


基于学习的基数估计方法的可用性分析

- 2021.10.08



目录

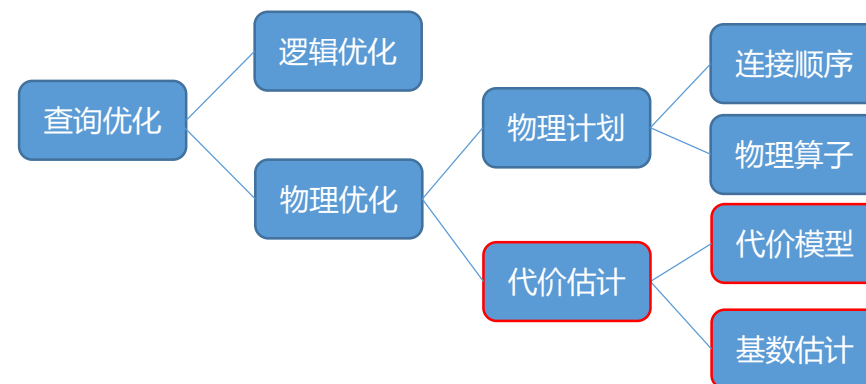
- 
- A horizontal bar with a blue segment on the left and an orange segment on the right.
- 背景与研究进展
 - 现有方法介绍
 - 静态环境中学习方法的可用性分析
 - 动态环境中学习方法的可用性分析
 - 学习方法什么时候出错?
 - 研究方向与机会

基数估计问题描述

什么是基数 (Cardinality) ?

- 对关系/中间结果：关系的行 (tuples) 数 (可重复)

```
SELECT COUNT(*) FROM R
WHERE 0 ≤ A1 ≤ 20 AND 20 ≤ A2 ≤ 100
```



基数估计 (Cardinality Estimation) 在查询优化中的位置?

- 代价估计 = 基数估计 + 代价模型

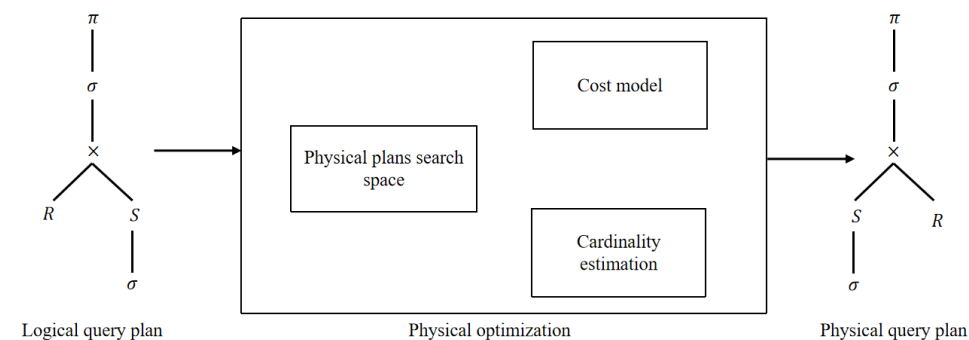
- 代价模型

总代价 = IO代价 + CPU代价

$$\text{cost} = P * a_page_cpu_time + T * W$$

预计访问页面数 每个页面读取时间 访问元组数 权重因子

- 对物理执行计划或其子计划 (中间结果) 基数 (行数) 的估计。





基数估计方法总览

传统方法

- **Histogram**
- **MHIST(multi-dimensional histogram)**
- **Sample**
- **KDE**
- Sketch
- Bayes
- QuickSel
- ...

学习方法

- 查询语句驱动(Query-Driven)
 - MSCN**
 - LW-XGB/NN**
 - End-to-End
 - DQM-Q
 - ...
- 数据驱动(Data-Driven)
 - Naru**
 - DeepDB**
 - DQM-D
 - NeuroCard
 - ...

目录



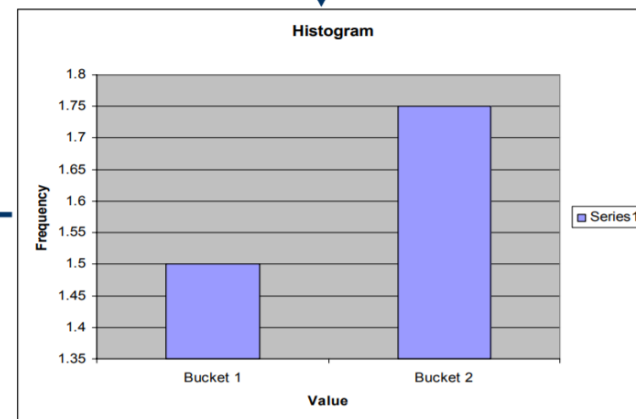
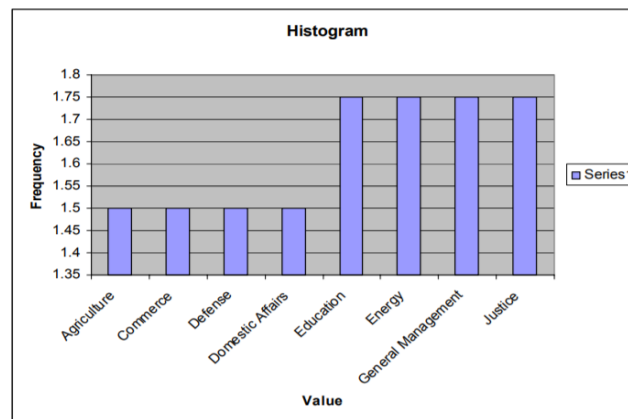
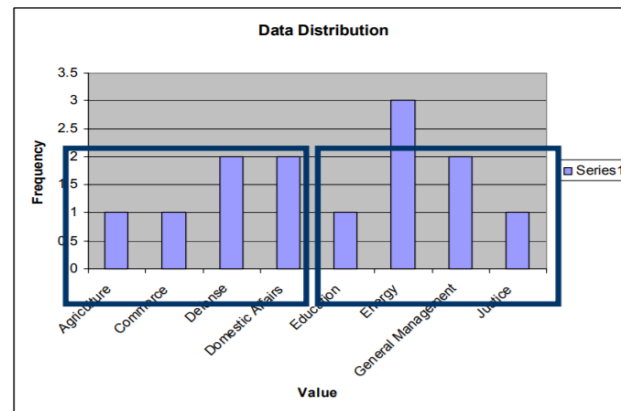
- 背景与研究进展
- 现有方法介绍
- 静态环境中学习方法的可用性分析
- 动态环境中学习方法的可用性分析
- 学习方法什么时候出错?
- 研究方向与机会

传统方法

一维直方图

- 思想：对一个列的采样，将数据按等频率分为不同的Bucket，使用均匀分布来模拟同一个Bucket的值的分布
- 优点
 - 离线统计,几乎不占用运行时开销
 - 占用空间小, 在只涉及单属性查询误差小
 - 被广泛应用于商业DBMSs
- 缺点
 - 多属性查询依赖于独立性假设, 不能抓取属性之间关联性

Department	Histogram H1	
	Frequency in Bucket	Approximate Frequency
Agriculture	1	1.5
Commerce	1	1.5
Defense	2	1.5
Domestic Affairs	2	1.5
Education	①	1.75
Energy	③	1.75
General Management	②	1.75
Justice	①	1.75



