基于数据概率分布的基数估计方法

- 2021.5.14

目录

基数估计与概率分布关系

回顾基数估计流程

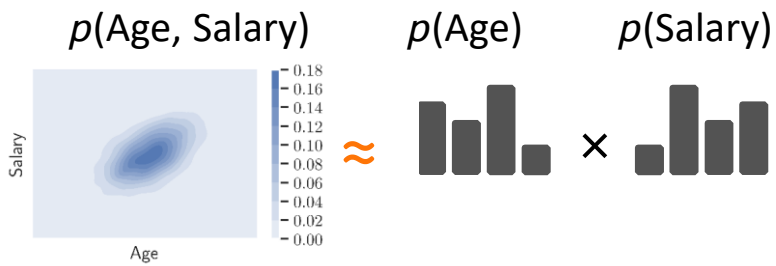
- 单表查询基数估计

```
SELECT *
FROM Employee
WHERE Age < 25 and Salary < 15k;
```

  - P(Age<25)=40%, P(Salary<15k)=60%
  - P(Salary<15k|Age<25)=40%, Total rows:N
  - Rows = N * P(predicates), 后者是选择谓词的选择率
- 基于独立性假设
  - 真实基数=N * p(Salary<15k|Age<25)=0.4N
  - 估计基数=N * P(Age<25) * P(Salary<15k)=24%N
  - 低估 (under estimate)

Employee
ID int
Name string
Sex string
Age int
Salary int

Independence Assumption

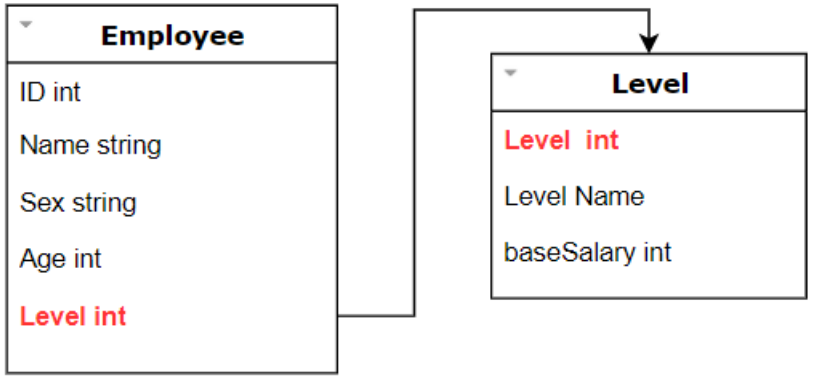


基数估计与概率分布关系

回顾基数估计流程

- 多表查询基数估计

```
SELECT *
FROM Employee as e JOIN Level as l
ON e.level=l.level
WHERE e.Age < 25 and l.Salary < 15k
```



  - Rows = N1 * N2 * P(predicates), 后者是选择谓词的选择率

基数估计与概率分布关系

问题转换

- 基数估计->选择率估计->数据概率分布
  - 估计基数=N * P(Age<25) * P(Salary<15k)=24%N
  - 真实基数=N * p(Salary<15k, Age<25)=0.3N

$$p(X_1, X_2, ..., X_n)$$

- 通过链式法则，我们可以把 n 维的联合概率分布分解成：
  - 等价转换
  - 不依赖于独立性假设

$$p(X_1, X_2, ..., X_n) = p(X_1)\, p(X_2 | X_1) \dots p(X_n | X_1, ..., X_{n-1})$$

基数估计与概率分布关系

与基于查询负载的方法对比

Workload-Driven

- MSCN（2019）、E2E（2020）
- 有监督，需要执行查询获取标签
- 需要大量查询负载作为训练数据
- 当数据集更新之后需要重新训练

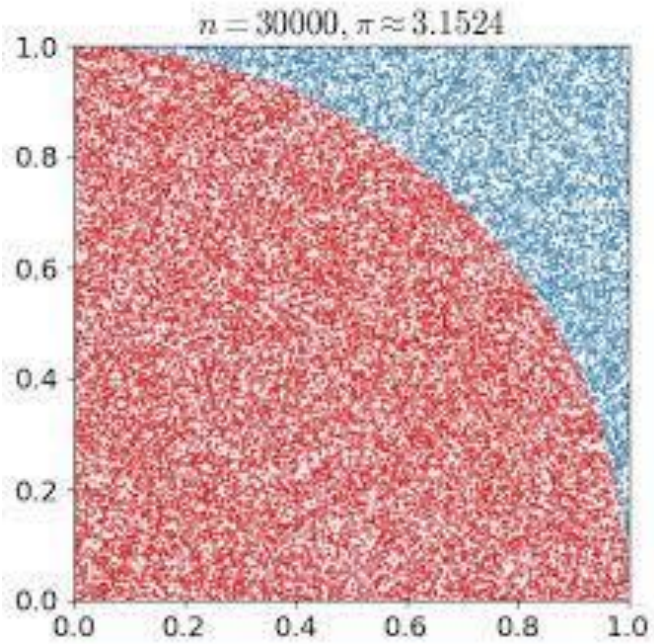
Data-Driven

- 采样、Histogram、KDE、Sum-Product Networks、AutoRegressive
- 无监督，不需要执行查询以获取标签
- 不依赖于查询负载
- 易于更新

