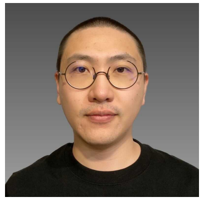
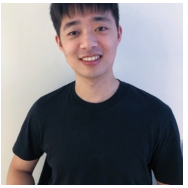
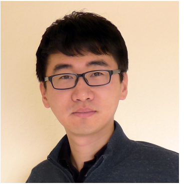
VLDB 2021 Best EA&B Paper Award

Are We Ready For Learned Cardinality Estimation?


Xiaoying Wang


Changbo Qu


Weiyuan Wu

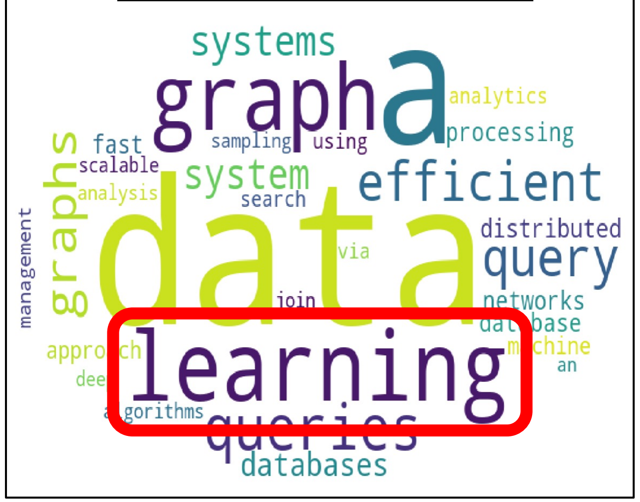

Jiannan Wang


Qingqing Zhou

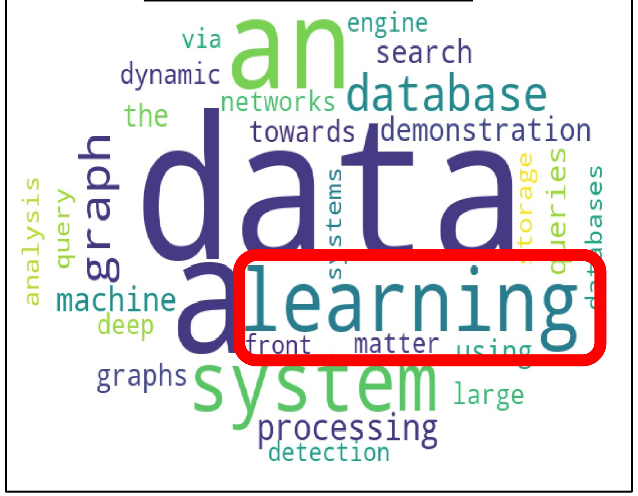
SFU SIMON FRASER UNIVERSITY

ML is dominating system research

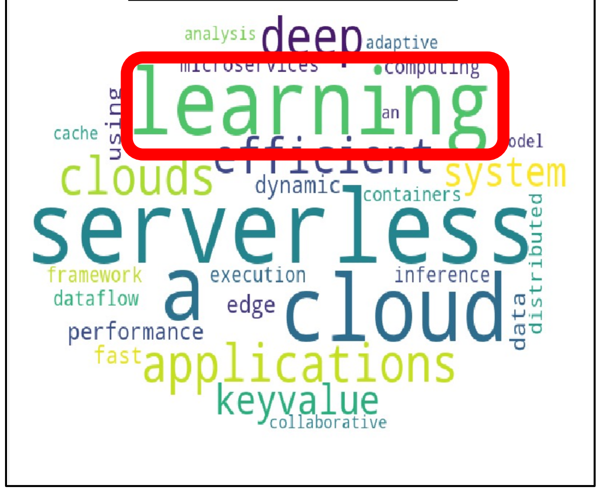
SIGMOD 2021



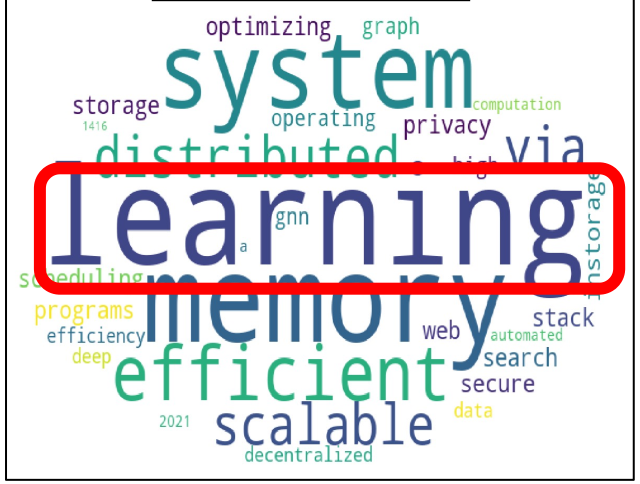
VLDB 2021



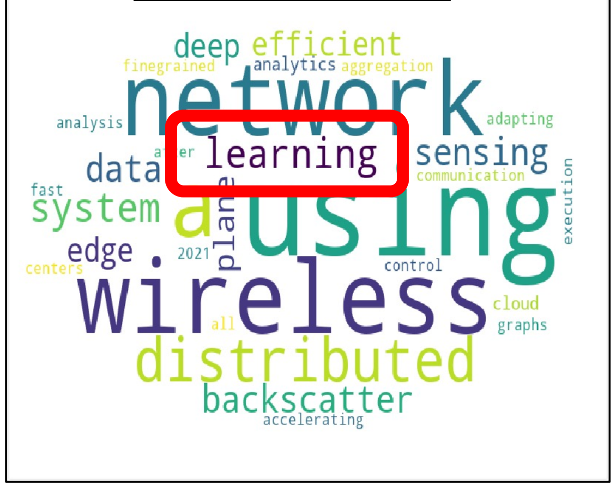
SOCC 2021



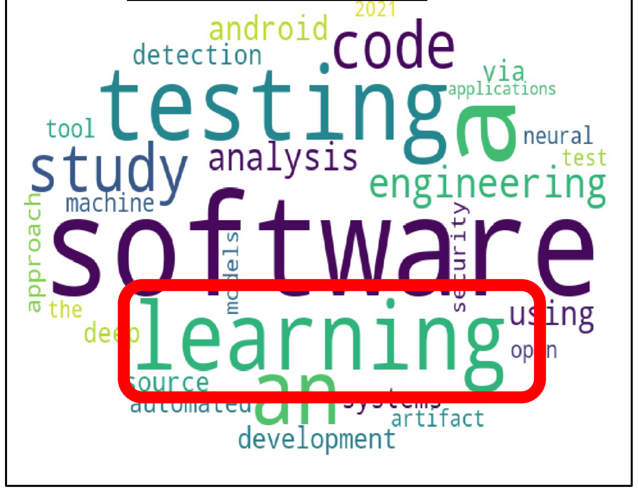
OSDI 2021



NSDI 2021



ICSE 2021



The impact of ML on Data Intensive Systems

Very Hot Topic!

Tutorial @ SIGMOD 2019

From Auto-tuning One Size Fits All to Self-designed and Learned Data-intensive Systems

Stratos Idreos
