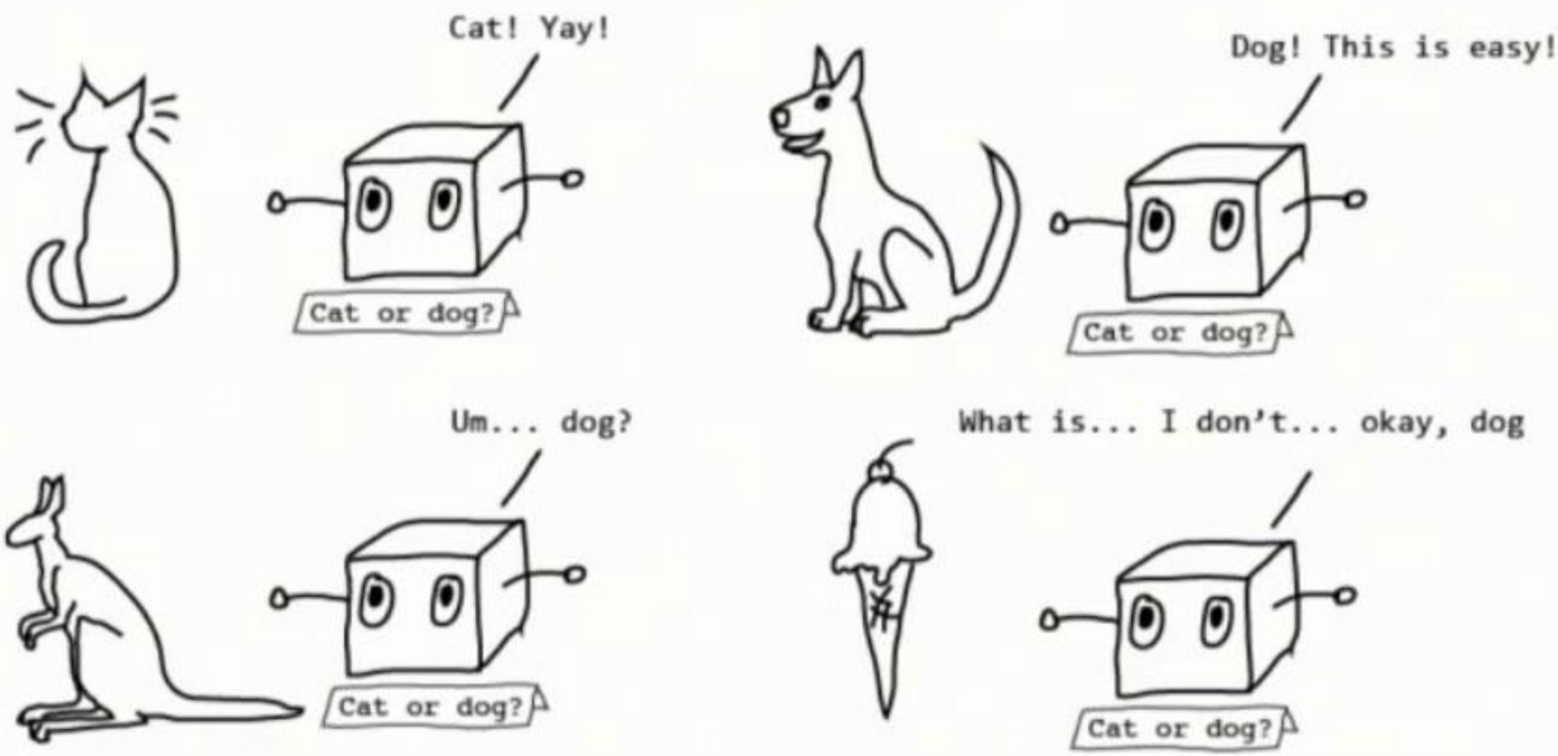
有理论保证的AI4DB算法
以NDV估计为例

魏哲巍
中国人民大学
高瓴人工智能学院

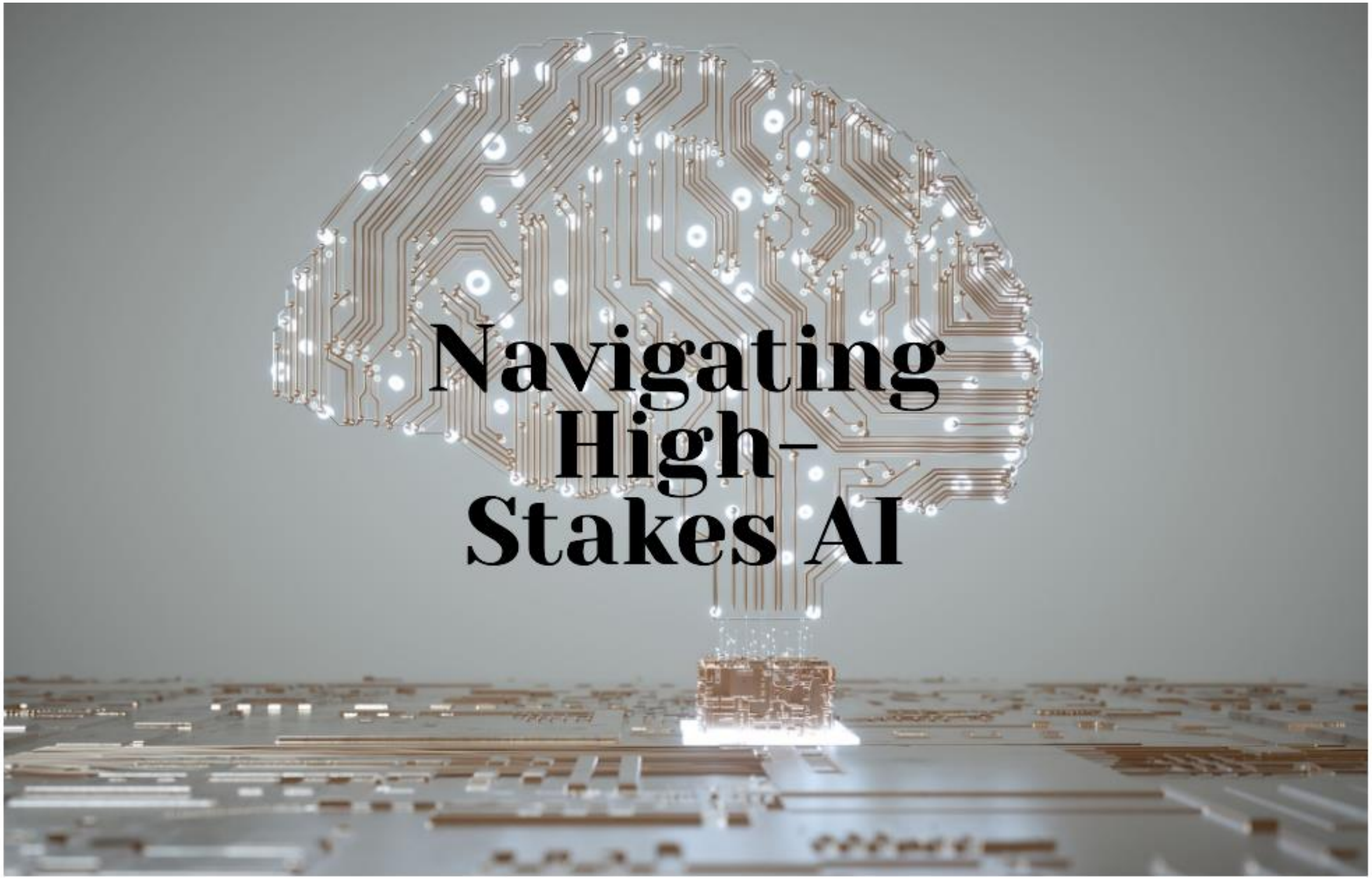
Pitfall of AI4DB

- 现有AI4DB方法往往不能提供理论保证



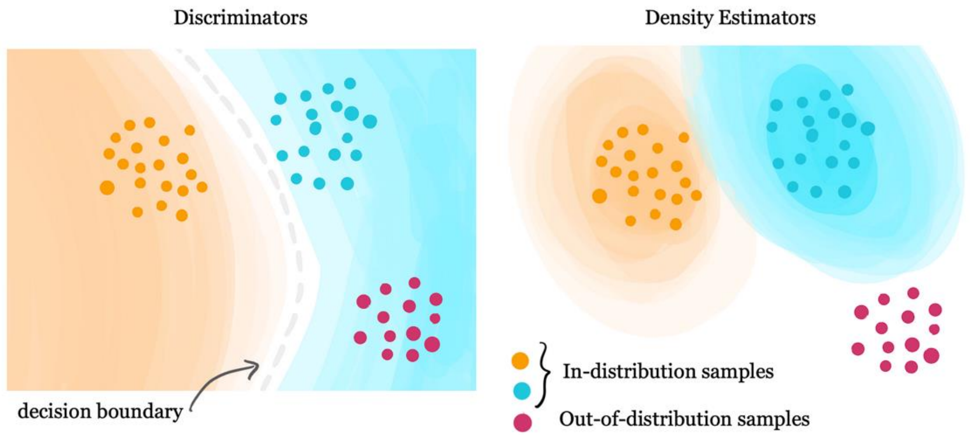
Pitfall of AI4DB

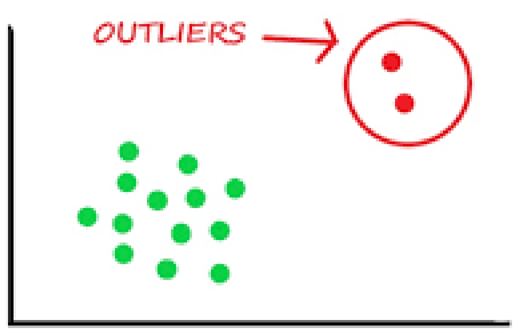
- AI4DB: High Stake AI



DB Meet ML

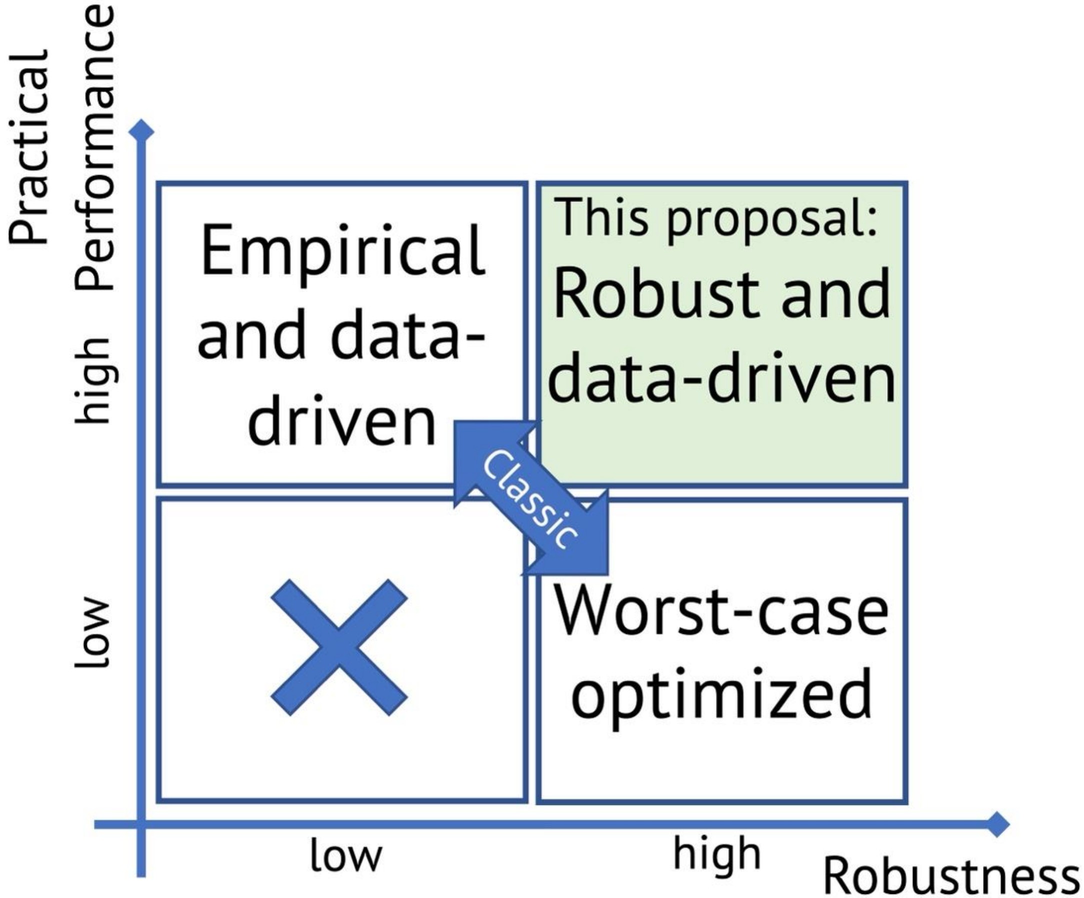
- AI4DB关心/遇到的问题，机器学习领域可能已经研究过
  - 统计量估计 v.s. Estimate Unseen
  - Workload/distribution shift v.s. OOD & Outlier Detection





有理论保证的机器学习算法

- Learning augmented algorithm



Number of Distinct Values (NDV)

元素总数 $N = 12$

Data

1 2 5 4 3 2 1 6 2 3 6 4

- 排序: 1 1 2 2 2 3 3 4 4 5 6 6

$O(N \log N)$

- 去重: 1 2 3 4 5 6
- 不同元素个数(NDV)： $D = 6$
- 频率的频率 $F_i = \sum 1_{\{N_j = i\}}$ ： 刚好出现 i 次的元素个数
  - $F_1 = 1, F_2 = 4, F_3 = 1$
  - NDV： $D = \sum_i F_i = 6$
  - 熵： $H = -\sum_i F_i \cdot p_i \log p_i = 1.75$

NDV 的研究与应用

- 查询优化[1,2]
  - Cardinality Estimation: 分析每列不同元素个数
  - Cost Estimation: 生成不同查询计划
- 数据库压缩[3]
  - 智能选择列压缩顺序
- 统计机器学习[4,5]
  - 估计离散分布支撑集大小