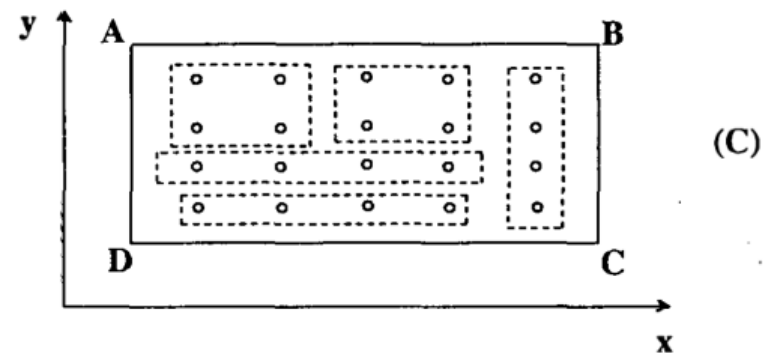
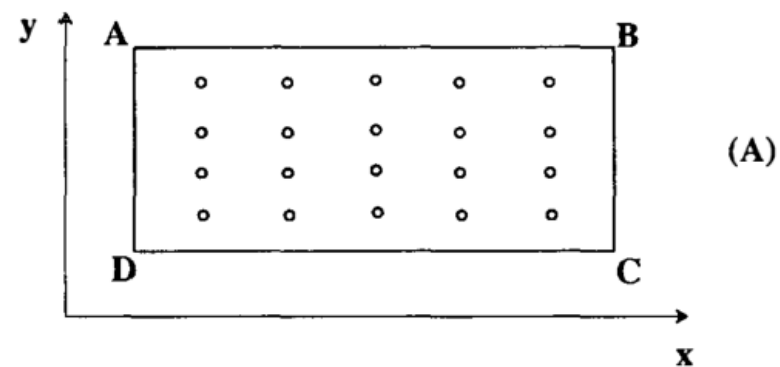
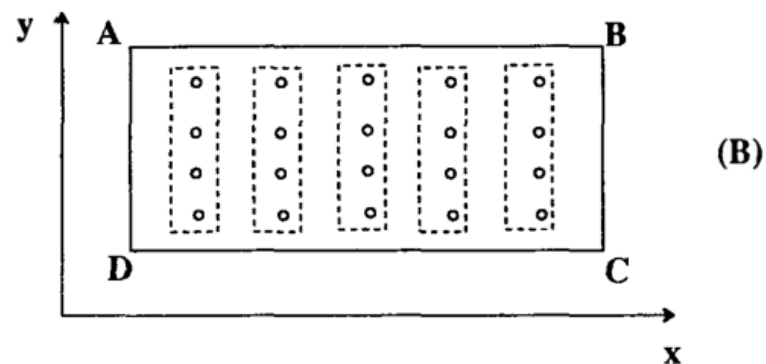
Independence Assumption
 $p(\text{Age, Salary})$ $p(\text{Salary})$



传统方法

多维直方图

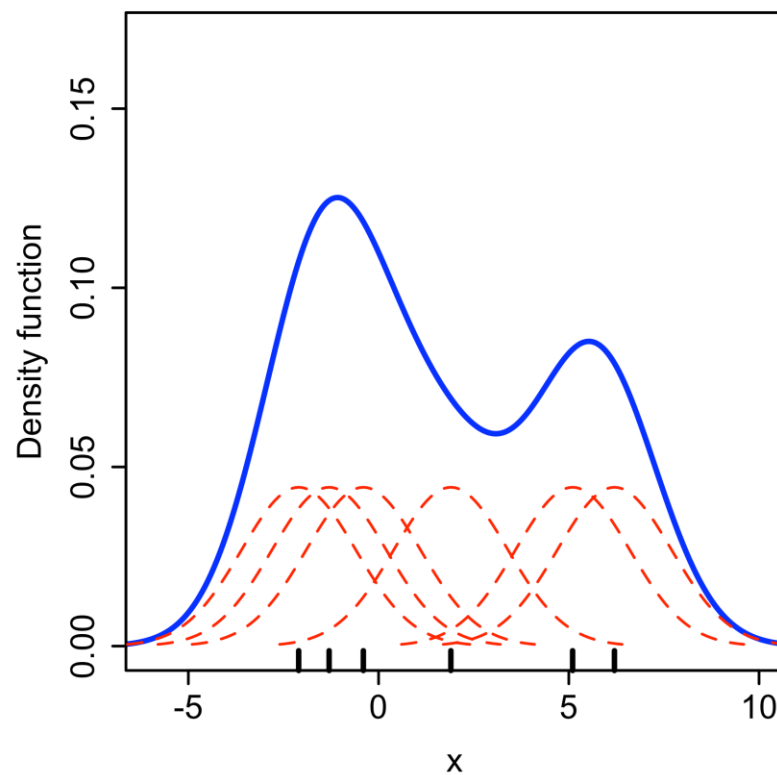
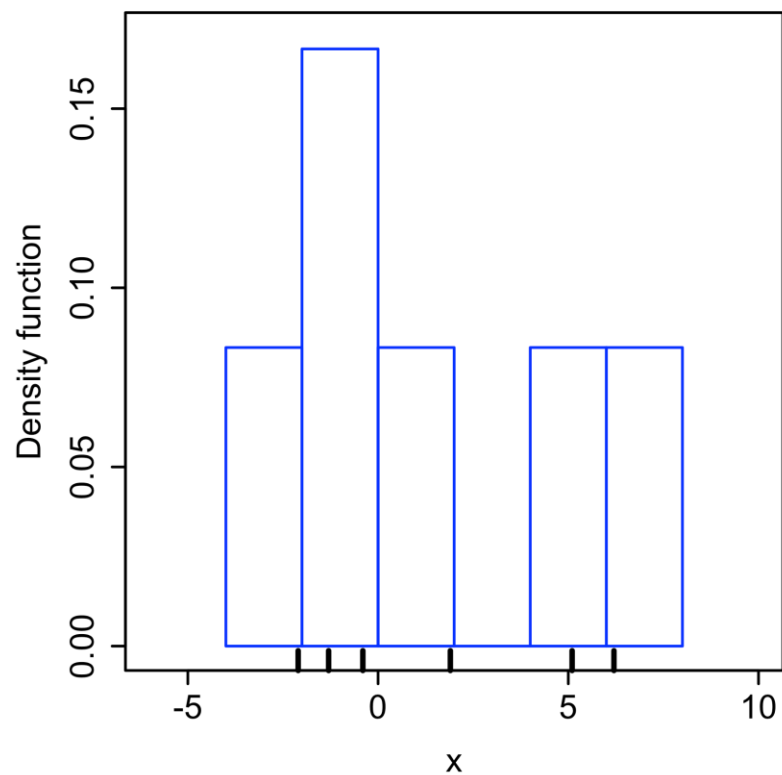
- 思想：按一定的顺序依次对每一个属性切分bucket
- 优点
 - 能够捕捉列于列之间的关联性
- 缺点
 - 不同的列顺序的bucket的质量不同
 - 依靠均匀分布假设



传统方法

一维核密度估计 kernel density estimation (KDE)