Harvard University

Tim Kraska
MIT

Tutorial @ VLDB 2021

Machine Learning for Cloud Data Systems: the Progress so far and the Path Forward

Alekh Jindal
Microsoft
alekh.jindal@microsoft.com

Matteo Interlandi
Microsoft
mainterl@microsoft.com

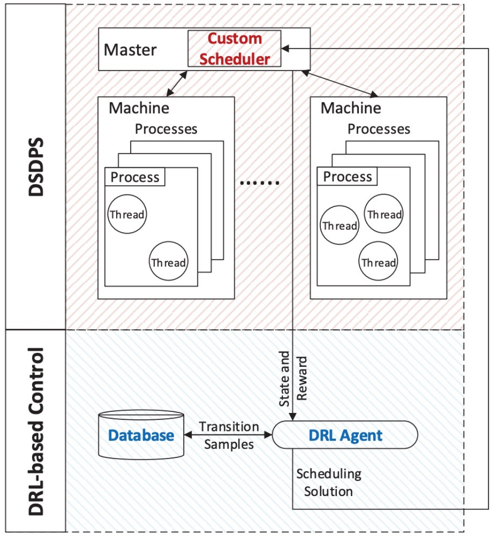
Tutorial @ VLDB 2021

Machine Learning for Databases

Guoliang Li, Xuanhe Zhou
Tsinghua University, Beijing, China
liguoliang@tsinghua.edu.cn,zhouxuan19@mails.tsinghua.edu.cn

Lei Cao
MIT, Cambridge, MA USA
lcao@csail.mit.edu

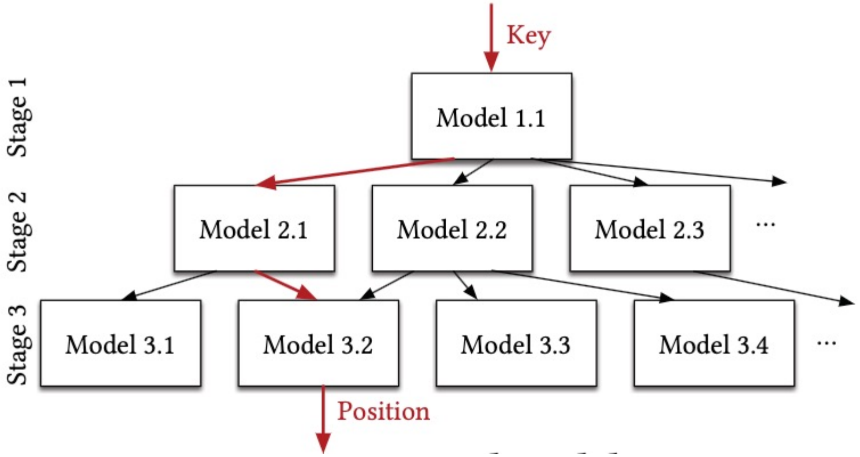
The Power of ML



Scheduler

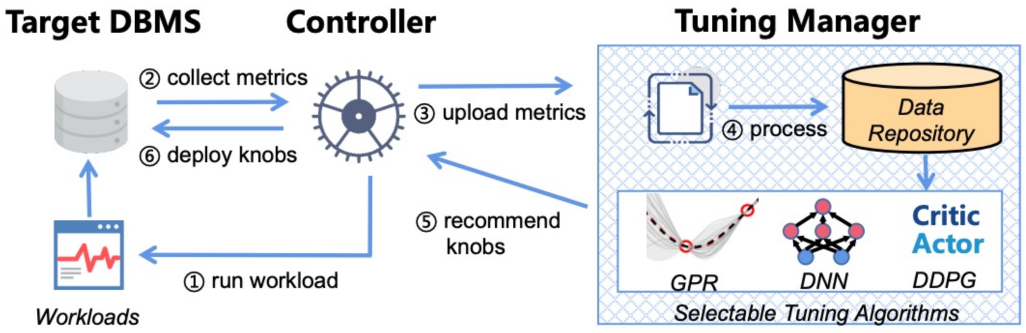
[Li, T et all. VLDB 18]

Index [Kraska, T et all. SIGMOD 18]

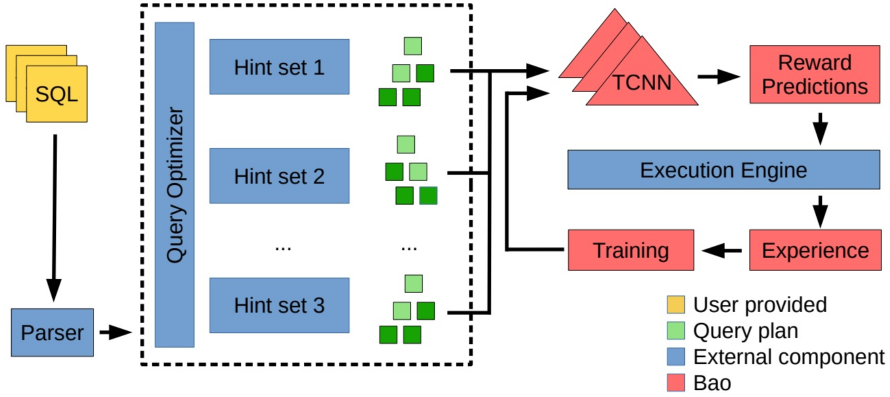


Knob Tuning

[Aken, D et all. VLDB 21]



Optimizer [Marcus, R et all. SIGMOD 21]



But what would happen in 5-10 years?

