Factorized Machine Learning: Paths and Roadblocks

Arun Kumar

UC San Diego
JACOBS SCHOOL OF ENGINEERING
Computer Science and Engineering

UC San Diego
HALICIOĞLU DATA SCIENCE INSTITUTE

Factorized Databases Workshop

Aug 3, 2022

Golden Age of ML Analytics

Golden Age of ML Analytics

Healthcare

Insurance

Retail

Sciences

Golden Age of ML Analytics

amazon FACEBOOK Google Microsoft

MAYO CLINIC
Healthcare

AMERICAN FAMILY INSURANCE
Insurance

Walmart
Retail


Sciences

$ 38 billion
in 2019*

$ 500 billion
by 2025*

*International Data Corporation

Golden Age of ML Analytics

amazon FACEBOOK Google Microsoft

MAYO CLINIC	AMERICAN FAMILY INSURANCE	Walmart	
Healthcare	Insurance	Retail	Sciences

$ 38 billion in 2019* → **$ 500 billion** by 2025*

scikit learn, dmlc XGBoost, Apache Spark ML, TensorFlow, PyTorch

*International Data Corporation

Golden Age of ML Analytics

amazon FACEBOOK Google Microsoft

MAYO CLINIC
Healthcare

AMERICAN FAMILY INSURANCE
Insurance

Walmart
Retail


Sciences

$ 38 billion
in 2019*

$ 500 billion
by 2025*

scikit learn
dmlc XGBoost
APACHE Spark ML
TensorFlow
PyTorch

amazon Google Microsoft

*International Data Corporation

Golden Age of ML Analytics

amazon FACEBOOK Google Microsoft

MAYO CLINIC
Healthcare

AMERICAN FAMILY INSURANCE
Insurance

Walmart
Retail


Sciences

$ 38 billion
in 2019*

$ 500 billion
by 2025*

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

*International Data Corporation

My Research

New abstractions, algorithms, and software systems
to "***democratize***" ML-based data analytics from
a data management/systems standpoint

My Research

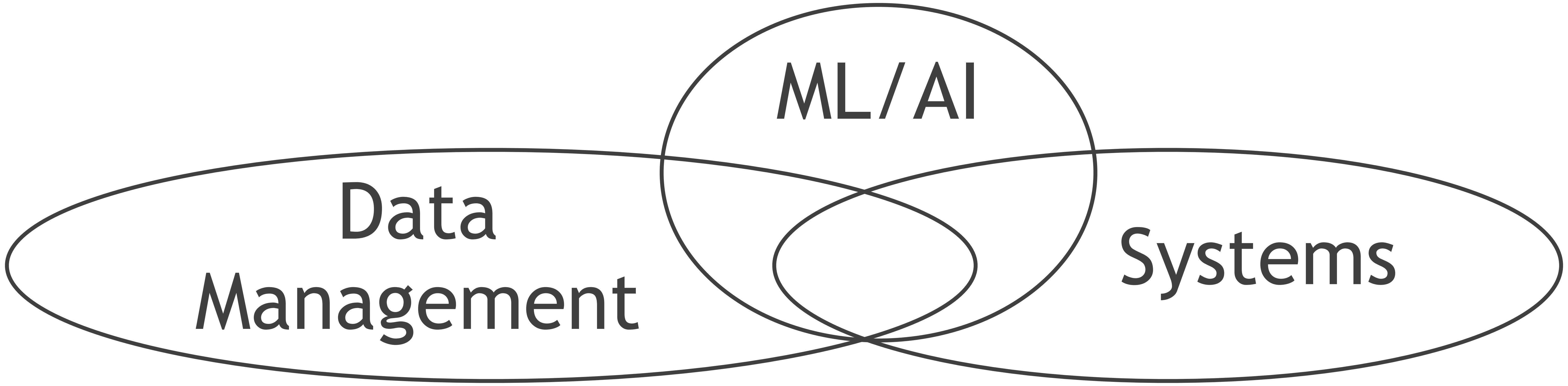
New abstractions, algorithms, and software systems to "***democratize***" ML-based data analytics from a data management/systems standpoint

Democratization = System Efficiency (Reduce costs) + Human Efficiency (Improve productivity)

My Research

New abstractions, algorithms, and software systems to "***democratize***" ML-based data analytics from a data management/systems standpoint

Democratization = System Efficiency (Reduce costs) + Human Efficiency (Improve productivity)



My Research

New abstractions, algorithms, and software systems to "*democratize*" ML-based data analytics from a data management/systems standpoint

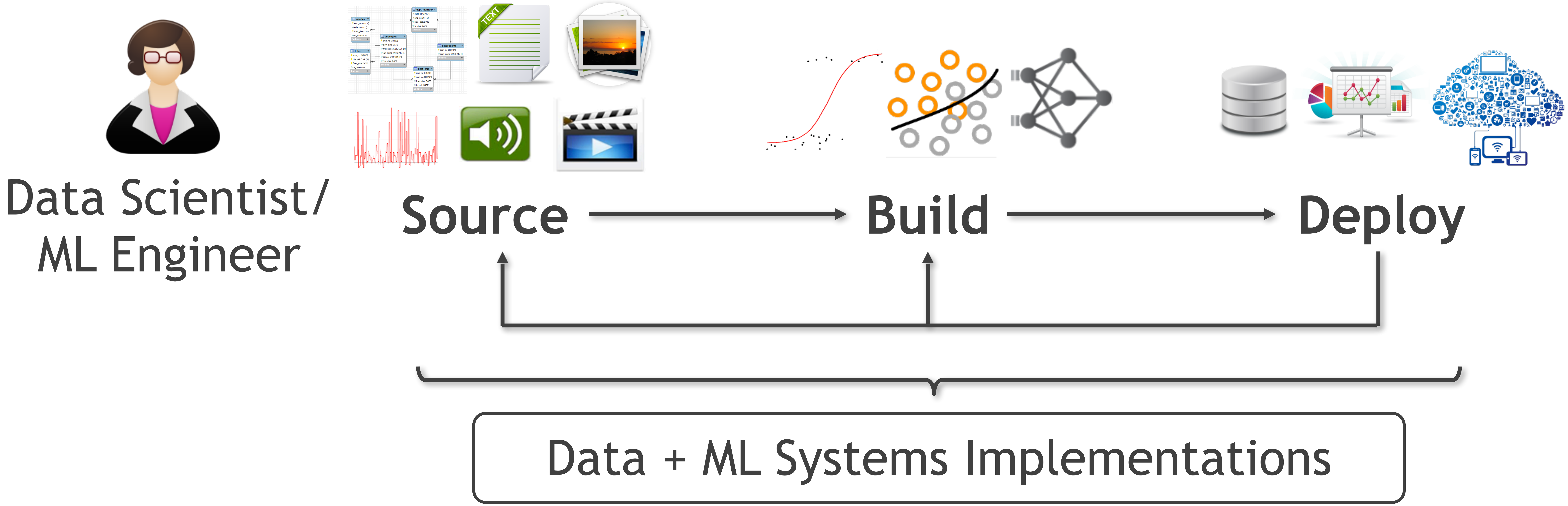
Democratization = System Efficiency (Reduce costs) + Human Efficiency (Improve productivity)

Practical and scalable data systems for ML analytics

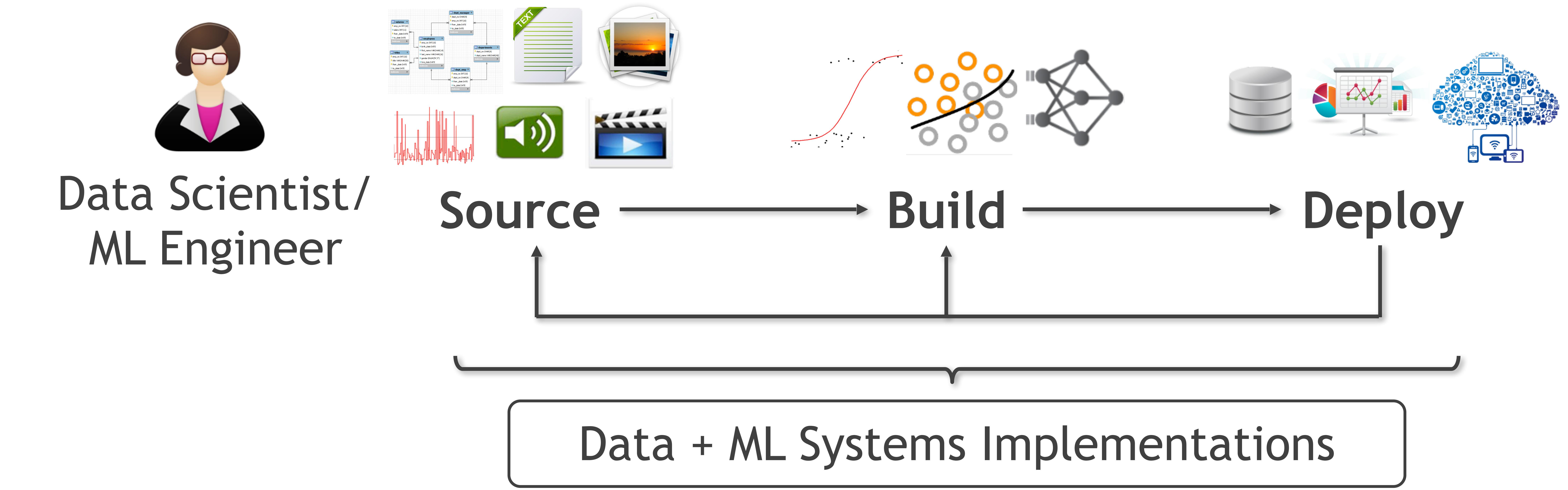
Inspired by *relational database systems* principles

Exploit insights from *learning theory* and *optimization theory*

End-to-End ML Application Lifecycle



End-to-End ML Application Lifecycle



Research Approach : ***Abstract*** key steps + ***Formalize*** computation + ***Automate*** grunt work + ***Optimize*** execution

https://ADALabUCSD.github.io

Outline

- Introducing ML over Joins
  - Orion: Factorized ML
  - Morpheus and Extensions
- Roadblocks and Musings

Outline

4m Introducing ML over Joins

4m Orion: Factorized ML

10m Morpheus and Extensions

4m Roadblocks and Musings

ML after Joins: The Problem

ML after Joins: The Problem

A fundamental bottleneck in feature engineering on structured data:

Many datasets are multi-table ⟷ ML toolkits assume single-table inputs

ML after Joins: The Problem

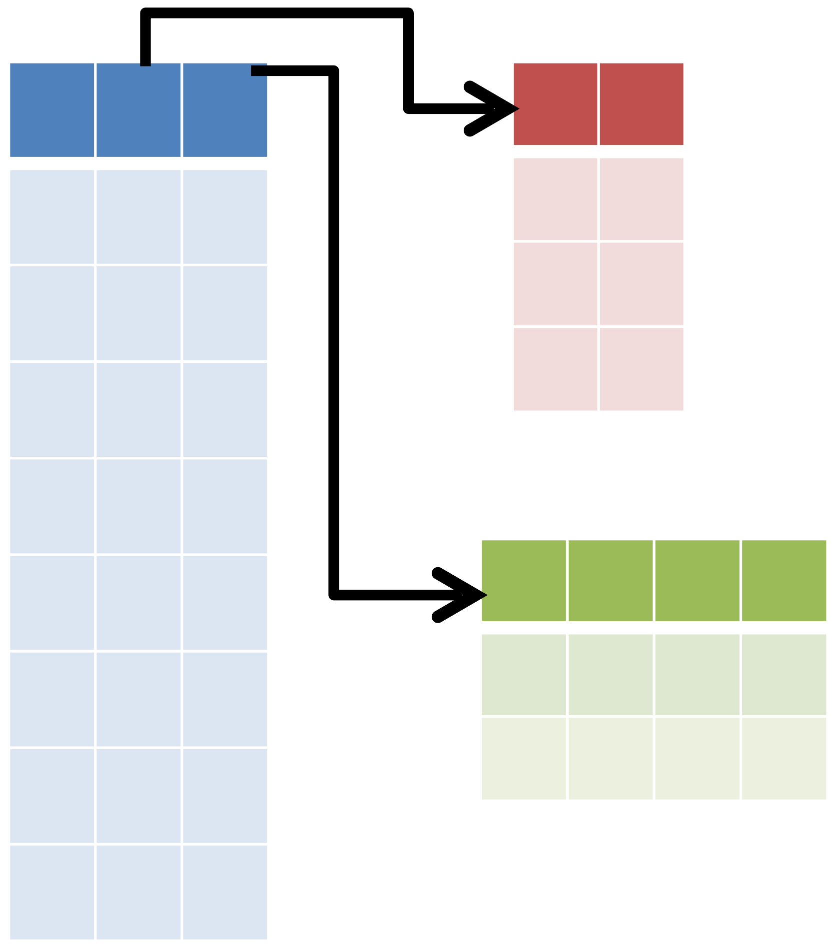
A fundamental bottleneck in feature engineering on structured data:

Many datasets are multi-table ⟷ ML toolkits assume single-table inputs ⇒ Materialize join output

ML <u>after</u> Joins: The Problem

A fundamental bottleneck in feature engineering on structured data:

Many datasets are <u>multi-table</u> ⟷ ML toolkits assume <u>single-table</u> inputs ⇒ <u>Materialize</u> join output



ML after Joins: The Problem

A fundamental bottleneck in feature engineering on structured data:

Many datasets are multi-table ⟷ ML toolkits assume single-table inputs ⇒ Materialize join output

ML after Joins: The Problem

A fundamental bottleneck in feature engineering on structured data:

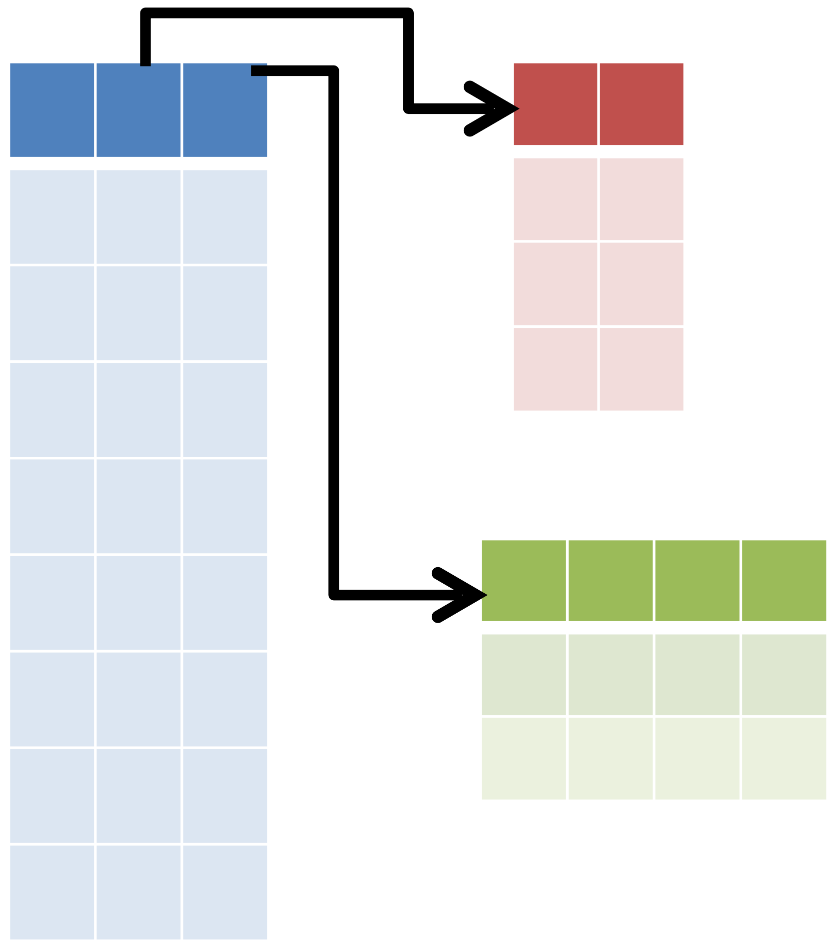
Many datasets are multi-table ⟷ ML toolkits assume single-table inputs ⇒ Materialize join output

AMERICAN FAMILY INSURANCE NETFLIX Walmart Microsoft Expedia

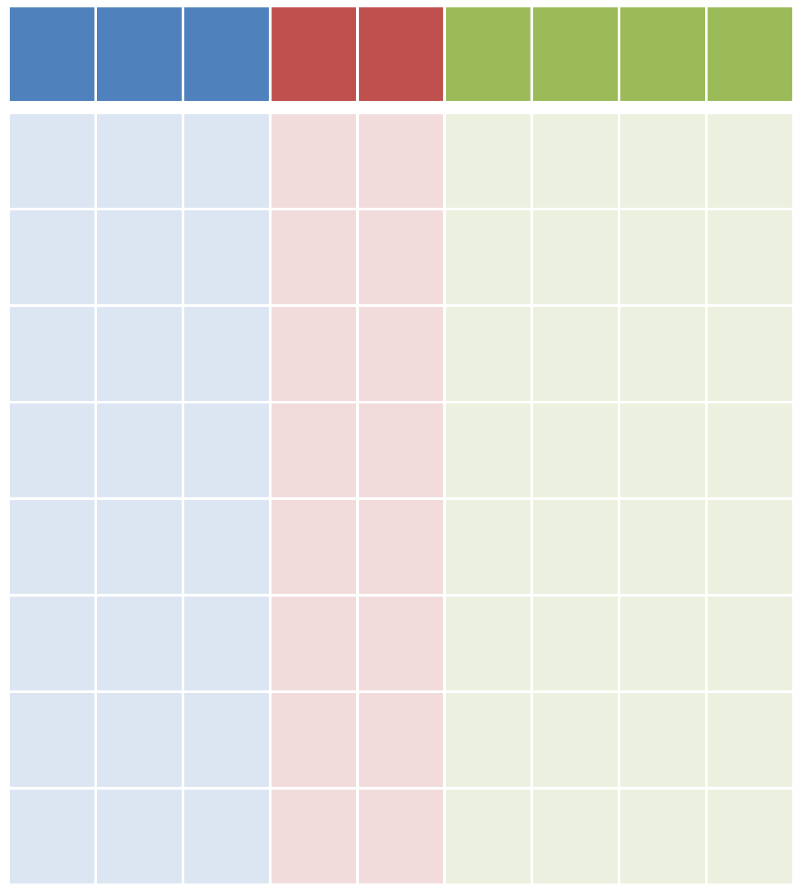
ML after Joins: The Problem

A fundamental bottleneck in feature engineering on structured data:

Many datasets are multi-table ⟷ ML toolkits assume single-table inputs ⇒ Materialize join output



