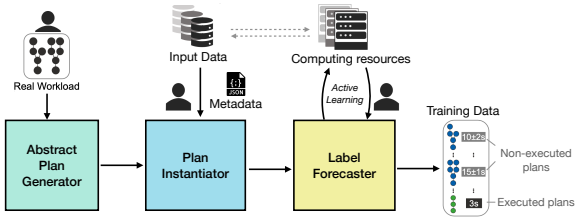

Figure 1: DataFarm**'s training data generation process.**

Human in the loop. To further enhance the accuracy of the label forecaster we have introduced the human in the active learning step. The intuition is that users often know their desired query workload and, thus, given the right insights, can guide the system on which plans to execute to get better-estimated labels. As the human alone cannot manually select among thousands of plans, the label forecaster iteratively suggests to the user a small set of candidate plans for execution. Then, the user can inspect these candidates and remove or add new plans via an intuitive graphical user interface (GUI). The GUI of DataFarm provides useful insights, such as feature importance and model explanation analysis, such that the users can take informed decisions.

DataFarm provides to users the ability to download the generated query plans with or without labels to be used either as a benchmark or as training data, respectively. Involving the human in the training data generation leads to higher-quality labeled training data improving the downstream ML task which they are used for.

[1] https://www.agora-ecosystem.com

REFERENCES

[1] D. Agrawal et al. RHEEM: Enabling Cross-Platform Data Processing - May The Big Data Be With You! -. *Proc. VLDB Endow.*, 11(11):1414–1427, 2018.
[2] Z. Kaoudi, J. Quiané-Ruiz, B. Contreras-Rojas, R. Pardo-Meza, A. Troudi, and S. Chawla. ML-based Cross-Platform Query Optimization. In *ICDE*, 2020.
[3] J. Traub, Z. Kaoudi, J.-A. Quiane-Ruiz, and V. Markl. Agora: Bringing together datasets, algorithms, models and more in a unified ecosystem [vision]. *SIGMOD Record*, 49(4), 2020.
[4] F. Ventura, Z. Kaoudi, J. Quiané-Ruiz, and V. Markl. Expand your Training Limits! Generating and Labeling Jobs for ML-based Data Management. In *SIGMOD*, 2021.