Two Possible Worlds

This topic is

Dead

OR

ML in Clickhouse
ML in Spark
ML in Snowflake
…

Are we ready to deploy **learned** X in production?

Cardinality Estimation
Index
Scheduler
…

Why Cardinality Estimation?

2014

IS QUERY OPTIMIZATION A "SOLVED" PROBLEM?

≡ Databases

Guy Lohman, IBM DB2 (40 years' experience)

"The root of all evil, the Achilles Heel of query optimization, is the estimation of the size of intermediate results, known as cardinalities."

2018 - 2021

Multiple research groups consistently reported that learned cardinality estimators show very **impressive** results

Massachusetts Institute of Technology

Berkeley UNIVERSITY OF CALIFORNIA

Technical University of Munich

Microsoft

Alibaba Group

What is Cardinality Estimation (CE)?

$\mathcal{Q}$: SELECT $*$
FROM $Student$
WHERE age > 15
AND $gender$ = 'Male';

$\text{Card}(\mathcal{Q}) = 4$

age	gender	GPA
21	Female	3.42
20	Male	2.58
18	Female	2.79
20	Female	3.98
24	Female	3.71
20	Male	3.50
21	Male	4.0
23	Female	3.66
22	Male	3.12

How Learned CE Methods work?

- Methodology 1: Query-driven
  - **Key Idea:** Model as a Regression problem

Query → Feature Vector → CE_result

- Methodology 2: Data-driven

Methodology 1: Query-Driven

Training

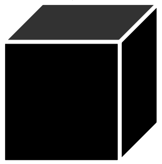
Query Pool | Labels

$Q1$: SELECT $*$FROM $Student$ WHERE age > 20; — 4
$Q2$: SELECT $*$FROM $Student$ WHERE GPA < 3.5 AND GPA > 3.0; — 2
$Q3$: SELECT $*$FROM $Student$ WHERE $gender$ = 'Female'; — 5
… — …

↓ **Featurize**

$Q1$: <0.8, 1.0, 0.0, 0.0, 0.0, 1.0> 4
$Q2$: <0.0, 1.0, 0.0, 1.0, 0.3, 0.6> 2
$Q3$: <0.0, 1.0, 1.0, 1.0, 0.0, 1.0> 5
…

↓ **Train**



Regression Model

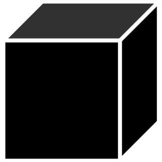
Inference

Q: SELECT $*$ FROM $Student$
WHERE age > 15 AND gender = "Male"

↓ **Featurize**

Q: <0.0, 0.9, 0.0, 1.0, 0.8, 1.0>

↓ **Inference**



Estimation: 4!

Methodology 1: Query-Driven

- **MSCN** [Kipf, A et all. CIDR 19]
  - Neural Network + Sampling
- **LW-XGB** [Dutt, A et all. VLDB 19]
  - Gradient Boosted Tree + Histogram
- **LW-NN** [Dutt, A et all. VLDB 19]
  - Neural Network + Histogram

How Learned CE Methods work?

- Methodology 1: Query-driven
  - **Key Idea:** Model as a Regression problem

Query → Feature Vector → CE_result

- Methodology 2: Data-driven
  - **Key Idea:** Model as a Joint Distribution Estimation problem

A_1	A_2	…	A_n

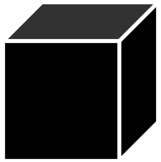
→ $P(A_{1,} A_{1,} \dots , A_n)$

Methodology 2: Data-Driven

Training

age	gender	GPA
21	Female	3.42
20	Male	2.58
18	Female	2.79
20	Female	3.98
24	Female	3.71
20	Male	3.50
21	Male	4.0
23	Female	3.66
22	Male	3.12

Train



P(age, gender, GPA)

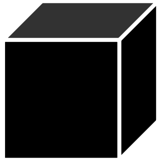
Joint Distribution Estimation Model

Inference

Q: SELECT * FROM $Student$
WHERE age > 15 AND gender = "Male"

$$P(age > 20, \text{gender} = \text{“Male”})$$

Inference



Estimation: 4!

Methodology 2: Data-Driven

- **Naru** [Yang, Z et all. VLDB 20]
  - Auto-regressive Model
- **DeepDB** [Hilprecht, B et all. VLDB 20]
  - Sum Product Network

Are we ready to deploy *learned cardinality estimation* in production?

Questions

- Are Learned Methods Ready for **Static** Environments?
- Are Learned Methods Ready for **Dynamic** Environments?
- When Do Learned Estimators Go **Wrong**?

Questions

- Are Learned Methods Ready for **Static** Environments?
- Are Learned Methods Ready for **Dynamic** Environments?
- When Do Learned Estimators Go **Wrong**?

Experiment Setup

- Evaluate Metric
  - $q\text{-}error = \frac{\max(est(q), act(q))}{\min(est(q), act(q))}$

Estimated CE: *est(q) = 1000*
Actual CE: *act(q) = 2000*

→ $q\text{-}error = \frac{\max(1000,2000)}{\min(1000,2000)} = 2$

4 real-world datasets

Dataset	Size(MB)	Rows	Cols/Cat	Domain
Census [16]	4.8	49K	13/8	10^{16}
Forest [16]	44.3	581K	10/0	10^{27}
Power [16]	110.8	2.1M	7/0	10^{17}
DMV [62]	972.8	11.6M	11/10	10^{15}

Comprehensive workload

	Predicate Number	Operator		Consider OOD
		Equal	Range	
MSCN	$0 \sim \|D\|$	✓	✓	×
LW-XGB/NN	$2 \sim \|D\|$	×	close range	✓
Naru	$5 \sim 11$	✓	open range	✓
DeepDB	$1 \sim 5$	✓	✓	×
DQM-D/Q	$1 \sim \|D\|$	✓	×	✓
Our Workload	$1 \sim \|D\|$	✓	✓	✓

Are Learned Methods Accurate?