基数估计与概率分布关系

最常用思想：采样（蒙特卡洛）

- 对于单属性、单表多属性或多表多属性查询，使用部分数据来预估整体数据



- 优点
  - 最直观方法
  - 其思想被许多其他方法广泛应用（构建直方图、KDE）
- 缺点
  - 偶然性强
  - 0-tuple，采样消失问题

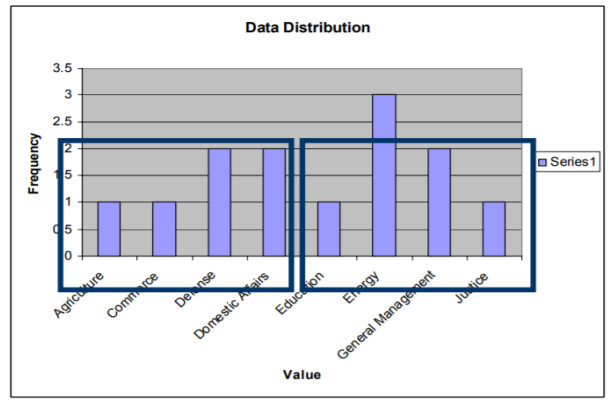
目录

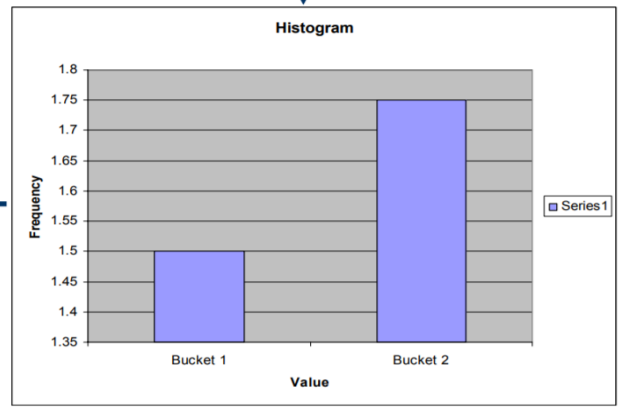
单属性概率分布估计

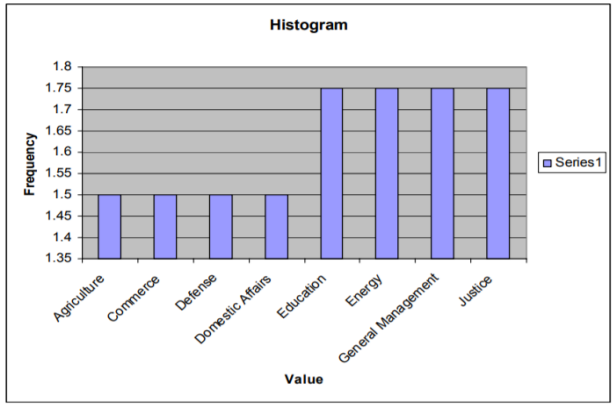
一维直方图

- 基于均匀分布假设
- 被广泛应用于商业DBMSs
- 优点
  - 离线统计，几乎不占用运行时开销
  - 不需要数据拟合一个概率分布
  - 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

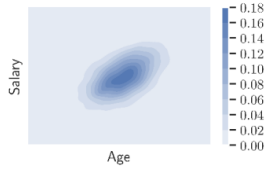






Independence Assumption

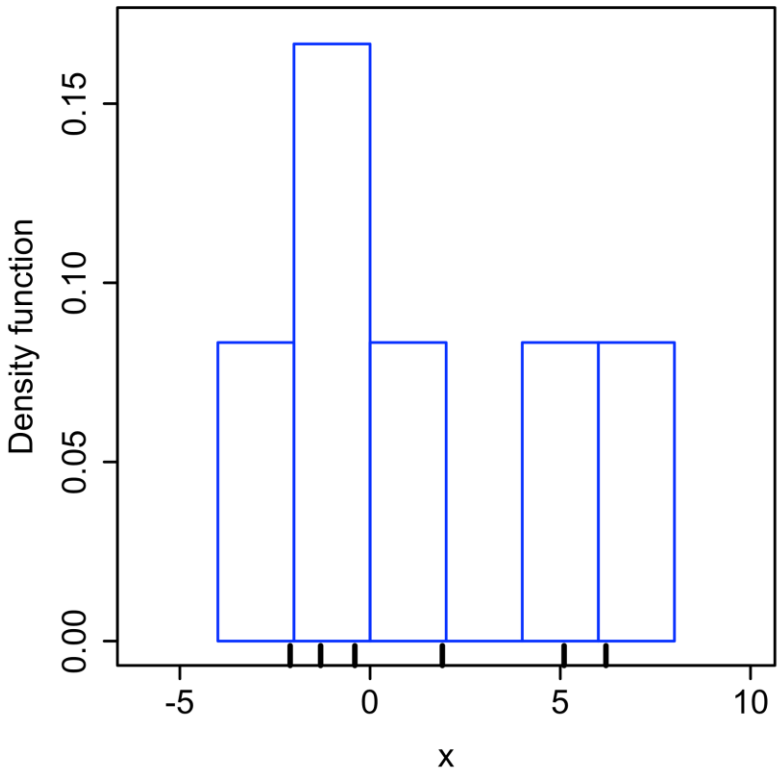
p(Age, Salary) $\approx$ p(Age) $\times$ p(Salary)

