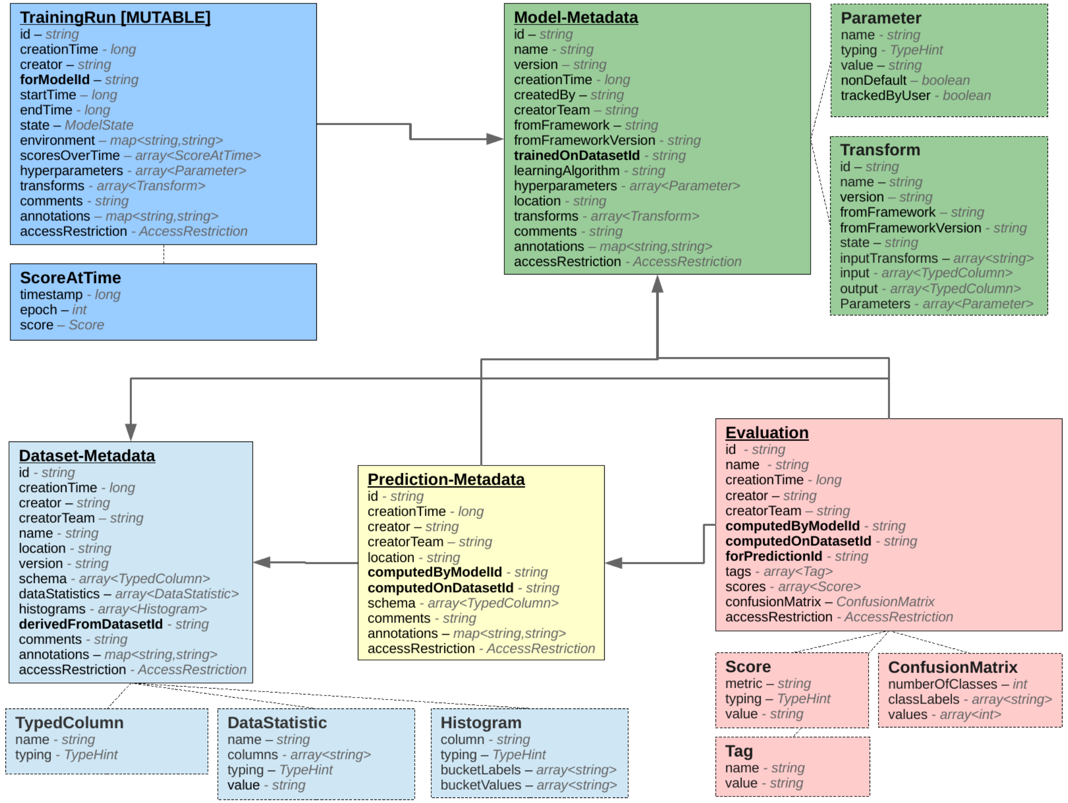
Optimization of Machine Learning Workloads with Experiment Databases

Behrouz Derakhshan

Behrouz.derakhshan@dfki.de

Experiment Database (ED)

- Contains meta-data of machine learning experiments about the datasets, models, transformations, and evaluations
- Based on [1]
- Workload Optimizations using ED:
    - Reuse of existing models
    - Materialization of transformed datasets
    - Multi-query optimization
    - Warm-starting
    - Efficient data transformation
    - Efficient hyper-parameter tuning
    - Efficient multi-model training



Database schema for storing declarative description of machine learning experiments [1]

Categories of Machine Learning Workloads

Next paper

1. Interactive batch workload

- Description:
  - Multi-user (from tens to thousands of users)
  - Primarily consists of repeated data preprocessing and data transformations
  - Aggressive Hyper-parameter tuning and model evaluation
- Optimizations using ED:
  - Reuse
  - Materialization
  - Multi-query optimization
  - Warm-starting
  - Decreasing the search space of hyper-parameters

Future Work

2. Incremental Training Workload

- Description:
  - Multi-user (tens of users)
  - Incremental improvement of existing data processing pipelines and ML models
  - Retrain models and pipelines (typically daily)
- Optimizations using ED:
  - Speed up in data transformation
  - Decreasing the search space of hyper-parameters
  - Multi-model training
  - Warm-starting

3. Continuous Training Workload

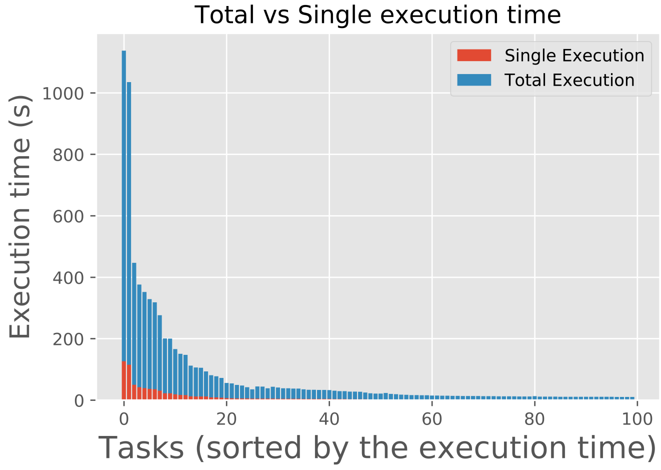
- Description:
  - Real-time (or near real-time) data processing
  - Online ML models
- Optimizations using ED:
  - Speed up in data preprocessing
  - Multi-model training
  - Fast detection of model quality degradation

Plan for next Paper

- VLDB 2018 (Deadline: 1st of March)
- Focus on Workload 1 (Interactive batch)
- Plan:
  - Develop a simple database based on [1]
  - Experiments:
    - Focus only on a few frameworks (scikit-learn, R mlr package)
    - Popular Kaggle competitions [3], OpenML datasets and tasks [2]
    - I expect a reduction in the processing time and the development time

- I currently have a paper under review for SIGMOD2018. Depending on the result of the review, this work may have to be pushed back to SIGMOD2019/VLDB2018

- Early Results:
  - The Most popular pipeline in OpenML (scikit-learn) consists of:
    - Missing Value Imputer
    - Dimensionality reduction using PCA
    - Random Forest Classifier
  - It is repeatedly executed on 100 different tasks (average of 9 times on each task)
  - Figure below shows that a simple reuse can save about 2 hours of processing time



This figure may include executions performed by bots, therefore, we cannot reach a conclusion about the reduction on the development time

References

[1] *Sebastian Schelter, Joos-Hendrik Böse, Johannes Kirschnick, Thoralf Klein, Stephan Seufert*, **Automatically Tracking Metadata and Provenance of Machine Learning Experiments**, Machine Learning Systems workshop at the conference on Neural Information Processing Systems (NIPS) 2017

[2] https://www.openml.org/

[3] https://www.kaggle.com/