Database Learning: Toward a Database that Becomes Smarter Every Time

Presented by: Huanyi Chen

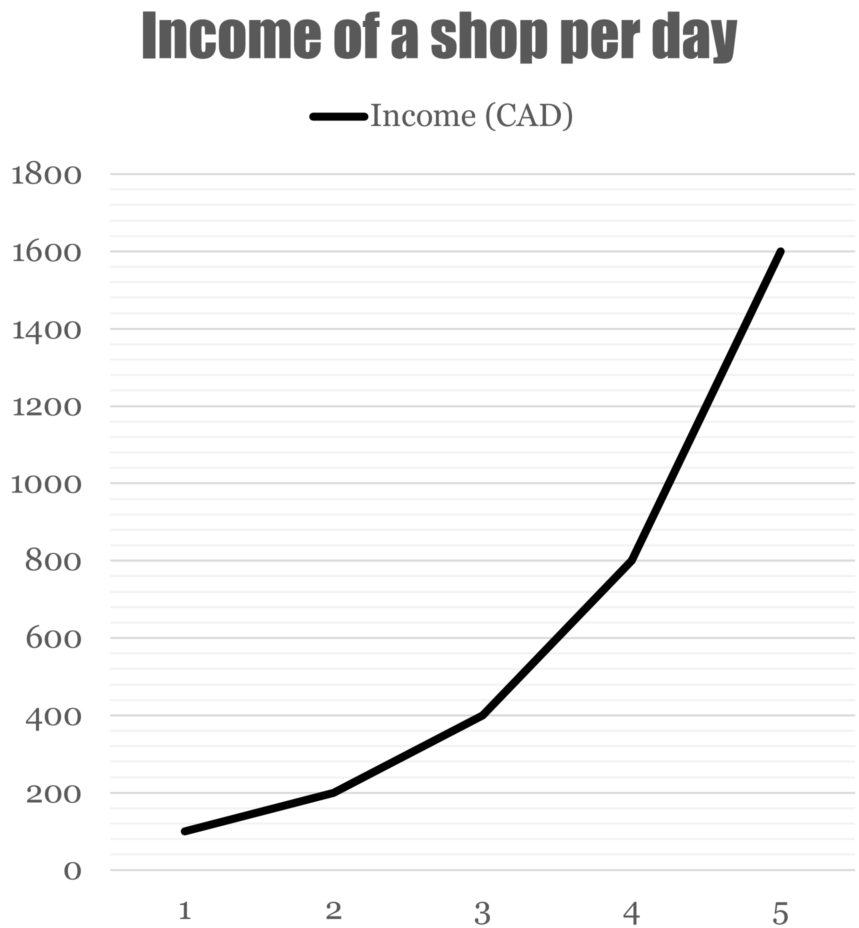
UNIVERSITY OF WATERLOO

Where does the data come from?