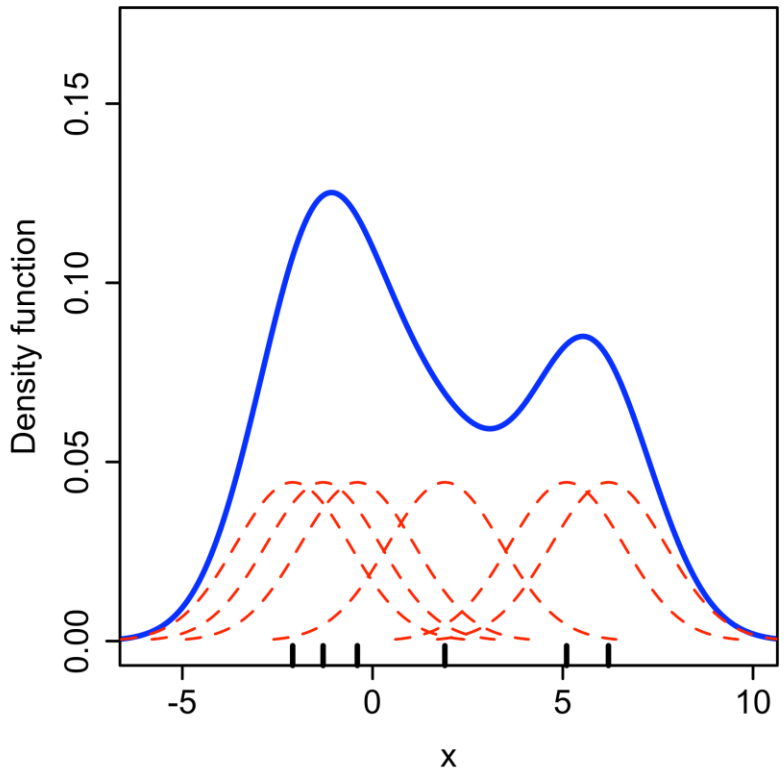


单属性概率分布估计

核密度估计 kernal density estimation (KDE)

- 类似于直方图
  - 使用Kernal（如正太分布）来代替均匀假设





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

单表单属性概率分布估计

小结

- 一维直方图
  - 均匀分布假设
- 核密度估计
  - 正太分布等不同Kernal来估计其他点分布

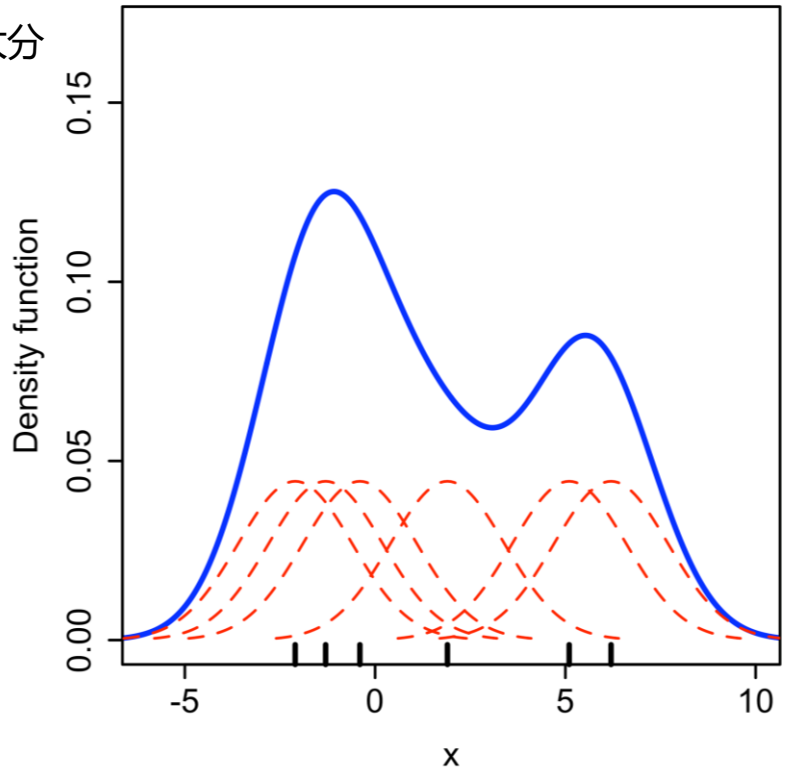
- 怎么拓展到单个表多个属性?

