Lecture #20

Carnegie Mellon University

Special Topics:

Self-Driving Database Management Systems

Autonomous Systems III

Kushagra Singh // 15-799 // Spring 2022

Discussion so far

- **Database Optimisation**
- Improve upon various aspects of the DB
  - Better physical layout decisions
  - Better knob configurations

Today

- **Database Learning**
- Not the same as database optimization
    - We’re not “tuning” database internals
    - Rather modelling the underlying data

Database Learning: Toward a Database that Becomes Smarter Every Time
Park et al., SIGMOD 2017

Motivation

- Work done while executing queries is lost
- Some exceptions:
  - Caching tuples
  - View selection / Adaptive Indexing

Database Learning: Toward a Database that Becomes Smarter Every Time
Park et al., SIGMOD 2017

Motivation

- Work done while executing queries is lost
- Some exceptions:
  - Caching tuples
  - View selection / Adaptive Indexing
- Can we reuse work done?

Database Learning: Toward a Database that Becomes Smarter Every Time
Park et al., SIGMOD 2017

Database Learning

- Underlying data follows some unknown distribution

- Each query answer reveals information about the underlying distribution

- If we could perfectly model this distribution, we wouldn't even need the database!

The answer to each query reveals some fuzzy knowledge *about the answers to other queries, even if each query accesses a different subset of tuples and columns.*

Database Learning: Toward a Database that Becomes Smarter Every Time
Park et al., SIGMOD 2017

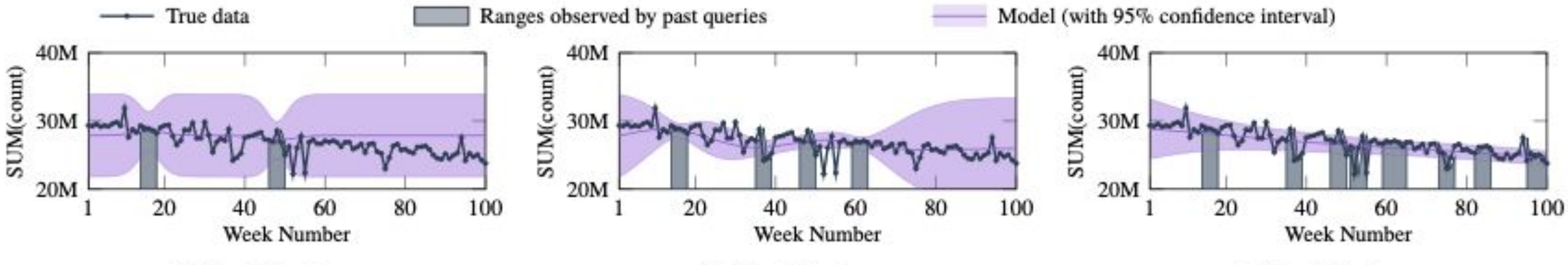
Database Learning

- Unfortunately, modelling the distribution perfectly is not always feasible.
- Approximate models can still be beneficial!
- Can improve sample based answers in AQP settings

> We call the above goal *Database Learning* (DBL), as it is reminiscent of the inferential goal of Machine Leaning (ML) whereby past observations are used to improve future predictions [15,16,66]. Likewise, our goal in DBL is to enable a similar principle by **learning from past observations, but in a query processing setting**. Specifically, in DBL, we plan to treat approximate answers to past queries as observations, and use them to refine our posterior knowledge of the underlying data, which in turn can be used to speed up future queries.

Database Learning: Toward a Database that Becomes Smarter Every Time
Park et al., SIGMOD 2017

Database Learning



(a) After 2 Queries (b) After 4 Queries (c) After 8 Queries

Each query answer reveals information about the underlying distribution

Database Learning: Toward a Database that Becomes Smarter Every Time
Park et al., SIGMOD 2017

Database Learning

- Goal – Given previous query answers, improve future query answers

- How – Query answers could potentially be correlated; exploit this!

Database Learning: Toward a Database that Becomes Smarter Every Time
Park et al., SIGMOD 2017

