Faster ML over Joins of Tables

Arun Kumar

UC San Diego
Jacobs School of Engineering
Computer Science and Engineering

Strata Data Conference Mar 28, 2019
About Myself

2009: Bachelors in CSE from IIT Madras

Summer: 110F!

2009–16: MS and PhD in CS from UW-Madison

Winter: -40F!

2016–Now: Asst. Prof. at UC San Diego CSE

Ahh... 😊
Why Am I Here?

1. To share 2 interesting ideas from my recent research that may be relevant to data+ML systems practitioners

2. To learn from your questions, use cases, and experience

3. To explore connections and potential future collaborations
New abstractions, algorithms, and systems to **accelerate** ML applications from a data management/systems standpoint

**My Research**

**System Efficiency**

**Human Efficiency**

**ML/AI**

**Data Management**

**Systems**
End-to-End ML Application Lifecycle

Data Scientist/ML Engineer

Source → Build → Deploy

Data + ML Systems Implementations

Research Approach:

Abstract key steps + Formalize data model + Automate grunt work + Optimize execution

https://adalabucsd.github.io
An end-to-end system to simplify data sourcing and ML model selection

“Database perception” to simplify deployment of deep nets for analytics

https://adalabucsd.github.io
ML after Joins: The Problem

A fundamental bottleneck in feature engineering on structured data:

Many datasets are multi-table \iff ML toolkits assume single-table inputs \implies Materialize join output

Key-Foreign Key (KFK) Joins

\x System efficiency
\x Human efficiency
ML over Joins: Overview

Avoid Joins Physically

Avoid Joins Logically

Key-Foreign Key (KFK) Joins

Runs faster, same accuracy

Orion [SIGMOD 2015]

Morpheus [VLDB 2017]

Hamlet [SIGMOD 2016]

Hamlet++ [VLDB 2018]

Even faster, similar accuracy

System efficiency

Human efficiency
Running Example for ML over Joins

**ML Task:** Classify if a customer will *churn* or not

<table>
<thead>
<tr>
<th>CID</th>
<th>Churn?</th>
<th>Gender</th>
<th>Age</th>
<th>EmpID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Yes</td>
<td>Female</td>
<td>33</td>
<td>AMZN</td>
</tr>
<tr>
<td>2</td>
<td>No</td>
<td>Male</td>
<td>51</td>
<td>GOOG</td>
</tr>
<tr>
<td>3</td>
<td>Yes</td>
<td>Other</td>
<td>46</td>
<td>GOOG</td>
</tr>
<tr>
<td>4</td>
<td>No</td>
<td>Female</td>
<td>27</td>
<td>MSFT</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Employers</th>
</tr>
</thead>
<tbody>
<tr>
<td>EmpID</td>
</tr>
<tr>
<td>AMZN</td>
</tr>
<tr>
<td>GOOG</td>
</tr>
<tr>
<td>MSFT</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>

More joins possible, e.g., with neighborhood data, weather data, etc.

*Materializing such joins can blow up the data, even by 10x!*
Outline

4m   ML over Joins

12m  Avoiding Joins Physically

8m   Avoiding Joins Logically

4m   Ongoing/Future Work
**ORION: Factorized ML**

**Insight:** Decompose ML computations and push them down through joins

**Focus:** Generalized Linear Models (GLMs) solved using batch gradient descent methods

\[ \nabla L(w) = \sum_{i=1}^{N} g(w^T x_i, y_i)x_i \]

\[ x \equiv \begin{bmatrix} X_C & X_E \end{bmatrix} \]

\[ = \begin{bmatrix} w_C^T & w_E^T \end{bmatrix} \begin{bmatrix} X_C \\ X_E \end{bmatrix} = w_C^T X_C + w_E^T X_E \]

1 full iteration requires 2 scans of Employers, 1 scan of Customers

**Challenges Tackled:** Scalability, developability

*Learning Generalized Linear Models over Normalized Data. SIGMOD 2015*
ORION: Implementations and Adoption

Prototyped on PostgreSQL with UDFs; with MapReduce on Hive & Spark
Extended to probabilistic classifiers, clustering algorithms

Adopted/explored for production use by:

- Microsoft (Web security)
- LogicBlox (Retail)
- Google (Ads)
**MORPHEUS: Generalizing ORION**

**Q:** Can we avoid *manual rewriting* of each ML algorithm and “automate” factorized ML over ML systems?

**Idea:** Many ML algorithms are bulk *linear algebra* (LA) programs. Create a framework for rewrite rules for LA ops.

---

**Factorized ML: Prior Work**

- **ORION** (GLMs)
  - Stack 1
  - Stack 2
- **FDB-F** (OLS)
- **Other ML models**

**The MORPHEUS Approach**

- **GLMs**
- **OLS**
- **K-Means**
- **NMF**
- ... **MORPHEUS: Factorized LA**

- **R**
- **NumPy**
- **Oracle R Enterprise**
- **TensorFlow**
- **Apache SystemML**
- ... **Other ML models**