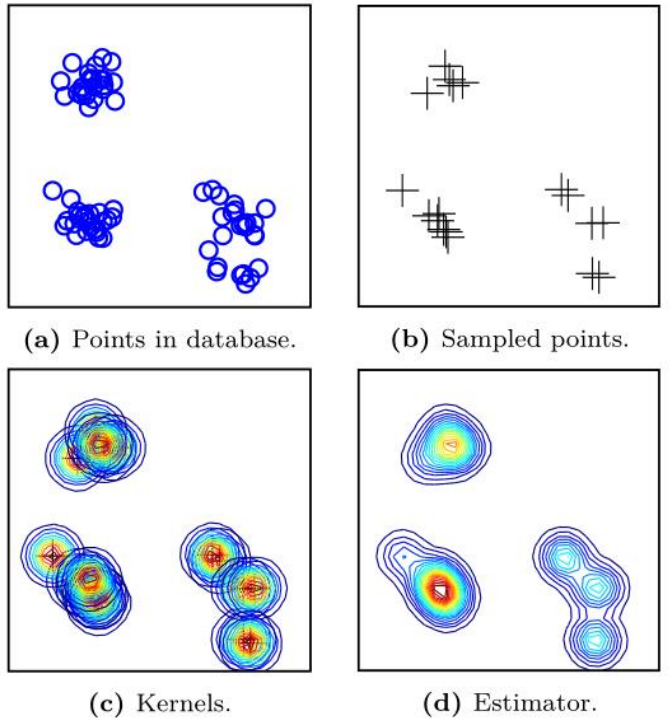
目录

单表多属性概率分布估计

多维核密度估计（KDE）

- 维度升级
  - 使用多维度的核：如多变量正太分布



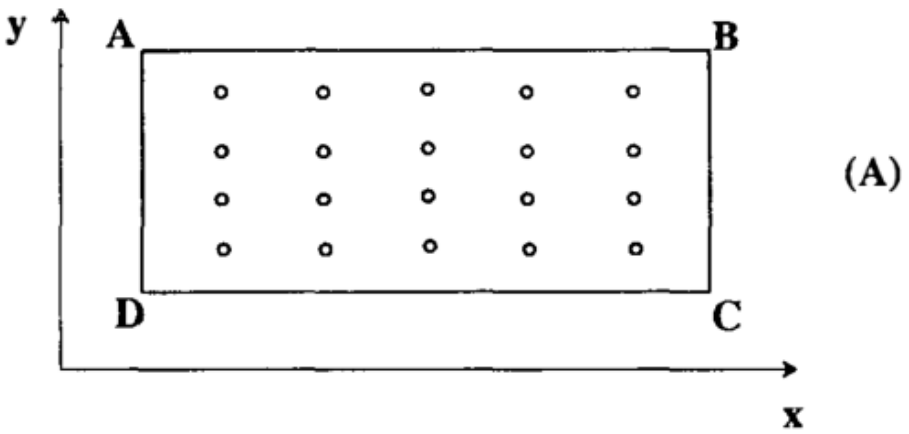


$$\hat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i) = \frac{1}{nh}\sum_{i=1}^{n} K(\frac{x - x_i}{h})$$

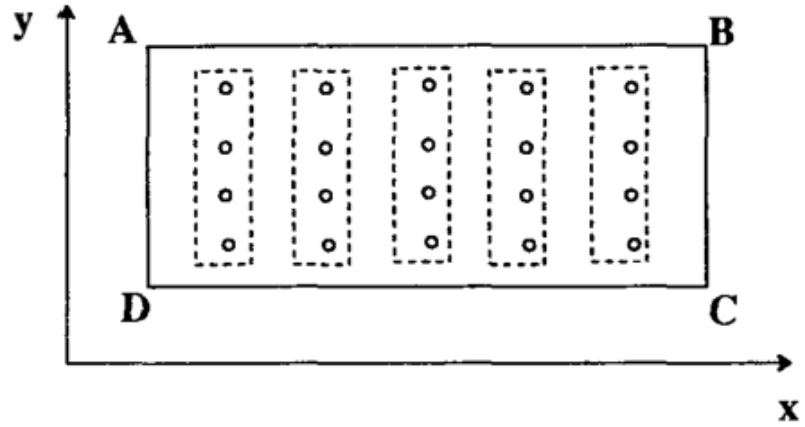
单表多属性概率分布估计

多维直方图

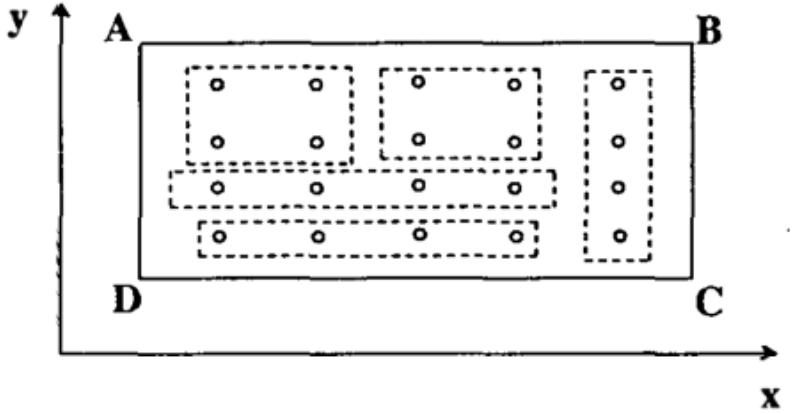
- 难点：如何划分空间
  - 因为直方图是基于均匀假设
  - 每个bucket的值组合最好频率相似，这样才能符合均匀假设，减小误差



(A)



(B)



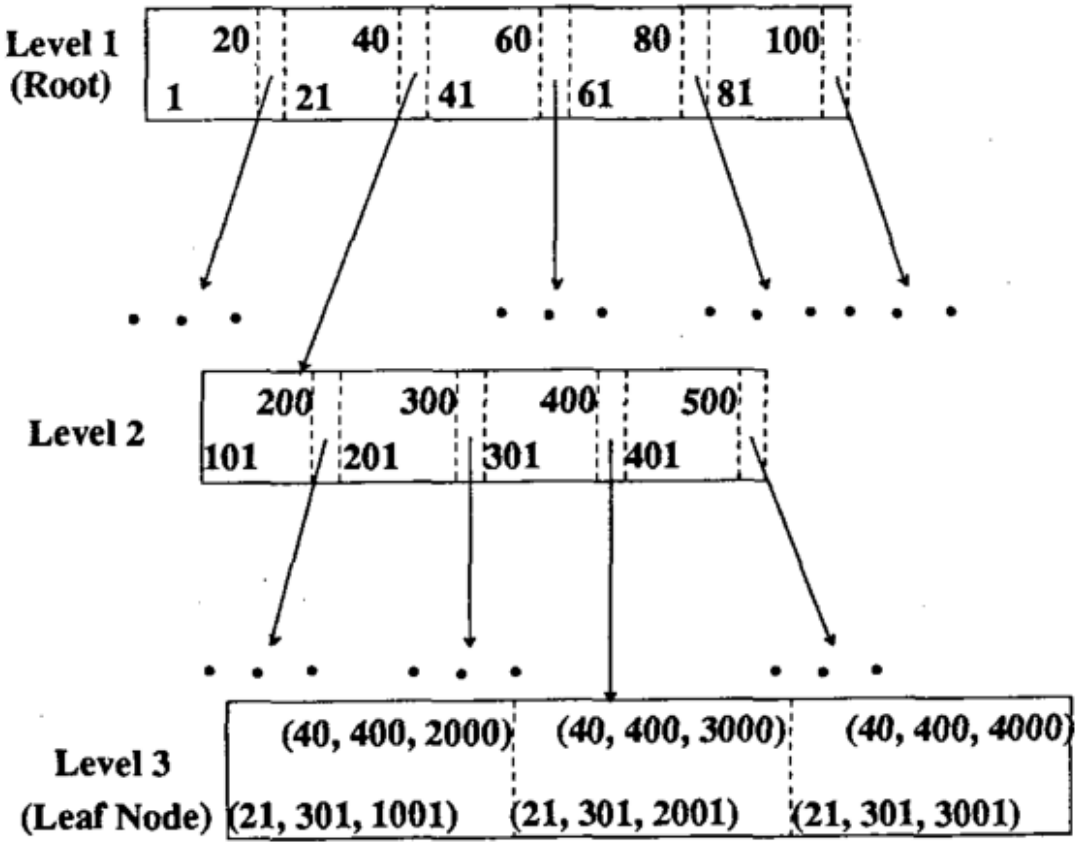
(C)