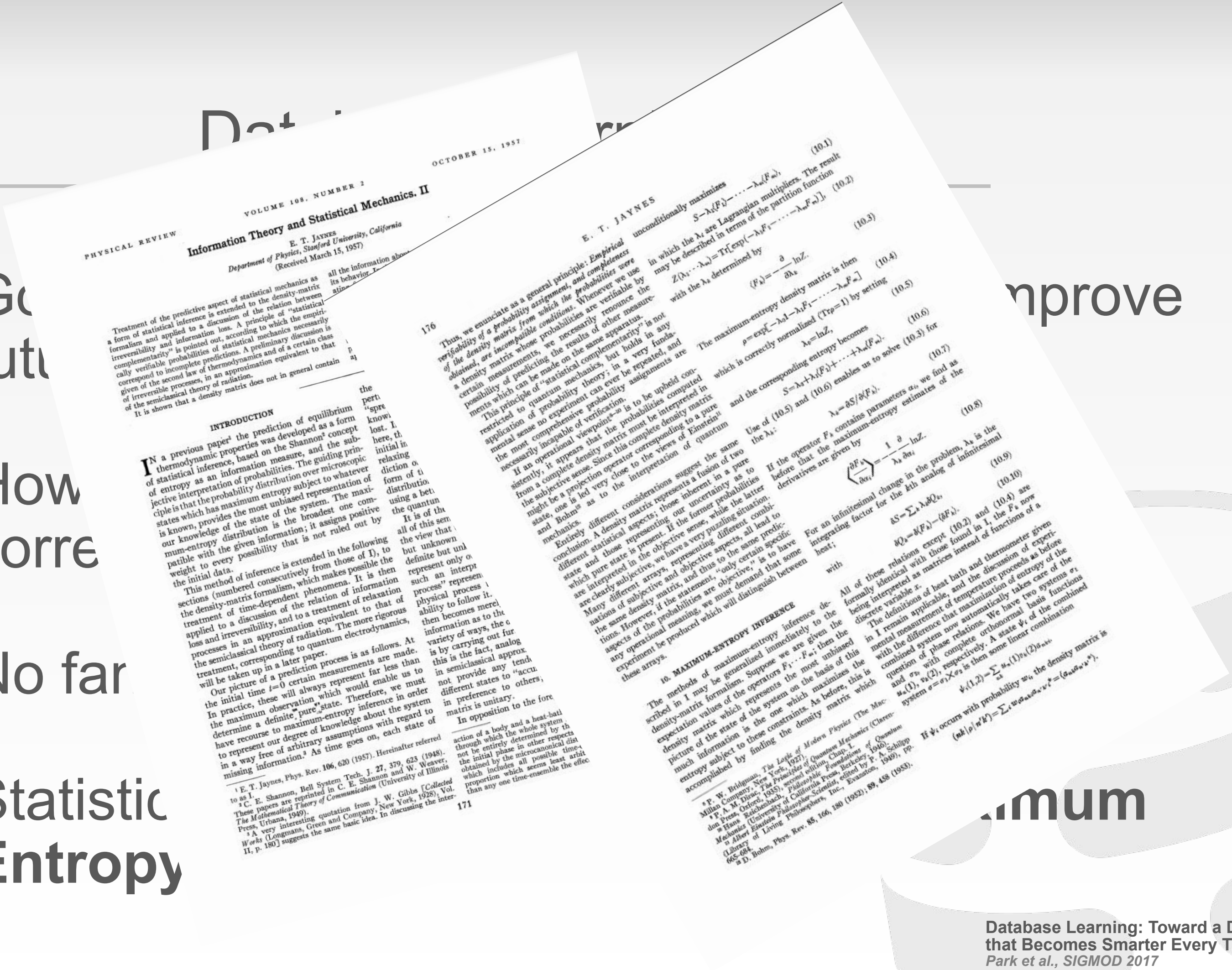
Database Learning

- Goal – Given previous query answers, improve future query answers
- How – Query answers could potentially be correlated; exploit this!
- No fancy deep learning
- Statistical approach – **Principle of Maximum Entropy** (1957)

Database Learning: Toward a Database that Becomes Smarter Every Time
Park et al., SIGMOD 2017

Dat… …r…

- Go… futu… …mprove
- How… corre…
- No far…
- Statistic… …imum **Entropy**



PHYSICAL REVIEW VOLUME 108, NUMBER 2 OCTOBER 15, 1957

Information Theory and Statistical Mechanics. II

E. T. Jaynes

Department of Physics, Stanford University, California

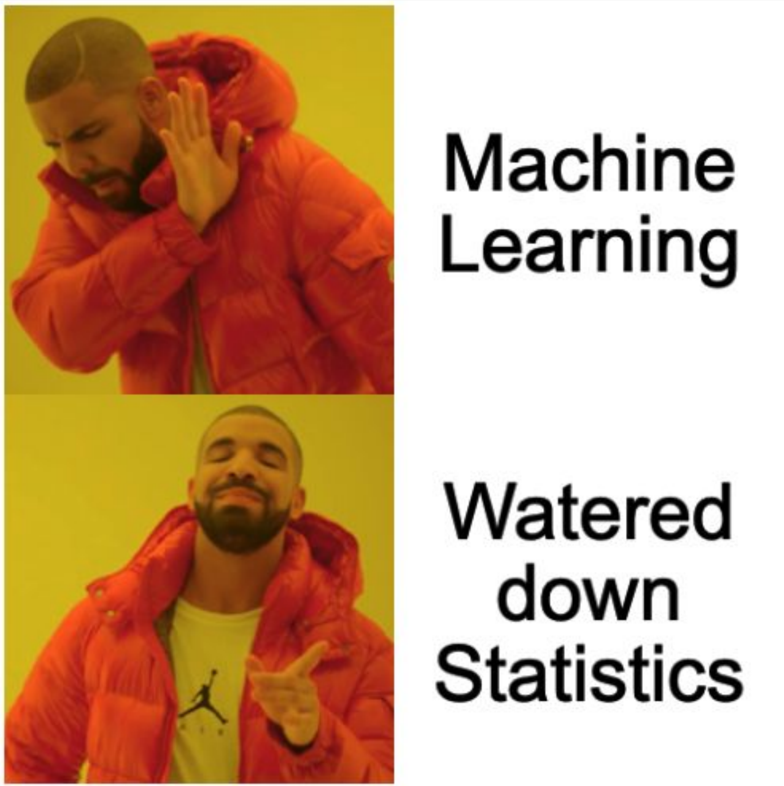
(Received March 15, 1957)