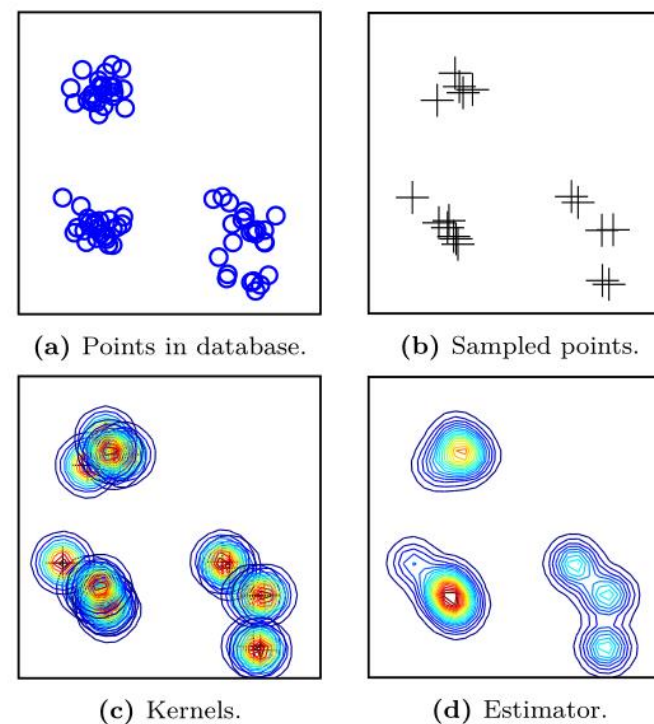
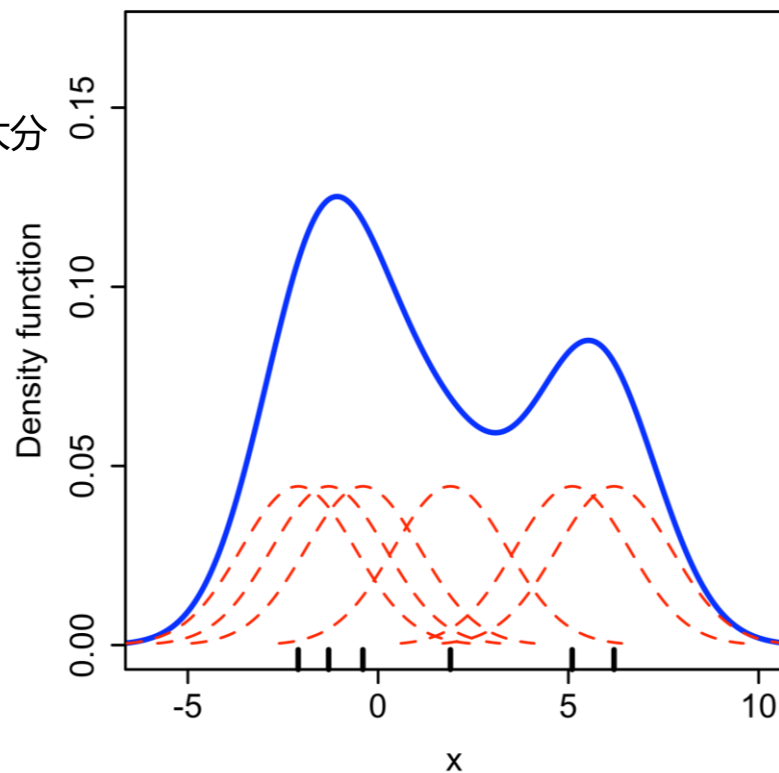
- 思想：类似于直方图，对于一个列的采样，使用Kernal（如正太分布）来模拟周边的值的分布



$$\hat{f}_h(x) = \frac{1}{n} \sum_{i=1}^n K_h(x - x_i) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right)$$

多维核密度估计 (KDE)

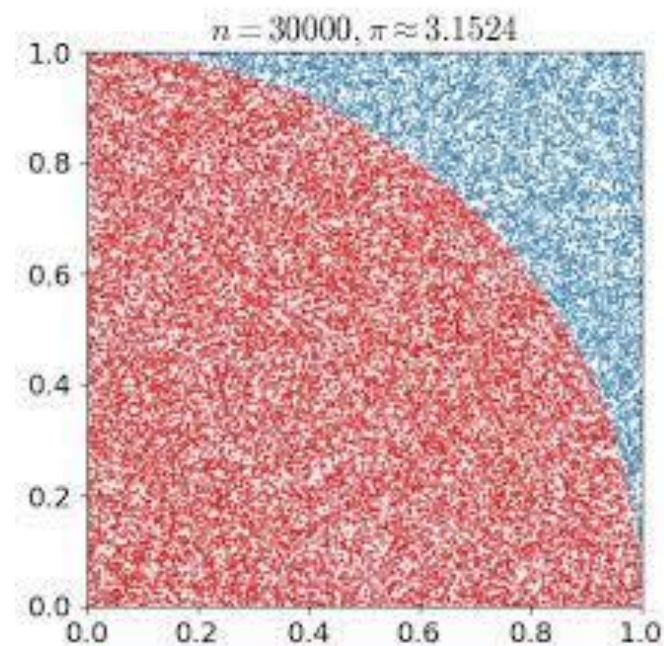
- 思想：使用高维核来模拟分布
- 维度升级
 - 使用多维度的核：如多变量正太分布



$$\hat{f}_h(x) = \frac{1}{n} \sum_{i=1}^n K_h(x - x_i) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right)$$

Sample 采样

- 思想：用部分数据预估整体数据
- 优点
 - 直观
 - 思想被许多其他方法广泛应用（构建直方图、KDE）
- 缺点
 - 运行时开销大
 - 偶然性强
 - 0-tuple, 采样消失问题



MSCN (Query-Driven)

- 思想：把查询语句的重要组成部分特征编码，以真实基数作为label，训练模型。

- Tables (用于查询的表格)
- Join (连接谓词)
- Predicate (选择谓词)
- Bitmaps(采样信息)

$$\text{损失函数} : \text{qerror} = \max\left(\frac{C}{\hat{C}}, \frac{\hat{C}}{C}\right)$$

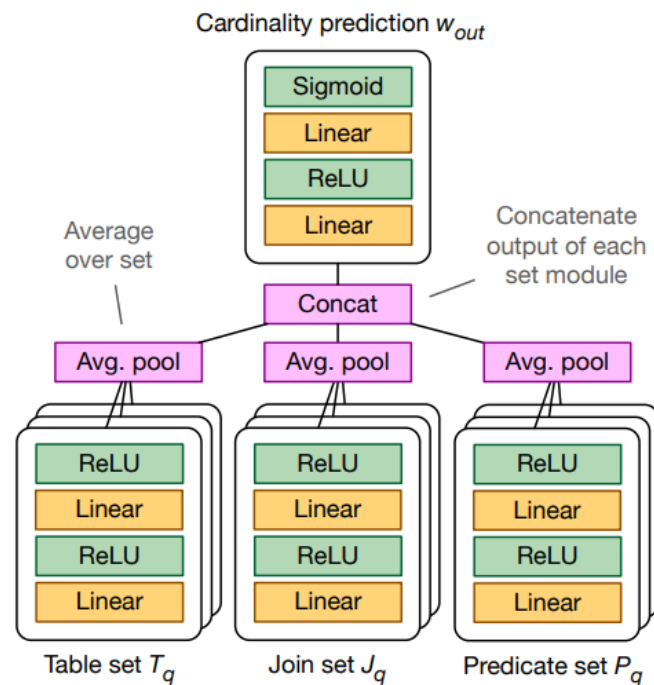
- 优点
 - 简单有效
 - 引导作用
- 缺点
 - 模型限制较大，准确率等方面有较大提升空间

SELECT COUNT(*) FROM title t, movie_companies mc WHERE t.id = mc.movie_id AND t.production_year > 2010 AND mc.company_id = 160

Table set { [0 1 0 1 ... 0], [0 0 1 0 ... 1] } Join set { [0 0 1 0] } Predicate set { [1 0 0 0 0 1 0 0 0.72], [0 0 0 1 0 0 1 0 0.14] }

table id samples join id column id value operator id

Figure 2: Query featurization as sets of feature vectors.