Estimator	Census				Forest				Power				DMV			
	50th	95th	99th	Max	50th	95th	99th	Max	50th	95th	99th	Max	50th	95th	99th	Max

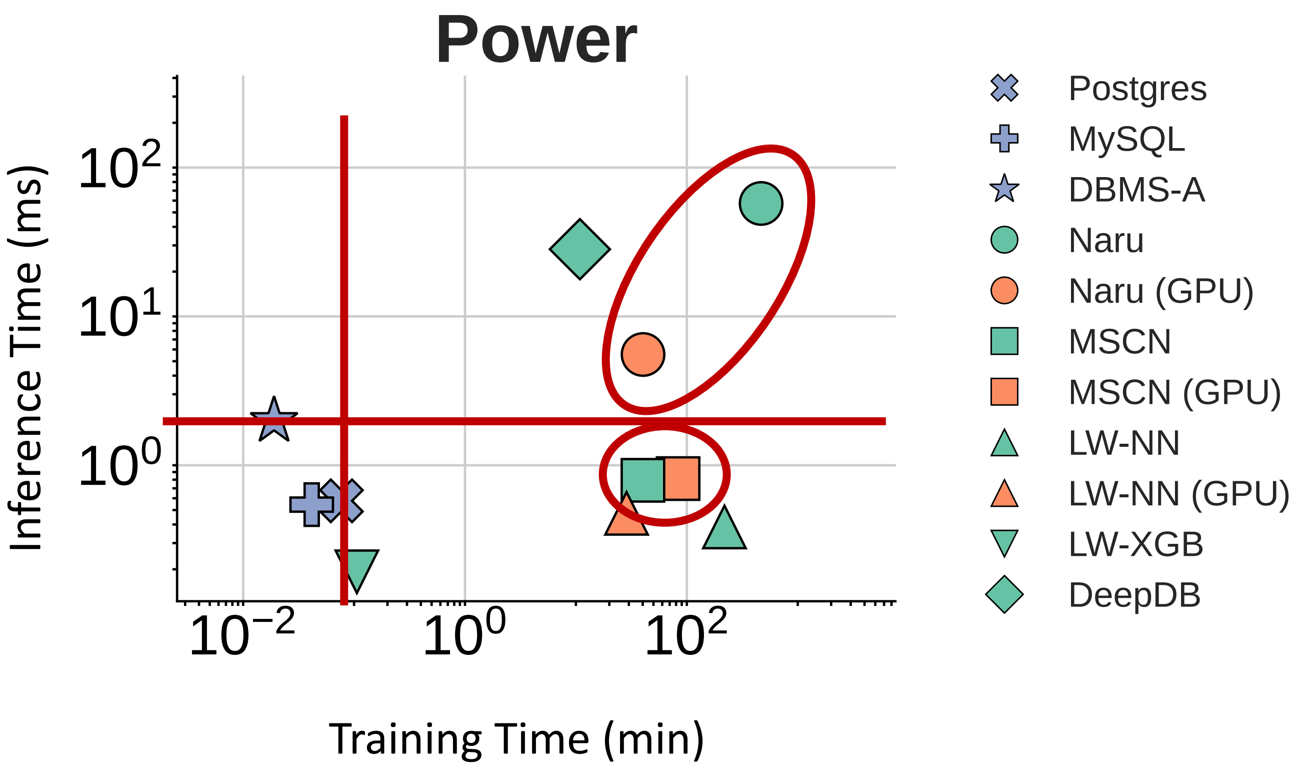
Are Learned Methods Accurate?

Estimator	Census				Forest				Power				DMV			
	50th	95th	99th	Max	50th	95th	99th	Max	50th	95th	99th	Max	50th	95th	99th	Max
Traditional Methods																
Postgres	1.40	18.6	58.0	1635	1.21	17.0	71.0	9374	1.06	15.0	235	$2 \cdot 10^5$	1.19	78.0	3255	$1 \cdot 10^5$
MySQL	1.40	19.2	63.0	1617	1.20	48.0	262	7786	1.09	26.0	2481	$2 \cdot 10^5$	1.40	1494	$3 \cdot 10^4$	$4 \cdot 10^5$
DBMS-A	4.16	122	307	2246	3.44	363	1179	$4 \cdot 10^4$	1.06	8.08	69.2	$2 \cdot 10^5$	1.46	23.0	185	$3 \cdot 10^4$
Sampling	1.16	31.0	90.0	389	**1.04**	17.0	67.0	416	**1.01**	**1.22**	**8.00**	280	**1.01**	**1.42**	19.0	**231**
MHIST	4.25	138	384	1673	3.83	66.5	288	$2 \cdot 10^4$	4.46	184	771	$1 \cdot 10^5$	1.58	13.8	90.8	$3 \cdot 10^4$
QuickSel	3.02	209	955	6523	1.38	15.0	142	7814	3.13	248	$1 \cdot 10^4$	$4 \cdot 10^5$	126	$1 \cdot 10^5$	$4 \cdot 10^5$	$4 \cdot 10^6$
Bayes	**1.12**	**3.50**	**8.00**	303	1.13	7.00	29.0	1218	1.03	2.40	15.0	$3 \cdot 10^4$	1.03	1.85	**12.9**	$1 \cdot 10^5$
KDE-FB	1.18	23.0	75.0	**293**	**1.04**	**5.00**	**17.0**	**165**	**1.01**	1.25	9.00	**254**	**1.01**	1.50	36.0	283
Learned Methods																
MSCN	1.38	7.22	15.5	**88.0**	1.14	7.62	20.6	377	**1.01**	2.00	9.91	**199**	1.02	5.30	25.0	351
LW-XGB	1.16	**3.00**	**6.00**	594	1.10	**3.00**	**7.00**	220	1.02	1.72	**5.04**	5850	**1.00**	1.68	**6.22**	$3 \cdot 10^4$
LW-NN	1.17	**3.00**	**6.00**	829	1.13	**3.10**	**7.00**	1370	1.06	1.88	**4.89**	$4 \cdot 10^4$	1.16	3.29	22.1	$3 \cdot 10^4$
Naru	**1.09**	**2.50**	**4.00**	**57**	1.06	**3.30**	**9.00**	**153**	**1.01**	**1.14**	**1.96**	**161**	**1.01**	**1.09**	**1.35**	**16.0**
DeepDB	**1.11**	4.00	8.50	**59.0**	1.06	**5.00**	**14.0**	1293	1.00	1.30	**2.40**	1568	1.02	1.86	**5.88**	5086
L v.s. T	**win**	**win**	**win**	**win**	lose	**win**	**win**	**win**	**win**	**win**	**win**	**win**	**win**	**win**	**win**	**win**