Towards Linear Algebra over Normalized Data. VLDB 2017
Bulk LA-based ML Algorithms

Ordinary Least Squares linear regression with normal equations

**Input:** Regular matrix $T$, $Y$, $w$

$$w = ginv(\text{crossprod}(T))(T^T Y)$$

Logistic regression with BGD; works for L-BFGS and Conjugate Gradient too

**Input:** Regular matrix $T$, $Y$, $w$, $\alpha$

```plaintext
for $i$ in 1 : max_iter do
    $w = w + \alpha \times (T^T (Y/(1 + \exp(Tw))))$
end
```
MORPHEUS: High-level Architecture

- Normalized schema and metadata
- ML algorithm expressed using LA ops on an LA system
- MORPHEUS
  Rewrite Rules for Factorized LA ops on an LA system
- Automatically factorized execution of ML algorithm on LA system

Towards Linear Algebra over Normalized Data. VLDB 2017
**MORPHEUS: Factorized LA Rewrite Rules**

**New Abstraction:** “Normalized Matrix” to represent join in LA

\[
\mathbf{T}(ID, X) \leftarrow \pi(\mathbf{S}(ID, X_S, FK) \bowtie_{FK=RID} \mathbf{R}(RID, X_R))
\]

\[
X \equiv [X_S X_R]
\]

\[
T_{n \times d} \quad S_{n \times d_S} \quad K_{n \times n_R} \quad R_{n_R \times d_R}
\]

\[
T = [S \quad KR] \quad K[i, j] = \begin{cases} 1, & \text{if } S[i].FK = j \\ 0, & o/w \end{cases}
\]

Framework of algebraic rewrite rules for many LA operations

**Left Matrix Multiplication:** \( Tw \rightarrow S w_S + K(Rw_R) \)

GLMs, K-means clustering, NMF, etc. automatically factorized

*Towards Linear Algebra over Normalized Data. VLDB 2017*
Automatically Factorized ML in MORPHEUS

Input: Regular matrix $T$, $Y$, $w$, $\alpha$

for $i$ in $1$ : $\text{max\_iter}$ do
  $w = w + \alpha \ast (T^T(Y/(1 + \exp(Tw))))$
end

$T \equiv (S, K, R)$

MORPHEUS

Input: Normalized matrix $(S, K, R)$, $Y$, $w$, $\alpha$

for $i$ in $1$ : $\text{max\_iter}$ do
  $P = (Y/(1 + \exp(Sw[1 : d_S, ] +
          K(Rw[d_S + 1 : d_S + d_R, ]))))^T$
  $w = w + \alpha \ast [PS, (PK)R]^T$
end

Towards Linear Algebra over Normalized Data. VLDB 2017
Prototype in R for a dozen LA ops (~800 LOC); commodity machine

Snapshots of Empirical Results

S: Ratings
R_1: Users
R_2: Businesses

S: Listings
R_1: Hotels
R_2: Search details

Runtime (sec)

Towards Linear Algebra over Normalized Data. VLDB 2017
MORPHEUS: Implementations and Adoption

Library released for both R and Python NumPy
Supports “star” schemas for many LA ops
Some data cleaning/prep ops also factorized

MORPHEUSFI: Second-order feature interactions in Morpheus
MORPHEUSFLOW: “Lazy join” for SGD in TensorFlow
TOC: Generalized data compression for SGD

Towards Linear Algebra over Normalized Data. VLDB 2017
When is MORPHEUS not likely to be beneficial?

**Short Answer:** When the join(s) do *not* introduce much redundancy

**Case 1:**
Fact table is *not much taller* than dimension table(s)

**Case 2:**
Dimension table has *much fewer* features than fact table

**Case 3:** MLPs do *not* have much computational redundancy (anyway)

*Towards Linear Algebra over Normalized Data. VLDB 2017*
Demo of Using MORPHEUS in R and Python

https://adalabucsd.github.io/morpheus.html

https://github.com/lchen001/Morpheus

https://github.com/ADALabUCSD/MorpheusPy
Outline

4m   ML over Joins
12m  Avoiding Joins Physically
8m   Avoiding Joins Logically
4m   Ongoing/Future Work
Avoiding Joins Logically

Observation: Given EmplID, features of Employers are fixed!

Q: Are State and Revenue really “needed,” given EmplID?

Avoid the join “logically”: Use Foreign Key as “representative” feature
Why not use just feature selection?

Improves **accuracy** and/or interpretability

---

**Q:** Why bother with avoiding KFK joins?

**A:** Fewer features, **lower runtime/memory**
  Avoid procuring some tables altogether!

Basically, “short-circuit” feature selection using DB schema information!