[1] Hilprecht, B., Schmidt, A., Kulessa, M., Molina, A., Kersting, K., & Binnig, C. (2019). Deepdb: Learn from data, not from queries!. arXiv preprint arXiv:1909.00607.
[2] Zhu, R., Wu, Z., Chai, C., Pfadler, A., Ding, B., Li, G., & Zhou, J. (2022). Learned Query Optimizer: At the Forefront of AI-Driven Databases. In EDBT (pp. 1-4).
[3] Lemire, D., & Kaser, O. (2011). Reordering columns for smaller indexes. Information Sciences, 181(12), 2550-2570.
[4] Wu, Y., & Yang, P. (2019). Chebyshev polynomials, moment matching, and optimal estimation of the unseen. The Annals of Statistics, 47(2), 857-883.
[5] Acharya, J., Das, H., Orlitsky, A., & Suresh, A. T. (2017, July). A unified maximum likelihood approach for estimating symmetric properties of discrete distributions. In International Conference on Machine Learning (pp. 11-21). PMLR.

NDV 的研究与应用

Calibrated Language Models Must Hallucinate

Adam Tauman Kalai* — OpenAI
Santosh S. Vempala† — Georgia Tech

March 21, 2024

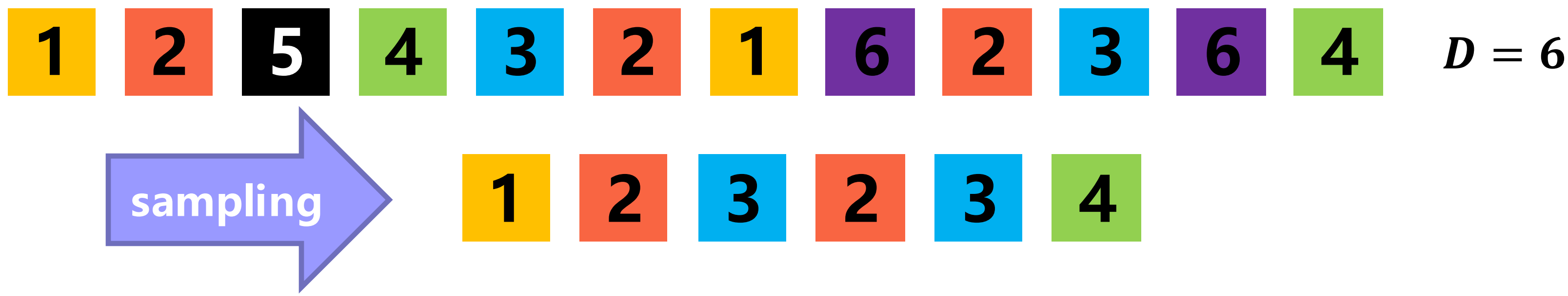
Abstract

Recent language models generate false but plausible-sounding text with surprising frequency. Such "hallucinations" are an obstacle to the usability of language-based AI systems and can harm people who rely upon their outputs. This work shows that there is an inherent statistical lower-bound on the rate that pretrained language models hallucinate certain types of facts, having nothing to do with the transformer LM architecture or data quality. For "arbitrary" facts whose veracity cannot be determined from the training data, we show that hallucinations must occur at a certain rate for language models that satisfy a statistical calibration condition appropriate for generative language models. Specifically, if the maximum probability of any fact is bounded, we show that the probability of generating a hallucination is close to the fraction of facts that occur exactly once in the training data (a "Good-Turing" estimate), even assuming ideal training data without errors.