MSCN: multi-set convolutional network



学习方法

LW-XGB/NN

- 思想：递归地将表的概率分布用树的sum和product节点表示，product节点的子节点概率分布相互独立
- 节点类型
 - sum节点：将表格按行切分，概率相加
 - product节点：将表格按列切分，概率相乘（两个属性之间满足独立性假设）

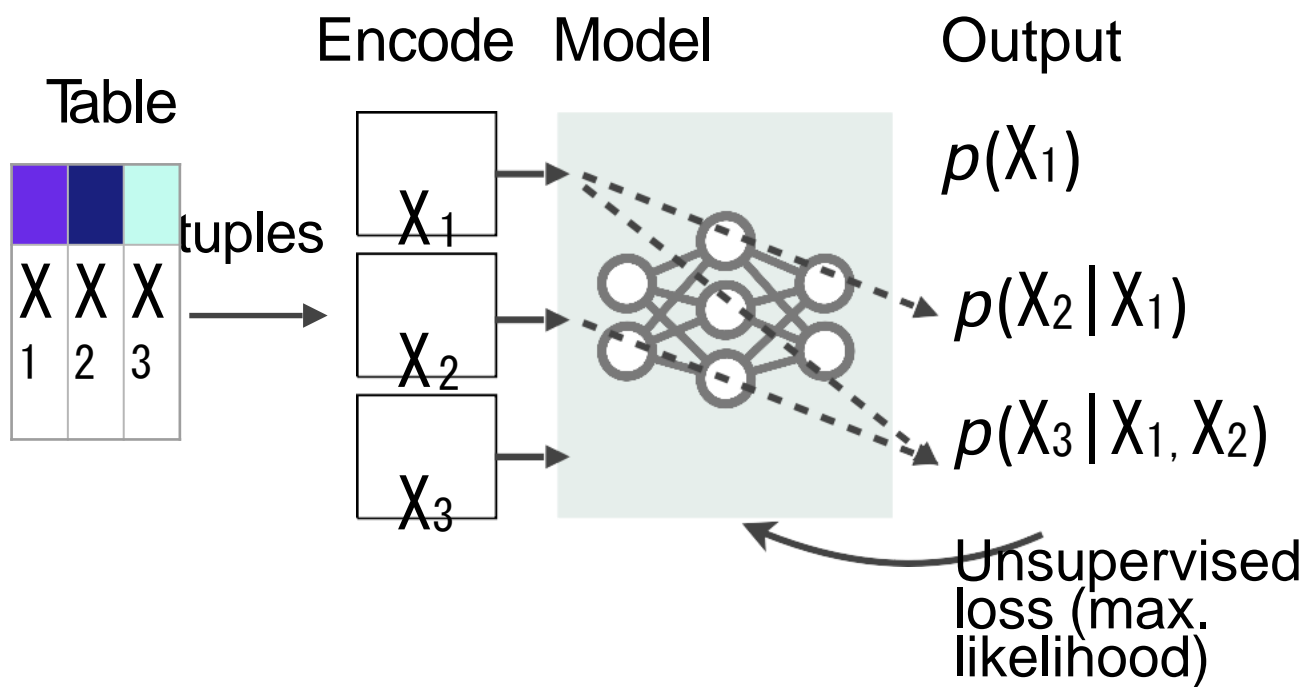
学习方法

Naru(Data-Driven)

- 思想：使用自回归模型（MADE）利用极大似然估计的原则对数据的概率分布进行学习

- 构建流程

- 指定一个表用于构建Naru estimator
- 将这个表的所有tuples按batch传入自回归模型（MADE）
- 以极大似然估计原则进行一次参数更新
- 经过训练后模型能够对任意等值连接的联合概率分布进行估计



$$P(X_1 = x_1, \dots, X_n = x_n) = \hat{P}(X_1 = x_1), \hat{P}(X_2 = x_2 | X_1 = x_1), \dots, \hat{P}(X_n = x_n | X_1 = x_1, \dots, X_{n-1} = x_{n-1})$$

Naru(Data-Driven)

- 等值查询

- 等值查询时, 每个属性的值已经被指定

$$P(X_1 = x_1, \dots, X_n = x_n)$$

- 根据链式法则

$$\hat{P}(X_1 = x_1), \hat{P}(X_2 = x_2 | X_1 = x_1), \dots, \hat{P}(X_n = x_n | X_1 = x_1, \dots, X_{n-1} = x_{n-1})$$

- 范围查询

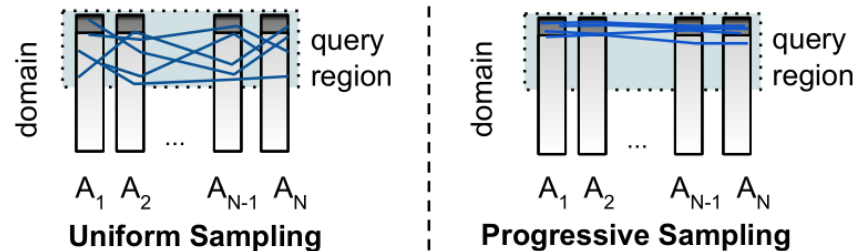
- 范围查询 (小范围)

- 枚举所有可能性

$$\text{sel}(X_1 \in R_1, \dots, X_n \in R_n) \approx \sum_{x_1 \in R_1} \dots \sum_{x_n \in R_n} \hat{P}(x_1, \dots, x_n)$$

- 范围查询 (范围比较大时)

- 采用采样(progressive sampling)的方式, 需要依次进行多次采样, 会带来也测时间开销





学习方法

Naru(Data-Driven)

- 优点
 - 基于数据，准确率高
- 缺点
 - 模型较重，基于大数据集时模型训练时间长，更新时间长。

概率图模型-Sum-Product Networks (DeepDB,Data-Driven)