---

**Q:** But what is the effect of avoiding KFK joins on ML accuracy?
Our Results: Bias will likely not increase, but Variance could

Basically, using Foreign Key as repr. may cause more overfitting

A novel accuracy-runtime tradeoff in ML

Q: How do we apply our theory of this tradeoff to practice?

Idea: Practical heuristic to bound rise in Variance: avoid join “safely”

Challenge: How to obtain a bound to avoid joins safely?

Q: But what is the effect of avoiding KFK joins on ML accuracy?
HAMLET: Technical Summary

- Adapted VC dimension-based bounds to avoid joins safely
- Simplified to a decision rule with “tuple ratio” (TR)

\[ TR = \frac{n}{|D_{FK}|} \]

- \( t \) depends on error tolerance and VC dim; set once with simulations
- For error tolerance 0.01, \( t = 20 \) for linear classifiers
- Surprisingly, for RBF-SVMs & MLPs, \( t = 7 \); for decision trees, \( t = 3! \)
- Upto 185x speed-ups on real data with similar errors

TR Rule: If \( TR > t \), safe to avoid join

---

To Join or Not to Join? Thinking Twice about Joins before Feature Selection. SIGMOD 2016
Are Key-Foreign Key Joins Safe to Avoid when Learning High-Capacity Classifiers? VLDB 2018
HAMLET: Adoption

Adopted/explored for production use by:

LogicBlox (Retail)

facebook (Friend recommendations)

make my trip (Customer churn)

Google (Brain/TFX)
When is HAMLET not likely to be beneficial?

1. Tuple ratio is less than threshold $t$ for given model

2. FK feature is judged “uninterpretable”

3. FK feature has monotonically growing domain

**Caveat:** Dropping FK features o/w may raise bias and hurt accuracy!
Datasets and Sample Code for HAMLET

https://adalabucsd.github.io/hamlet.html

https://github.com/arunkk09/hamlet

https://github.com/pvn25/Hamlet_Extension
Outline

4 minutes
ML over Joins

12 minutes
Avoiding Joins Physically

8 minutes
Avoiding Joins Logically

4 minutes
Ongoing/Future Work
Ongoing Work Related to ML over Joins

1. Lazy joins and factorized ML for XGBoost and latest SparkML

2. TOC-compressed ML for in-RDBMS ML and SparkML

3. Theoretical explanation of TR Rule for infinite VC-dim models
Other Major ADALab Projects

SortingHat
ML schema inference and auto data prep

Cerebro
High-throughput deep net model selection system

Krypton
Faster saliency maps to explain CNN predictions

SpeakSQL
Speech-driven multimodal data querying interface
Supun Nakandala, 2nd yr PhD

Vraj Shah, 3rd yr PhD

Yu Hao Zhang, 2nd yr MS->PhD

https://adalab.ucsd.github.io
Plug: First Textbook on ML Systems

Table of Contents
Introduction
ML Through Database Queries and UDFs
Multi-Table ML and Deep Systems Integration
Rewrites and Optimization
Execution Strategies
Data Access Methods
Resource Heterogeneity and Elasticity
Systems for ML Lifecycle Tasks
Conclusions
Bibliography
Authors' Biographies

THANK YOU!

https://adalabucsd.github.io

arunkk@eng.ucsd.edu

github.com/adalabucsd

@TweetAtAKK

ACKS: NSF, NIH NIDDK, Helmax, Oracle, Google, Opera Solutions, NVIDIA
Rate today’s session

Cyberconflict: A new era of war, sabotage, and fear

We’re living in a new era of constant sabotage, misinformation, and fear, in which everyone is a target, and you’re often the collateral damage in a growing conflict among states. From crippling infrastructure to sowing discord and doubt, cyber is now the weapon of choice for democracies, dictators, and terrorists.

David Sanger explains how the rise of cyberweapons has transformed geopolitics like nothing since the invention of the atomic bomb. Moving from the White House Situation Room to the dens of Chinese, Russian, North Korean, and Iranian hackers to the boardrooms of Silicon Valley, David reveals a world coming face-to-face with the perils of technological revolution—a conflict that the United States helped start when it began using cyberweapons against Iranian nuclear plants and North Korean missile launches. But now we find ourselves in a conflict we’re uncertain how to control, as our adversaries exploit vulnerabilities in our hyperconnected nation and we struggle to figure out how to deter these complex, short-of-war attacks.

David Sanger
The New York Times

David E. Sanger is the national security correspondent for the New York Times as well as a national security and political contributor for CNN and a frequent guest on CBS This Morning, Face the Nation, and many PBS shows.
Backup Slides
Golden Age of ML Analytics

Still, fundamental **usability and efficiency bottlenecks** in the **end-to-end process** of building and deploying ML applications

*Forbes, IDC*
Background: Bias-Variance Tradeoff

Statistical learning theory view of test (prediction) error:

Test Error = **Bias** + **Variance** + Irreducible Noise

Aka "Overfitting"