单表多属性概率分布估计

多维直方图-一种simple的划分方法(PHASED)

- 表格情况
  - 以三维（三个属性的情况为例）
  - 属性A [1,100]、属性B [101,500]、属性C [1001,4000]

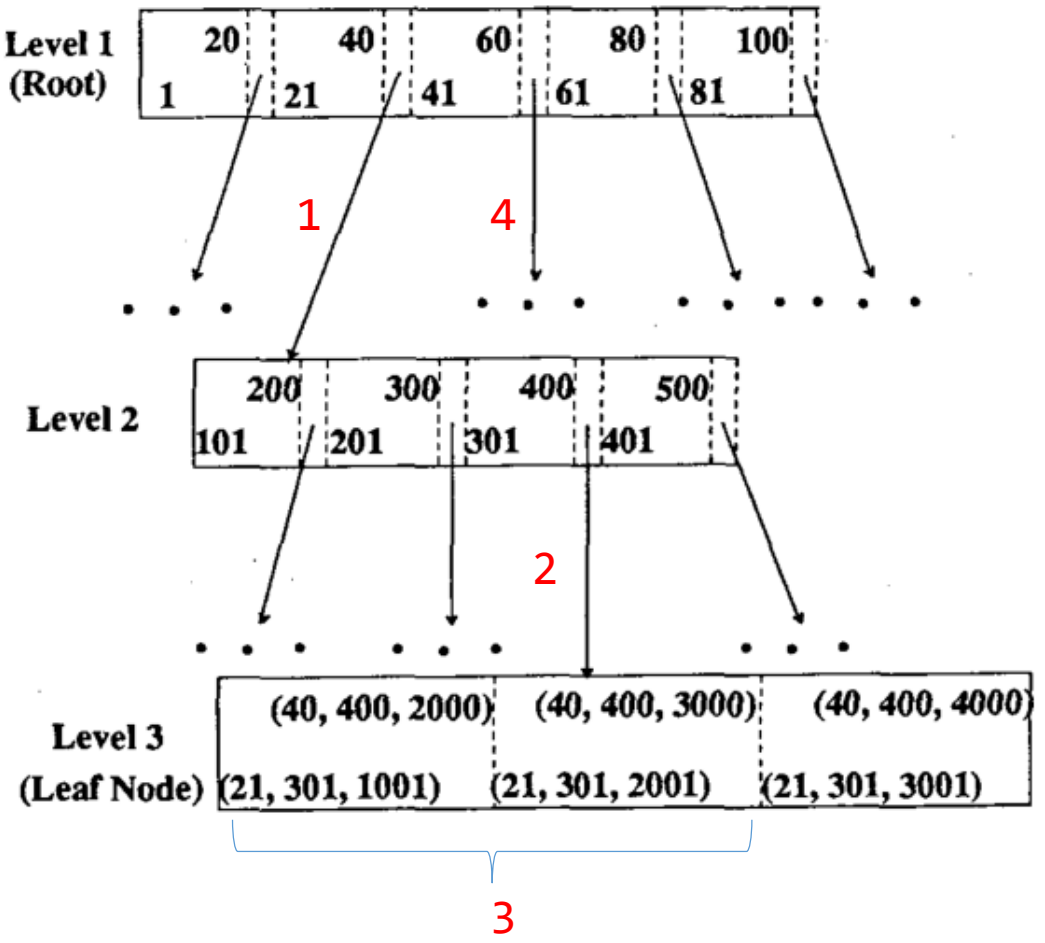
- 划分步骤
  - 表格按第一个属性排序，划分成5个bucket -- (1)
  - 对（1）中的5个bucket分别按第二个属性排序，每个划分成4个bucket，此时一共有5*4=20个bucket -- (2)
  - 对（2）中的20个bucket分别按第三个属性排序，每个划分成3个bucket，此时一共有20*3=60个bucket -- (3)



单表多属性概率分布估计

多维直方图-查询举例(PHASED)

- 查询范围((31, 325, 1250), (50, 375, 2500))举例
  - 步骤1：属性A (31, 50) 与 Level1的 2, 3Block匹配,先处理Block2跟随指针来到Level2
  - 步骤2：属性B(325,375)只与Block3匹配，跟随指针来到Level3
  - 步骤3：属性C(1250,2500)与Bucket1, 2匹配，根据占比估算其概率分布。
  - 步骤4：以此类推处理Level1的Block3。



单表多属性概率分布估计

多维直方图(PHASED vs MHIST)