Key-Foreign Key (KFK) Joins



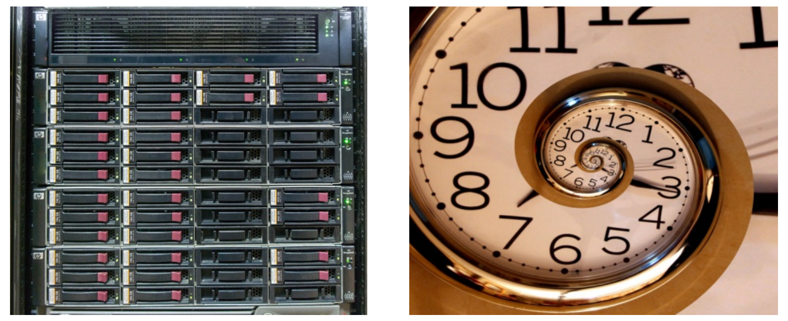
✖ System efficiency

AMERICAN FAMILY INSURANCE · NETFLIX · Walmart · Microsoft · Expedia

ML after Joins: The Problem

A fundamental bottleneck in feature engineering on structured data:

Many datasets are multi-table ⟷ ML toolkits assume single-table inputs ⇒ Materialize join output



✖ System efficiency

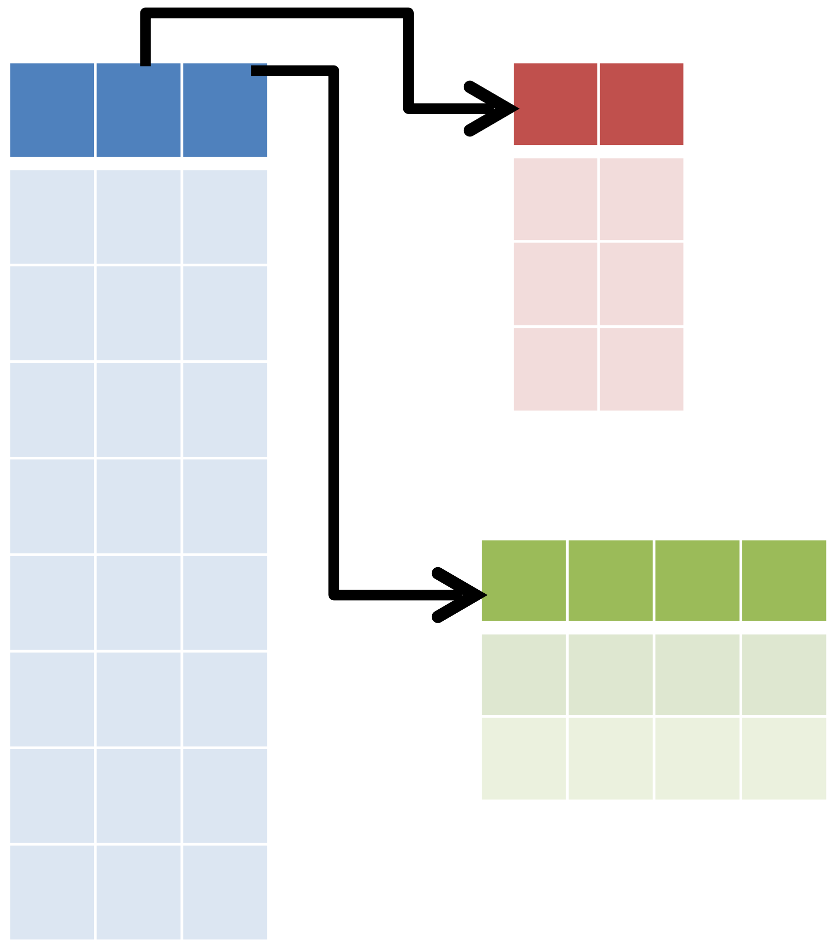
✖ Human efficiency

AMERICAN FAMILY INSURANCE NETFLIX Walmart Microsoft Expedia

ML <u>after</u> Joins: The Problem

A fundamental bottleneck in feature engineering on structured data:

Many datasets are <u>multi-table</u> ⟷ ML toolkits assume <u>single-table</u> inputs ⇒ <u>Materialize</u> join output