One conclusion is that models pretrained to be sufficiently good *predictors* (i.e., calibrated) may require post-training to mitigate hallucinations on the type of arbitrary facts that tend to appear once in the training set. However, our analysis also suggests that there is no statistical reason that pretraining will lead to hallucination on facts that tend to appear more than once in the training data (like references to publications such as articles and books, whose hallucinations have been particularly notable and problematic) or on systematic facts (like arithmetic calculations). Therefore, different architectures and learning algorithms may mitigate these latter types of hallucinations.

Kalai, A. T., & Vempala, S. S. (2024, June). Calibrated language models must hallucinate. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing* (pp. 160-171). [STOC 2024]

基于采样估计NDV



- 样本频率的频率 f_i: 样本中出现i次的元素个数
  - $f_1 = 2, f_2 = 2$
  - 样本NDV $d = \sum_i f_i = 4$
- NDV 估计器:
  - Plug-in: $\hat{D} = d = \sum_i f_i = 4$ ➡ **永远低估!** 原始数据NDV $D = 6$
  - Chao: $\hat{D}_{Chao} = d + \frac{f_1^2}{2f_2} = 4 + \frac{2^2}{2*2} \approx 5$ → **Estimate Unseen**

$\hat{D}_{GT} = d + \sum_{j=1}(-1)^{j+1} t^j f_j$

$\hat{D}_{WY} = \sum_{j=1}^{L} g_L(j) f_j + \sum_{j>L} f_j$

……

基于采样的NDV估计历史

~1990年

传统统计量
- 49 Goodman
- 53 Good-Toulmin
- 81 Shlosser
- ……