- PHASED
  - 生成方法简单粗暴（分裂顺序不好）
  - 生成的bukcet质量比较差，分布不均匀

- MHIST(multi-dimensional histogram)
  - 在每一步中选择和划分最“关键”的属性

- 多维直方图问题
  - 避免独立性假设的统计和空间开销太大，只能针对极小的数据集

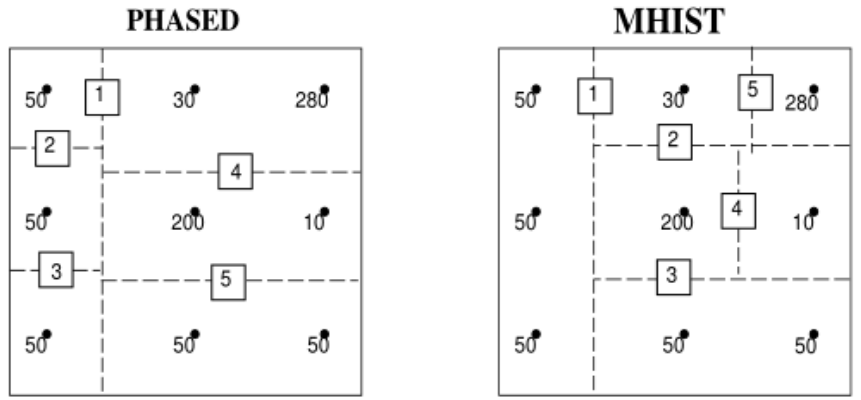


Figure 4: Two-dimensional MaxDiff(V,F) histograms

单表多属性概率分布估计

概率图模型-Sum-Product Networks（DeepDB单表）

- 概率图模型
  - 在图的基础上表示概率分布的模型
- 节点类型
  - sum节点：将表格按行切分成row cluster
  - product节点：将表格按列切分Column cluster
- 生成步骤
  - 步骤1：利用Kmeans等方法将元组分类切分成不同的sum节点
  - 步骤2：利用RDC(Randomized Dependency Coefficients)判断不同属性之间是的关联性，如果存在相关性则回到步骤1继续切分sum节点，当row cluster的行数小于阈值n时则认为两个列之间没有关联性

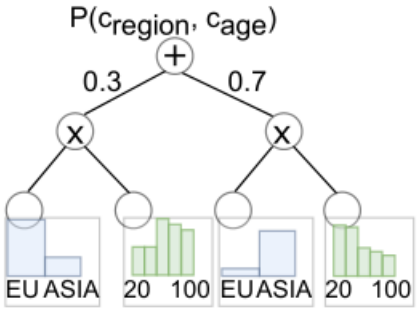
- 有多少客户时来自Europe并且小于30岁？
  - 12% · 0.3 + 2% · 0.7 = 5%

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

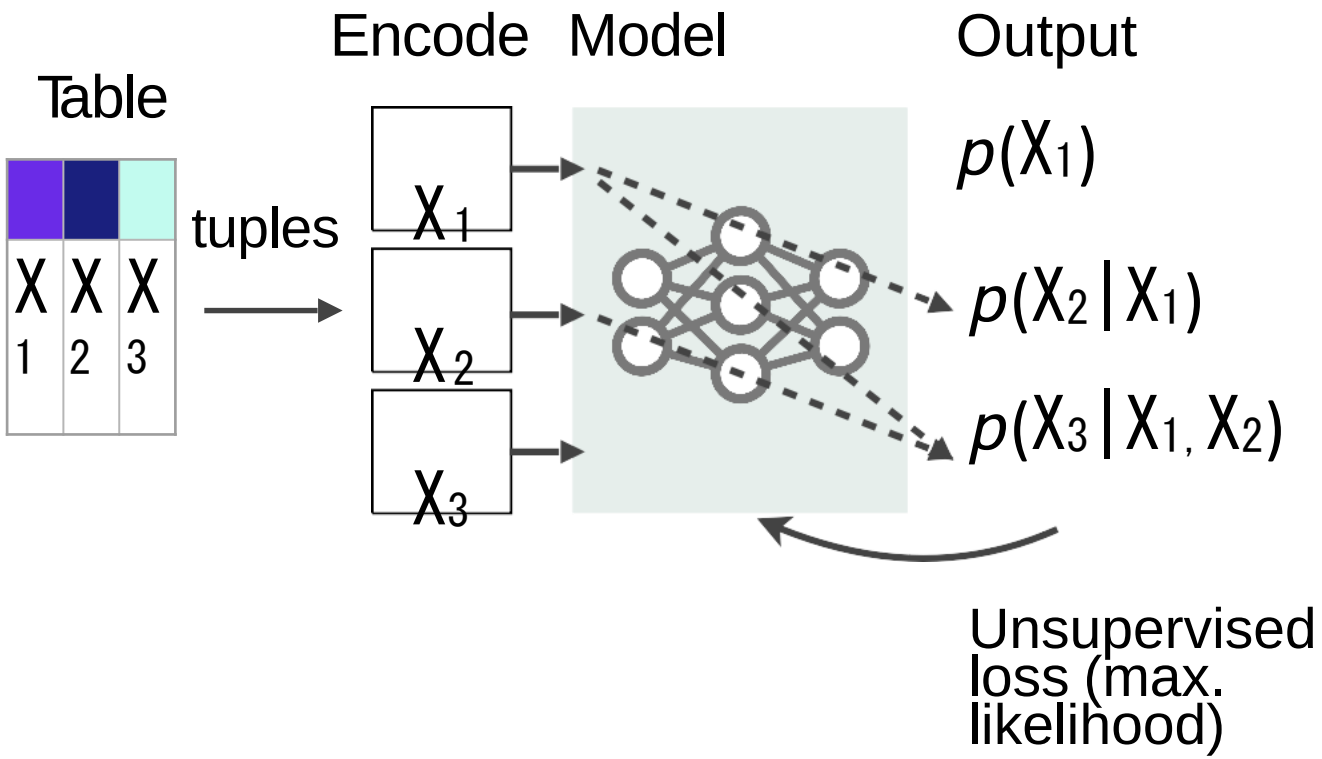
(d) Probability of European Customers younger than 30

Figure 2: Customer Table and corresponding SPN.