Naru performs the best among all estimators

Learned methods are more accurate

Are Learned Methods Efficient?



- Training time:
  - DBMS ≈ LW-XGB ≫ Others
- Inference time:
  - DBMS ≈ Query-Driven ≫ Data-Driven
- Benefit from GPU:
  - Limited and may introduce overhead

Takeaways in Static Environment

- Accuracy
  - Learned methods outperform traditional methods
  - Naru performs the best
- Learned methods are costly
  - Longer training time
  - Longer inference time
- Benefit from GPU is limited

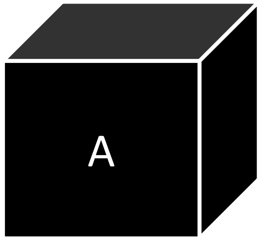
Questions

- Are Learned Methods Ready for **Static** Environments?
- Are Learned Methods Ready for **Dynamic** Environments?
- When Do Learned Estimators Go **Wrong**?

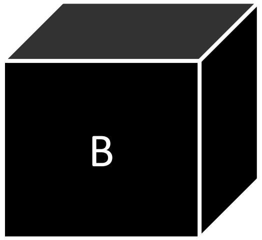
Update frequency matters

- q-error: 100
- Update time: **1 minutes**

- q-error: 10
- Update time: **1 hour**

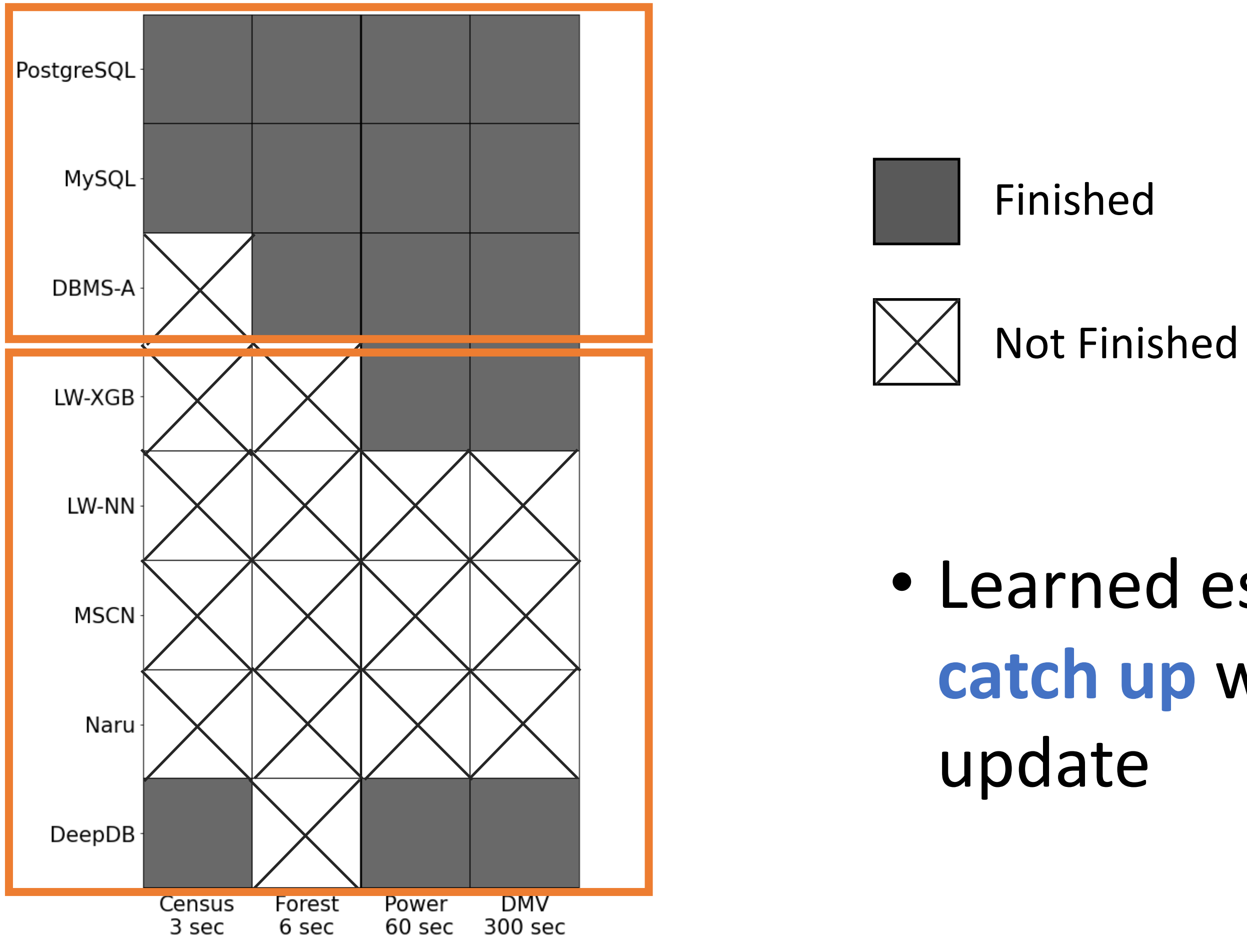


V.S.