1990年~2011年

NDV广泛应用到实践
- 数据库
- ……

针对分布假设设计统计量
- 91 Chao
- 00 GEE
- ……

2011 ~ 2014年

Gregory Valiant& Paul Valiant
引入线性方程以最优化方法求解

2019年

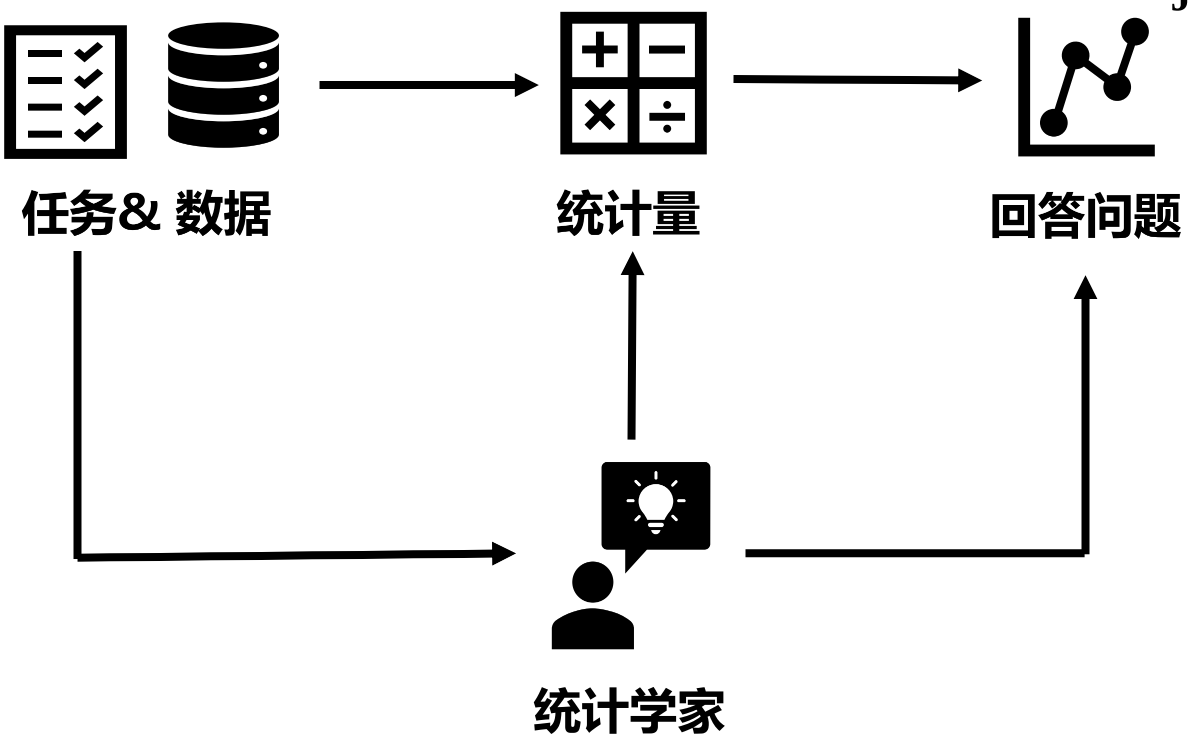
Profile Maximum Likelihood

Yihong Wu & Pengkun Yang
以Chebyshev Polynomial为切入

2021年

VLDB 2022
Renzhi Wu
基于学习的NDV估计

Learning-based Property Estimation with Polynomials

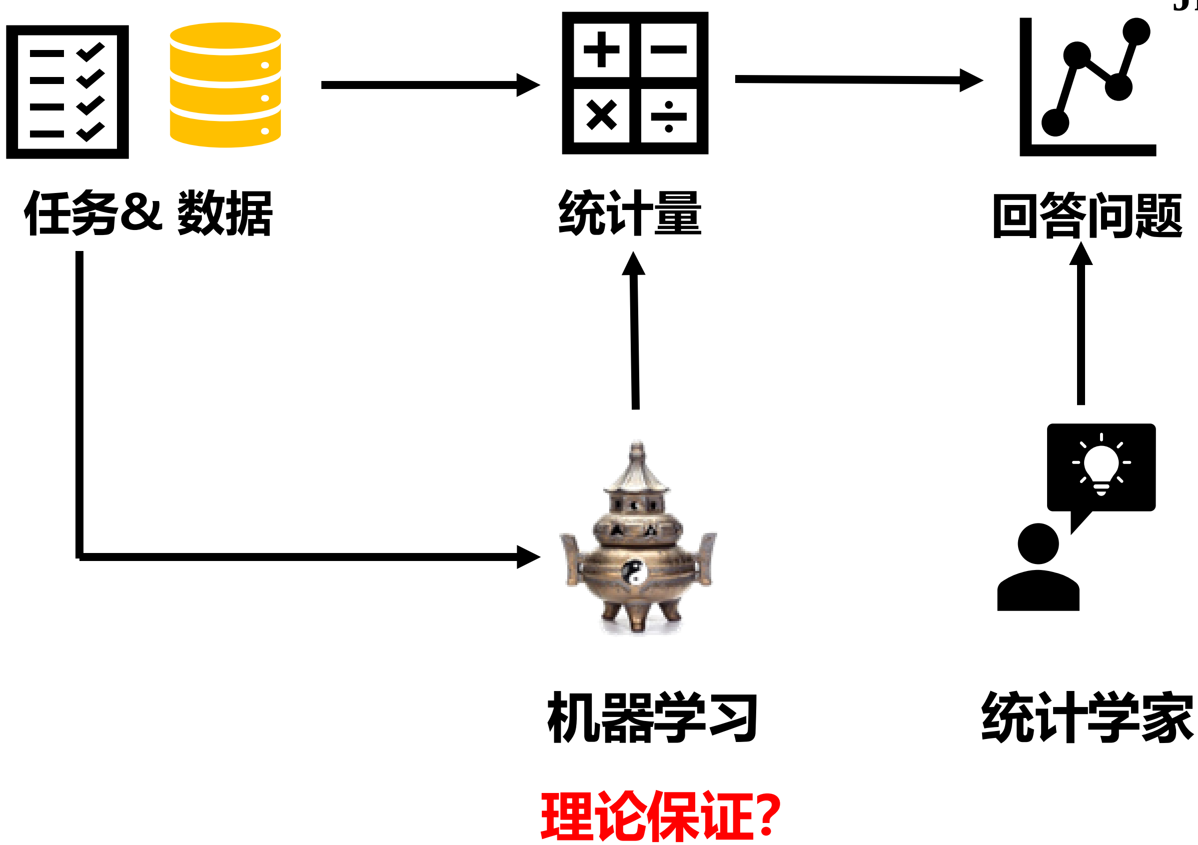


Jiajun Li[1,2], Runlin Lei[1], Sibo Wang[3], Zhewei Wei*[1], Bolin Ding[2]

[1]Renmin University of China
[2]Alibaba Group
[3] The Chinese University of Hong Kong

[SIGMOD 2024]

Learning-based Property Estimation with Polynomials



Jiajun Li[1,2], Runlin Lei[1], Sibo Wang[3], Zhewei Wei*[1], Bolin Ding[2]
[1]Renmin University of China
[2]Alibaba Group
[3] The Chinese University of Hong Kong

[SIGMOD 2024]

Learning-based Property Estimation with Polynomials

不同元素个数估计

$$D = \sum_{j=1} 1_{F_j} \cdot F_j$$

推广

性质估计

$$\Psi = \sum_{j=1} \psi\left(\frac{j}{N}\right) F_j$$

Jiajun Li[1,2], Runlin Lei[1], Sibo Wang[3], Zhewei Wei*[1], Bolin Ding[2]

[1]Renmin University of China
[2]Alibaba Group
[3] The Chinese University of Hong Kong

[SIGMOD 2024]

$$D = \sum_{j=1} 1_{F_j} \cdot F_j$$

NDV

$$H = \sum_{j=1} \frac{j}{N} \log \frac{N}{j} \cdot F_j$$

Entropy

$$PS = \sum_{j=1} \left(\frac{j}{N}\right)^{\alpha} \cdot F_j$$

α-power sum

是否存在一个统一的的可学习框架?