176 E. T. JAYNES

Database Learning: Toward a Database that Becomes Smarter Every Time

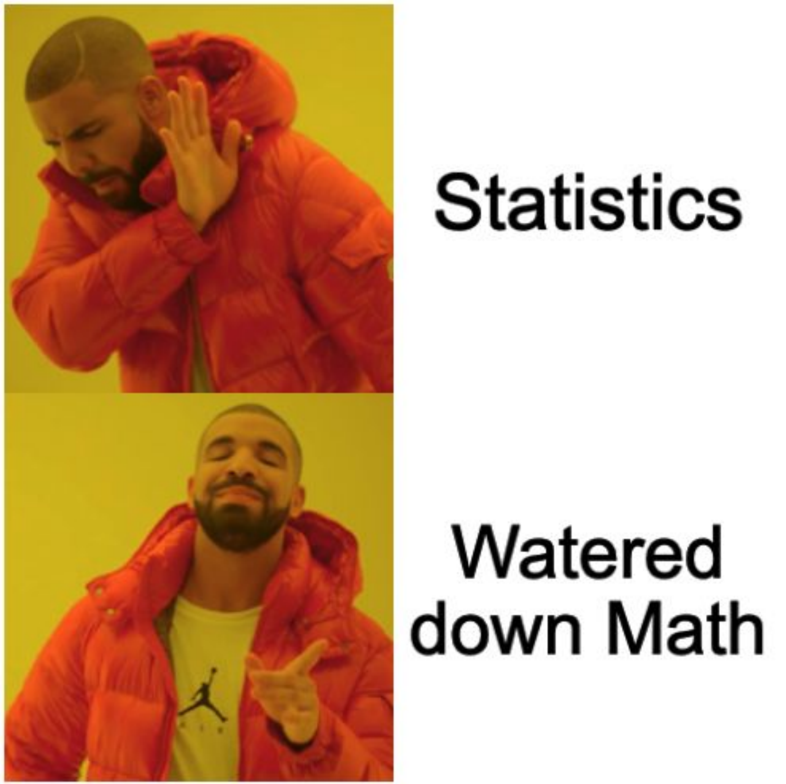
Park et al., SIGMOD 2017

Maybe contentious



~~Database Learning: Toward a Database that Becomes Smarter Every Time~~
~~*Park et al., SIGMOD 2017*~~

Definitely contentious



~~Database Learning: Toward a Database that Becomes Smarter Every Time~~
~~*Park et al., SIGMOD 2017*~~

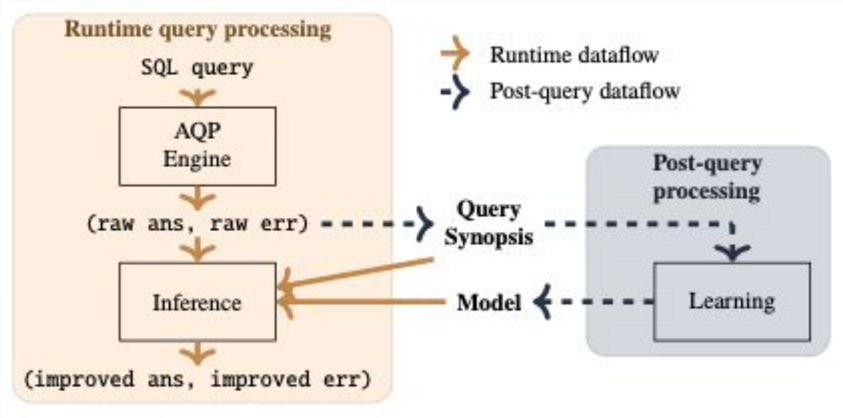
Limited Edition; HMU

Modelling credits: Anon friend

~~Database Learning: Toward a Database that Becomes Smarter Every Time~~
~~*Park et al., SIGMOD 2017*~~

Overall Architecture

- Query Synopsis
- Inference Model
- AQP engine (off the shelf)

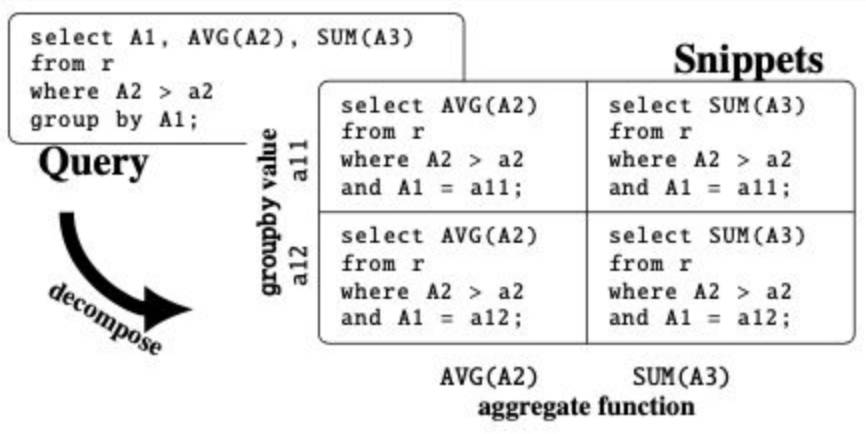


Database Learning: Toward a Database that Becomes Smarter Every Time
Park et al., SIGMOD 2017

Query Synopsis

Break down query into "snippets"

- Snippet – supported SQL query with a scalar result
- Single aggregate function



Database Learning: Toward a Database that Becomes Smarter Every Time
Park et al., SIGMOD 2017

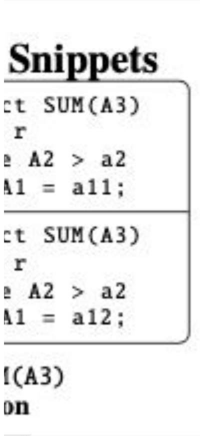
Break down query in “snippets”

- Snippet – suppor SQL query with a scalar result
- Single aggregate function

2.2 Supported Queries

Verdict supports aggregate queries that are flat (i.e., no derived tables or sub-queries) with the following conditions:

1. **Aggregates**. Any number of SUM, COUNT, or AVG aggregates can appear in the `select` clause. The arguments to these aggregates can also be a *derived attribute*.
2. **Joins**. Verdict supports foreign-key joins between a fact table[2] and any number of dimension tables, exploiting the fact that this type of join does not introduce a sampling bias [3]. For simplicity, our discussion in this paper is based on a denormalized table.
3. **Selections**. Verdict currently supports equality and inequality comparisons for categorical and numeric attributes (including the `in` operator). Currently, Verdict does not support disjunctions and textual filters (e.g., `like '%Apple%'`) in the `where` clause.
4. **Grouping**. `groupby` clauses are supported for both stored and derived attributes. The query may also include a `having` clause. Note that the underlying AQP engine may affect the cardinality of the result set depending on the `having` clause (e.g., subset/superset error). Verdict simply operates on the result set returned by the AQP engine.