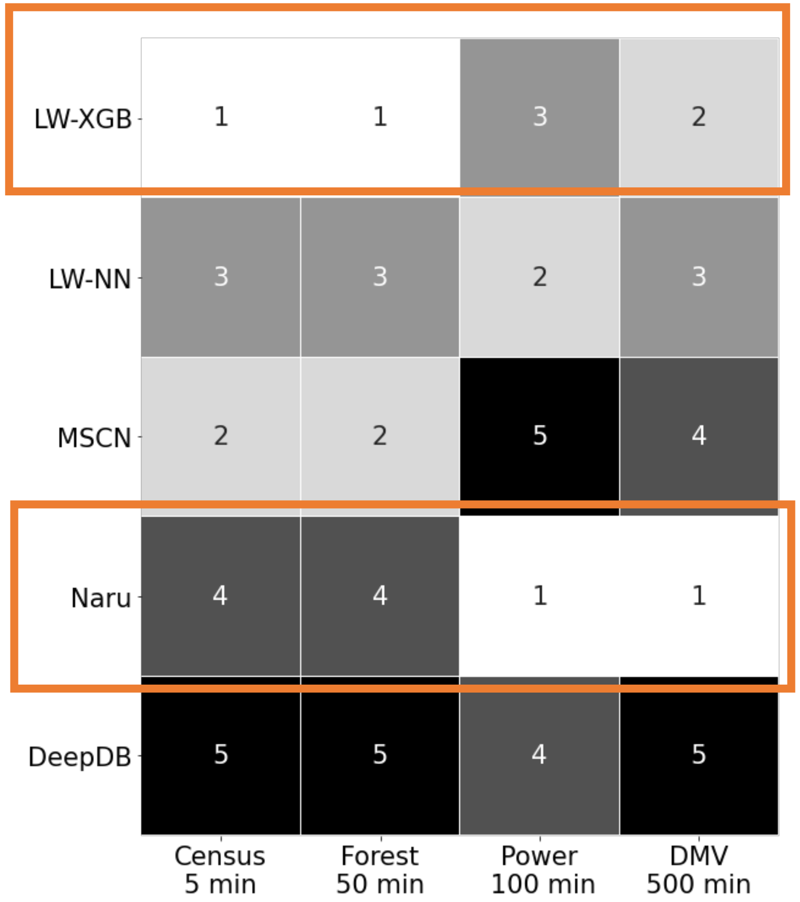
Which one is better?

Performance under fast data updates



- Learned estimators **cannot catch up** with fast data update

Who is the winner?



- There is **no clear winner** within learned estimators

Takeaways in Dynamic Environment

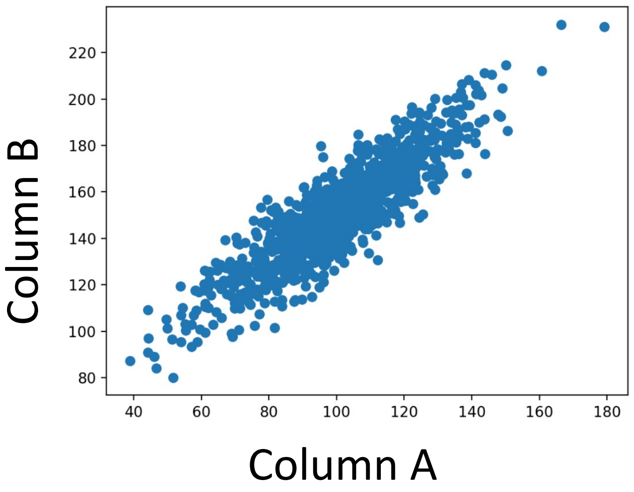
- Learned methods fail to catch up with fast data update
- There is no clear winner among learned estimators

Questions

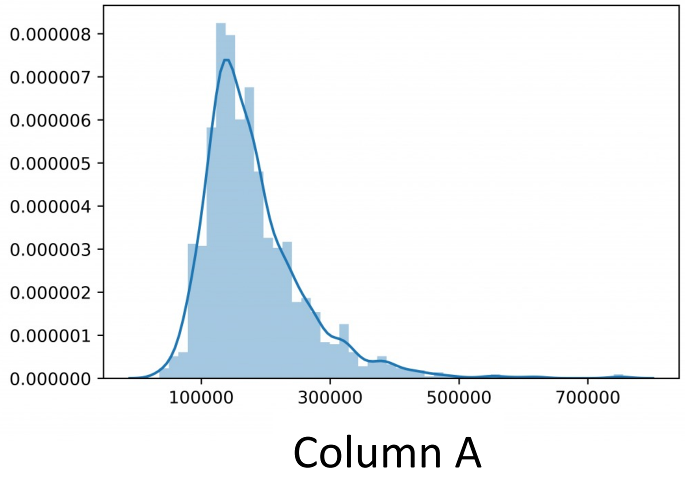
- Are Learned Methods Ready for **Static** Environments?
- Are Learned Methods Ready for **Dynamic** Environments?
- When Do Learned Estimators Go **Wrong**?

