Using Data-Driven and Zero-Shot Learning to learn DBMS Components

Abstract

Workload-driven learning is a technique to replace a DBMS component with a machine learning model

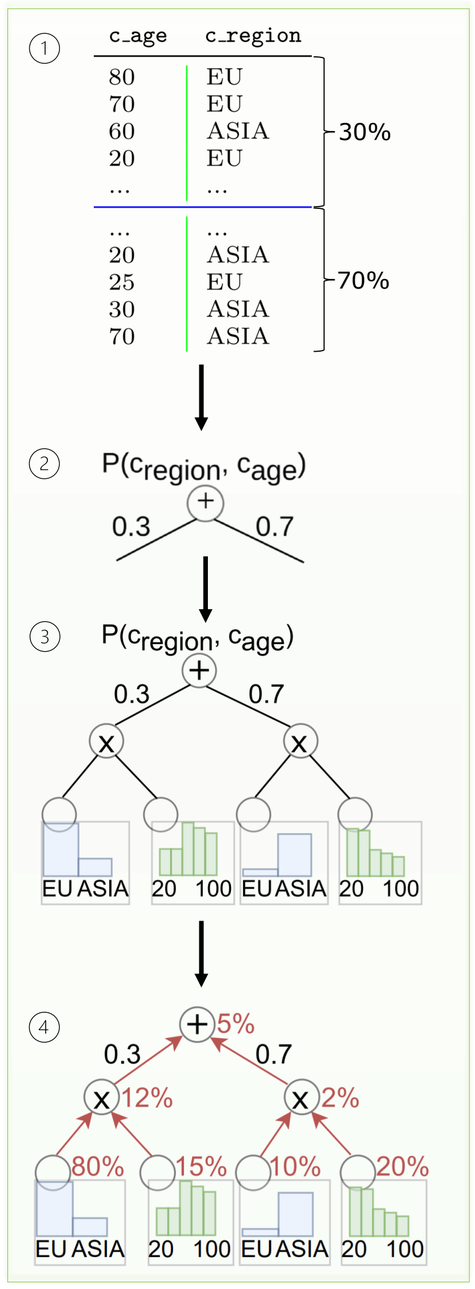
- **Issue**: For each new database or component, a new model must be trained. This makes it very inflexible and expensive to train.
- **Solution**: Use data-driven and transfer learning approaches to reduce training effort and make the model generalizable to unseen databases

Data-Driven Learning

Idea: Model **learns data characteristics** like the data's distribution and correlation across complex relational databases

- No training workload needed as the model relies on data only
- Retraining the model only takes a few minutes
- Support for tasks that do not consider workload (cardinality estimation, AQP, indexing)

Goal: Construct Relational Sum-Product Network (RSPN) from database



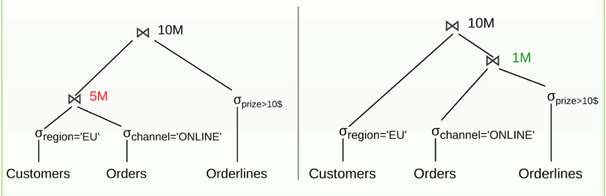
1. Split independent rows into row clusters (e.g. using KMeans)

2. Use sum node and add weights corresponding to the row cluster sizes to the edges

3.
   - In each row cluster, split independent columns into column clusters (product node)
   - If not all columns are independent, start again with the first step, otherwise continue

4.
   - Use RSPN to compute probabilities on arbitrary attributes of the table
   - Example: `SELECT COUNT(*) FROM Customer C WHERE c_region='EU' AND c_age<30` yields 5%

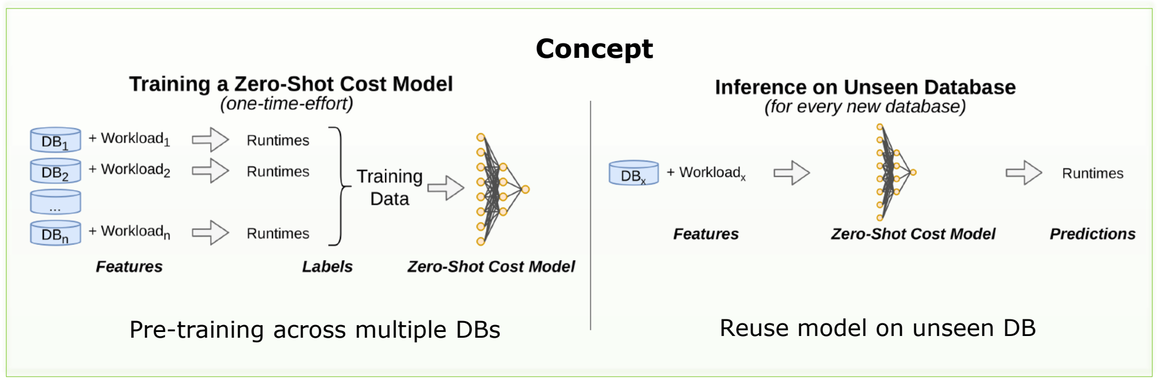
5. Estimated value can be used to **select optimal query plan**



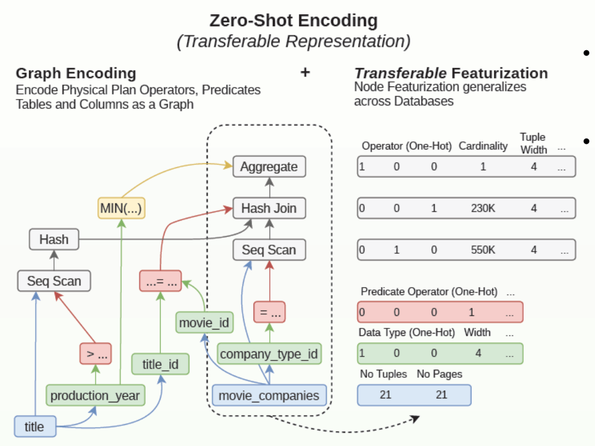
Zero-Shot Learning for Databases

Idea: Inline to other zero-shot approached (e.g. GPT-3), train a model that can **generalize to unseen databases out-of-the-box**.

