Special Topics:
Self-Driving Database Management Systems
Autonomous Systems IV
@Ying_Jiang // 15-799 // Spring 2022
LAST CLASS

openGauss
→ Learned Optimizer, Learned Advisor, Model Validator...
→ (Partly) adopted by real customers in China

Verdict: “Database Learning”
→ Enhances the Approximate Query Processing engine
→ Learns from past query answers to get smarter
A Unified Transferable Model for ML-Enhanced DBMS

Solving DBMS tasks using machine learning approaches.
Learned cardinality estimator,
Learned cost estimator,
Learned index, etc.

Unified: one model for multiple DBMS tasks.
Transferable: one model transferable across various datasets.

Source: Ziniu Wu
TODAY’S AGENDA

Overview
Multi-tasking meta-learning framework
Query Optimizer: A Case Study
Experiments
Parting Thoughts
TODAY’S AGENDA

Overview
Multi-tasking meta-learning framework
Query Optimizer: A Case Study
Experiments
Parting Thoughts
EXISTING ARCHITECTURES

One model for one task / one dataset (one DB)

Training from scratch is expensive
→ Requires huge amount of executed queries
→ Cold-start: not transferrable across DBs

Ignore relations between tasks
→ Redundant learning of common knowledge
→ Neither efficient nor effective

Each ML model serves a specific task

CardEst  CostEst  JoinSel  Index  …

Each ML model trained from scratch on each new dataset

Source: Zinu Wu
MOTIVATION: TRANSFERABILITY?

Learned knowledge decomposition: data and task

Data-agnostic meta-knowledge: common rules in DBs e.g., expert experiences and multi-table join rules

Data-specific knowledge: e.g., data distributions and join schema

Task specific knowledge: method to solve a specific task, e.g. join order selection rules

Task shared knowledge: data and query representations used in all tasks

Source: Ziniu Wu
OVERVIEW

Distill, share and reuse.

→ Identify and classify transferable and non-transferable knowledge

→ MTMLF, the mighty “multi-tasking meta-learning framework”
  
  Multitask training procedure: Train different tasks simultaneously
  
  Meta-learning paradigm: Pretrain + finetune to adapt quickly to new DB

→ Case study: a concrete model in Query Optimization

→ Experiments

→ Future research directions
  
  More tasks incorporated!
  
  Pre-train + finetune paradigm motivates federated learning
TODAY’S AGENDA

Overview
Multi-tasking meta-learning framework
Query Optimizer: A Case Study
Experiments
Parting Thoughts
A Unified Model

Detailed model architecture

Shared Representation Module

- CardEst model
- CostEst model
- JoinSel model
- Scheduling model

Query optimization tasks

Task-specific knowledge

Cross-DB Meta-Learning

Task-shared knowledge

Database-agnostic Meta knowledge

Database-specific knowledge

Featurization and encoding

Data (Tables in DB)

Queries (SQL, plan tree)

Other inputs (e.g., machine property)

Source: Ziniu Wu
AN ENVISIONED DB SERVICE

Service provider
→ Train on multiple anonymized users’ DBs
→ Provide users the shared representation and task-specific modules
→ Periodically collect anonymized information from users to update the model as service upgrade

Users
→ Train featurization module on data, plug in to the provided modules
→ Execute a small number of queries to fine-tune the entire model
→ Optionally provide anonymized information to the service provider

Source: Ziniu Wu
WORKFLOW

Task-specific module

Featurization module
For data 1

Shared representation module

Data 1

Service provider:
provide pre-trained modules, fixed during training for various datasets

Users: train featurization module for their own DB

Source: Ziniu Wu
**UNIFIED MODEL: ADVANTAGES**

**Architecture**
- More efficient without redundant learning
- More effective task modeling
- Transferability

**DB service**
- Recyclable computation: pre-trained large models provided as a fundamental tool
- Continuous optimization: see more (anonymized) data from users
- Robust against data update and workload shift: user update F module

Source: Ziniu Wu
TODAY’S AGENDA

Overview
Multi-tasking meta-learning framework
Query Optimizer: A Case Study
Experiments
Parting Thoughts
QO: A CASE STUDY

Jointly learn multiple tasks in Query Optimization together

Technical contribution on join order selection task:

→ Details in [Wu et al. 2021]

1. Propose a new encoding method for a join order
2. Design a novel decoding method to ensure the legitimacy of output
3. New join order loss function to empower end-to-end training

Source: Ziniu Wu
QO: A CASE STUDY

Jointly learn multiple tasks in QO together

(F) Featurization and encoding

(F.i) Featurization:
- $t_f(T)/j(T_i, T_j)$

(F.ii) Encoding:
- $E(f(T_j))$
- $E(f(T_n))$

(F.iii) Tree structure embedding:
- Serializer

(S) Shared Representation

(S) CardEst Model

(T) Task-specific Module

(T.i) CardEst Model
- $M_{CardEst}$

(T.ii) CostEst Model
- $M_{CostEst}$

(T.iii) JOIN order Model
- $Trans_{JO}$

(L) Loss Criteria and training

(L.i) CardEst loss
- $Q_{error}(\text{Card}, \text{Card})$

(L.ii) CostEst loss
- $Q_{error}(\text{Cost}, \text{Cost})$

(L.iii) JOIN order loss
- CrossEntropy($\hat{p}_1$, $p_1$)

(L.iv) Gradient backpropogation
QO: A CASE STUDY

Jointly learn multiple tasks in QO together

- **(I) Inputs**
  - I.i) Data Tables
  - I.ii) Query

- **(F) Featurization and encoding**
  - F.i) Featurization
  - F.ii) Encoding
  - F.iii) Tree structure embedding

- **(S) Shared Representation**
  - S.i) Featurization
  - S.ii) Encoding

- **(T) Task-specific Module**
  - T.i) CardEst Model
  - T.ii) CostEst Model
  - T.iii) JO loss

- **(L) Loss Criteria and training**
  - L.i) CardEst loss
  - L.ii) CostEst loss
  - L.iii) JO loss

**DB-Specific knowledge**

**DB-Agnostic Task-shared knowledge**

**DB-Agnostic Task-specific knowledge**

**Loss Criterion**

A UNIFIED TRANSFERABLE MODEL FOR ML-ENHANCED BMS

CIDR 2022
QO: A CASE STUDY

Jointly learn multiple tasks in QO together
QO: A CASE STUDY

Jointly learn multiple tasks in QO together

For each table, each col, each op (j/f), provide a col value embedding (stored as .npy in code)

Notation:
- k nodes
- m timesteps / tables touched
- n total tables

DB-Specific knowledge

A UNIFIED TRANSFERABLE MODEL FOR ML-ENHANCED BMS
CIDR 2022

For each table, each col, each op (j/f), provide a col value embedding (stored as .npy in code)
QO: A CASE STUDY

Jointly learn multiple tasks in QO together

(F) Featurization and encoding
- F.i) Featurization: \( t/f(T_j)/j(T_i,T_j) \)
- F.ii) Encoding
- F.iii) Tree structure embedding

(S) Shared Representation

(T) Task-specific Module
- T.i) CardEst Model
- T.ii) CostEst Model
- T.iii) Join order Model

(L) Loss Criteria and training
- L.i) CardEst loss: \( \text{Qerror}(\text{Card}) \)
- L.ii) CostEst loss: \( \text{Qerror}(\text{Cost}) \)
- L.iii) JO loss: \( \text{CrossEntropy}(P_i, P) \)

Transformer \( \text{Enc}_i \) encode distribution of \( T_i \) after applying \( f(T_i) \).

\( \text{Enc}_i \) trained separately with a CardEst task on \( T_i \).
QO: A CASE STUDY

Jointly learn multiple tasks in QO together

Serializer: Transformers’ tree positional embedding [Shiv and Quirk 2019]
Converting tree structured plan [tables + op type + j(N_i) / E(f(N_i)) for node i] into vector E(P) = (E(N_1), …, E(N_k))
QO: A CASE STUDY

Jointly learn multiple tasks in QO together

Trans_Share: \( E(P) \rightarrow (S_1, S_2, \ldots S_k) \)
- From single table to multi-table distributions
- Model the interactions among tables and nodes
- Jointly trained with (T)

Notation:
- \( k \) nodes
- \( m \) timesteps / tables touched
- \( n \) total tables

