In-Database Machine Learning
Using Gradient Descent and Tensor Algebra

Maximilian Schüle, Frédéric Simonis, Thomas Heyenbrock, Alfons Kemper, Stephan Günnewann, Thomas Neumann
\{schuele, simonis, kemper, guennemann, neumann\}@in.tum.de

HyPer + Tensors + Gradient Descent

**Machine Learning:** Data in tensors and a loss function

- Operator for gradient descent:
  - Gradient needed for gradient descent: automatic differentiation necessary for arbitrary loss functions
  - Integration in relational algebra
  - Representation of a loss function

\[ \lambda(R,S)(R.a * S.x + R.b - S.y)^2 \]

- Tensors: datatype with algebra
- Optimisation problems solvable in the core of database systems

Materializing

- Allows any optimization method
  - Tuples need to be materialized

Pipelined

- No materialization required
  - Iterations must be precompiled

Combined

- Precomputes weights in pipelines
  - Little performance gains

Evaluation

**Linear Regression**

**Multiple Linear Regression**

**Logistic Regression**

**k-Means**

---

Materializing

- Initial Weights
- Training Data
- Loss Function
- Initial Weights

- TensorFlow
- PSQL
- MariaDB

Pipeline

- Stochastic GD
- Training Data
- Test Data
- Calculated Weights
- Labeling

Evaluation

- Time in s
- Number of Tuples
- Number of threads

- TensorFlow-GPU
- HyPer
- MAXDB
- PSQl

Combined

- Stochastic GD
- Training Data
- Test Data
- Calculated Weights
- Labeling

Evaluation

- Time in s
- Number of Tuples
- Number of iterations

- TensorFlow-GPU
- HyPer
- MAXDB
- PSQl

---

In-Database Machine Learning
Using Gradient Descent and Tensor Algebra