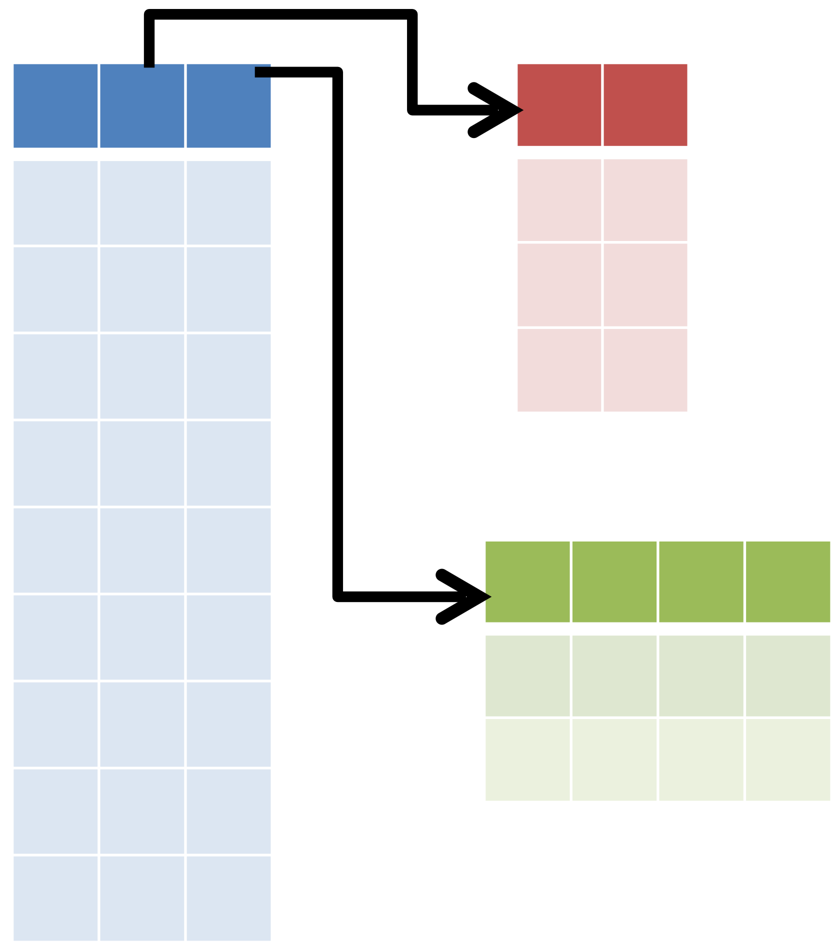
Key-Foreign Key (KFK) Joins

✖ System efficiency

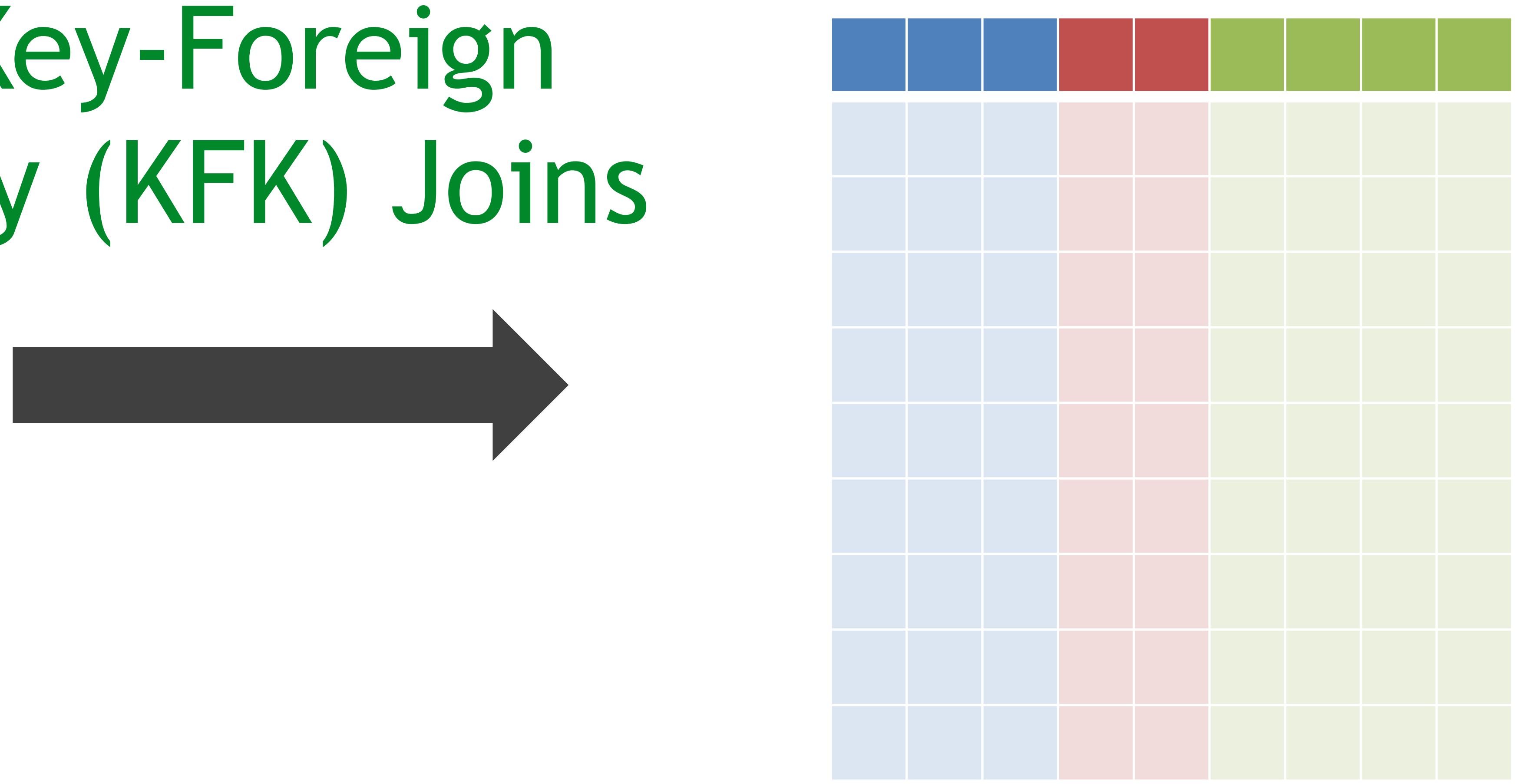
✖ Human efficiency

AMERICAN FAMILY INSURANCE   NETFLIX   Walmart   Microsoft   Expedia

ML over Joins: Overview



Key-Foreign Key (KFK) Joins



✗ System efficiency

✗ Human efficiency

ML over Joins: Overview

Avoid Joins **Physically**

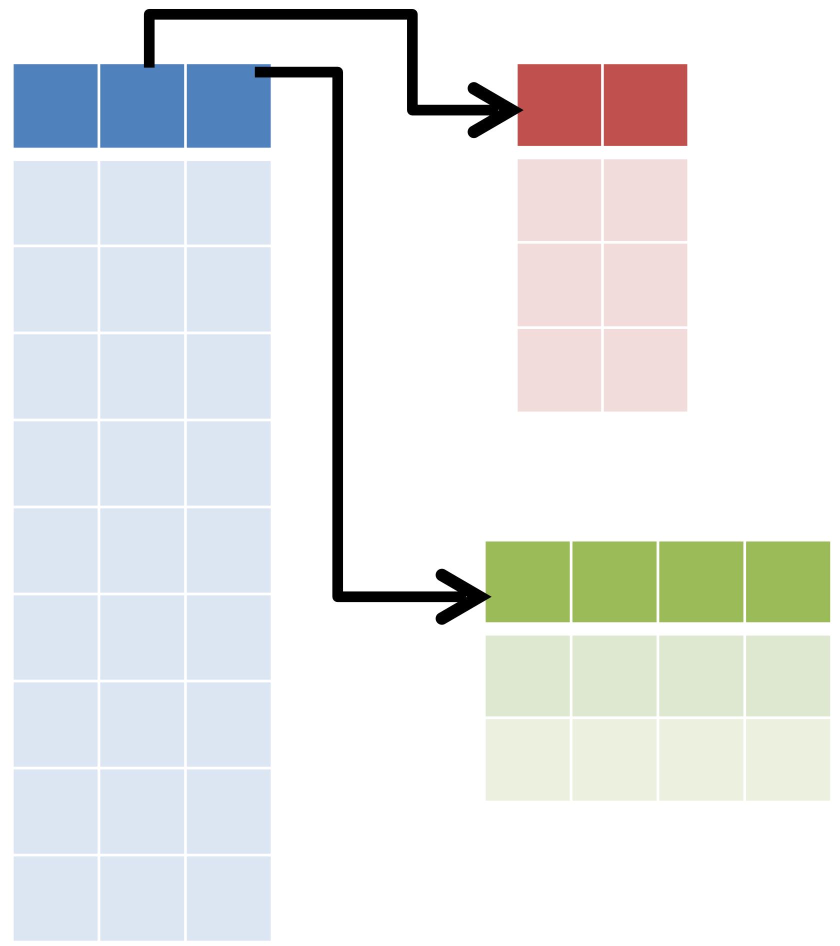
ORION, MORPHEUS

✖ System efficiency

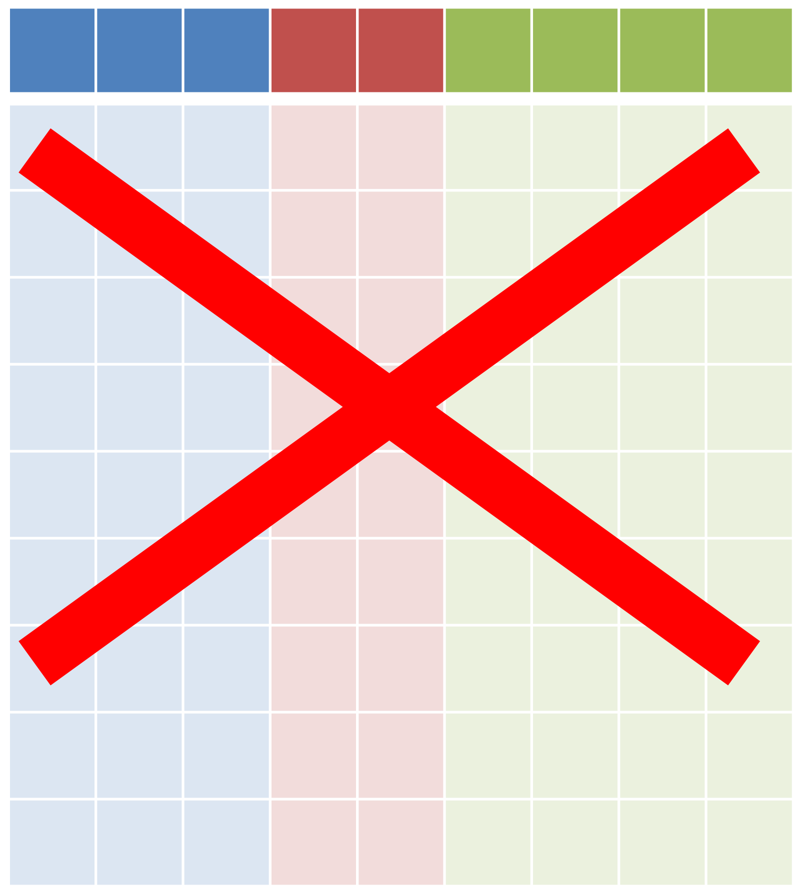
✖ Human efficiency

ML over Joins: Overview

Avoid Joins **Physically** ORION, MORPHEUS



Key-Foreign Key (KFK) Joins

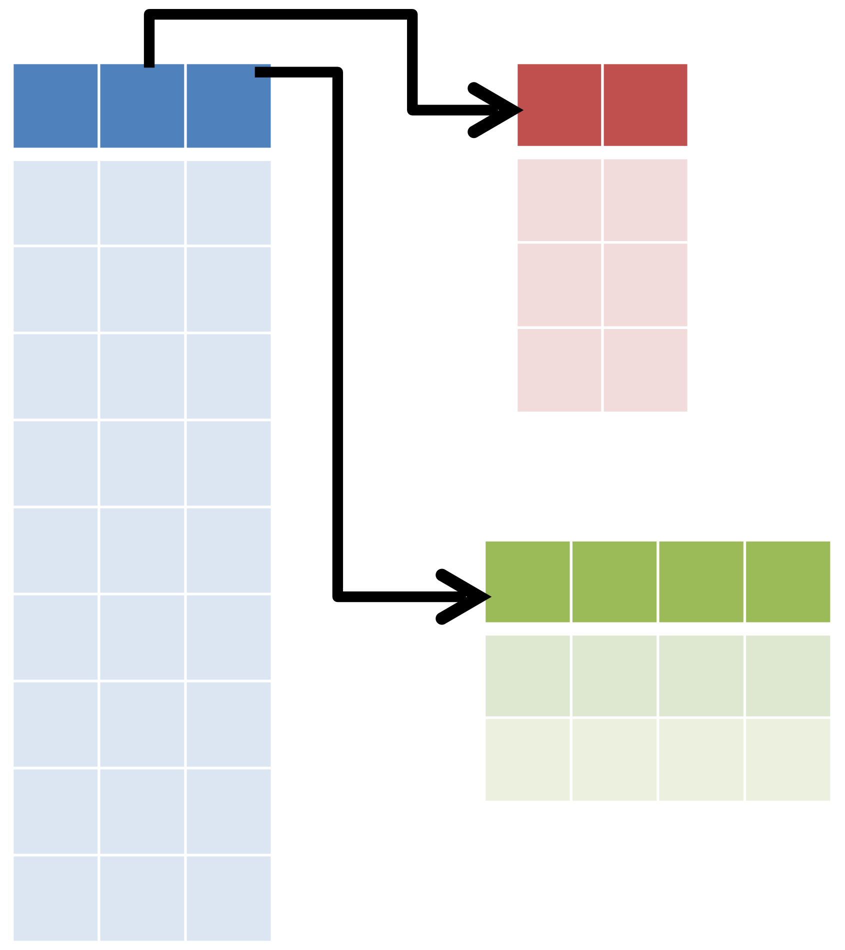


✖ System efficiency

✖ Human efficiency

ML over Joins: Overview

Avoid Joins **Physically** ORION, MORPHEUS



✖ System efficiency

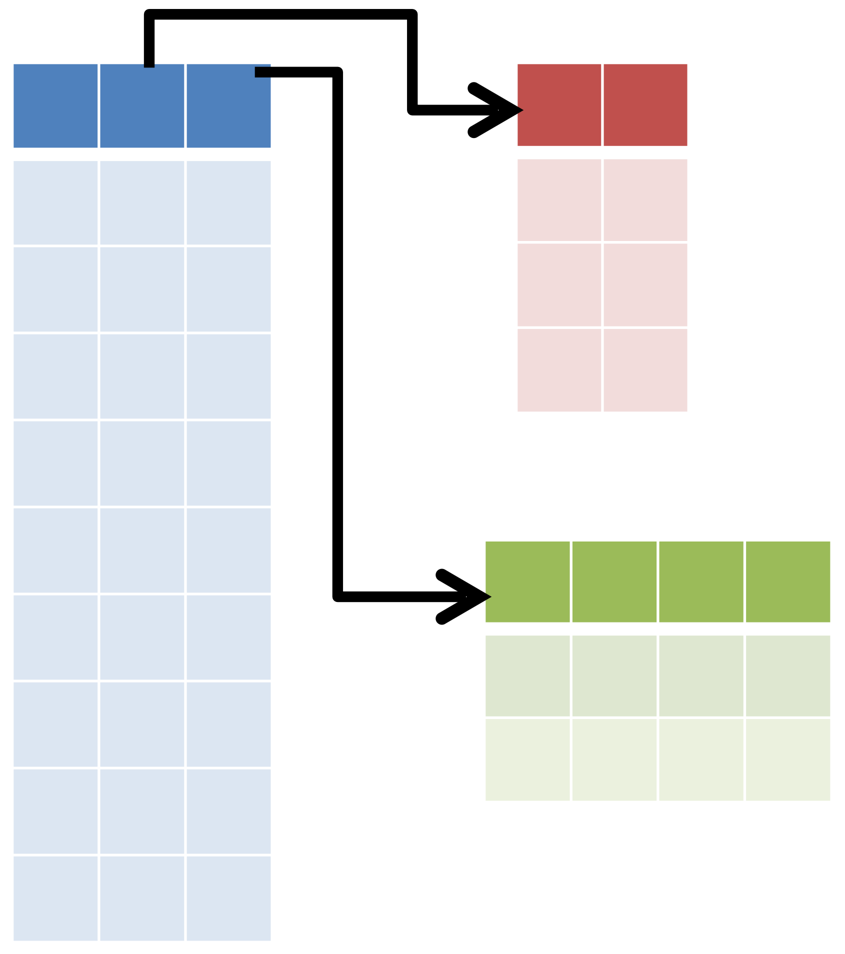
✖ Human efficiency

ML <u>over</u> Joins: Overview

Avoid Joins **Physically**

ORION, MORPHEUS

Runs faster, same accuracy



✖ System efficiency

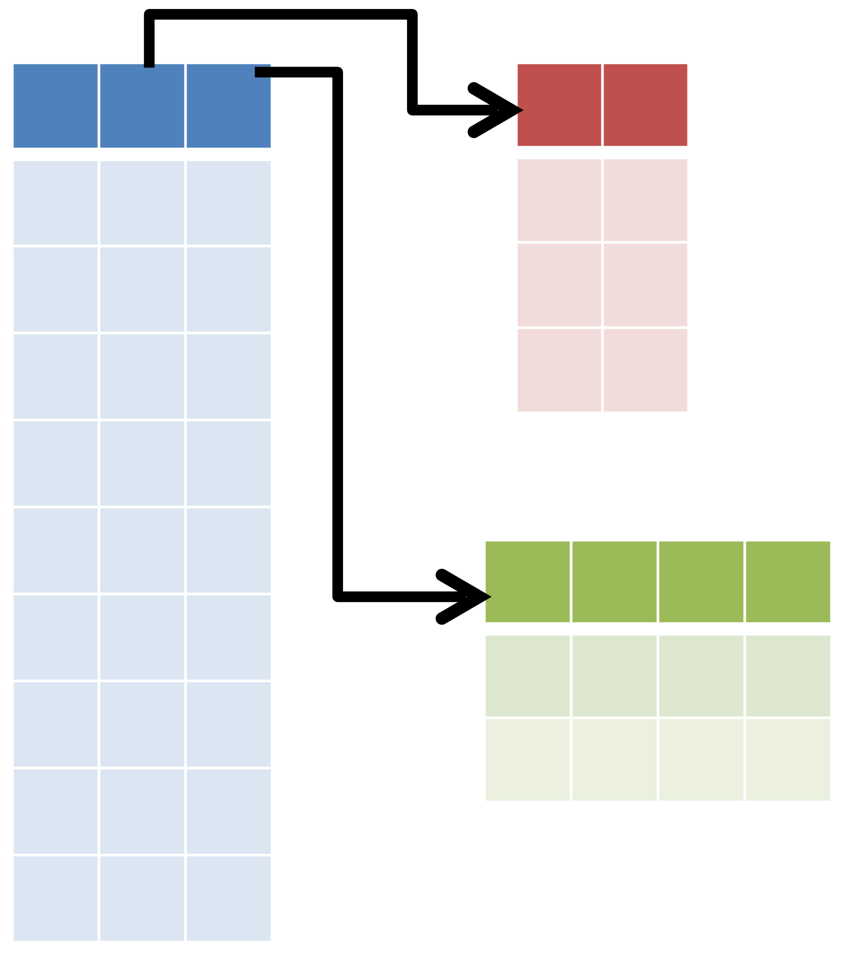
✖ Human efficiency

ML <u>over</u> Joins: Overview

Avoid Joins **Physically**

ORION, MORPHEUS

Runs faster, same accuracy



✖ System efficiency

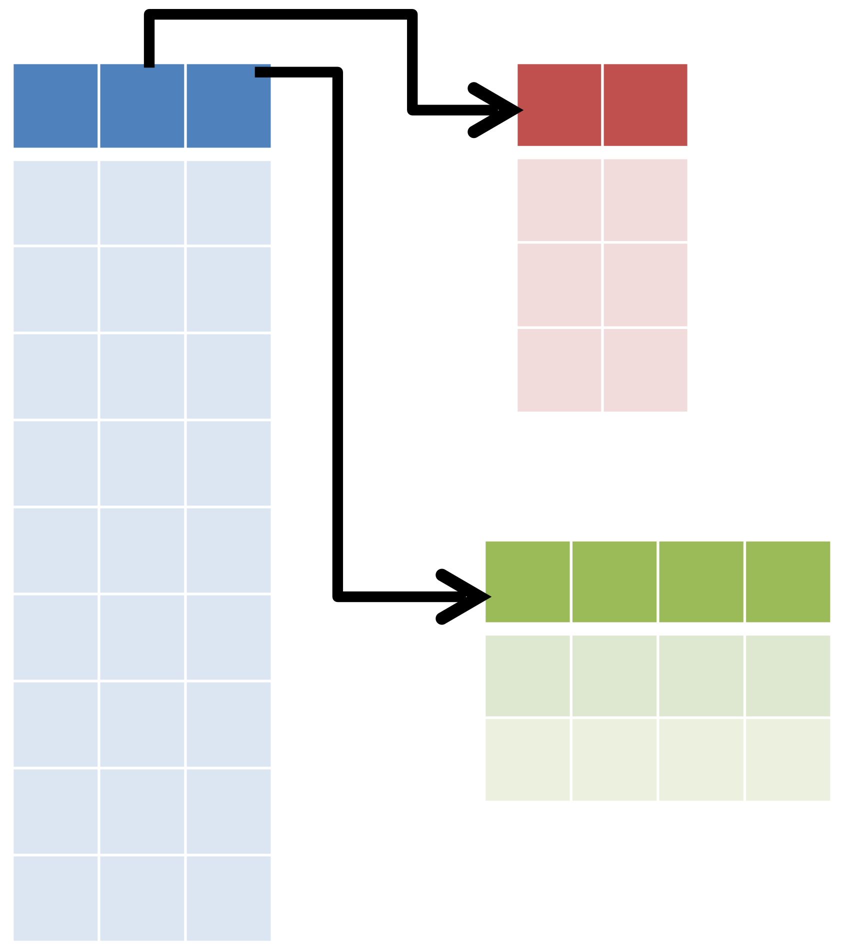
✚ Human efficiency

ML <u>over</u> Joins: Overview

Avoid Joins **Physically**

ORION, MORPHEUS

Runs faster, same accuracy



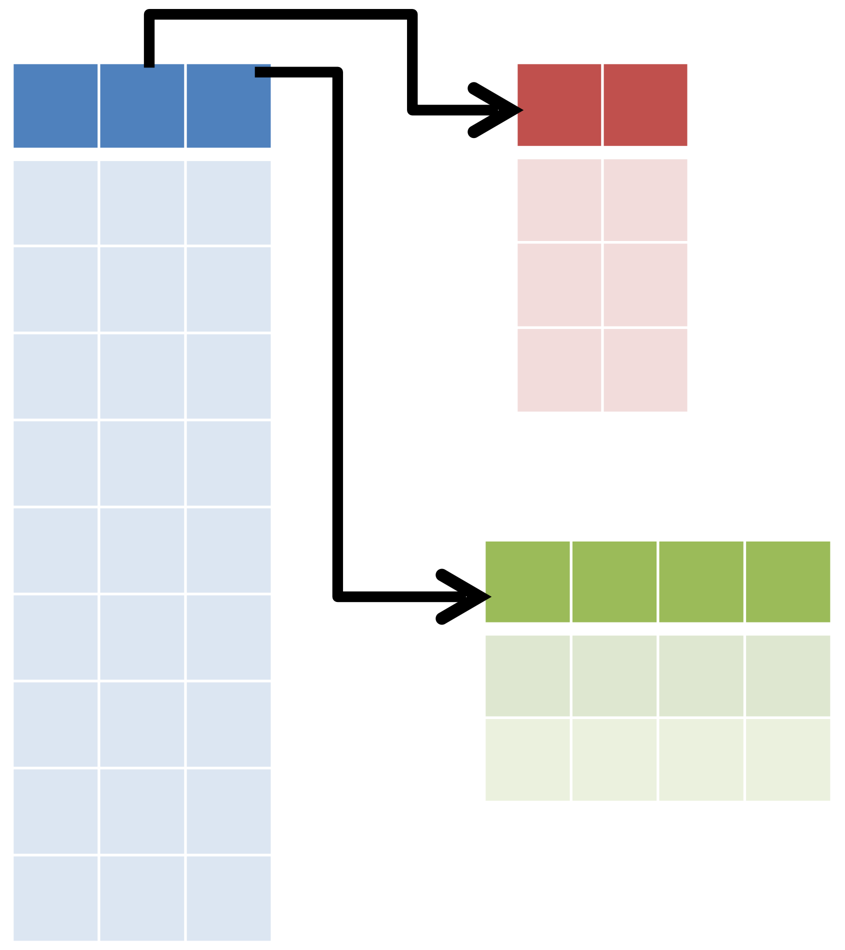
+ System efficiency

+ Human efficiency

ML <u>over</u> Joins: Overview

Avoid Joins **Physically** — ORION, MORPHEUS — *Runs faster, same accuracy*

Avoid Joins **Logically** — HAMLET, HAMLET++



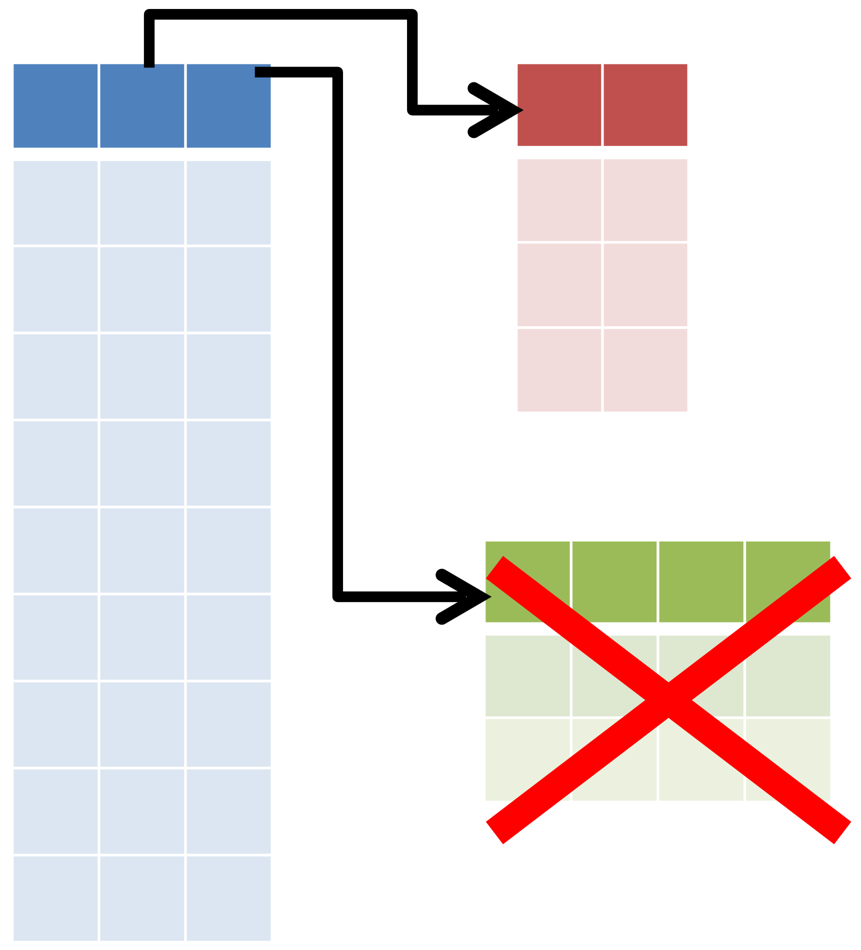
+ System efficiency

+ Human efficiency

ML over Joins: Overview

Avoid Joins **Physically** ORION, MORPHEUS *Runs faster, same accuracy*

Avoid Joins **Logically** HAMLET, HAMLET++

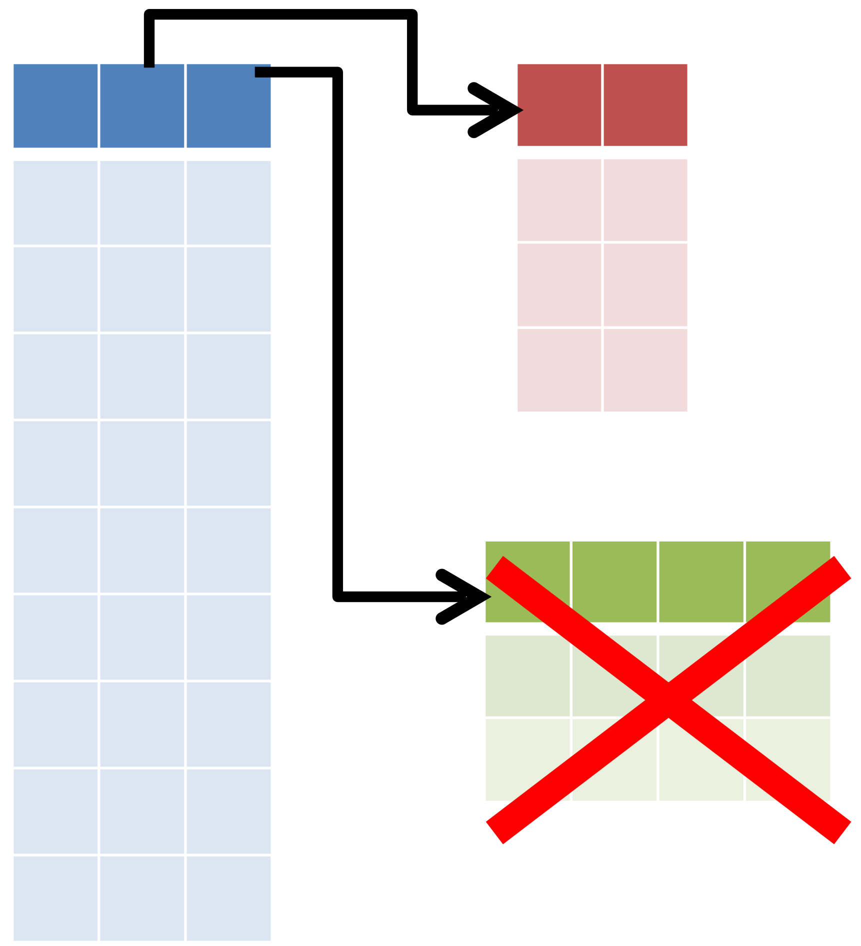


+ System efficiency

+ Human efficiency

ML ~~over~~ Joins: Overview

Avoid Joins **Physically**	**ORION, MORPHEUS**	*Runs faster, same accuracy*
Avoid Joins **Logically**	**HAMLET, HAMLET++**	*Even faster, similar accuracy*