- Real world
- The entire dataset follows certain underlying distribution

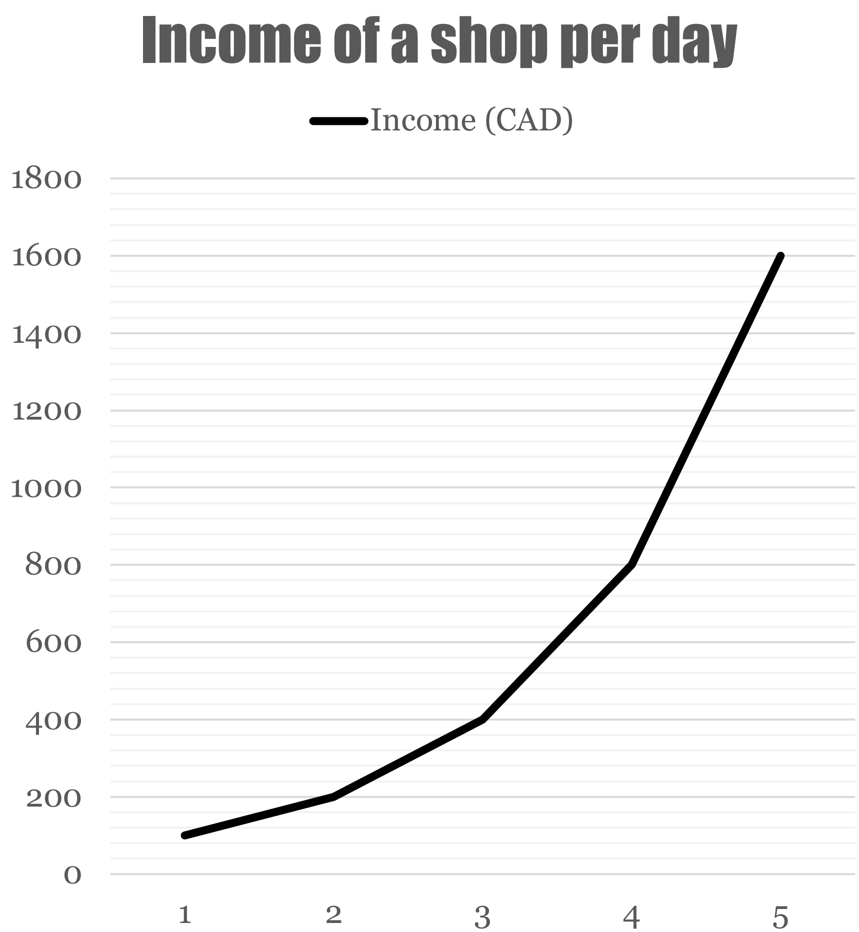
Income of a shop

# of Day	Income (CAD)
1	100
2	200
3	400
4	800
5	1600



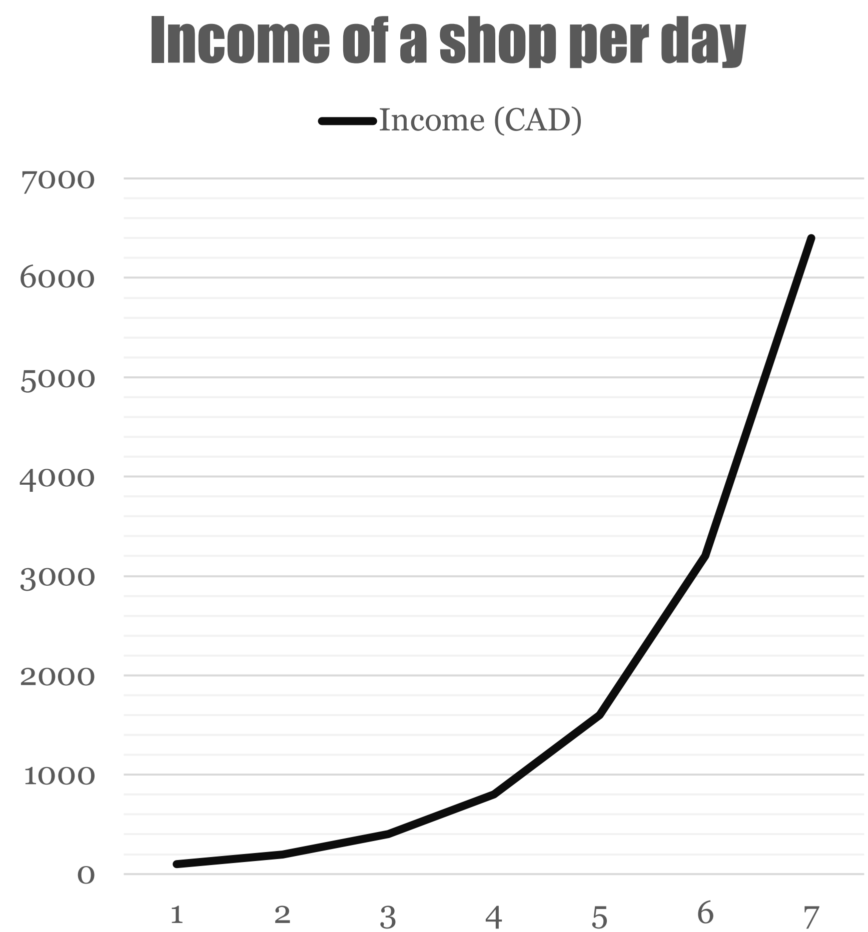
Income of a shop

# of Day	Income (CAD)
1	100
2	200
3	400
4	800
5	1600
6	?



Income of a shop

- $Income = 50 * 2^n \ \ (\mathrm{n} = 1, 2, 3 \ \ldots)$
- No database needed if we can find the underlying distribution



Which distribution do we care?

- The exact underlying distribution that generates the entire dataset and future unseen data?
  - Not possible
- An exact underlying distribution that generates the entire dataset but excludes future unseen data?
  - Benefits nothing. One can always make a model by using every value of a column, but this model is not able to predict anything. We still need to store future data in order to answer queries.
- A possible distribution that generates the entire dataset and future unseen data!