Three aspects

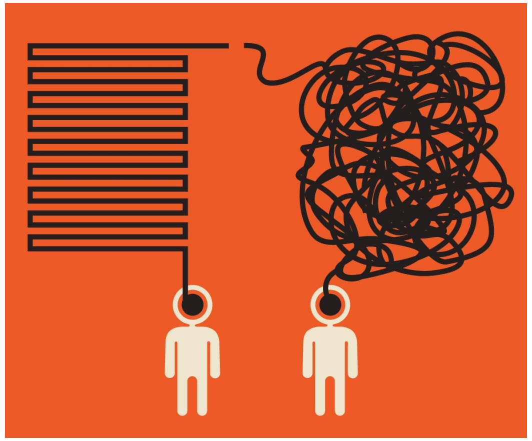
1. Correlation



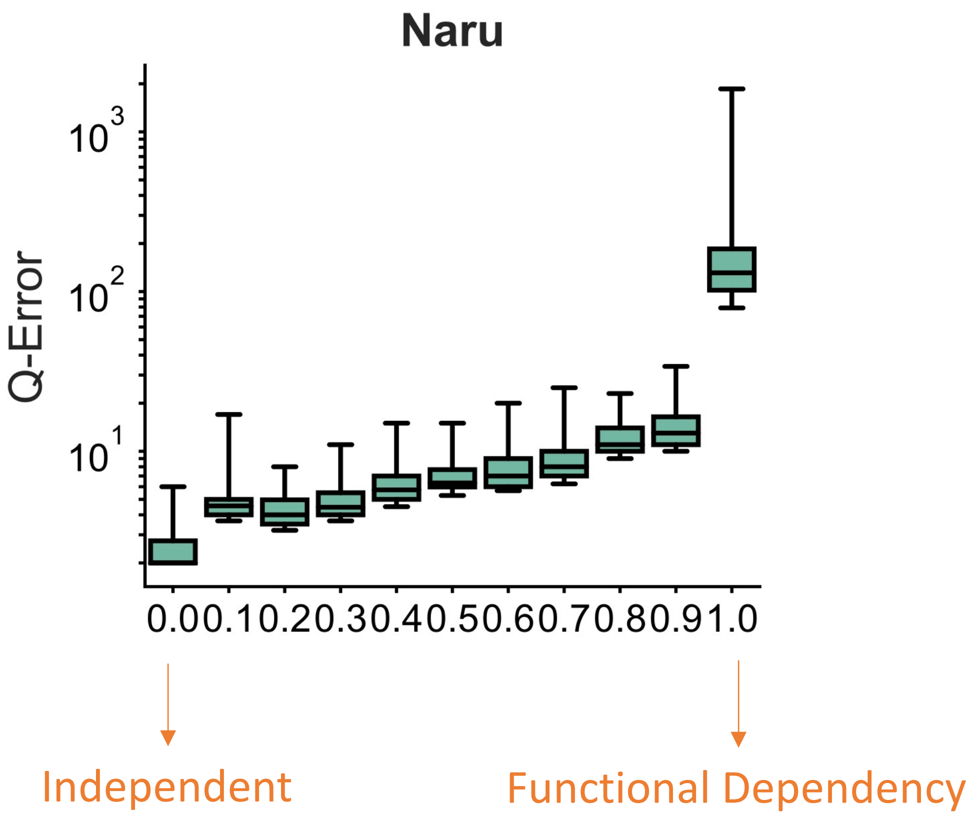
2. Distribution



3. Logical or Illogical

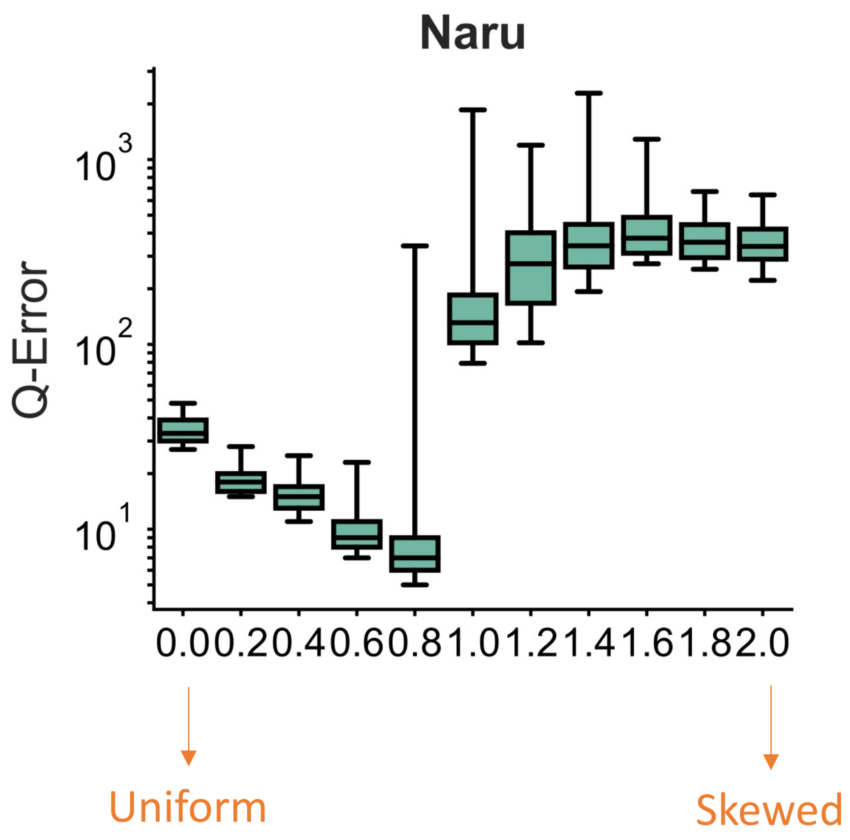


Correlation



- Error becomes larger on more correlated dataset

Distribution



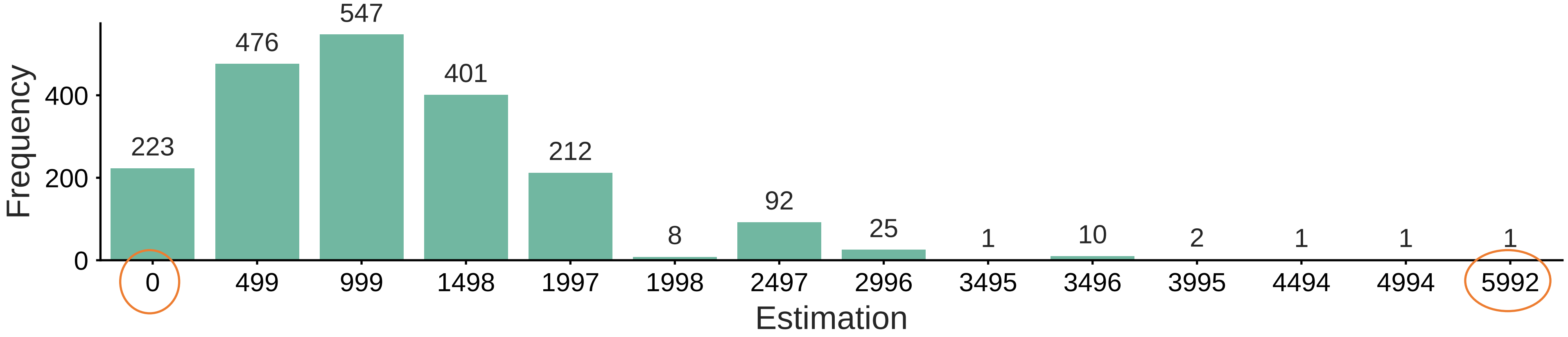
- No clear pattern, hard to explain

Logical or Illogical

- Example 1: Estimation results are not monotonic
  - Q1: SELECT * FROM R WHERE A >= 320 AND A <= 800 AND ...
  - Q2: SELECT * FROM R WHERE A >= **340** AND A <= **740** AND ...

Card(Q2) is **larger than** Card(Q1) by **61%** on LW-XGB

- Example 2: Estimation result can be **unstable** using Naru



Rules for Logical Cardinality Estimator

Rule	Naru	MSCN	LW-XGB	LW-NN	DeepDB
Monotonicity	x	x	x	x	√
Consistency	x	x	x	x	√
Stability	x	√	√	√	√
Fidelity-A	√	x	x	x	√
Fidelity-B	√	x	x	x	√

- Except for DeepDB, all learned methods violate some of the rules

What Will Happen in Multi-Table?

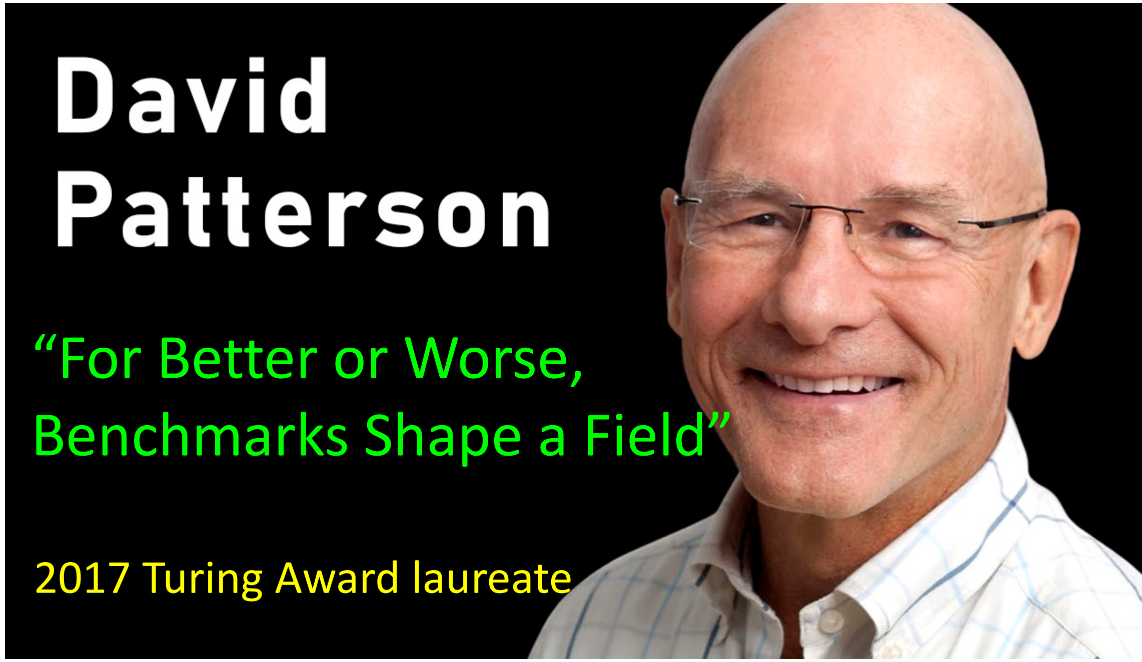
- Issues (inefficiency and untrustworthy) still exist in multi-table scenarios
- Estimate on join queries:
  - Learn a large model on (a sample of) full outer join **– Poor Scalability** [1]
  - Get estimation for single or a few tables and derive with assumptions