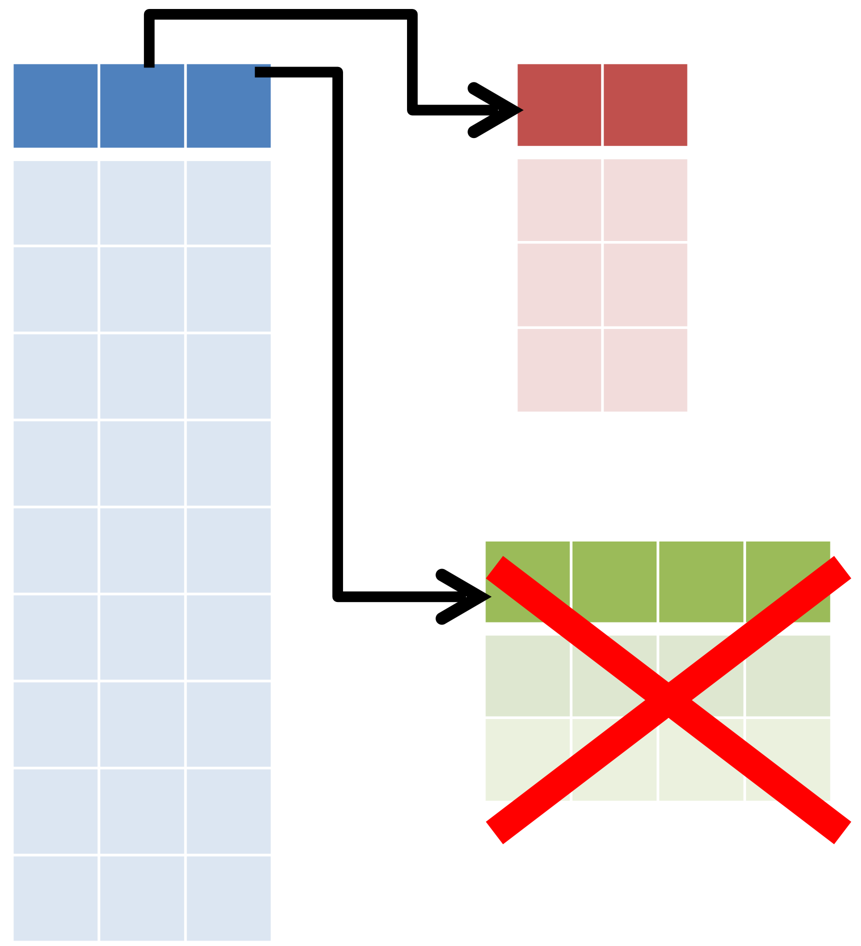
+ System efficiency

+ Human efficiency

ML over Joins: Overview

Avoid Joins **Physically** — ORION, MORPHEUS — *Runs faster, same accuracy*

Avoid Joins **Logically** — HAMLET, HAMLET++ — *Even faster, similar accuracy*



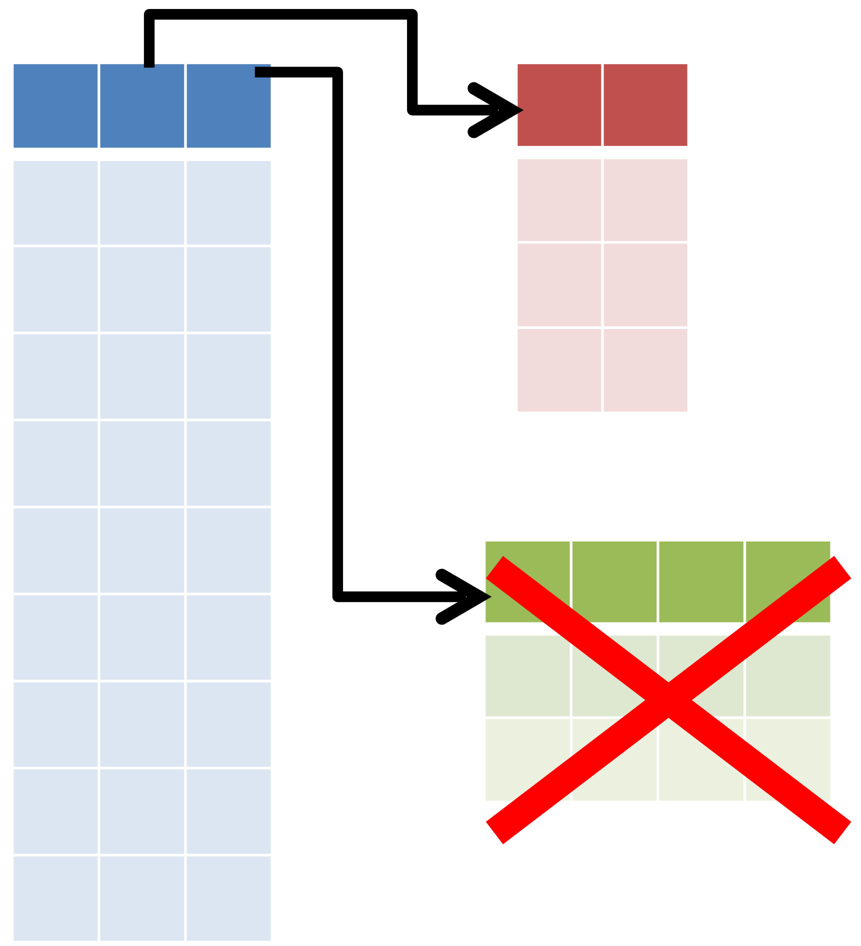
++ System efficiency

++ Human efficiency

ML over Joins: Overview

Avoid Joins **Physically** ORION, MORPHEUS *Runs faster, same accuracy*

Avoid Joins **Logically** HAMLET, HAMLET++ *Even faster, similar accuracy*



++ System efficiency

++ Human efficiency

Running Example for ML over Joins

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

AMERICAN FAMILY INSURANCE

Customers — Foreign Key → Employers

CID	Churn?	Gender	Age	EmpID
1	Yes	Female	33	AMZN
2	No	Male	51	GOOG
3	Yes	Other	46	GOOG
4	No	Female	27	MSFT
…	…	…	…	…

EmpID	State	Revenue
AMZN	WA	136b
GOOG	CA	89b
MSFT	WA	85b
…	…	…

Running Example for ML over Joins

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

Customers

Foreign Key

CID	Churn?	Gender	Age	EmpID
1	Yes	Female	33	AMZN
2	No	Male	51	GOOG
3	Yes	Other	46	GOOG
4	No	Female	27	MSFT
…	…	…	…	…

Employers

EmpID	State	Revenue
AMZN	WA	136b
GOOG	CA	89b
MSFT	WA	85b
…	…	…

More features!

Running Example for ML over Joins

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

Customers — Foreign Key: EmpID → **Employers**

CID	Churn?	Gender	Age	EmpID
1	Yes	Female	33	AMZN
2	No	Male	51	GOOG
3	Yes	Other	46	GOOG
4	No	Female	27	MSFT
…	…	…	…	…

EmpID	State	Revenue
AMZN	WA	136b
GOOG	CA	89b
MSFT	WA	85b
…	…	…

More features!

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

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

Outline

4m Introducing ML over Joins

4m Orion: Factorized ML

10m Morpheus and Extensions

4m Roadblocks and Musings

ORION: Factorized ML

Learning Generalized Linear Models over Normalized Data. **SIGMOD 2015**

ORION: Factorized ML

Insight: Decompose ML computations and push them down through joins

Learning Generalized Linear Models over Normalized Data. **SIGMOD 2015**

ORION: Factorized ML

Insight: Decompose ML computations and push them down through joins

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

$$\mathbf{X} \equiv [\mathbf{X_C}\ \mathbf{X_E}]$$

Learning Generalized Linear Models over Normalized Data. **SIGMOD 2015**

ORION: Factorized ML

Insight: Decompose ML computations and push them down through joins

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

$$\nabla L(\boldsymbol{w}) = \sum_{\mathrm{i}=1}^{\mathrm{N}} g\left(\boldsymbol{w}^{\mathrm{T}}\mathbf{x}_{\mathbf{i}}, \mathrm{y}_{\mathrm{i}}\right)\mathbf{x}_{\mathbf{i}} \qquad \mathbf{X} \equiv [\mathbf{X}_{\mathbf{C}}\ \mathbf{X}_{\mathbf{E}}]$$

Learning Generalized Linear Models over Normalized Data. **SIGMOD 2015**

ORION: Factorized ML

Insight: Decompose ML computations and push them down through joins

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

$$\nabla L(\boldsymbol{w}) = \sum_{i=1}^{N} g\left(\boxed{\boldsymbol{w}^{T}\mathbf{x}_{\mathbf{i}}}, y_i\right)\mathbf{x}_{\mathbf{i}} \qquad \mathbf{X} \equiv [\mathbf{X}_{\mathbf{C}} \ \mathbf{X}_{\mathbf{E}}]$$

Learning Generalized Linear Models over Normalized Data. **SIGMOD 2015**

ORION: Factorized ML

Insight: Decompose ML computations and <u>push them down</u> through joins

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

$$\nabla L(\boldsymbol{w}) = \sum_{i=1}^{N} g\left(\boxed{\boldsymbol{w}^T \mathbf{x_i}}, y_i\right)\mathbf{x_i} \qquad \mathbf{X} \equiv [\mathbf{X_C}\ \mathbf{X_E}]$$

$$\boldsymbol{w}^T \mathbf{x} = [\boldsymbol{w}_C^T\ \boldsymbol{w}_E^T]\begin{bmatrix}\mathbf{x_C}\\ \mathbf{x_E}\end{bmatrix} = \boldsymbol{w}_C^T \mathbf{x_C} + \boldsymbol{w}_E^T \mathbf{x_E}$$

Learning Generalized Linear Models over Normalized Data. **SIGMOD 2015**

ORION: Factorized ML

Insight: Decompose ML computations and <u>push them down</u> through joins

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

$$\nabla \mathrm{L}(\boldsymbol{w}) = \sum_{\mathrm{i}=1}^{\mathrm{N}} g\left(\boldsymbol{w}^{\mathrm{T}}\mathbf{x}_{\mathrm{i}}, y_{\mathrm{i}}\right)\mathbf{x}_{\mathrm{i}} \qquad \mathbf{X} \equiv [\mathbf{X}_{\mathrm{C}}\ \mathbf{X}_{\mathrm{E}}]$$

$$\boldsymbol{w}^{\mathrm{T}}\mathbf{x} = [\boldsymbol{w}_{\mathrm{C}}^{\mathrm{T}}\ \boldsymbol{w}_{\mathrm{E}}^{\mathrm{T}}]\begin{bmatrix}\mathbf{x}_{\mathrm{C}} \\ \mathbf{x}_{\mathrm{E}}\end{bmatrix} = \boldsymbol{w}_{\mathrm{C}}^{\mathrm{T}}\mathbf{x}_{\mathrm{C}} + \boxed{\boldsymbol{w}_{\mathrm{E}}^{\mathrm{T}}\mathbf{x}_{\mathrm{E}}}$$

Learning Generalized Linear Models over Normalized Data. **SIGMOD 2015**

ORION: Factorized ML

Insight: Decompose ML computations and push them down through joins

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

$$\nabla L(\boldsymbol{w}) = \sum_{i=1}^{N} g(\boldsymbol{w}^T \mathbf{x_i}, y_i) \boxed{\mathbf{x_i}} \qquad \mathbf{X} \equiv [\mathbf{X_C}\ \mathbf{X_E}]$$

$$\boldsymbol{w}^T \mathbf{x} = [\boldsymbol{w}_C^T\ \boldsymbol{w}_E^T] \begin{bmatrix} \mathbf{x_C} \\ \mathbf{x_E} \end{bmatrix} = \boldsymbol{w}_C^T \mathbf{x_C} + \boldsymbol{w}_E^T \mathbf{x_E}$$

Learning Generalized Linear Models over Normalized Data. **SIGMOD 2015**

ORION: Factorized ML

Insight: Decompose ML computations and <u>push them down</u> through joins

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

$$\nabla \mathrm{L}(\boldsymbol{w}) = \sum_{\mathrm{i}=1}^{\mathrm{N}} g(\boldsymbol{w}^{\mathrm{T}}\mathbf{x}_{\mathrm{i}}, \mathrm{y}_{\mathrm{i}})\boxed{\mathbf{x}_{\mathrm{i}}} \qquad \mathbf{X} \equiv [\mathbf{X}_{\mathrm{C}}\ \mathbf{X}_{\mathrm{E}}]$$

$$\boldsymbol{w}^{\mathrm{T}}\mathbf{x} = [\boldsymbol{w}_{\mathrm{C}}^{\mathrm{T}}\ \boldsymbol{w}_{\mathrm{E}}^{\mathrm{T}}]\begin{bmatrix}\mathbf{x}_{\mathrm{C}} \\ \mathbf{x}_{\mathrm{E}}\end{bmatrix} = \boldsymbol{w}_{\mathrm{C}}^{\mathrm{T}}\mathbf{x}_{\mathrm{C}} + \boldsymbol{w}_{\mathrm{E}}^{\mathrm{T}}\mathbf{x}_{\mathrm{E}}$$

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

Learning Generalized Linear Models over Normalized Data. **SIGMOD 2015**

ORION: Factorized ML

Insight: Decompose ML computations and <u>push them down</u> through joins

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

$$\nabla L(\boldsymbol{w}) = \sum_{i=1}^{N} g(\boldsymbol{w}^T \mathbf{x}_i, y_i)\boxed{\mathbf{x}_i} \qquad \mathbf{X} \equiv [\mathbf{X}_C \ \mathbf{X}_E]$$

$$\boldsymbol{w}^T\mathbf{x} = [\boldsymbol{w}_C^T \ \boldsymbol{w}_E^T]\begin{bmatrix}\mathbf{x}_C \\ \mathbf{x}_E\end{bmatrix} = \boldsymbol{w}_C^T\mathbf{x}_C + \boldsymbol{w}_E^T\mathbf{x}_E$$

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

Learning Generalized Linear Models over Normalized Data. **SIGMOD 2015**
Demonstration of Santoku: Optimizing Machine Learning over Normalized Data. **VLDB 2015**

ORION: Implementations

Prototyped on PostgreSQL with UDAFs (MADlib style)

Distributed prototype with MapReduce on Hive & Spark (MLlib style)

Extended to Naive Bayes, k-means clustering, decision trees as R package

Learning Generalized Linear Models over Normalized Data. **SIGMOD 2015**
Demonstration of Santoku: Optimizing Machine Learning over Normalized Data. **VLDB 2015**

ORION: Implementations

Prototyped on PostgreSQL with UDAFs (MADlib style)

Distributed prototype with MapReduce on Hive & Spark (MLlib style)

Extended to Naive Bayes, k-means clustering, decision trees as R package

Explored for production use cases:

Microsoft (Web security)

LogicBlox (Retail)

Google (Ads)

Learning Generalized Linear Models over Normalized Data. **SIGMOD 2015**
Demonstration of Santoku: Optimizing Machine Learning over Normalized Data. **VLDB 2015**

Q: *Can we avoid manual rewriting of each ML algorithm and "automate" factorized ML on top of ML tools?*

Outline

4m Introducing ML over Joins

4m Orion: Factorized ML

10m Morpheus and Extensions

4m Roadblocks and Musings

MORPHEUS: Generalizing ORION

Towards Linear Algebra over Normalized Data. **VLDB 2017**

MORPHEUS: Generalizing ORION

Goal: *Automate* factorized ML to many ML algorithms in a unified way

Towards Linear Algebra over Normalized Data. **VLDB 2017**

MORPHEUS: Generalizing ORION

Goal: *Automate* factorized ML to many ML algorithms in a unified way

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

Towards Linear Algebra over Normalized Data. **VLDB 2017**

MORPHEUS: Generalizing ORION

Goal: *Automate* factorized ML to many ML algorithms in a unified way

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

Factorized ML: Prior Work

Towards Linear Algebra over Normalized Data. **VLDB 2017**

Morpheus: Generalizing Orion

Goal: *Automate* factorized ML to many ML algorithms in a unified way

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

Towards Linear Algebra over Normalized Data. **VLDB 2017**

MORPHEUS: Generalizing ORION

Goal: *Automate* factorized ML to many ML algorithms in a unified way

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)	FDB-F (OLS)
↓	↓
Stack 1	Stack 2

Towards Linear Algebra over Normalized Data. **VLDB 2017**

Morpheus: Generalizing Orion

Goal: *Automate* factorized ML to many ML algorithms in a unified way

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)	FDB-F (OLS)	Other ML models
↓	↓	↓
Stack 1	Stack 2	…

Spark

Towards Linear Algebra over Normalized Data. **VLDB 2017**

Morpheus: Generalizing Orion

Goal: *Automate* factorized ML to many ML algorithms in a unified way

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)	FDB-F (OLS)	Other ML models
↓	↓	↓
Stack 1	Stack 2	…


The Morpheus Approach

Towards Linear Algebra over Normalized Data. **VLDB 2017**

Morpheus: Generalizing Orion

Goal: *Automate* factorized ML to many ML algorithms in a unified way

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)	FDB-F (OLS)	Other ML models
↓	↓	↓
Stack 1	Stack 2	…

Spark

The Morpheus Approach

GLMs OLS K-Means NMF …

Towards Linear Algebra over Normalized Data. **VLDB 2017**

MORPHEUS: Generalizing ORION

Goal: *Automate* factorized ML to many ML algorithms in a unified way

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

FDB-F (OLS) → Stack 2

Other ML models → …

Spark

The MORPHEUS Approach

GLMs, OLS, K-Means, NMF, … → MORPHEUS: Factorized LA

Towards Linear Algebra over Normalized Data. **VLDB 2017**

MORPHEUS: Generalizing ORION

Goal: *Automate* factorized ML to many ML algorithms in a unified way

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

FDB-F (OLS) → Stack 2

Other ML models → ...

Spark

The MORPHEUS Approach

GLMs, OLS, K-Means, NMF, ... → MORPHEUS: Factorized LA

R, NumPy, Oracle R Enterprise, TensorFlow, Apache SystemML, ...

