Technical Perspective: Declarative Recursive Computation on an RDBMS

Matthias Boehm
Graz University of Technology
m.boehm@tugraz.at

From a historical perspective, relational database management systems (RDBMSs) have integrated many specialized systems and data models back into the RDBMS over time. New workloads motivated specialized systems for performance, but over time, general-purpose RDBMSs absorbed this functionality to avoid boundary crossing. We already witnessed this process for object-relational functionality, XML and JSON data types, OLAP/HTAP systems, and RDF/graph processing, while for natural language processing (NLP), time series, and machine learning (ML), the outcomes remain unclear. Interestingly, graph processing, NLP, and time series are largely ML workloads too. For this reason, integrating data management and ML is of high practical relevance and has been addressed by (1) integrating ML into RDBMSs, and (2) specialized ML systems. The paper "Declarative Recursive Computation on an RDBMS" [3] by Jankov et al. makes a very valuable contribution by reconciling these two areas and showing the potential of recursive computations on an RDBMS, as the backend—not necessarily frontend—for large-scale machine learning.

ML in RDBMSs: Integrating ML primitives for model training and prediction into RDBMSs has been a major focus area in research and practice for over a decade. Compelling key arguments are to bring analysis close to the data, declarative specification, support for mixed ML and query workloads, scalability, and reuse of existing technology. Existing work includes SQL- and UDF-based systems, factorized learning over joins, deep system integration approaches (e.g., for time series forecasting), tailor-made array databases, efficient and zero-copy data transfers, as well as extended data models and operations. Another long standing question is the effective combination of relational and linear algebra, which recently has been addressed from both systems and theory perspectives as well.

DM in ML Systems: A second major avenue of combining data management (DM) and ML systems is the integration and use of DM techniques—such as rewrites, physical operators, size propagation, query compilation, distributed and federated query processing, caching, incremental maintenance, partitioning, indexing, and compression—into modern ML systems. While such ML systems also leverage data flow graphs at their core, they offer more specialized, stateless domain-specific languages and operators, and directly process files or in-memory matrices or tensors.

Paper Context: The paper by Jankov et al. follows a line of influential work on large-scale statistical computation with SimSQL [1], a distributed RDBMS on Hadoop MapReduce. In this context, the paper reuses the authors' previous work on (1) a chunked matrix representation of blocks with fixed logical size (e.g., $1\text{K} \times 1\text{K}$, whose guaranteed alignment simplifies join processing), integrated as matrix and vector data types [4], as well as (2) ideas on recursive computation of variable dependencies in BUDS [2]. This foundation is interesting because it closely resembles distributed matrix representations of specialized ML systems on distributed computing frameworks like Spark, where lazy evaluation can similarly "unroll" loops into recursive computations.

Paper Contributions: A major problem, however, is the mapping of mini-batch ML training to such recursive computations. The paper by Jankov et al. addresses this problem by two central contributions. First, a succinct SQL extension allows accessing recursive versions of tables via array indexes in a declarative, data–independent manner that facilitates optimization. Such a query is then compiled into a single DAG of relational algebra operations. Second, the operator DAG is partitioned into—independent and thus, manageable—frames by minimizing materialization points, subject to a maximum number of operators per frame. Experiments with large mini-batches and models show very promising scalability results compared to TensorFlow. Overall, this paper has potential for high impact in multiple areas: (1) inspiring improved handling of recursion, DAGs, and large plans in RDBMSs, (2) inspiring frame-based execution in systems with lazy evaluation, and (3) reconciling the diverging areas of In-RDBMS ML and specialized ML systems.

1. REFERENCES

[1] Z. Cai, Z. Vagena, L. L. Perez, S. Arumugam, P. J. Haas, and C. M. Jermaine. Simulation of Database-Valued Markov Chains Using SimSQL. In *SIGMOD*, 2013.

[2] Z. J. Gao, S. Luo, L. L. Perez, and C. Jermaine. The BUDS Language for Distributed Bayesian Machine Learning. In *SIGMOD*, 2017.

[3] D. Jankov, S. Luo, B. Yuan, Z. Cai, J. Zou, C. Jermaine, and Z. J. Gao. Declarative Recursive Computation on an RDBMS. *PVLDB*, 12(7), 2019.

[4] S. Luo, Z. J. Gao, M. N. Gubanov, L. L. Perez, and C. M. Jermaine. Scalable Linear Algebra on a Relational Database System. In *ICDE*, 2017.