Database Learning: Toward a Database that Becomes Smarter Every Time
Park et al., SIGMOD 2017

Inference Model - Problem setup

- Query snippet q_i – AQP returns (θ_i, β_i)

Sym.	Meaning
q_i	i-th (supported) query snippet
$n+1$	index number for a new snippet
$\boldsymbol{\theta}_i$	random variable representing our knowledge of the raw answer to q_i
θ_i	(actual) raw answer computed by AQP engine for q_i
β_i	expected error associated with θ_i
$\bar{\boldsymbol{\theta}}_i$	random variable representing our knowledge of the *exact* answer to q_i
$\bar{\theta}_i$	exact answer to q_i
$\widehat{\theta}_{n+1}$	improved answer to the new snippet
$\widehat{\beta}_{n+1}$	improved error to the new snippet

Database Learning: Toward a Database that Becomes Smarter Every Time
Park et al., SIGMOD 2017

Inference Model - Problem setup

- Query snippet q_i – AQP returns (θ_i, β_i)
- Q_n (synopsis) – AQP answers and errors for n snippets

Sym.	Meaning
q_i	i-th (supported) query snippet
$n+1$	index number for a new snippet
$\boldsymbol{\theta}_i$	random variable representing our knowledge of the raw answer to q_i
θ_i	(actual) raw answer computed by AQP engine for q_i
β_i	expected error associated with θ_i
$\bar{\boldsymbol{\theta}}_i$	random variable representing our knowledge of the *exact* answer to q_i
$\bar{\theta}_i$	exact answer to q_i
$\widehat{\theta}_{n+1}$	improved answer to the new snippet
$\widehat{\beta}_{n+1}$	improved error to the new snippet

Database Learning: Toward a Database that Becomes Smarter Every Time
Park et al., SIGMOD 2017

Inference Model - Problem setup

- Query snippet q_i – AQP returns (θ_i, β_i)
- Q_n (synopsis) – AQP answers and errors for *n* snippets
- Assumption: same aggregate and attribute e.g. AVG(A_k)

Sym.	Meaning
q_i	i-th (supported) query snippet
$n+1$	index number for a new snippet
$\boldsymbol{\theta}_i$	random variable representing our knowledge of the raw answer to q_i
θ_i	(actual) raw answer computed by AQP engine for q_i
β_i	expected error associated with θ_i
$\bar{\boldsymbol{\theta}}_i$	random variable representing our knowledge of the *exact* answer to q_i
$\bar{\theta}_i$	exact answer to q_i
$\widehat{\theta}_{n+1}$	improved answer to the new snippet
$\widehat{\beta}_{n+1}$	improved error to the new snippet

Database Learning: Toward a Database that Becomes Smarter Every Time
Park et al., SIGMOD 2017

Inference Model - Problem setup

- Query snippet q_i – AQP returns (θ_i, β_i)
- Q_n (synopsis) – AQP answers and errors for n snippets
- Assumption: same aggregate and attribute e.g. AVG(A_k)
- Problem: Given Q_n and (θ_{n+1}, β_{n+1}) – compte improved answer for q_{n+1}

Sym.	Meaning
q_i	i-th (supported) query snippet
$n+1$	index number for a new snippet
$\boldsymbol{\theta}_i$	random variable representing our knowledge of the raw answer to q_i
θ_i	(actual) raw answer computed by AQP engine for q_i
β_i	expected error associated with θ_i
$\bar{\boldsymbol{\theta}}_i$	random variable representing our knowledge of the *exact* answer to q_i
$\bar{\theta}_i$	exact answer to q_i
$\widehat{\theta}_{n+1}$	improved answer to the new snippet
$\widehat{\beta}_{n+1}$	improved error to the new snippet

Database Learning: Toward a Database that Becomes Smarter Every Time
Park et al., SIGMOD 2017

Inference Model - Problem setup

- Represent query snippet answers as a random vars
- n + 1 random vars; 1 for each snippet in synopsis and one for current snippet: $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_{n+1}$

Sym.	Meaning
q_i	i-th (supported) query snippet
$n+1$	index number for a new snippet
$\boldsymbol{\theta}_i$	random variable representing our knowledge of the raw answer to q_i
θ_i	(actual) raw answer computed by AQP engine for q_i
β_i	expected error associated with θ_i
$\bar{\boldsymbol{\theta}}_i$	random variable representing our knowledge of the *exact* answer to q_i
$\bar{\theta}_i$	exact answer to q_i
$\widehat{\theta}_{n+1}$	improved answer to the new snippet
$\widehat{\beta}_{n+1}$	improved error to the new snippet

Database Learning: Toward a Database that Becomes Smarter Every Time
Park et al., SIGMOD 2017

Inference Model - Problem setup

- Represent query snippet answers as a random vars
- n + 1 random vars; 1 for each snippet in synopsis and one for current snippet: $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_{n+1}$
- These vars form a joint PDF