Towards Linear Algebra over Normalized Data. **VLDB 2017**

Bulk LA-based ML Algorithms

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

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

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

Input: Regular matrix T, Y, w, α
for i in $1 : max\_iter$ **do**
$\quad w = w + \alpha * (T^{\intercal}(Y/(1 + \exp(Tw))))$
end

Towards Linear Algebra over Normalized Data. **VLDB 2017**

MORPHEUS: High-level Architecture

Towards Linear Algebra over Normalized Data. **VLDB 2017**

MORPHEUS: High-level Architecture

MORPHEUS

Rewrite Rules for
Factorized LA ops
on an LA tool

R Oracle R Enterprise NumPy TensorFlow

Towards Linear Algebra over Normalized Data. **VLDB 2017**

MORPHEUS: High-level Architecture

ML algorithm expressed using LA ops on an LA tool

→

MORPHEUS

Rewrite Rules for Factorized LA ops on an LA tool

Oracle R Enterprise

NumPy

TensorFlow

Towards Linear Algebra over Normalized Data. **VLDB 2017**

MORPHEUS: High-level Architecture

Normalized schema and metadata

ML algorithm expressed using LA ops on an LA tool

MORPHEUS

Rewrite Rules for Factorized LA ops on an LA tool

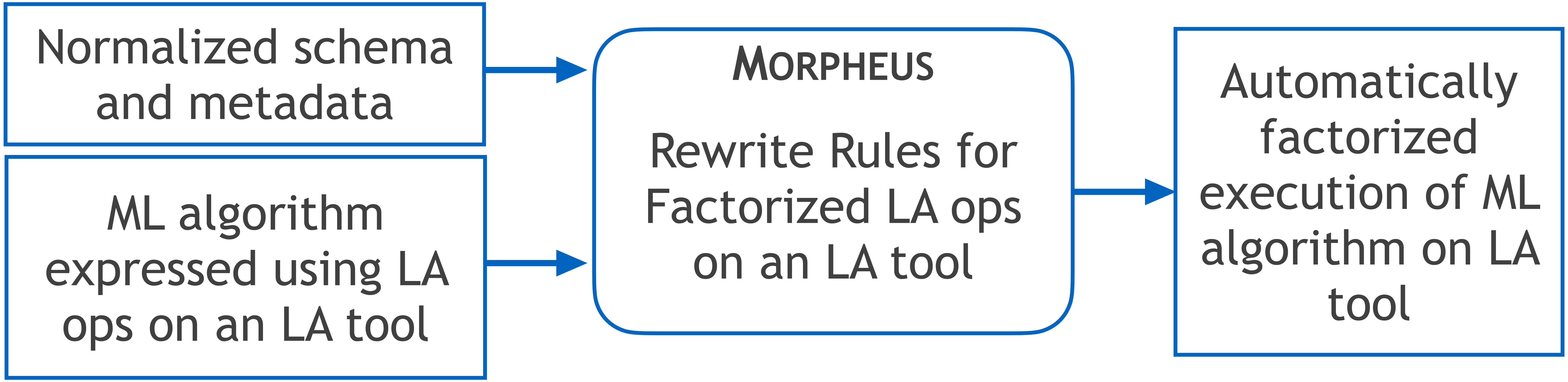
Oracle R Enterprise

NumPy

TensorFlow

Towards Linear Algebra over Normalized Data. **VLDB 2017**

MORPHEUS: High-level Architecture



Oracle R Enterprise NumPy TensorFlow

Towards Linear Algebra over Normalized Data. **VLDB 2017**

MORPHEUS: High-level Architecture

Normalized schema and metadata

ML algorithm expressed using LA ops on an LA tool

MORPHEUS

Rewrite Rules for Factorized LA ops on an LA tool

Automatically factorized execution of ML algorithm on LA tool

Oracle R Enterprise

NumPy

TensorFlow

Towards Linear Algebra over Normalized Data. **VLDB 2017**

MORPHEUS: Factorized LA Rewrite Rules

Towards Linear Algebra over Normalized Data. **VLDB 2017**

MORPHEUS: Factorized LA Rewrite Rules

New Abstraction: "Normalized Matrix" to represent join in LA

Towards Linear Algebra over Normalized Data. **VLDB 2017**

MORPHEUS: Factorized LA Rewrite Rules

New Abstraction: "Normalized Matrix" to represent join in LA

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

Towards Linear Algebra over Normalized Data. **VLDB 2017**

MORPHEUS: Factorized LA Rewrite Rules

New Abstraction: "Normalized Matrix" to represent join in LA

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

Employers

Towards Linear Algebra over Normalized Data. **VLDB 2017**

MORPHEUS: Factorized LA Rewrite Rules

New Abstraction: "Normalized Matrix" to represent join in LA

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

Customers Employers

Towards Linear Algebra over Normalized Data. **VLDB 2017**

MORPHEUS: Factorized LA Rewrite Rules

New Abstraction: "Normalized Matrix" to represent join in LA

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

$X \equiv [X_S X_R]$ Customers Employers

Towards Linear Algebra over Normalized Data. **VLDB 2017**

MORPHEUS: Factorized LA Rewrite Rules

New Abstraction: "Normalized Matrix" to represent join in LA

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

$X \equiv [X_S X_R]$

Customers

$S_{n \times d_S}$

Employers

Towards Linear Algebra over Normalized Data. **VLDB 2017**

MORPHEUS: Factorized LA Rewrite Rules

New Abstraction: "Normalized Matrix" to represent join in LA

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

$X \equiv [X_S X_R]$

Customers $S_{n \times d_S}$

Employers $R_{n_R \times d_R}$

Towards Linear Algebra over Normalized Data. **VLDB 2017**

MORPHEUS: Factorized LA Rewrite Rules

New Abstraction: "Normalized Matrix" to represent join in LA

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

$X \equiv [X_S X_R]$

Customers

Employers

$T_{n \times d}$

$S_{n \times d_S}$

$R_{n_R \times d_R}$

Towards Linear Algebra over Normalized Data. **VLDB 2017**

MORPHEUS: Factorized LA Rewrite Rules

New Abstraction: "Normalized Matrix" to represent join in LA

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

$X \equiv [X_S X_R]$ Customers Employers

$T_{n \times d}$ $S_{n \times d_S}$ $R_{n_R \times d_R}$

Towards Linear Algebra over Normalized Data. **VLDB 2017**

MORPHEUS: Factorized LA Rewrite Rules

New Abstraction: "Normalized Matrix" to represent join in LA

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

$X \equiv [X_S X_R]$

Customers Employers

$T_{n \times d}$ $S_{n \times d_S}$ $K_{n \times n_R}$ $R_{n_R \times d_R}$

Towards Linear Algebra over Normalized Data. **VLDB 2017**

MORPHEUS: Factorized LA Rewrite Rules

New Abstraction: "Normalized Matrix" to represent join in LA

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

$X \equiv [X_S X_R]$

Customers | Employers

$T_{n \times d}$ $S_{n \times d_S}$ $K_{n \times n_R}$ $R_{n_R \times d_R}$

$$K[i,j] = \begin{matrix} 1, \; if \;\; \mathbf{S}[i].FK = j \\ 0, \; o/w \end{matrix}$$

Towards Linear Algebra over Normalized Data. **VLDB 2017**

MORPHEUS: Factorized LA Rewrite Rules

New Abstraction: "Normalized Matrix" to represent join in LA

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

$$X \equiv [X_S X_R]$$

Customers Employers

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

$$T = [S \quad KR] \qquad K[i,j] = \begin{matrix} 1, \ if \ \mathbf{S}[i].FK = j \\ 0, \ o/w \end{matrix}$$

Towards Linear Algebra over Normalized Data. **VLDB 2017**

MORPHEUS: Factorized LA Rewrite Rules

New Abstraction: "Normalized Matrix" to represent join in LA

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

$X \equiv [X_S X_R]$ Customers Employers

$T_{n \times d}$ $S_{n \times d_S}$ $K_{n \times n_R}$ $R_{n_R \times d_R}$

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

Framework of algebraic rewrite rules for many LA operations

Towards Linear Algebra over Normalized Data. **VLDB 2017**

MORPHEUS: Factorized LA Rewrite Rules

New Abstraction: "Normalized Matrix" to represent join in LA

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

$X \equiv [X_S X_R]$

Customers Employers

$T_{n \times d}$ $S_{n \times d_S}$ $K_{n \times n_R}$ $R_{n_R \times d_R}$

$$T = [S \quad KR]$$

$$K[i,j] = \begin{cases} 1, & if\ \mathbf{S}[i].FK = j \\ 0, & o/w \end{cases}$$

Framework of <u>algebraic rewrite rules</u> for many LA operations

Left Matrix Multiplication:

Towards Linear Algebra over Normalized Data. **VLDB 2017**

MORPHEUS: Factorized LA Rewrite Rules

New Abstraction: "Normalized Matrix" to represent join in LA

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

$X \equiv [X_S X_R]$

Customers — Employers

$T_{n\times d}$ $S_{n\times d_S}$ $K_{n\times n_R}$ $R_{n_R\times d_R}$

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

Framework of <u>algebraic rewrite rules</u> for many LA operations

Left Matrix Multiplication: $Tw \rightarrow Sw_S + K(Rw_R)$

Towards Linear Algebra over Normalized Data. **VLDB 2017**

MORPHEUS: Factorized LA Rewrite Rules

New Abstraction: "Normalized Matrix" to represent join in LA

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

$X \equiv [X_S X_R]$

Customers | Employers

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

$$T = [S \quad KR] \qquad K[i,j] = \begin{matrix} 1, \; if \; \mathbf{S}[i].FK = j \\ 0, \; o/w \end{matrix}$$

Framework of algebraic rewrite rules for many LA operations

Left Matrix Multiplication: $Tw \rightarrow Sw_S + K(\boxed{Rw_R})$

Towards Linear Algebra over Normalized Data. **VLDB 2017**

MORPHEUS: Factorized LA Rewrite Rules

New Abstraction: "Normalized Matrix" to represent join in LA

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

$X \equiv [X_S X_R]$ Customers Employers

$T_{n \times d}$ $S_{n \times d_S}$ $K_{n \times n_R}$ $R_{n_R \times d_R}$

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

Framework of <u>algebraic rewrite rules</u> for many LA operations

Left Matrix Multiplication: $Tw \rightarrow Sw_S + K(Rw_R)$

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

Towards Linear Algebra over Normalized Data. **VLDB 2017**

Automatically Factorized ML in MORPHEUS

Towards Linear Algebra over Normalized Data. **VLDB 2017**

Automatically Factorized ML in MORPHEUS

Input: Regular matrix T, Y, w, α
for i in $1 : max\_iter$ **do**
$\quad w = w + \alpha * (T^{\intercal}(Y/(1 + \exp(Tw))))$
end

Towards Linear Algebra over Normalized Data. **VLDB 2017**

Automatically Factorized ML in MORPHEUS

Input: Regular matrix T, Y, w, α
for i *in* $1 : max\_iter$ **do**
| $w = w + \alpha * (T^{\intercal}(Y/(1 + \exp(Tw))))$
end

$T \equiv (S, K, R)$ ↓ MORPHEUS

Towards Linear Algebra over Normalized Data. **VLDB 2017**

Automatically Factorized ML in MORPHEUS

Input: Regular matrix T, Y, w, α
for i *in* $1 : max\_iter$ **do**
$\quad w = w + \alpha * (T^{\intercal}(Y/(1 + \exp(Tw))))$
end

$T \equiv (S, K, R)$ ↓ MORPHEUS

Input: Normalized matrix (S, K, R), Y, w, α
for i *in* $1 : max\_iter$ **do**
$\quad P = (Y/(1 + \exp(Sw[1 : d_S,]+ K(Rw[d_S + 1 : d_S + d_R,]))))^{\intercal}$
$\quad w = w + \alpha * [PS, (PK)R]^{\intercal}$
end

Towards Linear Algebra over Normalized Data. **VLDB 2017**

LA Operations Factorized in MORPHEUS

Table 1: Operators and functions of linear algebra handled in this paper over a normalized matrix T.

Op Type	Name	Expression	Output Type	Parameter X or x	Factorizable
Element-wise Scalar Op	Arithmetic Op ($\oslash = +, -, *, /, \hat{}$, etc)	$T \oslash x$ or $x \oslash T$	Normalized Matrix	A scalar	Yes
	Transpose	$T^{\intercal}$		N/A	
	Scalar Function f (e.g., log, exp, sin)	$f(T)$		Parameters for f	
Aggregation	Row Summation	rowSums(T)	Column Vector	N/A	
	Column Summation	colSums(T)	Row Vector		
	Summation	sum(T)	Scalar		
Multiplication	Left Multiplication	TX	Regular Matrix	$(d_S + d_R) \times d_X$ matrix	
	Right Multiplication	XT		$n_X \times n_S$ matrix	
	Cross-product	crossprod(T)		N/A	
Inversion	Pseudoinverse	ginv(T)			
Element-wise Matrix Op	Arithmetic Op ($\oslash = +, -, *, /, \hat{}$, etc)	$X \oslash T$ or $T \oslash X$		$n_S \times (d_S + d_R)$ matrix	No

Towards Linear Algebra over Normalized Data. **VLDB 2017**

Snapshot of Empirical Results

Towards Linear Algebra over Normalized Data. **VLDB 2017**

Snapshot of Empirical Results

Prototype in R (and Python) for listed LA ops; ~800 LOC; commodity machine

Towards Linear Algebra over Normalized Data. **VLDB 2017**

Snapshot of Empirical Results

Prototype in R (and Python) for listed LA ops; ~800 LOC; commodity machine

S: Ratings
R_1: Users
R_2: Businesses

Towards Linear Algebra over Normalized Data. **VLDB 2017**

Snapshot of Empirical Results

Prototype in R (and Python) for listed LA ops; ~800 LOC; commodity machine

yelp

S: Ratings
R_1: Users
R_2: Businesses

Expedia

S: Listings
R_1: Hotels
R_2: Search details

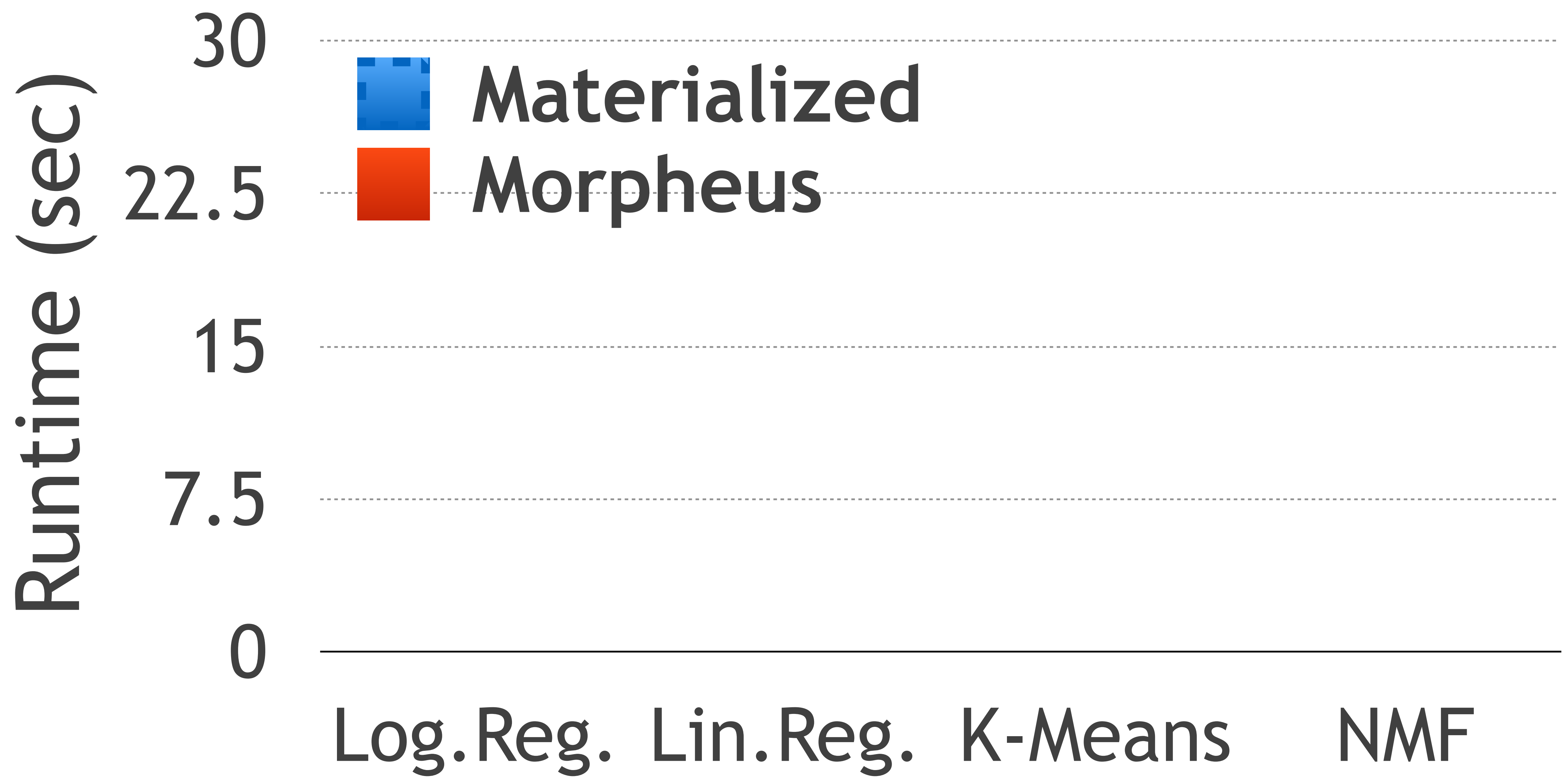
Towards Linear Algebra over Normalized Data. **VLDB 2017**

Snapshot of Empirical Results

Prototype in R (and Python) for listed LA ops; ~800 LOC; commodity machine

S: Ratings
R_1: Users
R_2: Businesses

S: Listings
R_1: Hotels
R_2: Search details



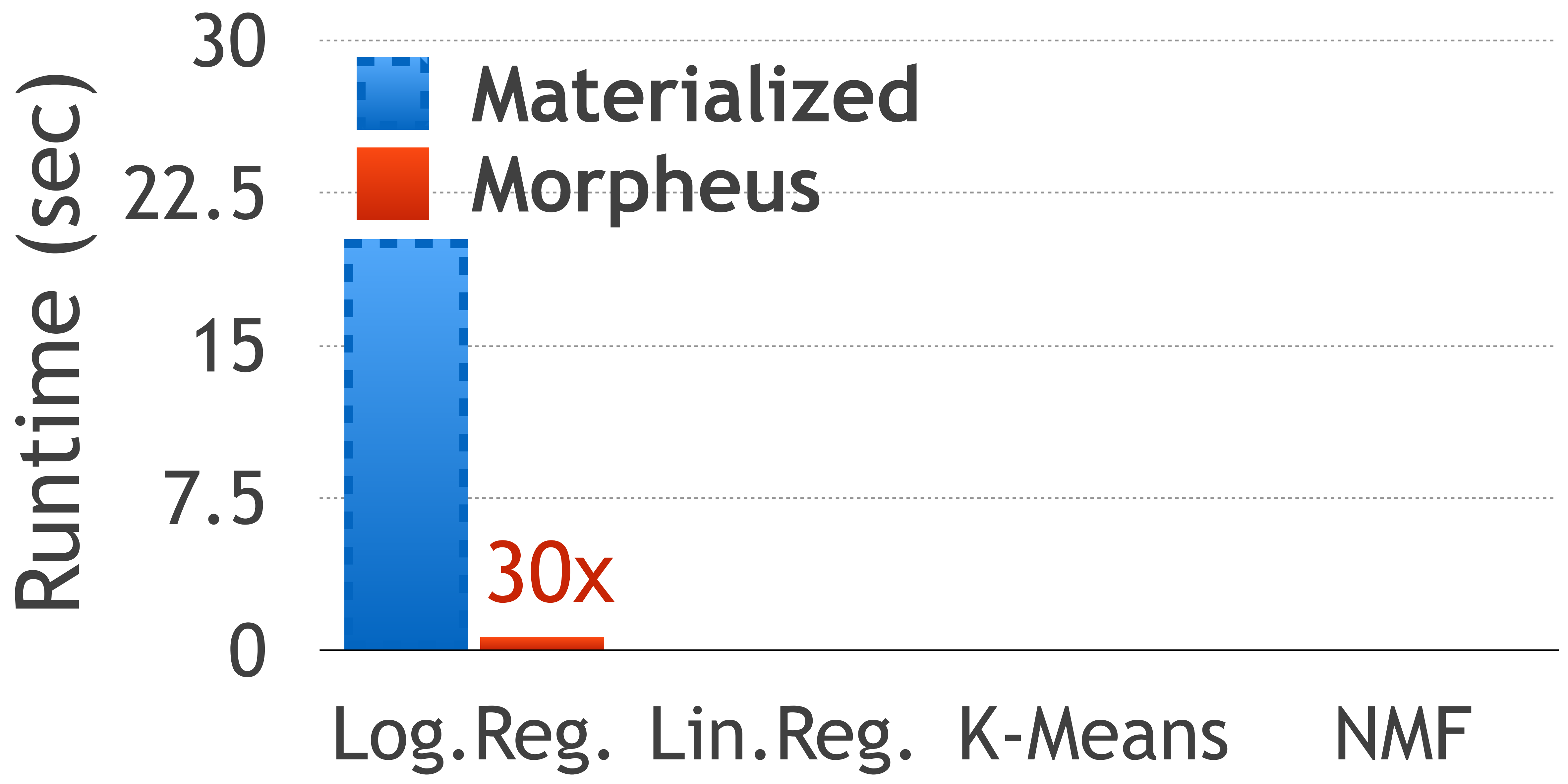
Towards Linear Algebra over Normalized Data. **VLDB 2017**