设计可学习的估计器

- 定义线性估计器

$$\widehat{\Psi} = \sum_{t=1}^{L} b_t f_t + \sum_{t=L+1} f_t$$

$$\begin{cases} \widehat{D}_{plug-in} = d \\ \widehat{D}_{GEE} = d + f_1\sqrt{N/n - 1} \\ \widehat{D}_{GT} = d + \sum_{j=1}(-1)^{j+1} t^j f_j \\ \widehat{D}_{WY} = \sum_{j=1}^{L} g_L(j) f_j + \sum_{j>L} f_j \\ \ldots\ldots \end{cases}$$

低频部分

高频部分

当t比较小时，

将b_t看作一组可学习的参数，寻找f_t与真实分布的联系

当 t 足够大时，

$$\frac{t}{n} \rightarrow \mathrm{Pr}[\text{被采样的概率}]$$

关于properties 的无偏估计

理论保证

- Lower bound [PODS2000]:

Case1: 1, 1, 1, … …1, 1, 1

Case2: 1, 1, 1, … 1, 2,3,…k

若未被采样
则难以区分两种case

Ratio Error：

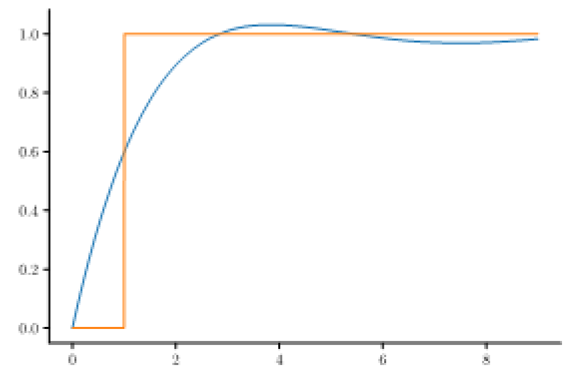
对于任何估计器，从N行数据中采样n列，对于任意的$\gamma > e^{-n}$，都存在一组数据使得以至少γ的概率，有

$$\textbf{Ratio Error} \geq \sqrt{\frac{N-n}{2n}\ln\frac{1}{\gamma}}.$$

- 切比雪夫多项式与最优采样数

[The Annals of Statistics 2019]:

$$\epsilon_D = \sum_{j=1}\left[\left(\sum_{t=1}^{L} Poly(N,n,j,t)b_t - 1\right)F_j\left(1-\frac{j}{N}\right)^n\right]$$



$$\text{n} = O\left(\frac{N}{\log N}\log^2\frac{1}{\epsilon}\right)$$

Charikar, M., Chaudhuri, S., Motwani, R., & Narasayya, V. (2000, May). Towards estimation error guarantees for distinct values. In *Proceedings of the nineteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems* (pp. 268-279).

Wu, Y., & Yang, P. (2019). Chebyshev polynomials, moment matching, and optimal estimation of the unseen. *The Annals of Statistics*, *47*(2), 857-883.

如何让可学习估计器保持最优采样数?

- 通过权重Chebyshev多项式插值近似学习F_j

$$\epsilon_\psi = \sum_{j=1}\left[\left(\sum_{t=1}^{L} Poly(N,n,j,t)b_t - 1\right)F_j\left(1-\frac{j}{N}\right)^n\right]$$

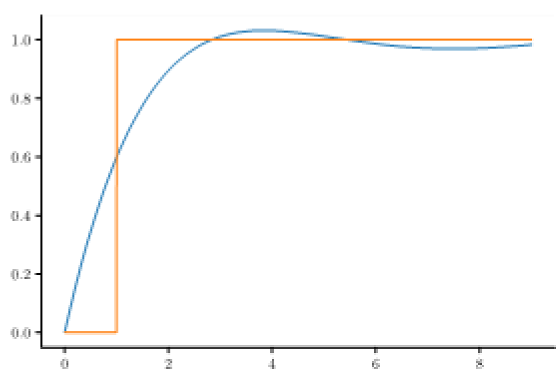
learning

Learning-based NDV Estimation

$$\epsilon_\psi = \sum_{j=1}\left[\left(\sum_{t=1}^{L} Poly(N,n,j,t)Net(f_j) - 1\right)w_j\right]$$

将系数转化为与f_j相关的可学习网络

从任意多项式插值变为权重多项式插值

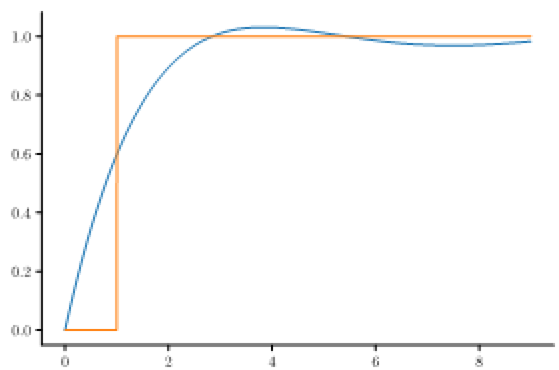


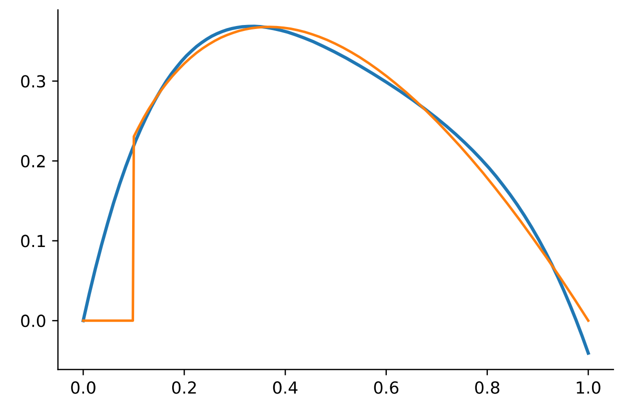
如何从NDV推广到其他性质估计?

$$D = \sum_{j=1} 1_{F_j} \cdot F_j$$

$$\epsilon_D = \sum_{j=1}\left[\left(\sum_{t=1}^{L} Poly(N,n,j,t)b_t - 1\right)F_j\left(1-\frac{j}{N}\right)^n\right]$$

NDV





$$H = \sum_{j=1} \frac{j}{N}\log\frac{N}{j} \cdot F_j$$

$$\epsilon_H = \sum_{j=1}\left[\left(\sum_{t=1}^{L} Poly(N,n,j,t)b_t - \frac{j}{N}\log\frac{N}{j}\right)F_j\left(1-\frac{j}{N}\right)^n\right]$$