- No queries on database are required for training
- Broader applicability to different tasks (physical cost estimation, knob tuning, physical design tuning)

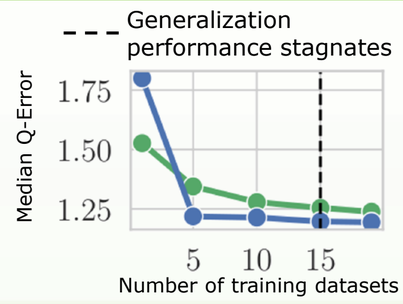


Key Challenges



1. Challenge: Query encoding that generalizes across databases

- **Problem**: Can't use representation of workload-driven models as they don't allow transfer across DBs
- **Solution**:
  - Capture query plans as graph encoding
  - Lean how expensive a certain operation on a dataset is
  - Encode information (e.g. data type, tuple width) of data where operations are executed on
  - Annotate operators with the intermediate sizes they are executed on (⟶ informed by data-driven models)



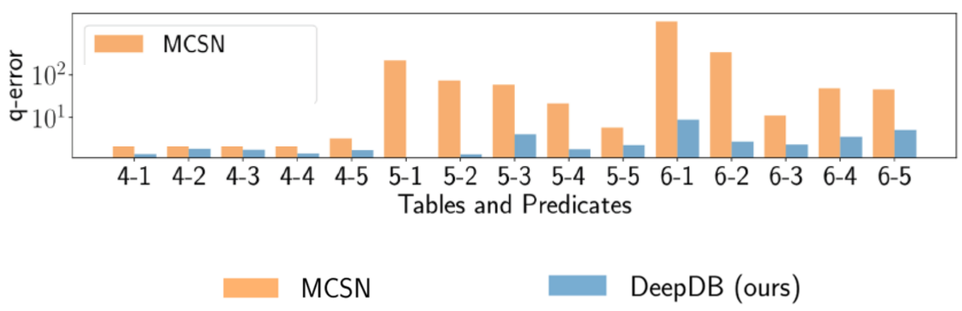
2. Challenge: Sufficient training

- **Goal**: Find out when model is sufficiently trained
- **Idea**: Estimate using holdout databases. If estimated performance is acceptable or stagnates, stop
- **Assumption**: Every DB and workload is sampled i.i.d.

Evaluation

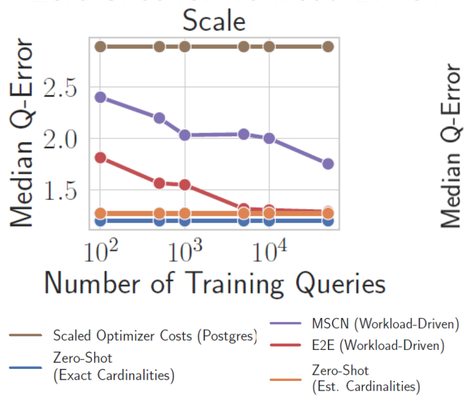
Generalizability to larger joins

Data-Driven vs. Workload-Driven

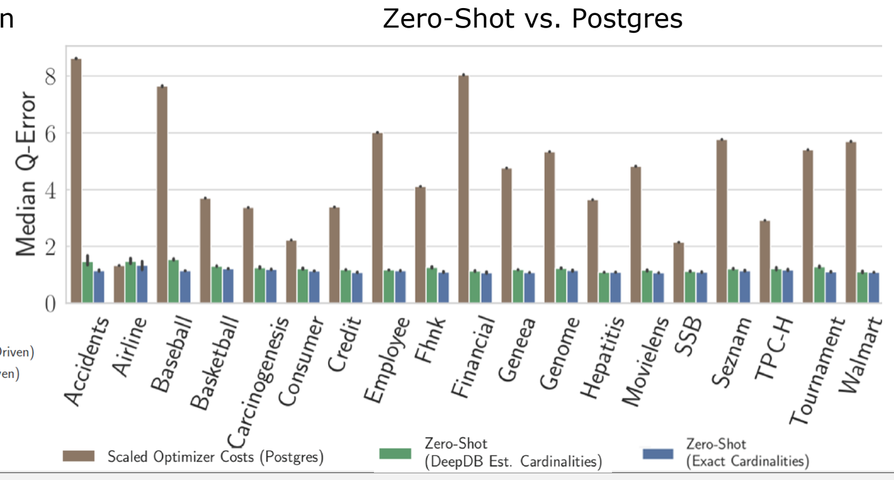


Query runtime estimation

Zero-Shot vs. Workload-Driven



Zero-Shot vs. Postgres



Vincent Melisch
Diploma Computer Science Student at the TU Dresden

E-Mail: vincent.melisch@tu-dresden.de

Resources

Based on Prof. Dr. Carsten Binnig's lecture *Learned DBMS Components* as part of the Lecture Series on Database Research

Graphics are taken from the slides of Prof. Dr. Carsten Binnig

HPI Hasso Plattner Institut