Sym.	Meaning
q_i	i-th (supported) query snippet
$n+1$	index number for a new snippet
$\boldsymbol{\theta}_i$	random variable representing our knowledge of the raw answer to q_i
θ_i	(actual) raw answer computed by AQP engine for q_i
β_i	expected error associated with θ_i
$\bar{\boldsymbol{\theta}}_i$	random variable representing our knowledge of the *exact* answer to q_i
$\bar{\theta}_i$	exact answer to q_i
$\widehat{\theta}_{n+1}$	improved answer to the new snippet
$\widehat{\beta}_{n+1}$	improved error to the new snippet

Database Learning: Toward a Database that Becomes Smarter Every Time
Park et al., SIGMOD 2017

Inference Model

If we have $f(\boldsymbol{\theta_1} = \theta'_1, \ldots, \boldsymbol{\theta_{n+1}} = \theta'_{n+1}, \bar{\boldsymbol{\theta}}_{n+1} = \bar{\theta}'_{n+1})$

then the prediction is the value of $\bar{\theta}'_{n+1}$ that maximizes

$$f(\bar{\boldsymbol{\theta}}_{n+1} = \bar{\theta}'_{n+1} \mid \boldsymbol{\theta_1} = \theta_1, \ldots, \boldsymbol{\theta_{n+1}} = \theta_{n+1})$$

$$\ddot{\theta}_{n+1} = \underset{\bar{\theta}'_{n+1}}{\text{Arg Max}} \; f(\bar{\theta}'_{n+1} \mid \boldsymbol{\theta}_1 = \theta_1, \ldots, \boldsymbol{\theta}_{n+1} = \theta_{n+1})$$

Slide credits: Huanyi Chen
CS848, University of Waterloo

Database Learning: Toward a Database that Becomes Smarter Every Time
Park et al., SIGMOD 2017

Inference Model

- Joint PDF derived from answer statistics – means, variances, and covariances
- ME principle: Joint PDF will maximise entropy

$$h(f) = -\int f(\vec{\theta}) \cdot \log f(\vec{\theta})\, d\vec{\theta}$$

***θ**s* are the observations (query answers)

- Constraint: satisfy means, variances, and covariances (derived from observations)

Database Learning: Toward a Database that Becomes Smarter Every Time
Park et al., SIGMOD 2017

Inference Model

- Leap: joint PDF that maximises entropy while satisfying means, variances, and covariances is a **normal distribution** with same statistics

$$h(f) = -\int f(\vec{\theta}) \cdot \log f(\vec{\theta})\, d\vec{\theta}$$

θ_s are the observations (query answers)

$$f(\vec{\theta}) = \frac{1}{\sqrt{(2\pi)^{n+2}|\Sigma|}} \exp\left(-\frac{1}{2}(\vec{\theta}-\vec{\mu})^\top \Sigma^{-1}(\vec{\theta}-\vec{\mu})\right)$$

- So if we have statistics, we can get the distribution!

Database Learning: Toward a Database that Becomes Smarter Every Time
Park et al., SIGMOD 2017

Inference Model

- For a normal pdf, the value that maximises it is the mean – which would be the new “improved” answer

- Similarly, the variance is the error bound

$$\mu_c = \bar{\mu}_{n+1} + \vec{k}_{n+1}^{\intercal} \Sigma_{n+1}^{-1} (\vec{\theta}_{n+1} - \vec{\mu}_{n+1}) \tag{4}$$

$$\sigma_c^2 = \bar{\kappa}^2 - \vec{k}_{n+1}^{\intercal} \Sigma_{n+1}^{-1} \vec{k}_{n+1} \tag{5}$$

where:

- $\vec{k}_{n+1}$ is a column vector of length $n + 1$ whose i-th element is $(i, n + 2)$-th entry of Σ;
- Σ_{n+1} is a $(n + 1) \times (n + 1)$ submatrix of Σ consisting of Σ's first $n + 1$ rows and columns;
- $\vec{\theta}_{n+1} = (\theta_1, \ldots, \theta_{n+1})^{\intercal}$;
- $\vec{\mu}_{n+1} = (\mu_1, \ldots, \mu_{n+1})^{\intercal}$; and
- $\bar{\kappa}^2$ is the $(n + 2, n + 2)$-th entry of Σ

Database Learning: Toward a Database that Becomes Smarter Every Time
Park et al., SIGMOD 2017

Inference Model - Computing Stats

- Goal: compute means, variances, and covariances of the random vars $\theta_1,\ldots,\theta_{n+1}$
- Means: historical average
- Covariance?

$$f(\vec{\theta}) = \frac{1}{\sqrt{(2\pi)^{n+2}|\Sigma|}} \exp\left(-\frac{1}{2}(\vec{\theta} - \vec{\mu})^\intercal \Sigma^{-1} (\vec{\theta} - \vec{\mu})\right)$$

Target function

Database Learning: Toward a Database that Becomes Smarter Every Time
Park et al., SIGMOD 2017

Inference Model - Computing Stats

- Covariance (θ_i, θ_j)?
- Can be decomposed down to computing inter-tuple covariance

As a result, the covariance between query answers can be broken into an integration of the covariances between tuple-level function values, which we call *inter-tuple covariances*.

$$\mathrm{cov}(\bar{\theta}_i, \bar{\theta}_j) = \mathrm{cov}\left(\int_{t \in F_i} \nu_g(\mathbf{t})\, d\mathbf{t}, \int_{t' \in F_j} \nu_g(\mathbf{t}')\, d\mathbf{t}'\right)$$
$$= \int_{t \in F_i} \int_{t' \in F_j} \mathrm{cov}(\nu_g(\mathbf{t}), \nu_g(\mathbf{t}'))\, d\mathbf{t}\, d\mathbf{t}'$$

Computing $\mathrm{cov}(\bar{\theta}_i, \bar{\theta}_j)$ relies on a straightforward observation: *the covariance between two query snippet answers is computable using the covariances between the attribute values involved in computing those answers.* For instance, we can easily compute the covariance between (i) the average revenue of the years 2014 and 2015 and (ii) the average revenue of the years 2015 and 2016, as long as we know the covariance between the average revenues of every pair of days in 2014–2016.