Mismatching data

- A possible distribution that generates the entire dataset and future unseen data is not able to match every data in the dataset
  - Not work when the accurate query results needed
  - Works in Approximate query processing (AQP)

Approximate Query Processing (AQP)

- Trade accuracy for response time
- Results are based on samples
- Previous query results have no help in future queries
  - so it comes Database Learning - learning from past query answers!

Database Learning Engine: Verdict

Target

- Improve future query answers by using previous query answers from an AQP engine

Workflow

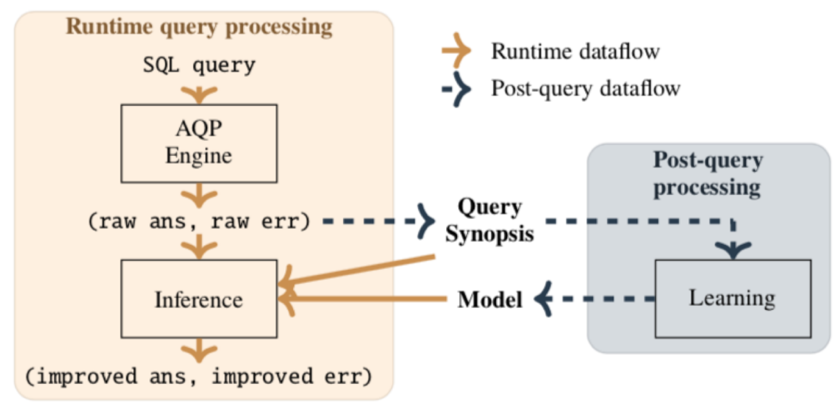


Figure 2: Workflow in Verdict. At query time, the Inference module uses the *Query Synopsis* and the *Model* to improve the query answer and error computed by the underlying AQP engine (i.e., *raw answer/error*) before returning them to the user. Each time a query is processed, the raw answer and error are added to the *Query Synopsis*. The Learning module uses this updated *Query Synopsis* to refine the current *Model* accordingly.

Verdict

- A query is decomposed into possibly multiple query snippets
  - the answer of a snippet is a single scalar value

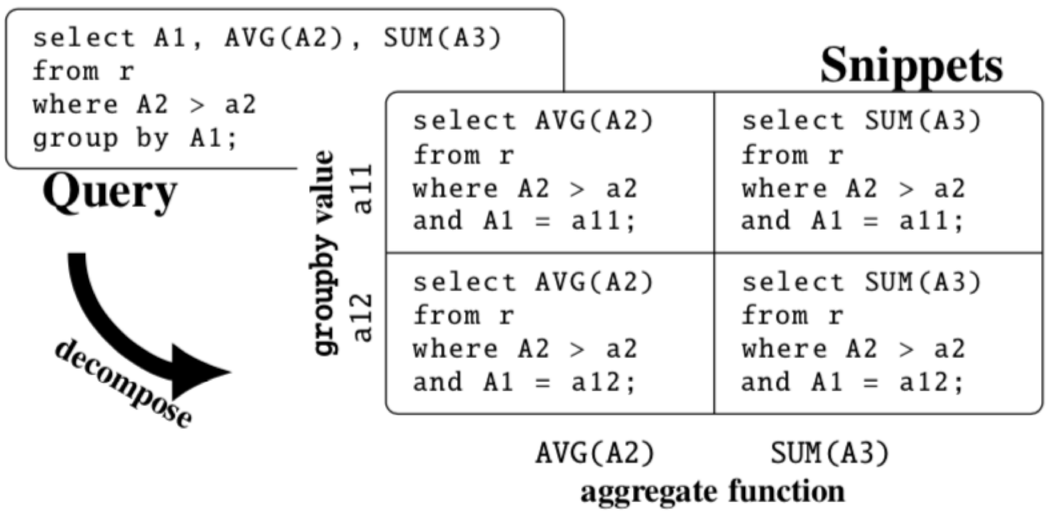


Figure 3: Example of a query's decomposition into multiple snippets.

