Data-Related Challenges in End-to-End Machine Learning

Sebastian Schelter
New York University
sebastian.schelter@nyu.edu

I moved back to academia after working on a set of real-world machine learning (ML) deployments in industry for some years. A recent article discusses some of the data management challenges encountered during that time [3], and I would like to present a poster summarizing some of these challenges with the goal to provide directions for future research.

Often, successful ML systems contain a lot of "tribal" knowledge that practitioners acquire over time and that is the foundation of customly designed data and ML pipelines in industry [5, 2]. It would be beneficial to convert this knowledge into general abstractions for ML systems, and thereby automate many of these tasks which are currently addressed with hand-crafted solutions by experts. Examples of some of the challenges are:

- *Guaranteeing consistent feature transformations in training and serving systems.* Model training and model serving systems have conflicting design goals, as training operates on large batches of data and aims for high throughput, while serving consumes single examples and aims for low latency responses. However, for a given model the feature transformations applied to the data must be consistent across training and serving, which is for example a huge problem for the SparkML pipeline abstraction which is implemented with a batch-orientation.
- *Execution of pipelines with mixed operations from relational and linear algebra.* End-to-end ML pipelines often consist of different stages which apply conflicting abstractions and data types. E.g., in data preparation, relational and map-reduce-like operations are applied to tuple data while model training applies linear algebraic operations on tensor data. In practice this often leads to pipelines comprised of different systems (often in different programming languages) that are "glued" together, which is the source of many problems. It would be highly beneficial to find a common language for flexibly applying relational and linear algebraic operations [1].
- *Validation of assumptions on input data distribution by trained models.* Due to the i.i.d.-assumption inherent in many ML algorithms, models contain implicit assumptions on the distribution of the data on which they are applied after training. Erroneous data or a distribution shift can crash models or lead to drop in prediction quality. Therefore it is desirable to find ways to explicitly model and automatically validate these assumptions [4].

References

[1] A. Kunft, A. Alexandrov, A. Katsifodimos, and V. Markl. Bridging the gap: towards optimization across linear and relational algebra. 2016.

[2] N. Polyzotis, S. Roy, S. E. Whang, and M. Zinkevich. Data management challenges in production machine learning. In *SIGMOD*, pages 1723–1726, 2017.

[3] S. Schelter, F. Biessmann, T. Januschowski, D. Salinas, S. Seufert, G. Szarvas, et al. On challenges in machine learning model management. *IEEE Data Engineering Bulletin*, 2018.

[4] S. Schelter, D. Lange, P. Schmidt, M. Celikel, F. Biessmann, and A. Grafberger. Automating large-scale data quality verification. *PVLDB*, 11(12):1781–1794, 2018.

[5] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, J.-F. Crespo, and D. Dennison. Hidden technical debt in machine learning systems. *NIPS*, pages 2503–2511, 2015.

North East Database Day 2019, Boston.