Database Learning: Toward a Database that Becomes Smarter Every Time
Park et al., SIGMOD 2017

Inference Model - Validation

- Negative Estimates for FREQ(*) – discard
- Unlikely improved answer
    - If not in a certain range of AQP answer, discard

Formally, let $t \geq 0$ be the value for which the AQP engine's answer would fall within $(\ddot{\theta}_{n+1} - t, \ddot{\theta}_{n+1} + t)$ with probability δ_v (0.99 by default) if $\ddot{\theta}_{n+1}$ were the exact answer. We call the $(\ddot{\theta}_{n+1} - t, \ddot{\theta}_{n+1} + t)$ range the likely region. To compute t, we must find the value closest to $\ddot{\theta}_{n+1}$ that satisfies the following expression:

$$\Pr\left(|X - \ddot{\theta}_{n+1}| < t\right) \geq \delta_v \qquad (14)$$

Database Learning: Toward a Database that Becomes Smarter Every Time
Park et al., SIGMOD 2017

Deployment scenarios

- **Accuracy bounded**

  Keep processing till error within certain bound

- **Time bounded**

  Give the best answer in limited time

Database Learning: Toward a Database that Becomes Smarter Every Time
Park et al., SIGMOD 2017

Experiment 1 setup – Workloads

Customer1

- 310 tables,15.5K queries – 3.3K analytical supported by Spark SQL
- Only access to data distribution – used to generate a 536 GB dataset

Database Learning: Toward a Database that Becomes Smarter Every Time
Park et al., SIGMOD 2017

Experiment 1 setup ds

Customer1

- 310
  analy
- Only a d
  to gene

Database Learning: Toward a Database that Becomes Smarter Every Time
Park et al., SIGMOD 2017

Experiment 1 setup – Workloads

TPC-H

- Scale factor 100 – 100 GB
- 21 templates have aggregations
- 500 queries generated

Experiment 1 setup – Workloads

Dataset	Total # of Queries with Aggregates	# of Supported Queries	Percentage
Customer1	3,342	2,463	73.7%
TPC-H	21	14	63.6%

Table 3: Generality of Verdict. Verdict supports a large fraction of real-world and benchmark queries.

Database Learning: Toward a Database that Becomes Smarter Every Time
Park et al., SIGMOD 2017

Experiment 1 setup – Implementations

NoLearn

- Online aggregation engine
- Randomly samples and splits data
- Error bound improves as more batches get processed

Database Learning: Toward a Database that Becomes Smarter Every Time
Park et al., SIGMOD 2017

Experiment 1 setup – Implementations

Verdict

- Uses NoLearn as AQP engine
- Takes NoLearn's answer and improves

Experiment 1 setup – Two Flavours

- Cached – All samples exist in memory
- Not cached – Samples read from SSD

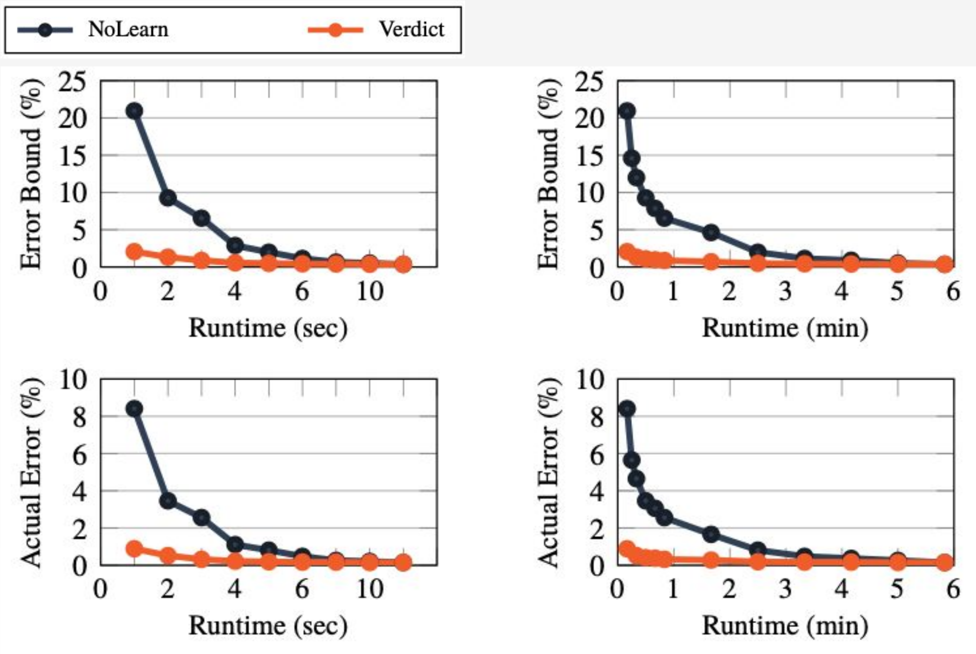
Database Learning: Toward a Database that Becomes Smarter Every Time
Park et al., SIGMOD 2017

Experiment 1 – Key takeaways

- Verdict produced smaller errors even when runtime was very large
- Verdict showed faster runtime for the same target errors