单表多属性概率分布估计

Naru(Neural Relation Understanding)

- Workflow
  - 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
  - 以极大似然估计原则进行一次参数更新



- Naru使用自回归模型(AutogRegressiv)

无监督学习数据的概率分布

  - MADE
  - ResMADE
  - Transformer

单表多属性概率分布估计

Naru(Neural Relation Understanding)

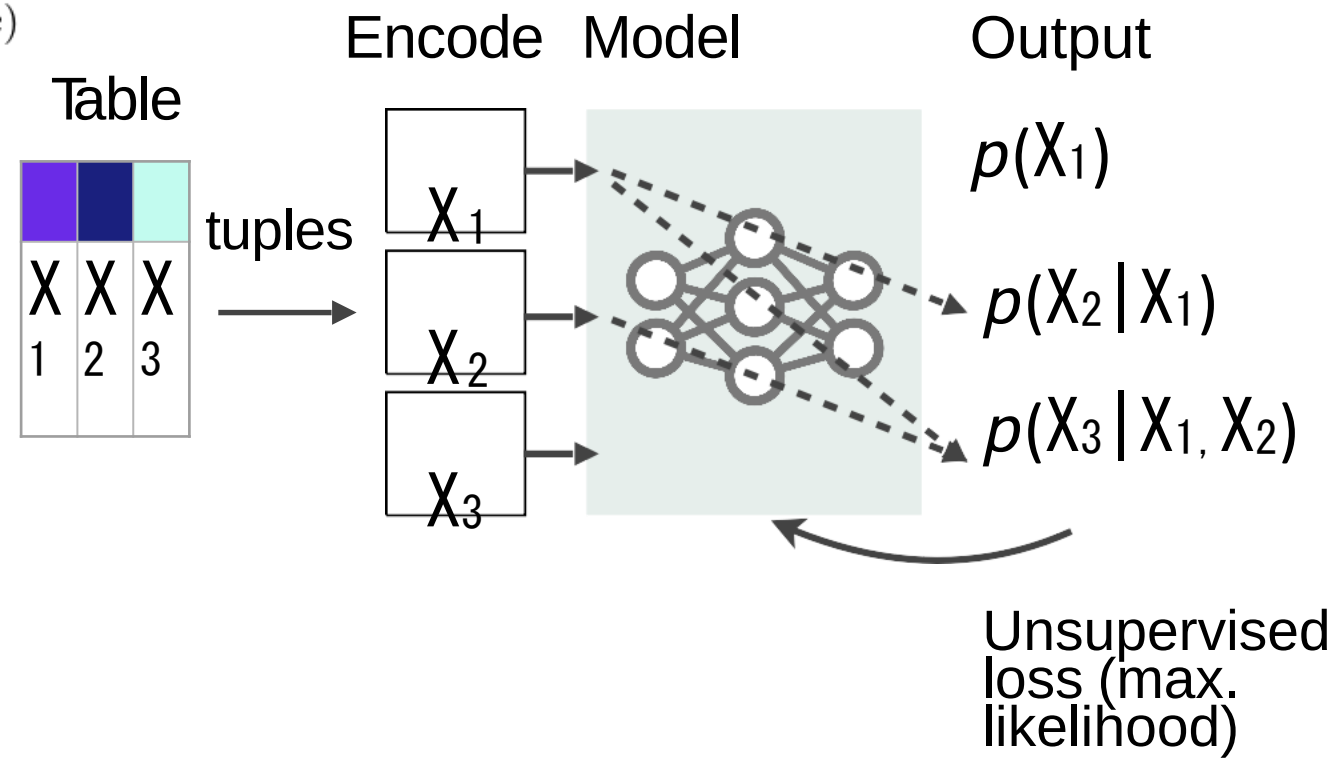
- 极大似然估计，最小化KL散度
  - 信息熵

$$\mathcal{H}(P, \widehat{P}) = -\sum_{\boldsymbol{x} \in T} P(\boldsymbol{x}) \log \widehat{P}(\boldsymbol{x}) = -\frac{1}{|T|} \sum_{\boldsymbol{x} \in T} \log \widehat{P}(\boldsymbol{x})$$