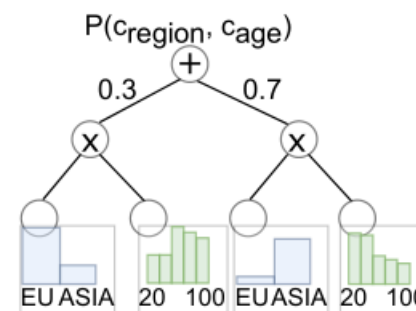
- 思想：递归地将表的概率分布用树的sum和product节点表示，product节点的子节点概率分布相互独立
- 节点类型
 - sum节点：将表格按行切分，概率相加
 - product节点：将表格按列切分，概率相乘（两个属性之间满足独立性假设）
 - 最底层使用直方图
- 优点
 - 即使是数据驱动，因为底层使用的是直方图，训练开销不大
 - 基于数据准确率高
- 缺点
 - 1

c_id	c_age	c_region
1	80	EU
2	70	EU
3	60	ASIA
4	20	EU
...
998	20	ASIA
998	25	EU
999	30	ASIA
1000	70	ASIA

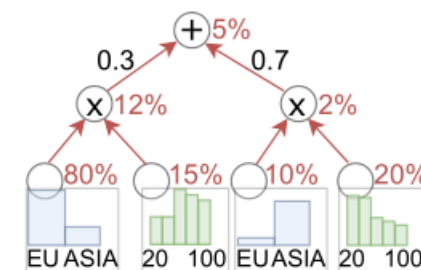
(a) Example Table

c_age	c_region
80	EU
70	EU
60	ASIA
20	EU
...	...
20	ASIA
25	EU
30	ASIA
70	ASIA

(b) Learning with Row/Column Clustering



(c) Resulting SPN



(d) Probability of European Customers younger than 30

Figure 2: Customer Table and corresponding SPN.

学习方法

现有实验的局限

- 学习方法之间没有直接对比
 - 出现年份较为接近
- 没有统一的数据集(dataset)和工作负载(workload)
- 现有的工作大多集中在静态的环境
 - 没有数据更新
- 现有的工作缺少对模型个阶段时间开销的研究
 - 训练时间(training time)
 - 预测时间(inference time)
 - 更新时间(update time)
- 现有的工作缺少对影响模型准确率因素的研究

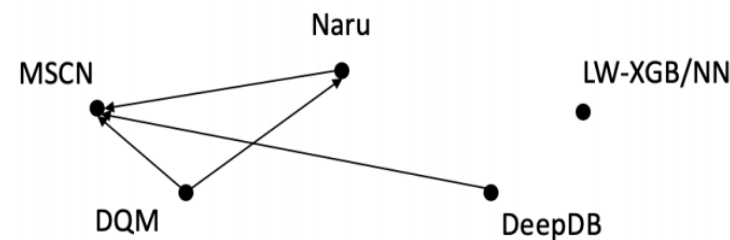


Figure 2: Comparison results available in existing studies.



综合研究 Are We Ready For Learned Cardinality Estimation? [Experiment, Analysis & Benchmark]

- **单表多属性**
- 相同的数据集和工作负载
- 静态环境下的可用性
- 动态环境下的可用性
- 并分析了学习方法在什么情况下会出现较大的误差
- 下一步的研究方向与机会

目录



- 背景与研究进展
- 现有方法介绍
- 静态环境中学习方法的可用性分析
- 动态环境中学习方法的可用性分析
- 学习方法什么时候出错?
- 研究方向与机会

静态环境中学习方法的可用性分析

实验设置

- dataset
 - 真实世界数据集
 - 大小不同，列与数据分布不同
 - 每个数据集被用于评价至少一个学习方法
- workload
 - 同一个实验中使用相同的工作负载
 - 支持1~|列数量|的谓词数量
 - 同时支持等值和范围谓词
 - OOD -> 不同列之间的谓词生成支持独立性

Table 3: Dataset characteristics. “Cols/Cat” means the number of columns and categorical columns; “Domain” is the product of the number of distinct values for each column.

Dataset	Size(MB)	Rows	Cols/Cat	Domain
Census [15]	4.8	49K	13/8	10^{16}
Forest [15]	44.3	581K	10/0	10^{27}
Power [15]	110.8	2.1M	7/0	10^{17}
DMV [59]	972.8	11.6M	11/10	10^{15}

Table 2: Workload used in existing experimental studies.

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

[illegible]

静态环境中学习方法的可用性分析

训练时间

- 学习方法
 - 训练模型的时间
- 数据库系统
 - 运行统计信息收集命令的时间
- 观察
 - 数据库系统因为通过采样的方式收集统计信息，能在几秒内完成训练
 - 学习方法中最快的是使用gradient boosted tree的LW-XGB
 - 使用了神经网络的模型(MSCN, LW-NN)训练时间与数据集大小无关，与训练数据量有关，30-100minute
 - 其中MSCN因为使用了if-else语句导致GPU和CPU耗时差不多
 - Data-Driven的DeepDB和Naru训练时间与数据集大小有关，小数据集上几分钟，大数据集上甚至需要几个小时
 - DeepDB因为使用的是SPN网络，其综合速度居学习方法第二

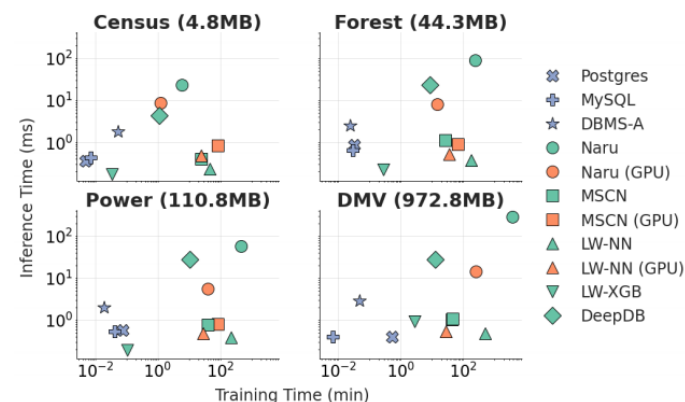


Figure 6: Training and inference time comparison between learned methods and real database system (MSCN's CPU and GPU results on DMV are overlapped).

方法	数据集	预测时间量级
数据库系统（采样）	无关	10s内
LW-XGB	小数据集 (Census)	10s内
LW-NN (GPU)	无关	30minute
MSCN (GPU, CPU)	无关	100minute
DeepDB	小数据集 (Census)	10minute
DeepDB	大数据集 (DMV)	20minute
Naru (GPU)	小数据集 (Census)	1minute
Naru (GPU)	大数据集 (DMV)	4hour

静态环境中学习方法的可用性分析

预测时间

- 学习方法
 - 模型调用时间
- 数据库系统
 - EXPLAIN获取执行计划的时间
- 观察
 - 数据库系统能够在1-2ms内完成基数估计（此时间还包含解析等其他操作）
 - Query-Driven（MSCN,LW_XGB/NN）预测时间与DMBS接近
 - Data-Driven（DeepDB）因为其SPN模型导致预测时间较大
 - Data-Driven（Naru）因为采样需要依次进行，导致预测时间较大
 - 查询优化过程中，预测模型会被多次调用，时间差距会被放大

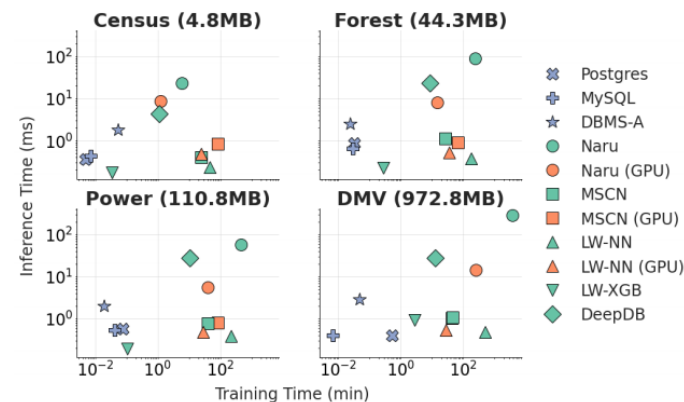


Figure 6: Training and inference time comparison between learned methods and real database system (MSCN's CPU and GPU results on DMV are overlapped).