(T) Task-specific Module
- T.i) CardEst Model
- T.ii) CostEst Model
- T.iii) Join order Model

(L) Loss Criteria and training
- L.i) CardEst loss
- L.ii) CostEst loss
- L.iii) JO loss
- L.iv) Gradient backpropogation

A UNIFIED TRANSFERABLE MODEL FOR ML-ENHANCED BMS CIDR 2022
**QO: A CASE STUDY**

Jointly learn multiple tasks in QO together

**M_CardEst, M_CostEst:**
- multiple-layer perceptrons,
  \( S \rightarrow \text{integer} \)
- Q-error loss: the factor \( L = \max(\text{pred}/\text{true}, \text{true}/\text{pred}) \)

**Notation:**
- \( k \) nodes
- \( m \) timesteps / tables touched
- \( n \) total tables
QO: A CASE STUDY

Jointly learn multiple tasks in QO together

**Trans_JO:**
- Seq2seq task using transformer decoder, beam-search for pruning
- \( S \rightarrow \text{seq } P_1 \ldots P_t \ldots P_m \)
- \( P_t \): Probability distribution of \( n \) tables
- Max entry indicates the next table for the left-deep join ordering
QO: A CASE STUDY

Jointly learn multiple tasks in QO together

Trans_JO:
- Token-level loss: cross entropy
- Seq-level loss: incorporated overall join order quality

Notation:
k nodes
m timesteps / tables touched
n total tables

DB-Agnostic Task-specific knowledge

(i) Inputs

I.i) Data Tables
Table T1, Table T2, ..., Table Tn

I.ii) Query Q
SELECT COUNT(*) FROM T1, T2, T3, ..., TN WHERE (joins) AND(f(T1)) AND ... AND(f(Tn))

(ii) Initial plan P:
Node: N1
Merge join on f(T1, T2)
Seq Scan T1 on f(T1)
Node: N2
Hash join on f(T1, T2)
Seq Scan T1 on f(T1)

(F) Featurization and encoding

(F.i) Featurization
\[ t/f(T_i)/j(T_i, T_j) \]

(S) Shared Representation

(S.i) CardEst Model M_CardEst

(S.ii) CostEst Model M_CostEst

(S.iii) Join order Model Trans_JO

(T) Task-specific Module

T.i) CardEst Model M_CardEst

T.ii) CostEst Model M_CostEst

T.iii) Join order Model Trans_JO

(L) Loss Criteria and training

L.i) CardEst loss Qerror(Card, Card)

L.ii) CostEst loss Qerror(Cost, Cost)

L.iii) JO loss CrossEntropy(\(\hat{p}_i, P_i\))

(L.iv) Gradient backpropogation

 Serializer

Loss Criterion

A UNIFIED TRANSFERABLE MODEL FOR ML-ENHANCED BMS

CMU-DB

CIDR 2022
QO: A CASE STUDY

Jointly learn multiple tasks in QO together

Trans_JO:
- Token-level loss: cross entropy
- Seq-level loss: incorporated overall join order quality

\[ L_{JO} = -\log(p(u^*|x)) + \sum_{u \in \mathcal{U}(x)} (1 - JOE(U(u, u^*))) \times \log(p(u|x)) \]
\[ + \lambda \times \log \sum_{u \in \mathcal{U}(x)} p(u|x) \]

Notation:
k nodes
m timesteps / tables touched
n total tables
QO: A CASE STUDY

Jointly learn multiple tasks in QO together

Notation:
k nodes
m timesteps / tables touched
n total tables

DB-Agnostic
Task-specific knowledge

(I) Inputs

I.i) Data Tables
Table T_1
Table T_2
... Table T_n

I.ii) Query Q

SELECT COUNT(*) FROM T_1, T_2, T_3, T_n WHERE (joins) AND (f(T_1)) AND ... AND (f(T_n))

Initial plan P:
Node: N1
Merge join on f(T_1, T_2)
Index Scan on f(T_3)
Scan T_1

Node: N2
Hash join on f(T_1, T_2)

Node: N3
Loop join on f(T_3, T_4)
Index Scan on f(T_5)
Scan T_3

(F) Featurization and encoding

F.i) Featurization
\(t/f(T_i)/j(T_i, T_j)\)