Snapshot of Empirical Results

Prototype in R (and Python) for listed LA ops; ~800 LOC; commodity machine

S: Ratings
R_1: Users
R_2: Businesses

S: Listings
R_1: Hotels
R_2: Search details



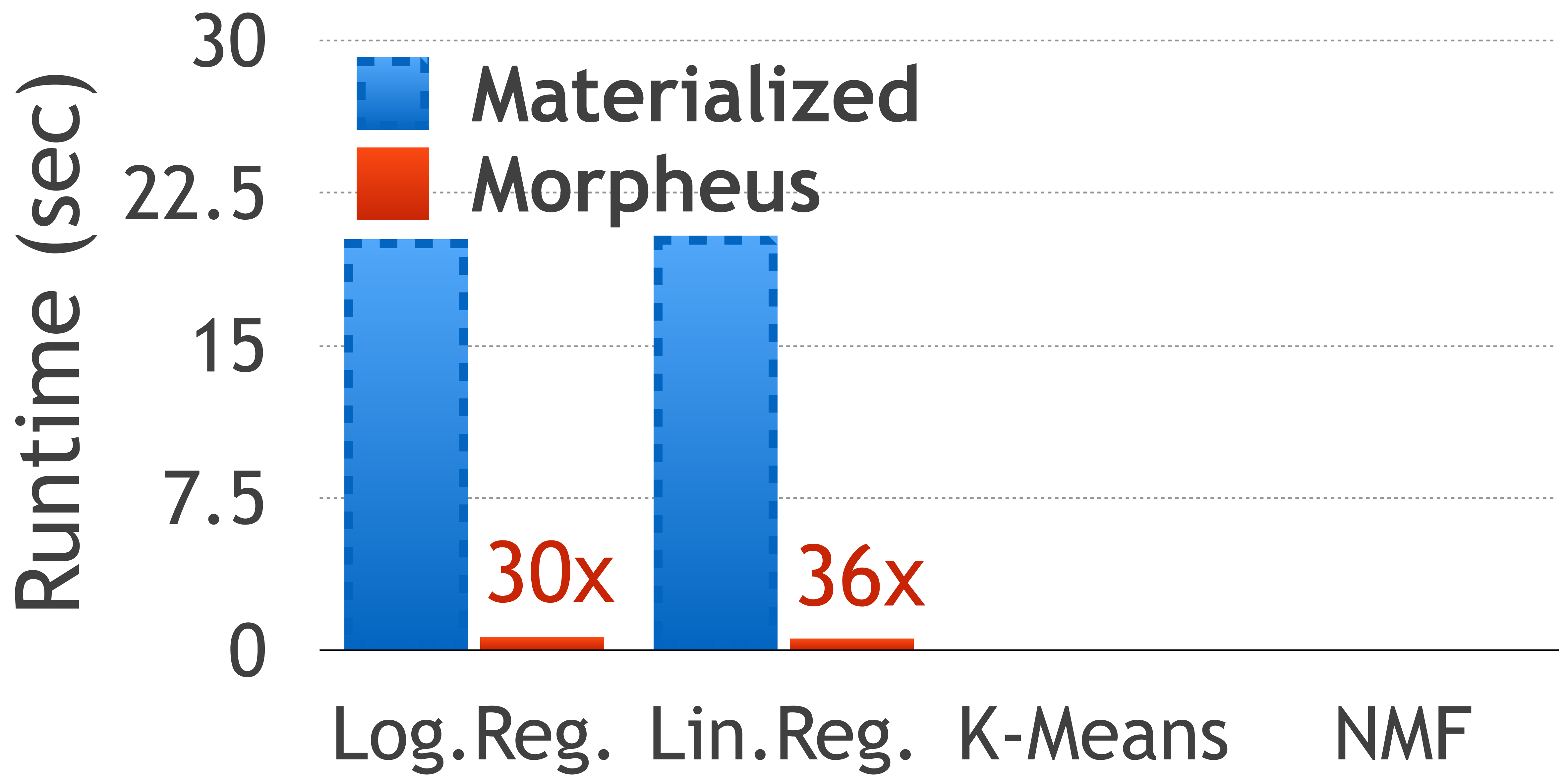
Towards Linear Algebra over Normalized Data. **VLDB 2017**

Snapshot of Empirical Results

Prototype in R (and Python) for listed LA ops; ~800 LOC; commodity machine

yelp — S: Ratings, R_1: Users, R_2: Businesses

Expedia — S: Listings, R_1: Hotels, R_2: Search details



Towards Linear Algebra over Normalized Data. **VLDB 2017**

Snapshot of Empirical Results

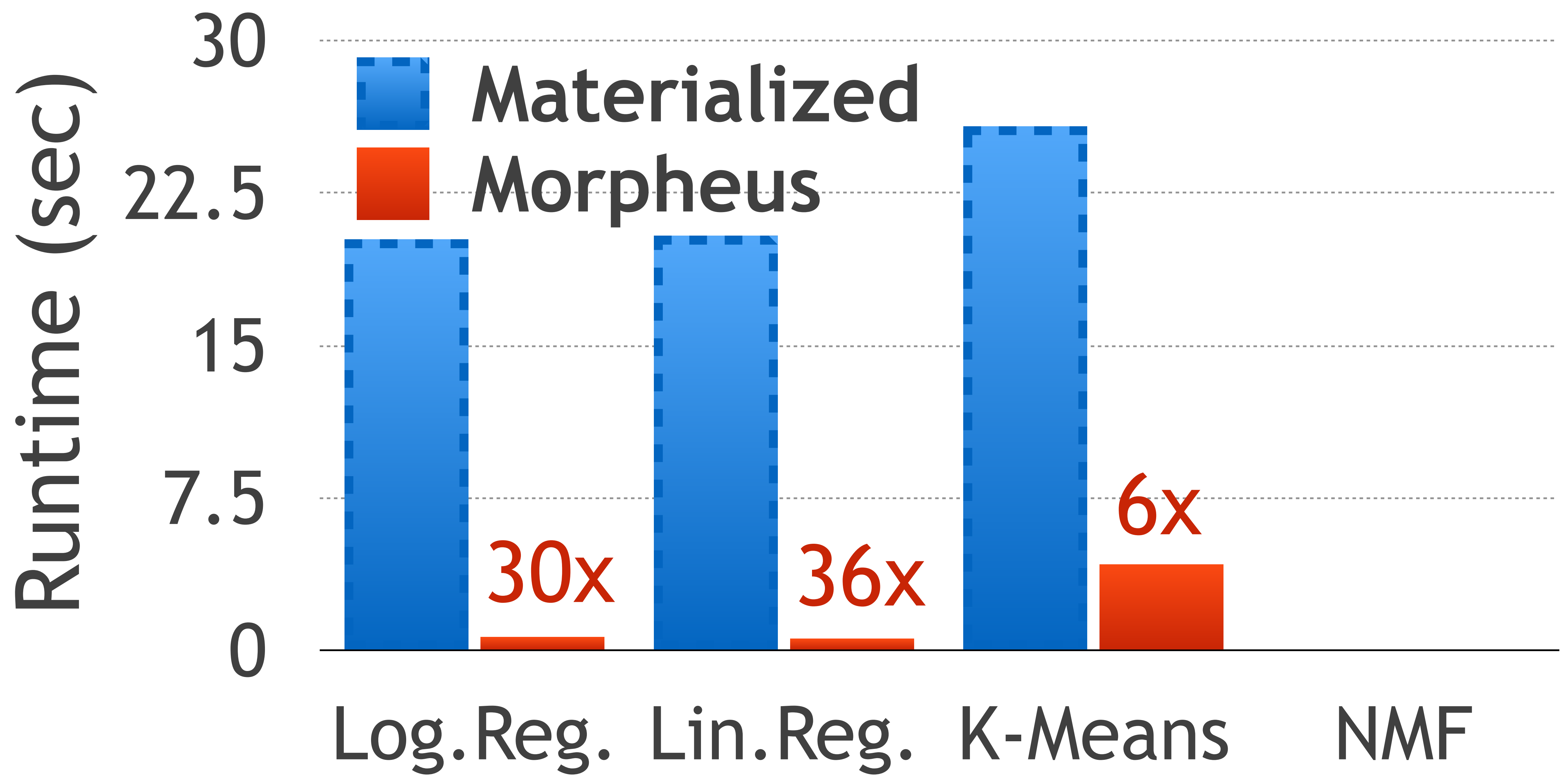
Prototype in R (and Python) for listed LA ops; ~800 LOC; commodity machine

yelp

S: Ratings
R_1: Users
R_2: Businesses

Expedia

S: Listings
R_1: Hotels
R_2: Search details



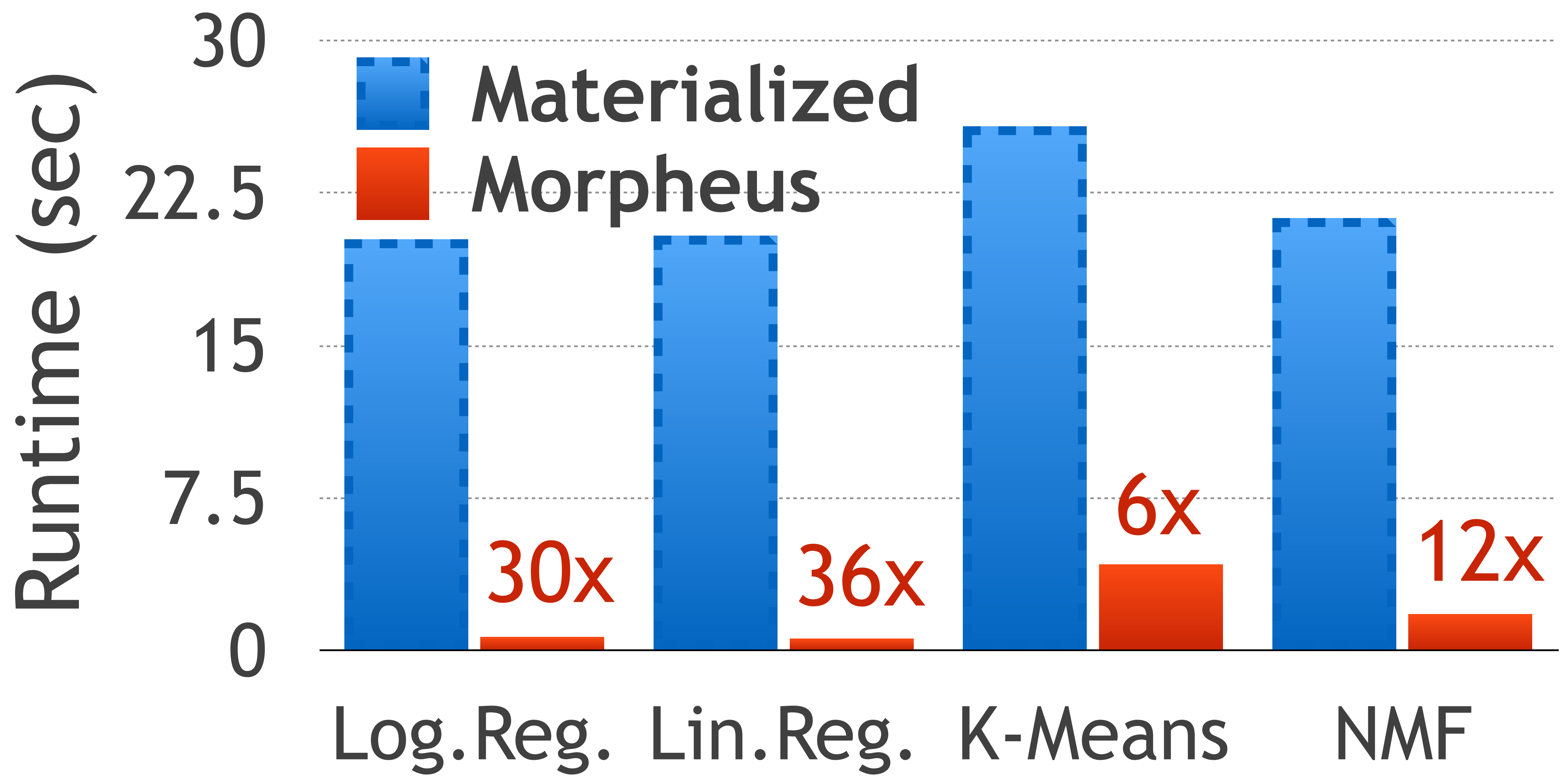
Towards Linear Algebra over Normalized Data. **VLDB 2017**

Snapshot of Empirical Results

Prototype in R (and Python) for listed LA ops; ~800 LOC; commodity machine

yelp
S: Ratings
R_1: Users
R_2: Businesses

Expedia
S: Listings
R_1: Hotels
R_2: Search details



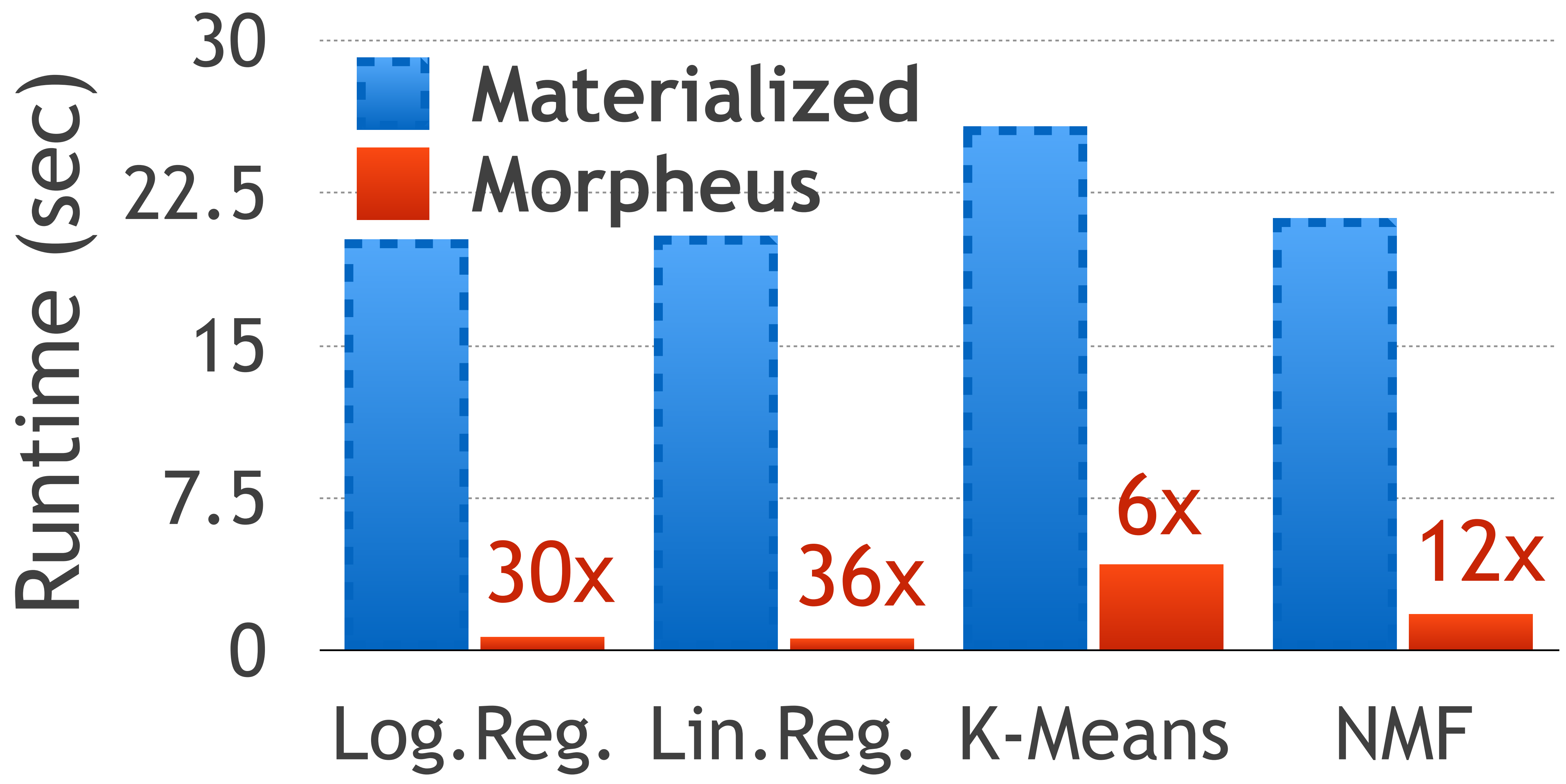
Towards Linear Algebra over Normalized Data. **VLDB 2017**

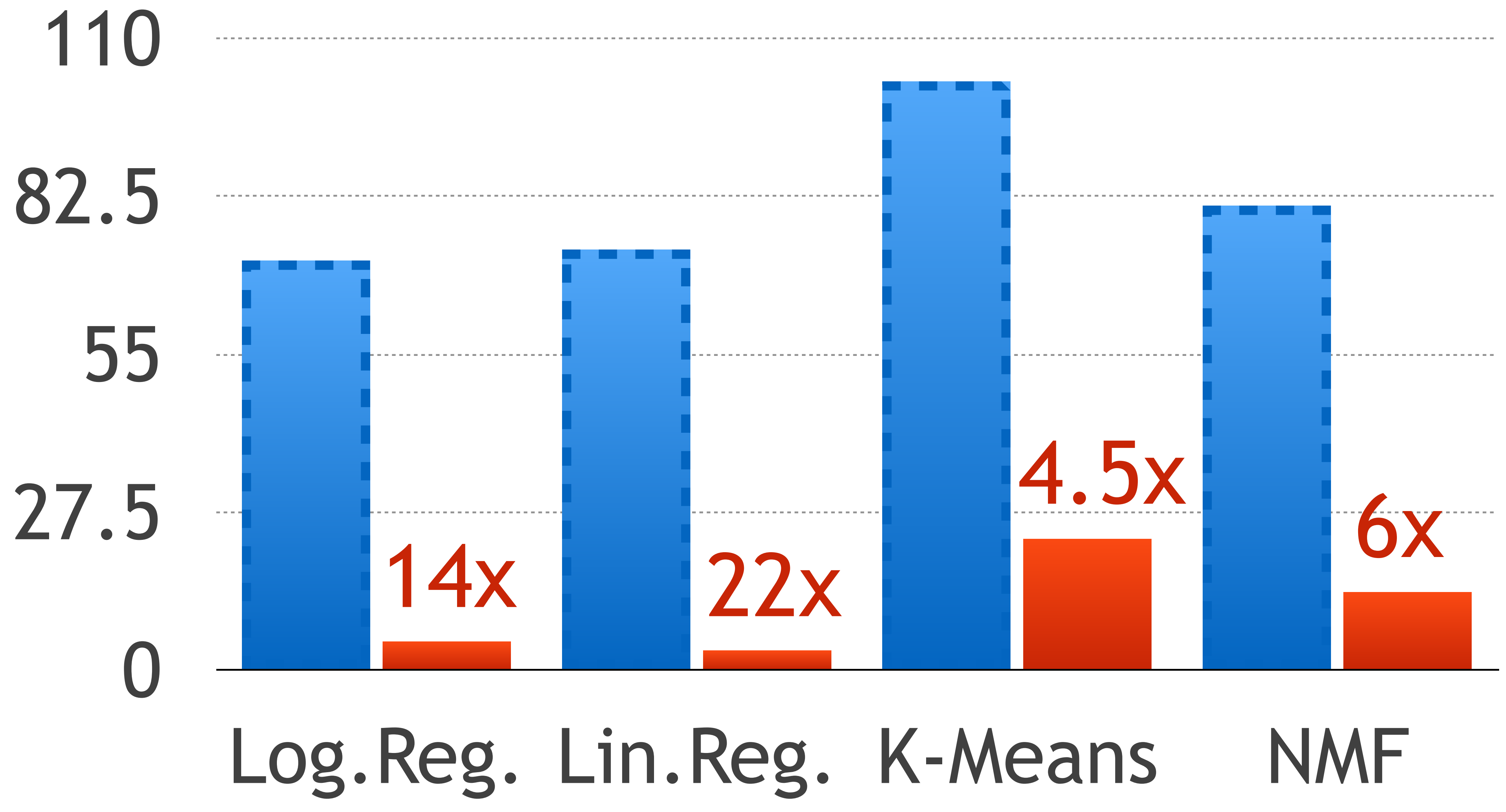
Snapshot of Empirical Results

Prototype in R (and Python) for listed LA ops; ~800 LOC; commodity machine

S: Ratings
R_1: Users
R_2: Businesses

S: Listings
R_1: Hotels
R_2: Search details





Towards Linear Algebra over Normalized Data. **VLDB 2017**

When is MORPHEUS not likely to be beneficial?