方法	数据集	预测时间量级
数据库系统	无关	1-2 ms
MSCN,LW-XGB/NN	无关	1-2ms
DeepDB	小数据集 (Census)	5ms
DeepDB	大数据集	25ms
Naru	无关	5ms-15ms



静态环境中学习方法的可用性分析

Hyper-parameter Tuning

- 超参数优化是学习方法的另一个代价
 - 上述对比中，都是已经经过超参数优化的模型
 - 如果没有超参数优化，同一个模型在准确率上的表现会相差较多
 - 如果没有超参数优化，同一个模型在训练时间上的表现会相差较多
 - 超参数优化的时间开销在生产应用者也需要被考虑到



静态环境中学习方法的可用性分析

主要结论

- 学习方法总体上比传统方法估计准确率高，其中Naru表现最为Robust。
- 在训练时间方面，学习方法比DBMS要久许多，LW-XGB除外。
- Query-Driven方法（MSCN和LW-XGB/NN）在预测时间上可以与现有数据库系统竞争，然而Data-Driven的方法，直接根据数据对联合分布的方法(Naru和DeepDB)需要更长的时间。
- 超参数优化是采用神经网络估计器时不可忽视的额外代价。

目录



- 背景与研究进展
- 现有方法介绍
- 静态环境中学习方法的可用性分析
- 动态环境中学习方法的可用性分析
- 学习方法什么时候出错?
- 研究方向与机会

动态环境中学习方法的可用性分析

实验设置

- dataset
 - 使用与静态环境相同的4个数据集
 - 添加20%的新数据，新增数据的关联关系与原数据不同
- workload
 - 根据新的数据集生成新的测试负载（谓词取值范围与数据集有关）
- 测试方法
 - 测试负载会在 $[0, T]$ 的时间内均匀执行（ T 为数据更新频率）
 - 旧模型+新数据： $n \cdot (1 - t_m/T)$
 - 新模型+新数据： $n \cdot [1 - (1 - t_m/T)]$
 - 模型更新的越慢，越多的测试负载会在旧模型新数据的情况下执行，导致误差增大

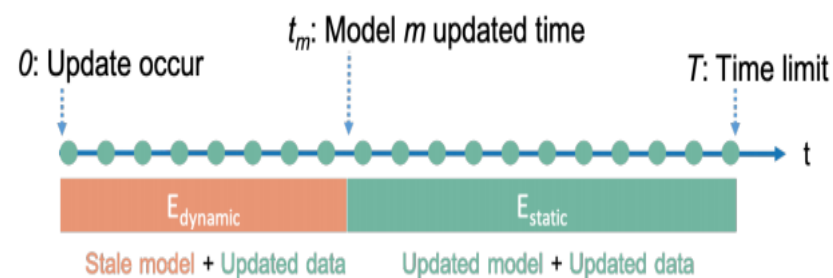


Figure 7: An illustration of a dynamic environment.

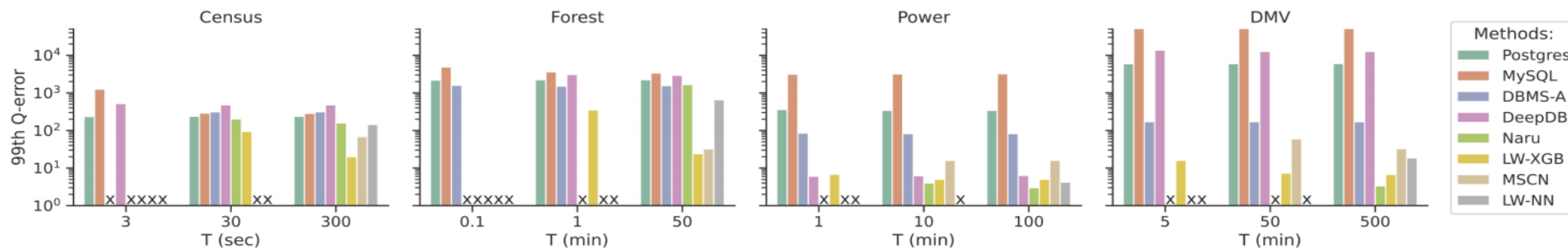
假设 $T = 100$ 分钟, Naru花 $t_m = 75$ 分钟更新模型。然后, Naru
75%的测试负载: 旧模型 + 新数据
25%的测试负载: 新模型 + 新数据

动态环境中学习方法的可用性分析

更新时间

- 学习方法
 - 训练模型的时间
- 数据库系统
 - 运行统计信息收集命令的时间
- 观察
 - 数据库系统因为模型更新时间极短，几乎所有的负载都是在新模型中被预测，即使这样，因为它们基数估计本身的准确率不高，表现整体上不如学习方法，尤其是在大数据集中。

方法	更新时间量级
数据库系统（采样）	10s内
LW-XGB	10s内
DeepDB	10minutes内
LW-NN(GPU)	30minutes内
MSCN(GPU,CPU)	100minutes内
Naru(GPU)	hours



x代表该模型不能在T时间内完成模型更新

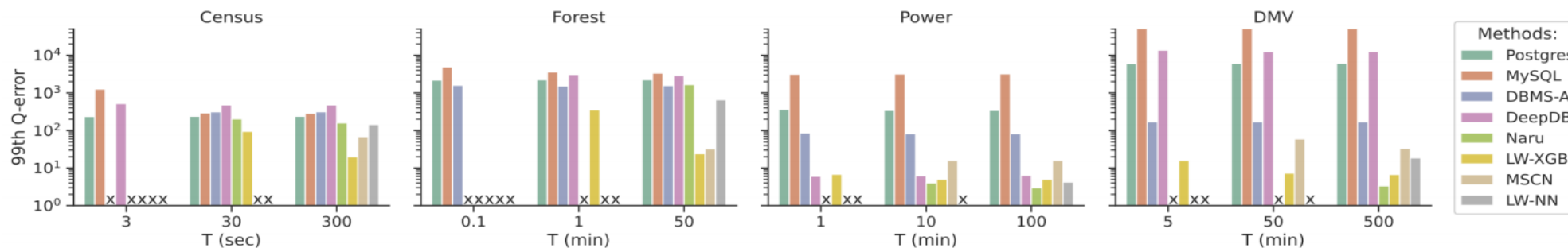
Figure 8: DBMSs vs learned methods under different dynamic environments on four datasets.