Entropy

如何从NDV推广到其他性质估计?

$$D = \sum_{j=1} 1_{F_j} \cdot F_j \qquad \epsilon_D = \sum_{j=1} \left[\left(\sum_{t=1}^{L} Poly(N,n,j,t) b_t - 1 \right) F_j \left(1 - \frac{j}{N}\right)^n \right]$$

NDV

$$H = \sum_{j=1} \frac{j}{N} \log \frac{N}{j} \cdot F_j \qquad \epsilon_H = \sum_{j=1} \left[\left(\sum_{t=1}^{L} Poly(N,n,j,t) b_t - \frac{j}{N} \log \frac{N}{j} \right) F_j \left(1 - \frac{j}{N}\right)^n \right]$$

Entropy

$$PS = \sum_{j=1} \left(\frac{j}{N}\right)^{\alpha} \cdot F_j \qquad \epsilon_{PS} = \sum_{j=1} \left[\left(\sum_{t=1}^{L} Poly(N,n,j,t) b_t - \left(\frac{j}{N}\right)^{\alpha} \right) F_j \left(1 - \frac{j}{N}\right)^n \right]$$

α − Power Sum

$$\Psi = \sum_{j=1} \psi\left(\frac{j}{N}\right) F_j \qquad \epsilon_{\Psi} = \sum_{j=1} \left[\left(\sum_{t=1}^{L} Poly(N,n,j,t) b_t - \psi\left(\frac{j}{N}\right) \right) F_j \left(1 - \frac{j}{N}\right)^n \right]$$

实验

- 效果 (NDV: Ratio Error, Entropy: Absolute Error)

Table 4: The performance of different NDV estimators (Ratio Error).

Methods	Kasandr				Airline				SSB				NCVR				Average
	0.001	0.005	0.01	Time(s)	0.001	0.005	0.01	Time(s)	0.001	0.005	0.01	Time(s)	0.001	0.005	0.01	Time(s)	
GEE	2.455	1.480	**1.335**	1.0	2.754	1.388	1.205	0.3	2.770	1.825	1.578	2.3	5.589	2.385	1.906	4.4	2.223
Chao	3.828	2.219	1.855	0.9	1.452	1.238	1.195	0.3	**1.069**	**1.053**	**1.046**	2.2	11.450	3.983	7.640	4.2	3.169
WY	4.143	1.642	1.370	8.4	**1.269**	1.345	1.323	3.0	4.019	1.538	1.268	20.5	8.641	2.774	2.401	37.6	2.645
GT	30.515	7.768	4.672	2.4	1.604	1.328	1.262	0.7	35.945	7.866	4.360	5.8	67.466	15.980	9.106	9.7	15.656
Shlosser	7.618	4.348	3.321	48.0	5.524	1.155	**1.074**	12.7	25.570	8.335	5.461	118.4	14.555	1.608	**1.274**	187.5	6.654
AE	33.231	7.494	4.427	109.8	1.293	1.156	1.133	12.2	39.452	8.575	4.710	295.8	59.450	12.617	6.979	221.8	15.043
WD	2.342	1.883	1.730	0.2	1.608	1.249	1.279	0.2	1.574	1.478	1.293	0.4	4.125	1.984	1.745	1.8	1.857
Ours	**2.085**	**1.297**	1.395	3.0	1.343	**1.102**	1.084	2.9	2.447	1.646	1.781	6.7	**2.796**	**1.478**	1.310	25.3	**1.647**

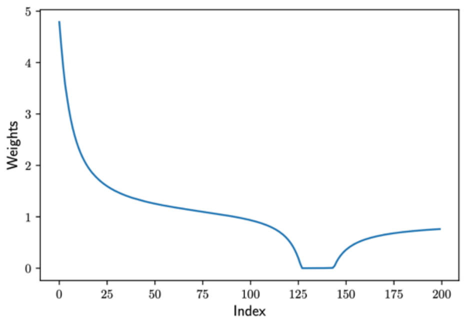
Table 5: The performance of different entropy estimators (Absolute Error).

Methods	Kasandr				Airline				SSB				NCVR				Average
	0.001	0.005	0.01	Time(s)	0.001	0.005	0.01	Time(s)	0.001	0.005	0.01	Time(s)	0.001	0.005	0.01	Time(s)	
Plug-in	1.151	0.651	0.475	0.046	0.025	0.007	0.004	0.077	1.502	0.901	0.679	0.033	0.529	0.358	0.301	0.315	0.549
MM	0.972	0.505	0.346	0.045	**0.008**	**0.003**	**0.002**	0.077	1.293	0.723	0.518	0.031	0.463	0.314	0.261	0.307	0.451
WY	19.040	3.774	1.887	0.108	20.467	4.087	2.044	0.169	17.266	3.367	1.678	0.178	21.782	4.220	2.068	0.836	8.473
Ours	**0.499**	**0.250**	**0.204**	2.589	0.025	0.007	0.004	1.971	**0.191**	**0.045**	**0.037**	6.355	**0.268**	**0.177**	**0.173**	17.115	**0.157**

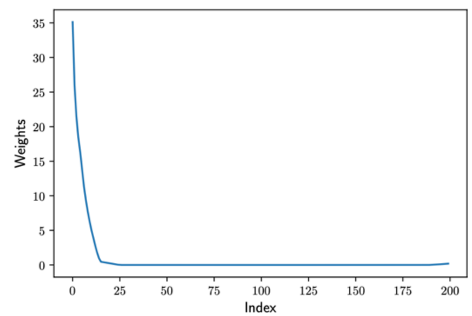
- 训练时间
  - 6000 s (Learn to be a statistician) → 300 s (Ours)

实验

- 不同训练数据下学习到的权重参数



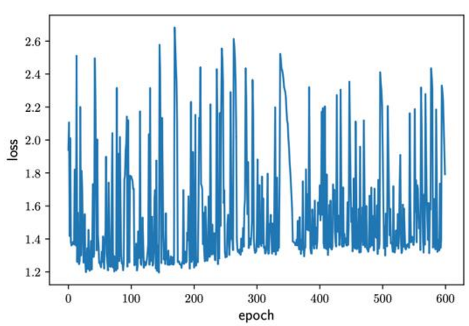
(a) Weights of $F_j = \frac{N}{j}, N = 100000, j \sim Uniform(50, 55)$.