F.ii) Encoding

Enc_1
... Enc_n

F.iii) Tree structure embedding

Serializer

(S) Shared Representation

(T) Task-specific Module

T.i) CardEst Model
M_CardEst

T.ii) CostEst Model
M_CostEst

T.iii) Join order Model
Trans_JO

(L) Loss Criteria and training

L.i) CardEst loss
Qerror(Card, Card)

L.ii) CostEst loss
Qerror(Cost, Cost)

L.iii) JO loss
CrossEntropy(\(\hat{P}_1, P_1\))

Backprop to (S) & (T)

\(L_{QO} = L_{\text{card}} + L_{\text{cost}} + L_{\text{jo}}\)

Notation:
k nodes
m timesteps / tables touched
n total tables

A UNIFIED TRANSFERABLE MODEL FOR ML-ENHANCED BMS
CIDR 2022
TODAY’S AGENDA

Overview
Multi-tasking meta-learning framework
Query Optimizer: A Case Study
Experiments
Parting Thoughts
QO: Experiments

Experimental objective:

1. Show effectiveness on multi-task training.
2. Show the transferability to unseen DBs.

Experimental setups:

• Evaluation of #1 & #2: single dataset
  IMDB single dataset, Join order benchmark (21 tables, 113 queries)
• Evaluation of #3:
  Artificially generated 11 DBs with workloads using various distribution and table join schemas
EFFECTIVENESS ON A SINGLE DATASET

Accurate performance

- Q-Error = max(Predict/True, True/Predict)
- Better than Tree-LSTM
- Comparison with other state-of-the-art models left as future work

<table>
<thead>
<tr>
<th>Method</th>
<th>Cardinality Q-Error</th>
<th>Cost Q-Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>median</td>
<td>max</td>
</tr>
<tr>
<td>PostgreSQL</td>
<td>184</td>
<td>670,000</td>
</tr>
<tr>
<td>Tree-LSTM</td>
<td>8.78</td>
<td>696.29</td>
</tr>
<tr>
<td>Ours</td>
<td>4.48</td>
<td>614.45</td>
</tr>
</tbody>
</table>

Source: Ziniu Wu
Multi-task learning > separate learning

- 3.5x speed-up over PostgreSQL baseline
- Comparison with the state-of-the-art models left as future work

<table>
<thead>
<tr>
<th>Method</th>
<th>JoinSel</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Execution Time (min)</td>
</tr>
<tr>
<td>PostgreSQL</td>
<td>1143.2</td>
</tr>
<tr>
<td>Optimal</td>
<td>209.1</td>
</tr>
<tr>
<td><strong>Multi-task</strong></td>
<td>318.3</td>
</tr>
<tr>
<td>One-task</td>
<td>450.4</td>
</tr>
</tbody>
</table>
Train the S and T modules on 10 datasets, plug in the F module from the test dataset without access to its query workload.
TRANSFERABILITY ACROSS DATASETS

Single: Trained from scratch on only the test dataset with access to the query workload on test dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Execution Time (min)</th>
<th>Improvement Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>PostgreSQL</td>
<td>393.3</td>
<td>---</td>
</tr>
<tr>
<td><strong>Transferable</strong></td>
<td>234.1</td>
<td>40.6%</td>
</tr>
<tr>
<td><strong>Single</strong></td>
<td>219.5</td>
<td>44.3%</td>
</tr>
</tbody>
</table>

Source: Ziniu Wu
CONCLUSION AND FUTURE WORK

A vision for a new form of ML-enhanced DBMS
→ More efficient and effective for multiple DB tasks
→ Pre-trained model transferable to new datasets

Future Work
→ Explore multi-task learning on other DBMS tasks
→ Design details and implement the new DB service
TODAY’S AGENDA

Overview
Solving Constrained Optimization
Boosting Tuning Process with Meta-Learning
Evaluation
Parting Thoughts
PARTING THOUGHTS

Revolutionary: classifying knowledge of DB
- Splitting stages of the unified model in a conceptually reasonable way, and show its effectiveness

Transferability between multi-tasks & multi-DB

Practical?
- A single gigantic model and massive data
  - One model for all?
  - Benefit coming from a large model
- Frequency to update components?
NEXT CLASS

Autonomous Systems V