动态环境中学习方法的可用性分析

更新时间

- 观察
 - MSCN和LW-NN表现不佳，因为它们需要更长的更新时间（重新执行查询语句获取真实基数作为标签），陈旧的模型处理太多的查询。
 - DeepDB的更新时间通常很短。但是，其更新后的模型不能很好地捕捉相关变化，因此在大多数情况下不能优于LW-XGB/NN。
 - Naru在静态环境中有很好的准确性。然而，在动态环境中，当更新频率较高或中等时，Naru因更新速度慢导致性能并不优于LW-XGB。

方法	更新时间量级
数据库系统（采样）	10s内
LW-XGB	10s内
DeepDB	10minutes内
LW-NN(GPU)	30minutes内
MSCN(GPU,CPU)	100minutes内
Naru(GPU)	hours



x代表该模型不能在T时间内完成模型更新

Figure 8: DBMSs vs learned methods under different dynamic environments on four datasets.

动态环境中学习方法的可用性分析

更新时间与准确率之间的关系

- 对照组
 - Stale 旧模型
 - Updated 新模型
 - Dynamic 旧模型+新模型
- 观察
 - Updated的准确率最好，Stale的效果最差
 - Dynamic的准确率先提升后减小，原因是，随着更新时间的增加，导致了旧模型处理的查询变多，误差变大。

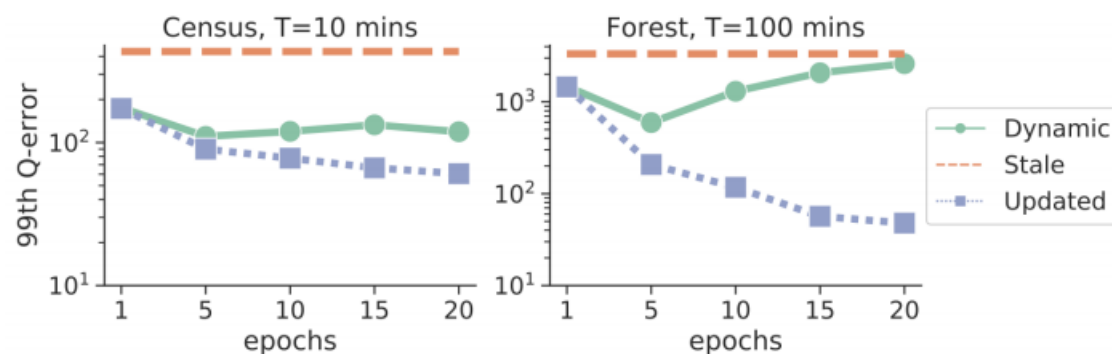


Figure 9: Trade-off (Naru): epochs vs accuracy.



动态环境中学习方法的可用性分析

主要结论

- 学习的方法不能跟上快速的日期更新。MSCN、LW-NN、Naru和DeepDB在动态环境中由于不同的原因返回较大的错误。
- 在更新时间方面，DeepDB是最快的数据驱动方法，LW-XGB是最快的查询驱动方法。
- 在学习的方法中，没有明确的赢家。Naru在数据更新间隔大时性能最好，而LW-XGB在数据频繁更新的环境中性能最好。
- 学习方法的更新时间和准确性之间存在权衡。要在实践中权衡利弊并不容易，需要在这方面进行更多的研究。

目录



- 背景与研究进展
- 现有方法介绍
- 静态环境中学习方法的可用性分析
- 动态环境中学习方法的可用性分析
- 学习方法什么时候出错?
- 研究方向与机会



学习方法什么时候出错?

实验设置

- dataset (只有两列数据)
 - 第一列 数据分布 (Distribution) $s \in [0, 2]$
0代表均匀分布, 随着s增大, 数据分布越倾斜
 - 第二列 关联关系 (Correlation) $c \in [0, 1]$
0代表第二列数据与第二列数据相互独立, 1代表第二列数据与第一列数据函数依赖
- workload
 - 与之前相同
- metric
 - top 1%的q-error分布 (更能体现变化)

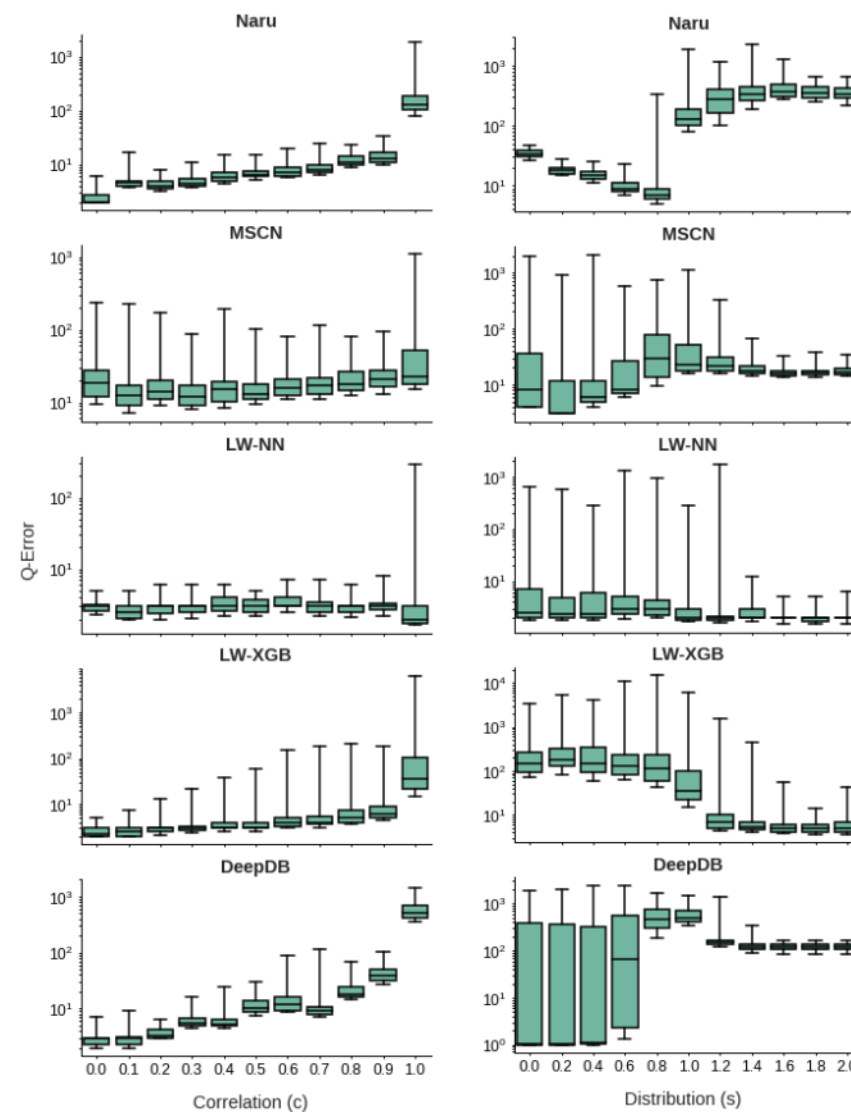
学习方法什么时候出错?

关联关系 (Correlation)

- 关联关系增大, 所有学习方法的误差增大

数据分布 (Distribution)

- 每种学习方法对数据分布变化的反应是不同的。
 - 需要进一步的研究, 让其效果更具有解释性。



(a) $s = 1.0, d = 1000$

(b) $c = 1.0, d = 1000$

学习方法什么时候出错?