Database Learning: Toward a Database that Becomes Smarter Every Time
Park et al., SIGMOD 2017

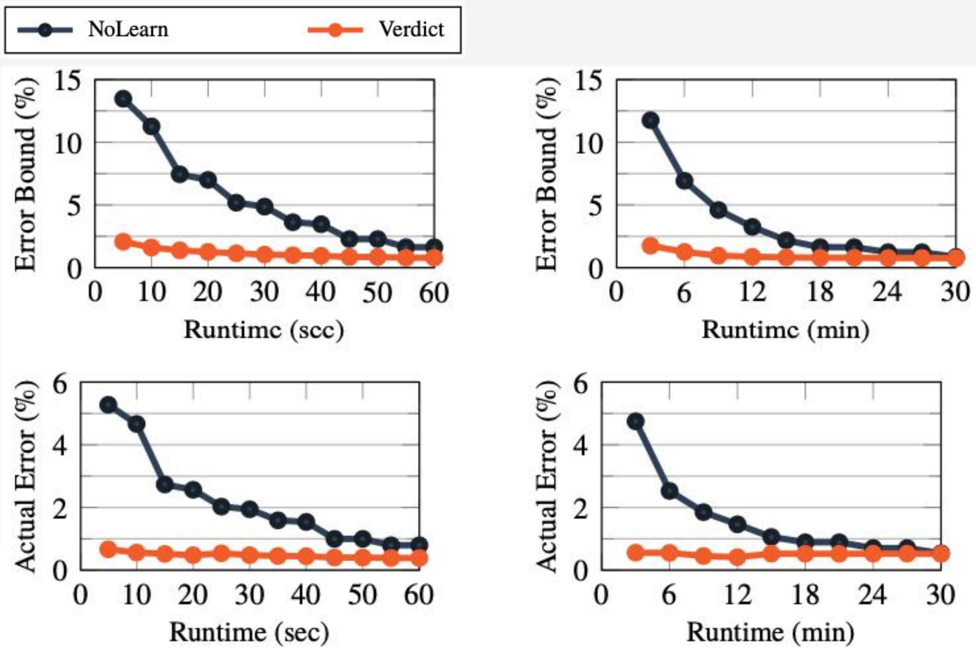
Experiment 1 – Results (Customer1)



(a) Cached, Customer1 (b) Not Cached, Customer1

Database Learning: Toward a Database that Becomes Smarter Every Time
Park et al., SIGMOD 2017

Experiment 1 – Results (TCP-H)



(c) Cached, TPC-H

(d) Not Cached, TPC-H

Database Learning: Toward a Database that Becomes Smarter Every Time
Park et al., SIGMOD 2017

Experiment 1 – Results summary

	Cached?	Error Bound	Time Taken NoLearn	Time Taken Verdict	Speedup
Customer1	Yes	2.5%	4.34 sec	0.57 sec	**7.7×**
		1.0%	6.02 sec	2.45 sec	**2.5×**
	No	2.5%	140 sec	6.1 sec	**23.0×**
		1.0%	211 sec	37 sec	**5.7×**
TPC-H	Yes	4.0%	26.7 sec	2.9 sec	**9.3×**
		2.0%	34.2 sec	12.9 sec	**2.7×**
	No	4.0%	456 sec	72 sec	**6.3×**
		2.0%	524 sec	265 sec	**2.1×**

	Cached?	Runtime	Achieved Error Bound NoLearn	Achieved Error Bound Verdict	Error Reduction
Customer1	Yes	1.0 sec	21.0%	2.06%	**90.2%**
		5.0 sec	1.98%	0.48%	**75.8%**
	No	10 sec	21.0%	2.06%	**90.2%**
		60 sec	6.55%	0.87%	**86.7%**
TPC-H	Yes	5.0 sec	13.5%	2.13%	**84.2%**
		30 sec	4.87%	1.04%	**78.6%**
	No	3.0 min	11.8%	1.74%	**85.2%**
		10 min	4.49%	0.92%	**79.6%**

Database Learning: Toward a Database that Becomes Smarter Every Time
Park et al., SIGMOD 2017

Experiment 2 setup

- RQ: Impact of having queries with diverse set of columns in selection predicates

Database Learning: Toward a Database that Becomes Smarter Every Time
Park et al., SIGMOD 2017

Experiment 2 setup

- RQ: Impact of having queries with diverse set of columns in selection predicates
- Synthetic dataset – 50 columns (10% categorical), 5M rows

Experiment 2 setup

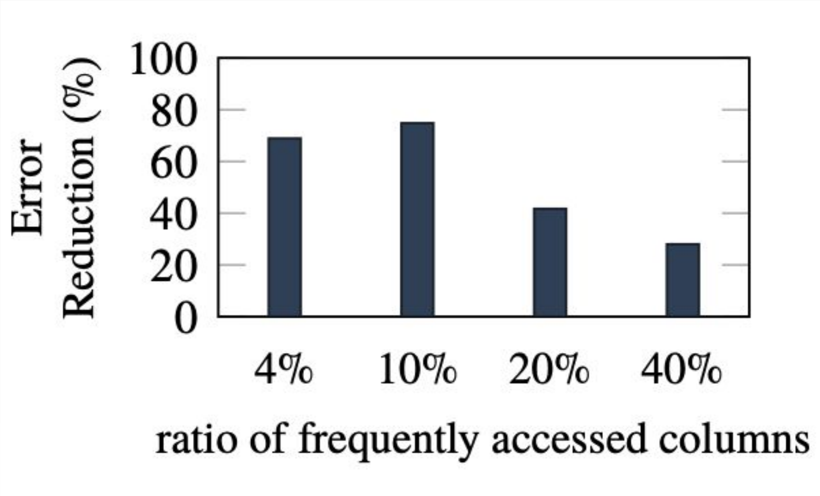
- RQ: Impact of having queries with diverse set of columns in selection predicates
- Synthetic dataset – 50 columns (10% categorical), 5M rows
- 4 synthetic workloads – varying proportions of frequently accessed columns