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

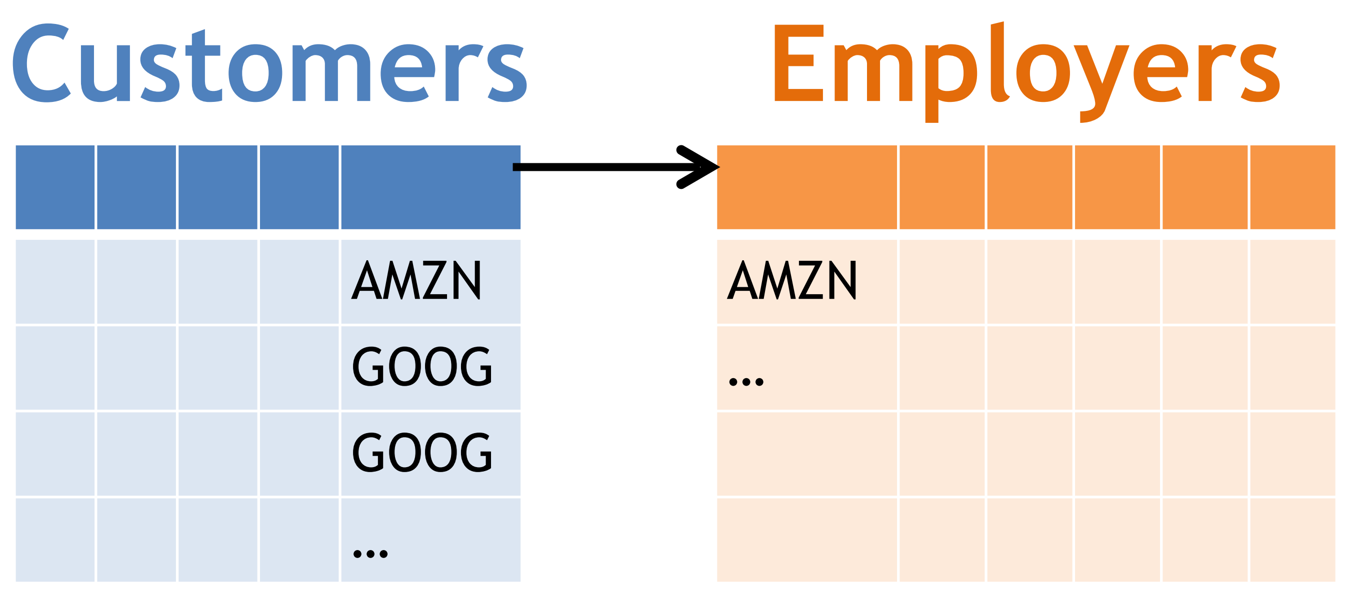
Towards Linear Algebra over Normalized Data. **VLDB 2017**

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)

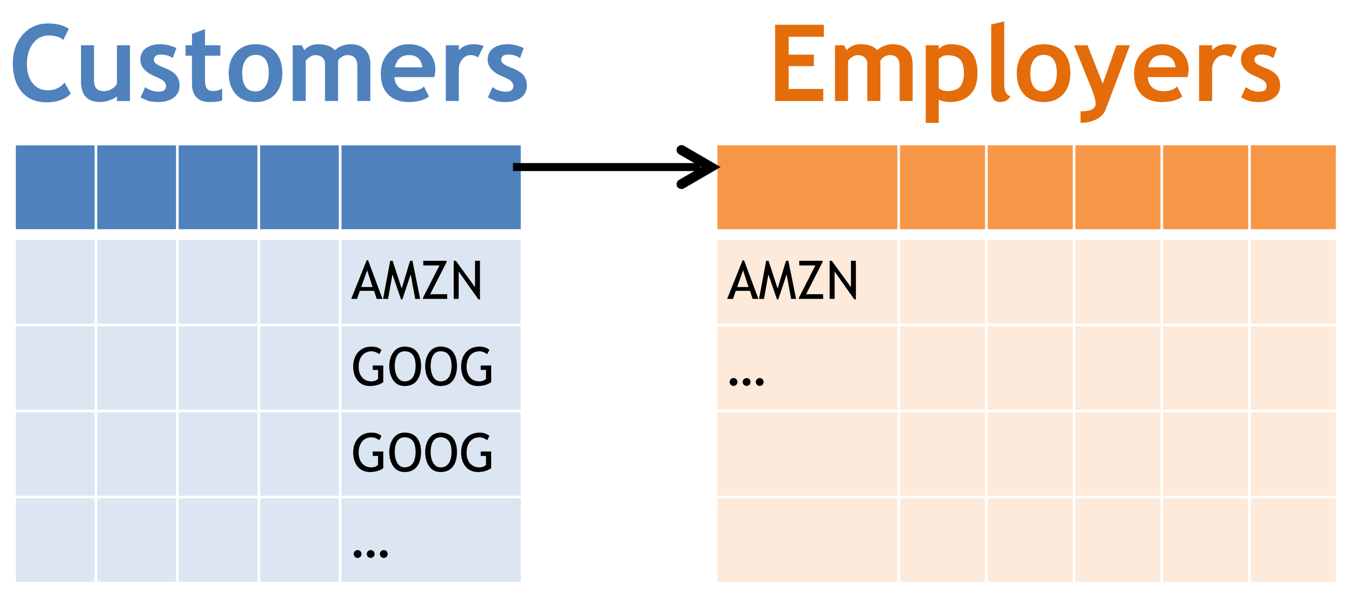


Towards Linear Algebra over Normalized Data. **VLDB 2017**

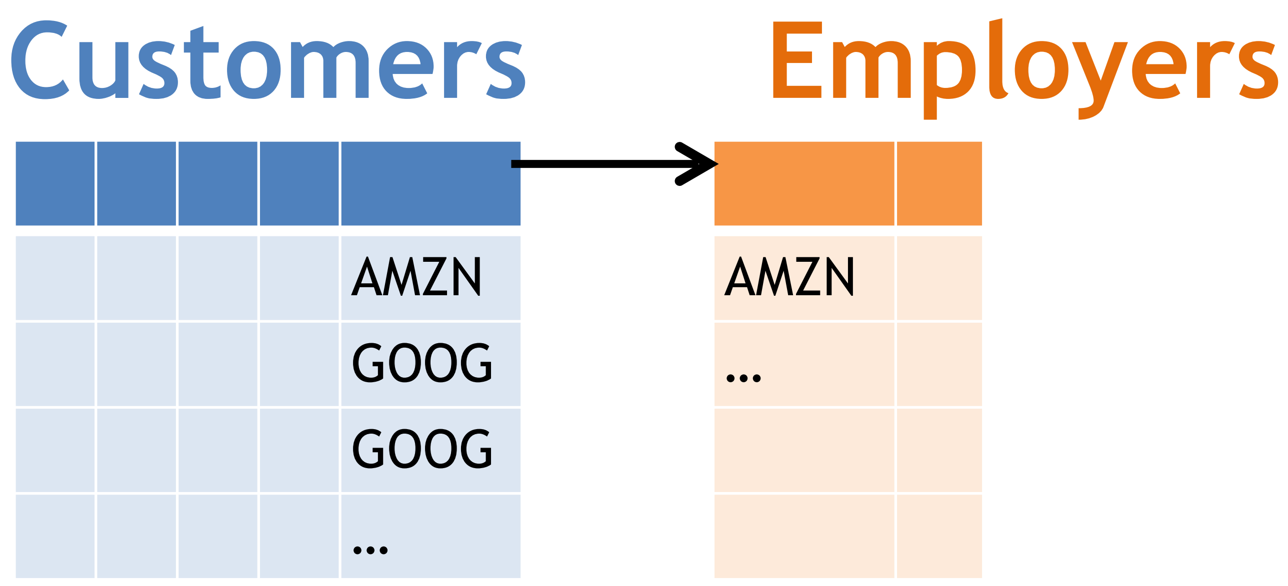
When is MORPHEUS not likely to be beneficial?

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

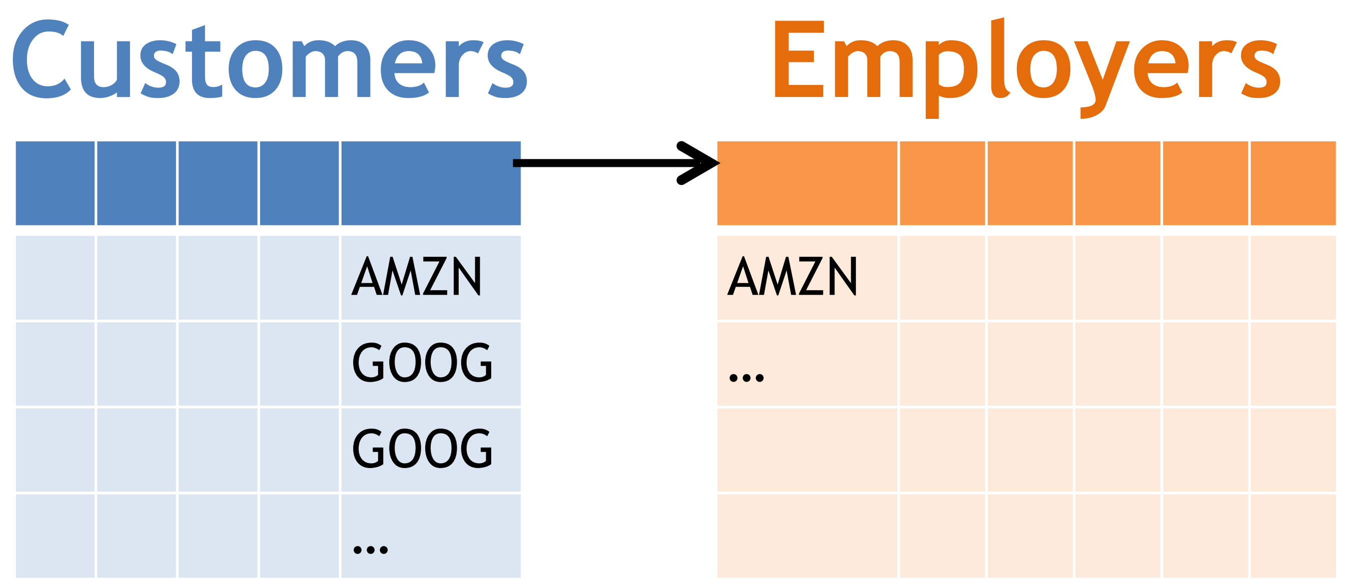


Towards Linear Algebra over Normalized Data. **VLDB 2017**

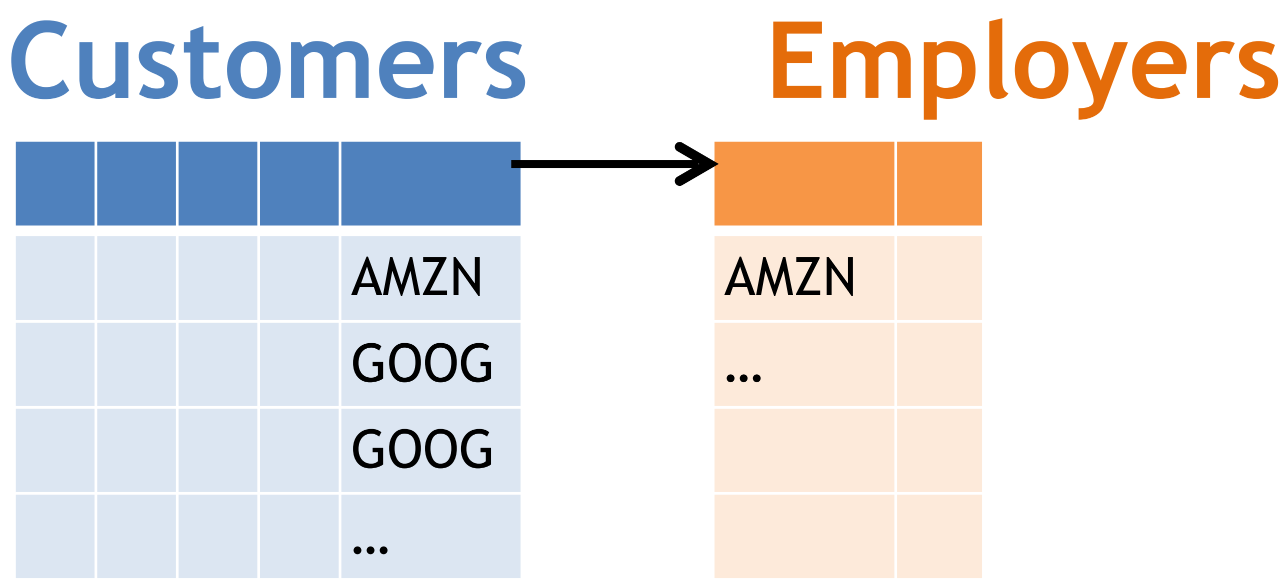
When is MORPHEUS not likely to be beneficial?

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**

MORPHEUS: Implementations and Extensions

Towards Linear Algebra over Normalized Data. **VLDB 2017**
Enabling and Optimizing Non-linear Feature Interactions in Factorized Linear Algebra. **SIGMOD 2019**
Tuple-oriented Compression for Large-scale Mini-batch Stochastic Gradient Descent. **SIGMOD 2019**

MORPHEUS: Implementations and Extensions

Library released for both R and Python NumPy

Towards Linear Algebra over Normalized Data. **VLDB 2017**
Enabling and Optimizing Non-linear Feature Interactions in Factorized Linear Algebra. **SIGMOD 2019**
Tuple-oriented Compression for Large-scale Mini-batch Stochastic Gradient Descent. **SIGMOD 2019**

MORPHEUS: Implementations and Extensions

Library released for both R and Python NumPy

Supports star schemas for many LA ops; snowflakes can be reduced to star

Towards Linear Algebra over Normalized Data. **VLDB 2017**
Enabling and Optimizing Non-linear Feature Interactions in Factorized Linear Algebra. **SIGMOD 2019**
Tuple-oriented Compression for Large-scale Mini-batch Stochastic Gradient Descent. **SIGMOD 2019**

MORPHEUS: Implementations and Extensions

Library released for both R and Python NumPy

Supports star schemas for many LA ops; snowflakes can be reduced to star

Some data cleaning/prep ops also factorized

Towards Linear Algebra over Normalized Data. **VLDB 2017**
Enabling and Optimizing Non-linear Feature Interactions in Factorized Linear Algebra. **SIGMOD 2019**
Tuple-oriented Compression for Large-scale Mini-batch Stochastic Gradient Descent. **SIGMOD 2019**

MORPHEUS: Implementations and Extensions

Library released for both R and Python NumPy

Supports star schemas for many LA ops; snowflakes can be reduced to star

Some data cleaning/prep ops also factorized

Avito Google ORACLE

Towards Linear Algebra over Normalized Data. **VLDB 2017**
Enabling and Optimizing Non-linear Feature Interactions in Factorized Linear Algebra. **SIGMOD 2019**
Tuple-oriented Compression for Large-scale Mini-batch Stochastic Gradient Descent. **SIGMOD 2019**

MORPHEUS: Implementations and Extensions

Library released for both R and Python NumPy

Supports star schemas for many LA ops; snowflakes can be reduced to star

Some data cleaning/prep ops also factorized

Avito Google ORACLE

MORPHEUSFI: Second-order feature interactions in Morpheus

Towards Linear Algebra over Normalized Data. **VLDB 2017**
Enabling and Optimizing Non-linear Feature Interactions in Factorized Linear Algebra. **SIGMOD 2019**
Tuple-oriented Compression for Large-scale Mini-batch Stochastic Gradient Descent. **SIGMOD 2019**

MORPHEUS: Implementations and Extensions

Library released for both R and Python NumPy

Supports star schemas for many LA ops; snowflakes can be reduced to star

Some data cleaning/prep ops also factorized

MORPHEUSFI: Second-order feature interactions in Morpheus

MORPHEUSFLOW: "Lazy join" for SGD in TensorFlow

Towards Linear Algebra over Normalized Data. **VLDB 2017**

Enabling and Optimizing Non-linear Feature Interactions in Factorized Linear Algebra. **SIGMOD 2019**

Tuple-oriented Compression for Large-scale Mini-batch Stochastic Gradient Descent. **SIGMOD 2019**

MORPHEUS: Implementations and Extensions

Library released for both R and Python NumPy

Supports star schemas for many LA ops; snowflakes can be reduced to star

Some data cleaning/prep ops also factorized

Avito Google ORACLE

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**
Enabling and Optimizing Non-linear Feature Interactions in Factorized Linear Algebra. **SIGMOD 2019**
Tuple-oriented Compression for Large-scale Mini-batch Stochastic Gradient Descent. **SIGMOD 2019**

TRINITY: MORPHEUS Meets Oracle GraalVM

Towards A Polyglot Framework for Factorized ML. **VLDB 2021**

TRINITY: MORPHEUS Meets Oracle GraalVM

Goal: *Automate* Morpheus itself to many PLs in a unified way

Towards A Polyglot Framework for Factorized ML. **VLDB 2021**

TRINITY: MORPHEUS Meets Oracle GraalVM

Goal: *Automate* Morpheus itself to many PLs in a unified way

Idea: Exploit GraalVM, an industrial-strength *polyglot* compiler + runtime for data science workloads (R, Py, Javascript, etc.)