Verdict

- A query is decomposed into possibly multiple query snippets
- Verdict exploits potential correlations between snippet answers to infer the answer of a new snippet

# of Day	Income (CAD)
1	100
2	200
3	400
4	800
5	1600

old avg

new avg

old avg and new avg are correlated

Inference

- $observations + rules = prediction$

	Observations	Rules	Prediction
Shop Income	100, 200, 400, 800, 1600	$Income = 50 * 2^n$	3200, 6400, …
Fibonacci	Initial: 1, 1	$F_n = F_{n-2} + F_{n-1}$	2, 3, 5, 8, …
Verdict	Past snippet answers from AQP + AQP answer for the new snippet	Maximize the conditional joint probability distribution function (pdf)	Improved answer and error for new snippet

UNIVERSITY OF WATERLOO

Inference: 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

Table 2: Mathematical Notations.

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

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

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

Inference: pdf

- How to find the pdf?
  - maximum entropy (ME) principal
    - $h(f) = -\int f(\vec{\theta}) \cdot \log f(\vec{\theta})\, d\vec{\theta}, \qquad where\ \vec{\theta} = (\theta_1', \dots, \theta_{n+1}', \bar{\theta}_{n+1}')$
- The joint pdf maximizing the above entropy differs depending on the kinds of given testable information
  - Verdict uses the first and the second order statistics of the random variables: mean, variances, and covariances.

Inference: pdf

Lemma 1. Let $\theta = (\theta_1, \ldots, \theta_{n+1}, \bar{\theta}_{n+1})^\intercal$ be a vector of n+2 random variables with mean values $\vec{\mu} = (\mu_1, \ldots, \mu_{n+1}, \bar{\mu}_{n+1})^\intercal$ and a $(n+2) \times (n+2)$ covariance matrix Σ specifying their variances and pairwise covariances. The joint pdf f over these random variables that maximizes $h(f)$ while satisfying the provided means, variances, and covariances is the following function:

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

and this solution is unique.

Inference: model-based answer and error

Generally

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

Computing above conditional pdf may be a computationally expensive task

Inference: model-based answer and error

However, computing the conditional pdf in lemma 1 is not expensive and computable; the result is another normal distribution.
The mean μ_c and variance σ_c^2 are given by:

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

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

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;
- $\vec{\theta}_{n+1} = (\theta_1, \ldots, \theta_{n+1})^\top$;
- $\vec{\mu}_{n+1} = (\mu_1, \ldots, \mu_{n+1})^\top$; and
- $\bar{\kappa}^2$ is the $(n+2, n+2)$-th entry of Σ

Inference: model-based answer and error

- Model-based answer
  - $\ddot{\theta}_{n+1} = \mu_c$
- Model-based error
  - $\ddot{\beta}_{n+1} = \sigma_c$
- Improved answer and error
  - $\left(\hat{\theta}_{n+1}, \hat{\beta}_{n+1}\right) = \left(\ddot{\theta}_{n+1}, \ddot{\beta}_{n+1}\right)$ (if validation succeed)
  - $\left(\hat{\theta}_{n+1}, \hat{\beta}_{n+1}\right) = \left(\theta_{n+1}, \beta_{n+1}\right)$ (if validation failed, return AQP answers)

Inference: means, variances, and covariances

- mean ($\vec{\mu}$)
  - the arithmetic mean of the past query answers for the mean of each random variable, $\boldsymbol{\theta_1}, \ldots, \boldsymbol{\theta_{n+1}}, \boldsymbol{\overline{\theta}_{n+1}}$.
- variances, and covariances (Σ)
  - the covariance between two query snippet answers is computable using the covariances between the attribute values involved in computing those answers

Inference: means, variances, and covariances

# of Day	Income (CAD)
1	100
2	200
3	400
4	800
5	1600

old avg

new avg

old avg and new avg are correlated

Inference: means, variances, and covariances

Inter-tuple Covariances

# of Day	Income (CAD)	Income (CAD)
1	100	100
2	200	200
3	400	400
4	800	800
5	1600	1600

Inference: means, variances, and covariances

- Estimate the inter-tuple covariances
  - analytical covariance functions
    - squared exponential covariance functions: capable of approximating any continuous target function arbitrarily closely as the number of observations (here, query answers) increases
  - compute variances, and covariances (Σ) efficiently

Experiments

- Up to 23× speedup for the same accuracy level
- Small memory and computational overhead

Summary

- An idea: Database Learning
  - learning from past query answers
- An implementation: Verdict
  - Given mean, variances, and covariances
  - Apply maximum entropy principal
  - Find a joint probability distribution function
  - Improve answer and error based on conditioning on snippet answers
  - https://verdictdb.org

Q & A

- Using testable information other than or in addition to mean, variances, covariances?
- Are there any other possible inferential techniques?
- Can we cut out training phase?

UNIVERSITY OF WATERLOO