Database Learning: Toward a Database that Becomes Smarter Every Time
Park et al., SIGMOD 2017

Experiment 2 results

- As % of frequently accessed columns increases, Verdict's relative error reduction over NoLearn gradually decreases

- Expected – Verdict constructs its model based on historical data



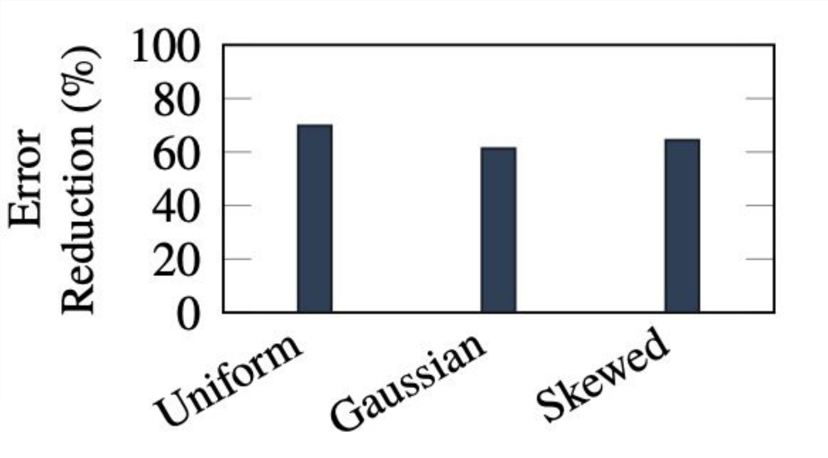
Database Learning: Toward a Database that Becomes Smarter Every Time
Park et al., SIGMOD 2017

Experiment 3 setup

- RQ: Verdict's sensitivity to underlying data distribution
- Synthetic dataset – tables generated using uniform, Gaussian, and log-normal

Experiment 3 results

- Performance remains roughly consistent
- Expected – No assumptions were made about the underlying data.
- Maximum Entropy Principle works for all



Database Learning: Toward a Database that Becomes Smarter Every Time
Park et al., SIGMOD 2017

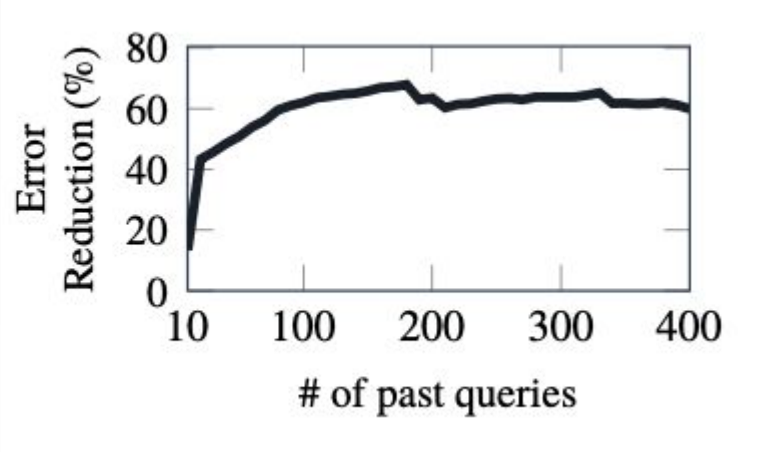
Experiment 4 setup

- RQ: Change in Verdict’s performance w.r.t. past queries

Experiment 4 results

- Error bound improves as the number of queries; eventually tapers off

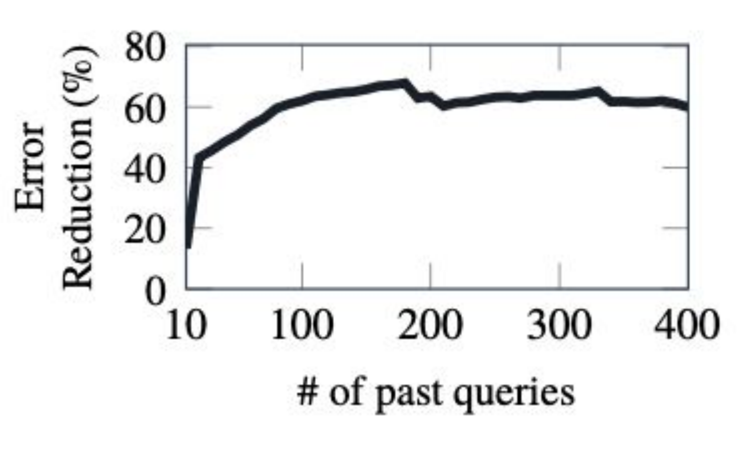
- Takeaway – Small number of queries are required to deliver a good performance



Database Learning: Toward a Database that Becomes Smarter Every Time
Park et al., SIGMOD 2017

Experiment 4 results

- Error bound improves as the number of queries; eventually tapers off
- Takeaway – Small number of queries are required to deliver a good performance
- A little sketchy – this would be highly data dependent



Database Learning: Toward a Database that Becomes Smarter Every Time
Park et al., SIGMOD 2017

Some questions

Verdict rejects the improved answer when it is not in the 99% interval of the AQP answer

- How often does this happen?
- Do the graphs include these queries, or is it plotted from cherry picked results