- The improvement space increase with the number of join tables [1]

[1] **Cardinality Estimation in DBMS: A Comprehensive Benchmark Evaluation**

Yuxing Han[1,#], Ziniu Wu[1,2,#], Peizhi Wu[3], Rong Zhu[1,*],
Jingyi Yang[3], Liang Wei Tan[3], Kai Zeng[1], Gao Cong[3], Yanzhao Qin[1,4],
Andreas Pfadler[1], Zhengping Qian[1], Jingren Zhou[1], Jiangneng Li[1,3], Bin Cui[4]
[1]*Alibaba Group,* [2]*MIT,* [3]*Nanyang Technological University,* [4]*Peking University*
[1]red.zr@alibaba-inc.com, [2]ziniuw@mit.edu, [3]gaocong@ntu.edu.sg, [4]bin.cui@pku.edu.cn

Summary

- We are **NOT** ready to deploy learned CE in production
  - Learned models tend to be **very costly**
  - Learned models are **hard to be trust**
- Impacts (**VLDB 2021 Best EA&B Paper Award)**
  - Construct the first benchmark to shape the field
  - Guide researchers and practitioners to work together to eventually push learned CE into production



Future Directions: ML for Systems

- Direction 1: Control the cost of learned models
- Direction 2: Make learned models trustworthy
- Direction 3: Solve data preparation

Thank you!

Q&A

"If 80 percent of our work is data preparation, then ensuring data quality is the important work of a machine learning team."

Andrew Ng

dataprep http://dataprep.ai

~1000 Stars

200K+ Downloads

25+ Contributors