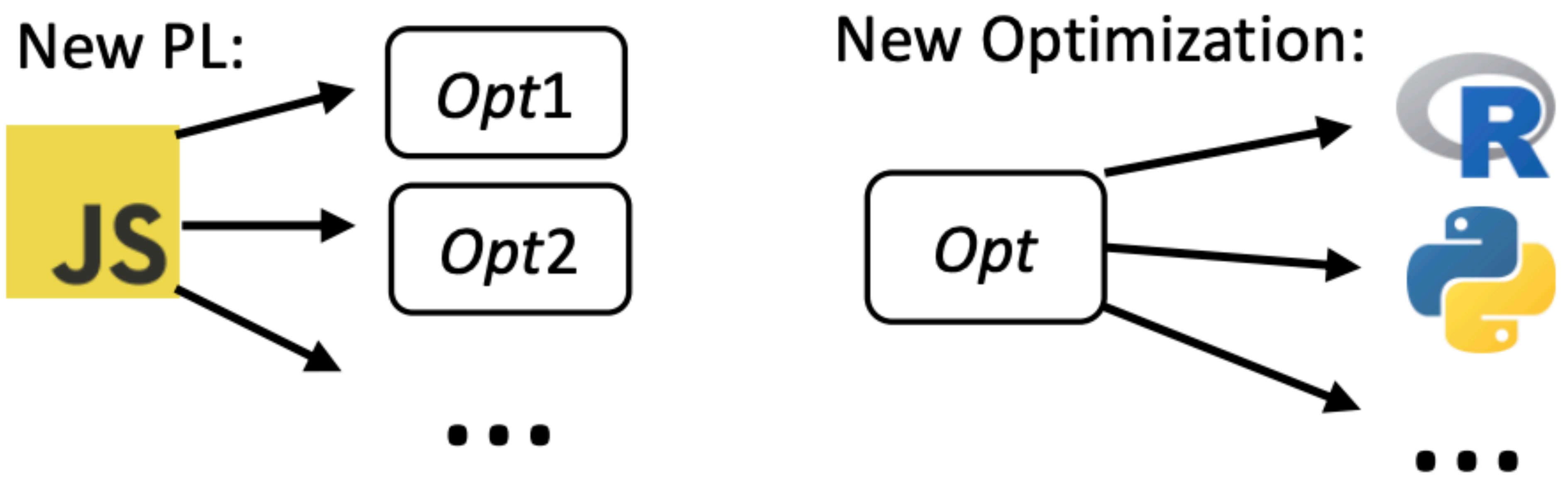
Towards A Polyglot Framework for Factorized ML. **VLDB 2021**

TRINITY: MORPHEUS Meets Oracle GraalVM

Goal: *Automate* Morpheus itself to many PLs in a unified way

Idea: Exploit GraalVM, an industrial-strength *polyglot* compiler + runtime for data science workloads (R, Py, Javascript, etc.)

World without Trinity



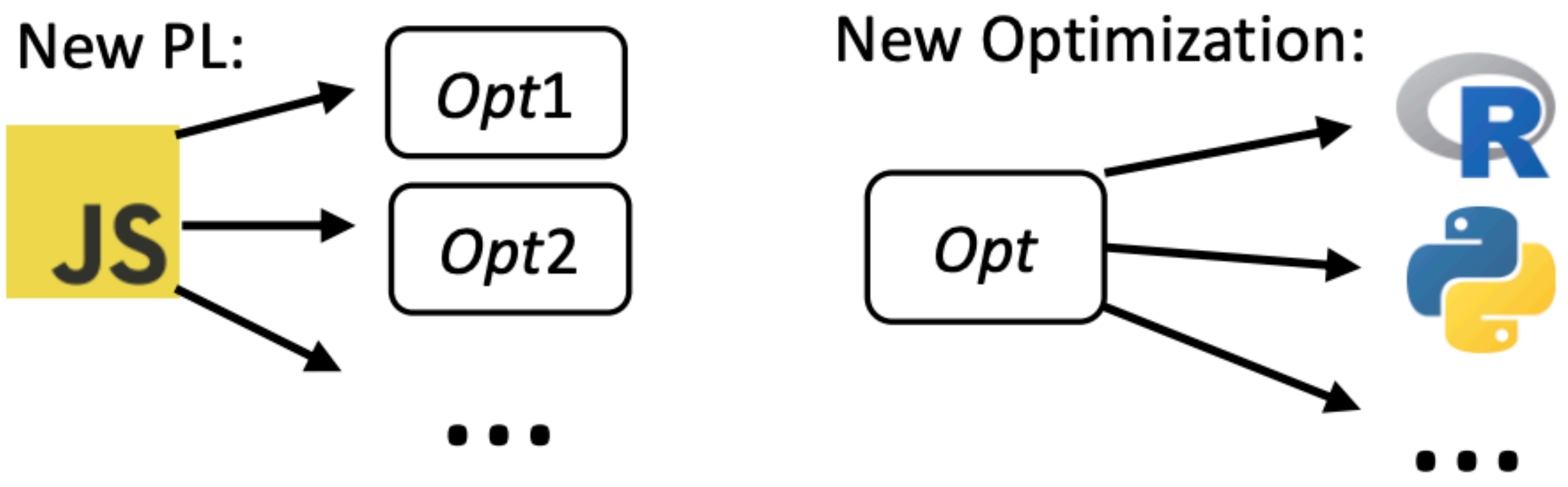
Towards A Polyglot Framework for Factorized ML. **VLDB 2021**

TRINITY: MORPHEUS Meets Oracle GraalVM

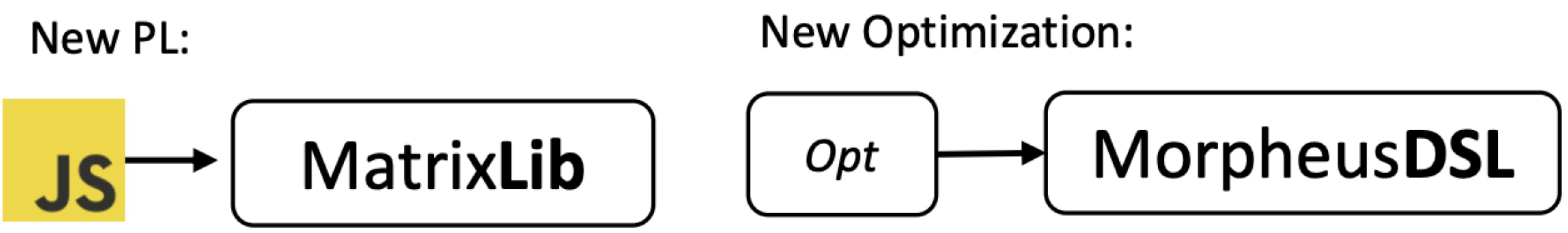
Goal: *Automate* Morpheus itself to many PLs in a unified way

Idea: Exploit GraalVM, an industrial-strength *polyglot* compiler + runtime for data science workloads (R, Py, Javascript, etc.)

World without Trinity



World with Trinity



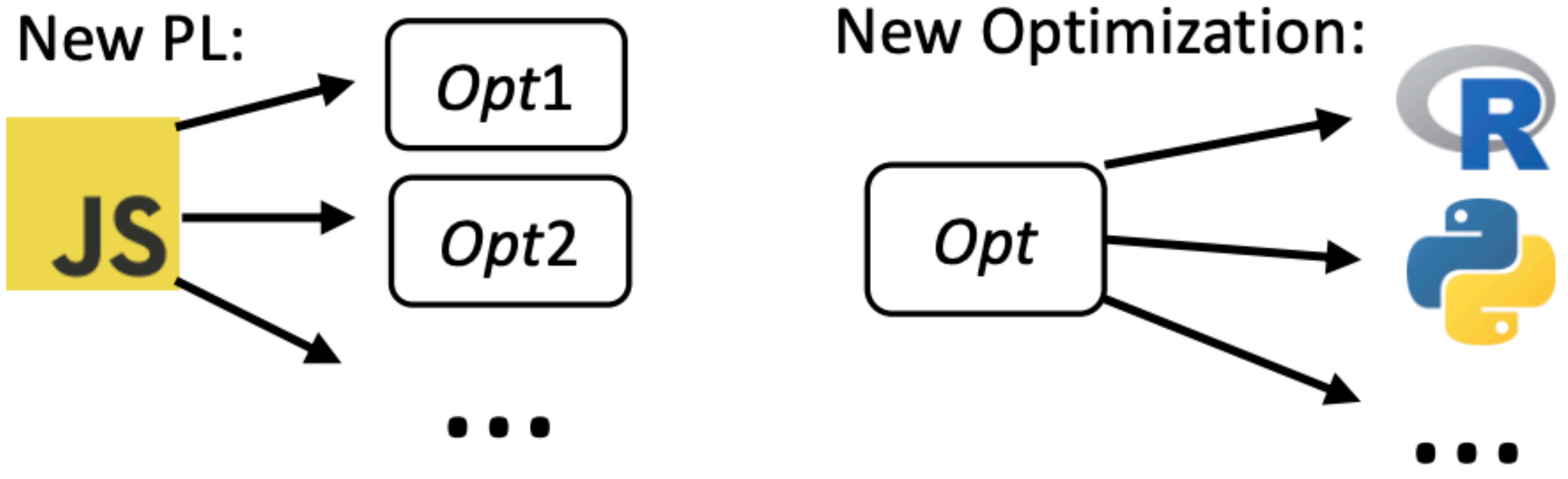
Towards A Polyglot Framework for Factorized ML. **VLDB 2021**

TRINITY: MORPHEUS Meets Oracle GraalVM

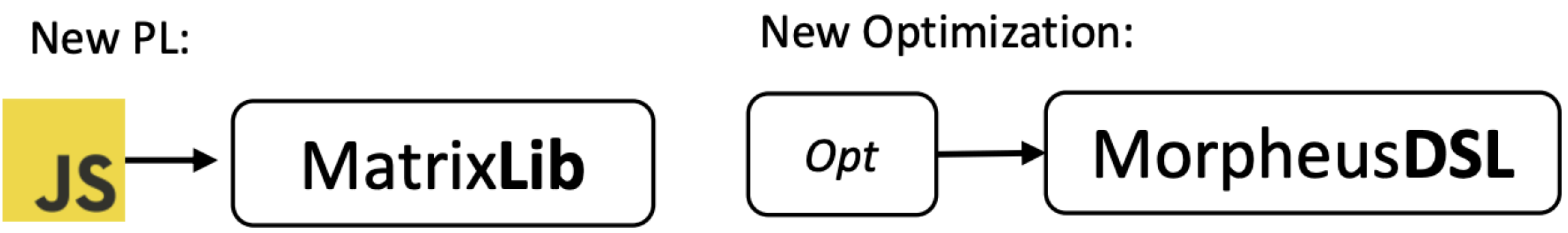
Goal: *Automate* Morpheus itself to many PLs in a unified way

Idea: Exploit GraalVM, an industrial-strength *polyglot* compiler + runtime for data science workloads (R, Py, Javascript, etc.)

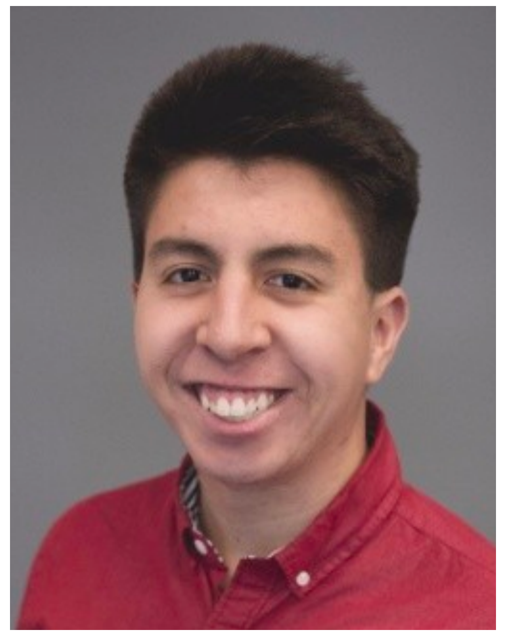
World without Trinity



World with Trinity



Learn more about Trinity from David Justo at 3:30pm today!



Towards A Polyglot Framework for Factorized ML. **VLDB 2021**

Outline

4m Introducing ML over Joins

4m Orion: Factorized ML

10m Morpheus and Extensions

4m Roadblocks and Musings

Roadblocks for Factorized ML

Roadblocks for Factorized ML

Observation: Factorized ML yet to have big practical impact on any path. :-/

Roadblocks for Factorized ML

Observation: Factorized ML yet to have big practical impact on any path. :-/

Reason 1: Applicability to *business-critical* ML algorithms limited

Tree ensembles rule tabular data; factorized ML gains marginal there

GLMs, clustering, etc. often not big bottleneck in real-world pipelines

Roadblocks for Factorized ML

Observation: Factorized ML yet to have big practical impact on any path. :-/

Reason 1: Applicability to *business-critical* ML algorithms limited

Tree ensembles rule tabular data; factorized ML gains marginal there

GLMs, clustering, etc. often not big bottleneck in real-world pipelines

Roadblocks for Factorized ML

Observation: Factorized ML yet to have big practical impact on any path. :-/

Reason 1: Applicability to *business-critical* ML algorithms limited

Tree ensembles rule tabular data; factorized ML gains marginal there

GLMs, clustering, etc. often not big bottleneck in real-world pipelines

Reason 2: *Implementation* effort to make it practical still non-trivial

Orion-style: UDFs too complex to implement/maintain on RDBMS/Spark

Morpheus-style: ML not always written as LA scripts; hidden C++ callouts

Trinity-style: Likely promising; over the wall at Oracle now! :)

Plug: First Textbook on ML Systems

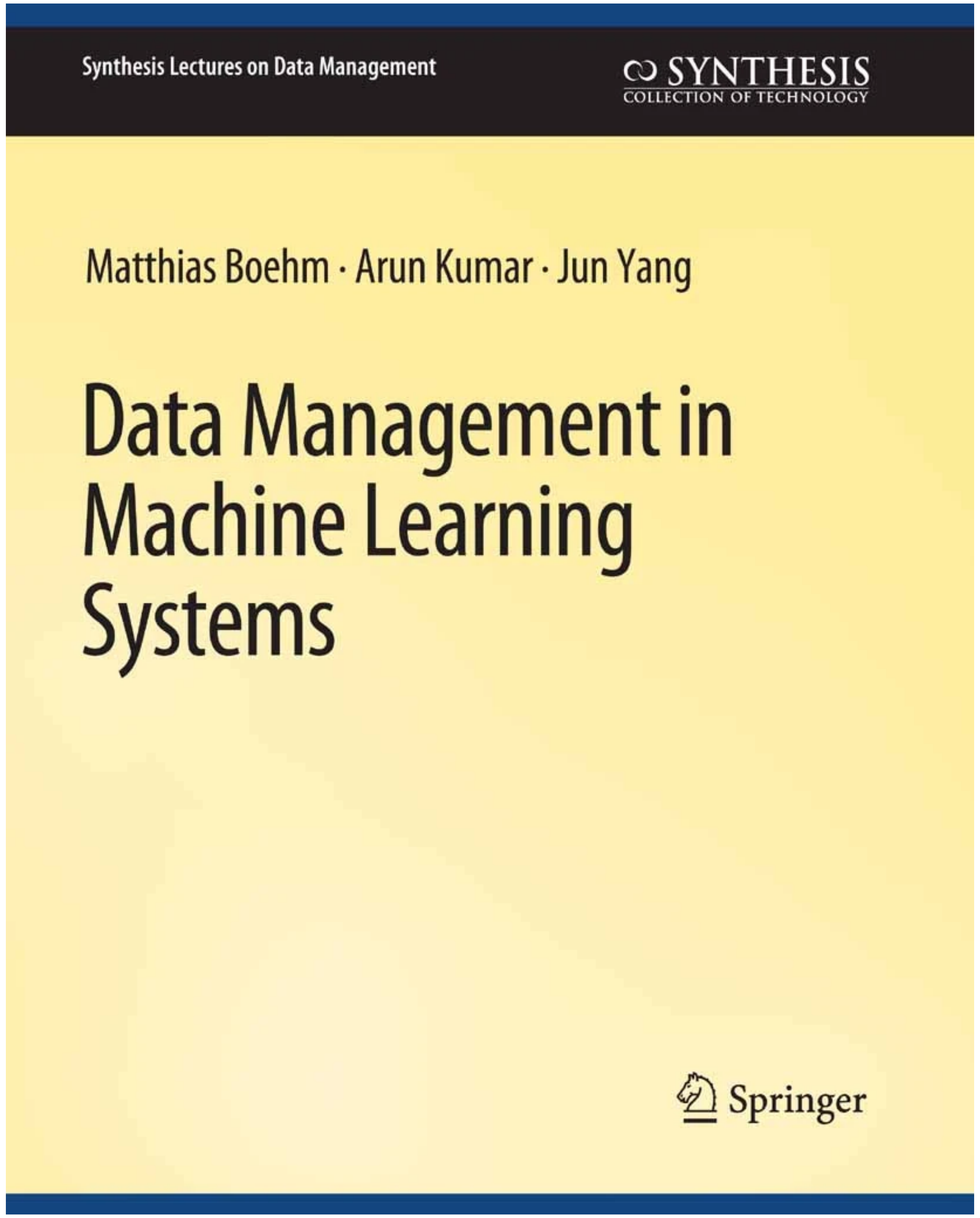


Table of Contents

https://tinyurl.com/MLSystemsBook

https://adalabucsd.github.io

arunkk@eng.ucsd.edu

github.com/ADALabUCSD

@TweetAtAKK

ACKS: NIH NIDDK NSF Hellman Fellows Fund Google ORACLE vmware