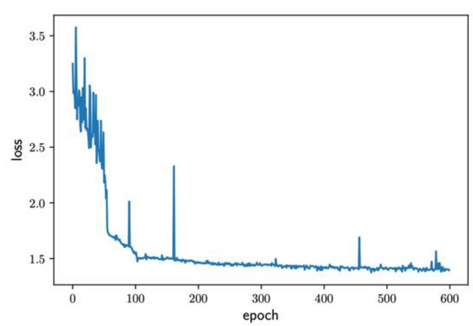
(b) Weights of our final model

符合$\psi\left(\frac{j}{N}\right)F_j\left(1-\frac{j}{N}\right)^n$

- 引入多项式近似，才能使模型收敛



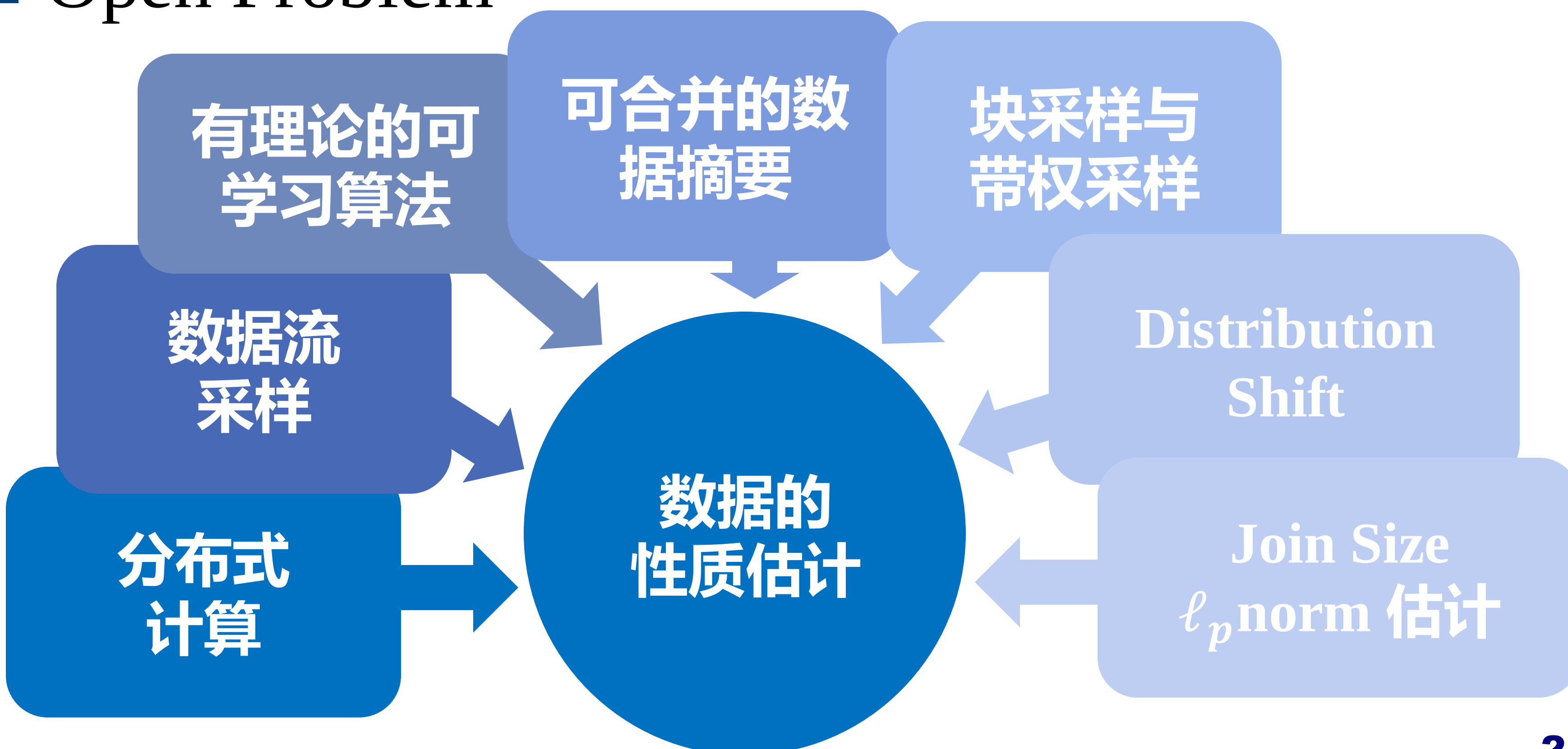
(a) Training loss without polynomial approximation.



(b) Training loss with polynomial approximation.

总结与展望

- 做有理论保证的AI4DB算法
  - 最优时间/采样/通讯复杂度/误差界、泛化界
- Open Problem



主要研究成员和合作者

■ 主要研究成员

Zhewei Wei

Jiajun Li

Runlin Lei

■ 合作者

Bolin Ding

Renzhi Wu

Sibo Wang

Thank you!
Q&A