列取值范围 (Domain Size)

- 除了LW-NN外，所有方法在更大的域上输出的误差都更大
 - 随着取值范围的增大，需要被学习的特征量也会增大，模型在有限的开销和训练内学习的不够充分，导致在误差增大

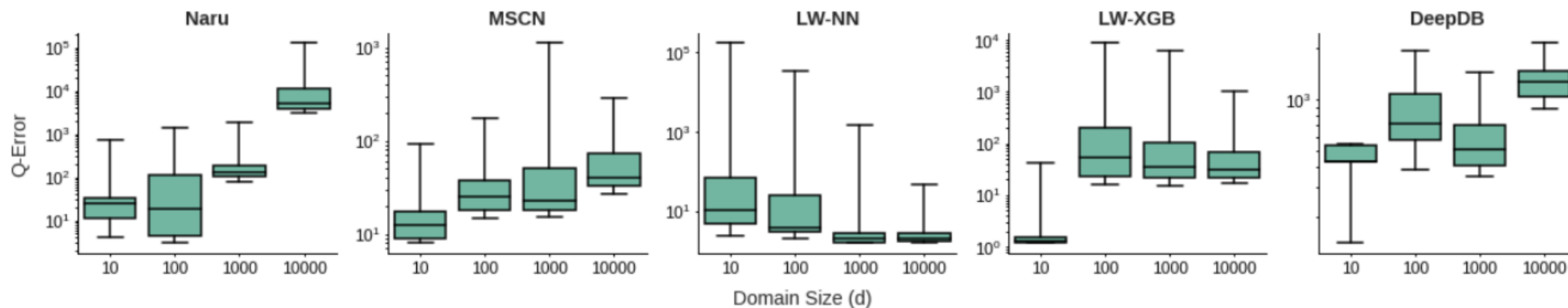


Figure 12: Top 1% error distribution under different domain size ($s = 1.0$, $c = 1.0$).



学习方法什么时候出错？

学习方法的可预测性

- 单调性 (Monotonicity)
 - 对于更严格(或更松散)的谓词，估计结果不应该增加(或减少)。
- 一致性 (consistency)
 - 查询的预测应该等于从查询中分离出来的查询的预测之和，如查询谓词(100、500)可以分成两个查询(100,200)和(200、500)
- 稳定性 (Stability)
 - 对于任何查询，来自同一模型的预测结果应该总是相同的。
- 原则A (Fidelity-A)
 - 选择性率估计之和应该等于1
- 原则B (Fidelity-B)
 - 对于带有无效谓词的查询，估计应该为0，如 `SELECT * FROM R WHERE 100 ≤ A ≤ 10`

Table 5: Satisfaction and violation of rules by learned estimators. (✓: satisfied, ×: violated)

Rule	Naru	MSCN	LW-XGB	LW-NN	DeepDB
Monotonicity	×	×	×	×	✓
Consistency	×	×	×	×	✓
Stability	×	✓	✓	✓	✓
Fidelity-A	✓	×	×	×	✓
Fidelity-B	✓	×	×	×	✓

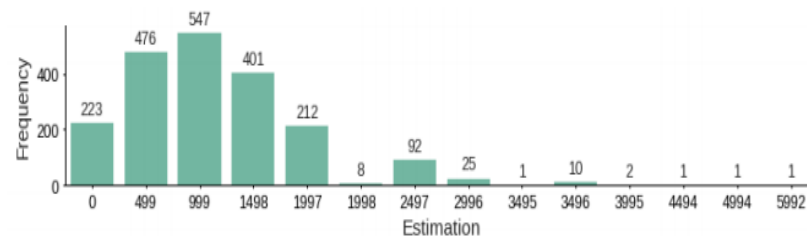


Figure 13: Prediction result of running Naru on the same query 2000 times ($s = 0.0$, $c = 1.0$, $d = 1000$).

学习方法什么时候出错？

学习方法的可预测性

- 所有学习方法中只有DeepDB能满足
 - 其底层用到的histogram，上部分的SPN是简单的加法和乘法
- MSCN,LW-XGB/NN因为是基于深度学习模型的只能满足稳定性
- Naru因为其采样方法导致其不满足连稳定性

Table 5: Satisfaction and violation of rules by learned estimators. (✓: satisfied, ×: violated)

Rule	Naru	MSCN	LW-XGB	LW-NN	DeepDB
Monotonicity	×	×	×	×	✓
Consistency	×	×	×	×	✓
Stability	×	✓	✓	✓	✓
Fidelity-A	✓	×	×	×	✓
Fidelity-B	✓	×	×	×	✓

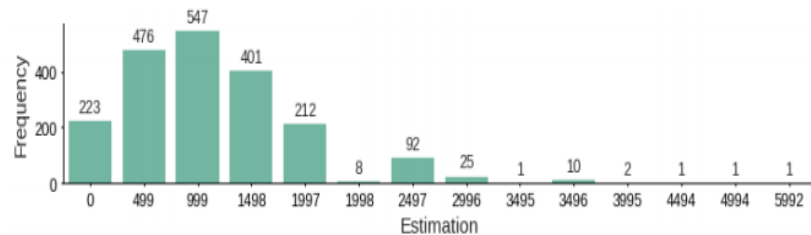


Figure 13: Prediction result of running Naru on the same query 2000 times ($s = 0.0$, $c = 1.0$, $d = 1000$).



学习方法什么时候出错？

可能对生产部署产生影响的地方

- 可调试性 (Debuggability)
 - 黑盒模型难以调试，难以判定误差来源于模型还是bug。
- 可解释性 (Explainability)
 - 黑盒模型缺乏可解释性。
- 可预测性 (Predicability)
 - 由于学习的方法不遵循一些基本的逻辑规则，数据库系统可能会表现出不合逻辑的行为，从而混淆数据库用户。
- 可再现性 (Reproducibility)
 - 数据库开发人员经常希望重现客户问题。但是，如果系统采用违反稳定性规则的Naru，由于随机推理过程，很难再现结果。



学习方法什么时候出错？

主要结论

- 所有的学习方法在相关性更高的数据上误差更大，当两列是函数依赖时，误差最大。
- 对于更倾斜的数据或更大的域大小的数据，不同的方法会有不同的反应。这可能是由于在模型、输入特性和损失函数的选择上的差异。
- 所有学习方法(DeepDB除外)都违反了5条直观规则。
- 在生产中部署时，学习方法的不透明性可能会在可调试性、可解释性、可预测性和可再现性方面造成麻烦。

目录



- 背景与研究进展
- 现有方法介绍
- 静态环境中学习方法的可用性分析
- 动态环境中学习方法的可用性分析
- 学习方法什么时候出错?
- 研究方向与机会



研究方向与机会

- 控制学习方法的训练和预测开销
 - 平衡效率-准确性的权衡 (训练时间、预测时间、更新时间、准确率)
 - 学习估计器的超参数调优 (寻求指导超参的规律)
- 提高学习方法可信度
 - 研究学习方法的可解释性
 - 使学习方法更符合经验规则
- 多表
 - 可行性分析
 - 优化方向研究



谢谢