  - KL散度作为损失函数（两个概率分布直接的距离）

$$\mathcal{H}(P, \widehat{P}) - \mathcal{H}(P)$$

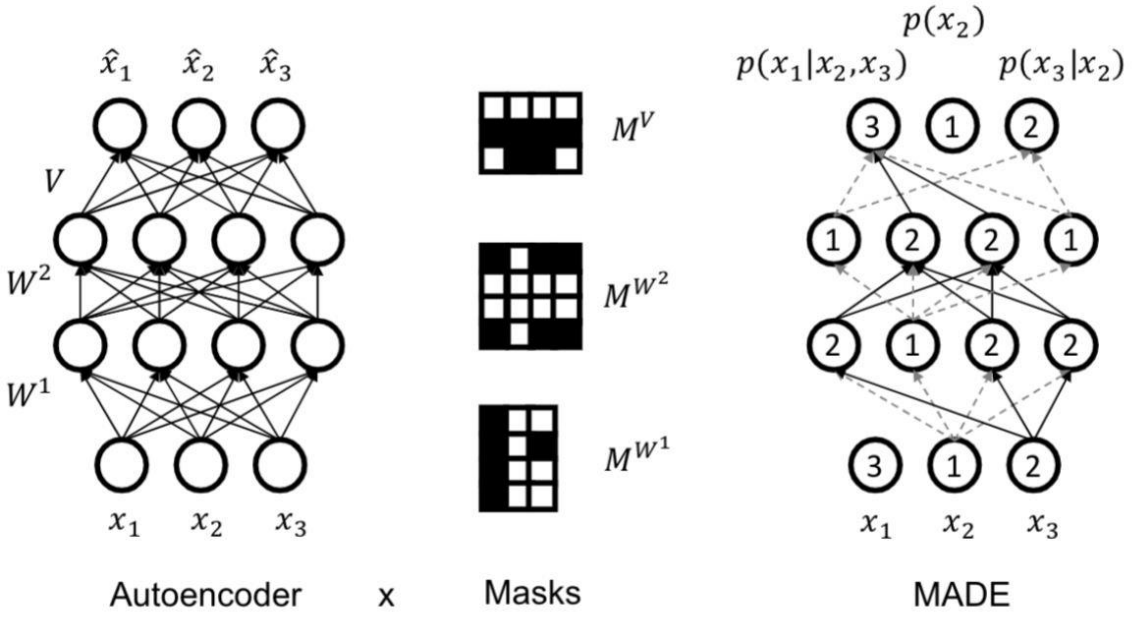
$$D_{\mathrm{KL}}(P|\widehat{P}) = -\sum_{i} P(i) \ln \frac{\widehat{P}(i)}{P(i)}$$



单表多属性概率分布估计

Naru(Neural Relation Understanding)

- Made
  - 在自编码器的基础上利用MLP作为Mask层
  - 在计算P(x1)的时候屏mask掉x2,x3
  - 在计算P(x2|x1)的时候mask掉x3...



$$p(X_1, X_2, ..., X_n) = p(X_1)\ p(X_2|X_1) \ ...\ p(X_n|\ X_1, ..., X_{n-1})$$

Not materialized; Emitted **on-demand** by model

单表多属性概率分布估计

Naru(Neural Relation Understanding)

- 等值查询
  - 等值查询时，每个属性的值已经被指定

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

  - 根据链式法则

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

- AR模型只能估计出等值情况，如何处理范围查询?

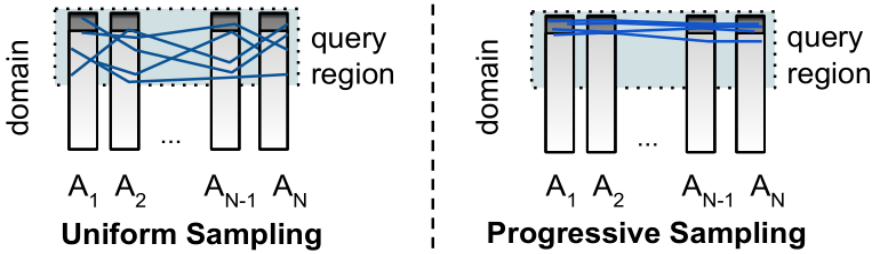
- 范围查询（小范围）
  - 枚举所有可能性

$$\text{sel}(X_1 \in R_1, \ldots, X_n \in R_n) \approx \sum_{x_1 \in R_1} \cdots \sum_{x_n \in R_n} \widehat{P}(x_1, \ldots, x_n)$$

单表多属性概率分布估计

Naru(Neural Relation Understanding)

- 范围查询（范围比较大时）
  - 枚举开销大最坏情况下，region $R = R_1 \times \cdots \times R_n$ 有 $O(\prod_i D_i)$ 种组合情况，其中 $D_i = |A_i|$ 是每个属性的取值个数
  - 采用采样的方式



一个具有N个具有相关性的表T，每个属性的值都不是均匀分布的，99%的频率属于1%的取值。使用范围谓词选择每个域的前50%的查询。

- uniform sampling
  - 容易缺失密度高的情况

$$1/(0.01/0.5)^n = 1/0.02^n$$

- progressive sampling
  - 使用Naru模型估计的条件概率有针对的对高频率数据进行抽样

```
# Query: sel(X1 in R1 ,..., Xn in Rn)

ProgressiveSample():

# Sample dim 1
s1 ~ Model(X1 | X1 in R1)

# Sample dim 2
s2 ~ Model(X2 | X2 in R2, X1=s1)

...
# Monte Carlo estimate
return p(X1 in R1)···p(Xn in Rn|s1, ···, sn-1)
```

单表多属性概率分布估计

小结

- 多维核密度估计
  - 使用高维的核，如高维正太分布
- 多维直方图
  - 难点在于bucket划分，介绍了PHASED和MHSIT两种方法
  - 统计和空间开销大
- Sum-Product Networks
  - 用由Sum和Product节点组成的树来存储表示

  数据概率分布
- Naru
  - 使用MADE自回归模型学习概率分布
  - Naru怎么扩展以支持范围查询

目录

多表概率分布估计

单表扩展到多表

- 查询中的Join连接会涉及到多表数据的联合概率分布
  - 可能可能存在连接的表，预先进行全外连接学习其分布

- 现有实现（DeepDB、NeuroCard）
  - DeepDB 检查不同表中的每一对属性是否可以认为是独立的，对存在关系的属性学习其RSPN（Relationnl Sum-Product Networks）
  - NeuroCard是Naru的多表版本，借鉴了DeepDB的思路，对所有表进行全外连接形成一个表，再使用自回归模型对其学习

- 全外连接(full outer join)
  - DeepDB 找关联性，按需求学习SPN
  - 借鉴了NeuroCard 将所有表full outer join 起来只学一个表

多表概率分布估计

DeepDB (Relational Sum-Product Networks RSPN)

- 单表扩展到多表
  - 对每一对主外键的全外连接学习其RSPN
  - 使用采样的方法检查不同表中的每一对属性之间是否存在关联性，如果存在则学习其其RSPN

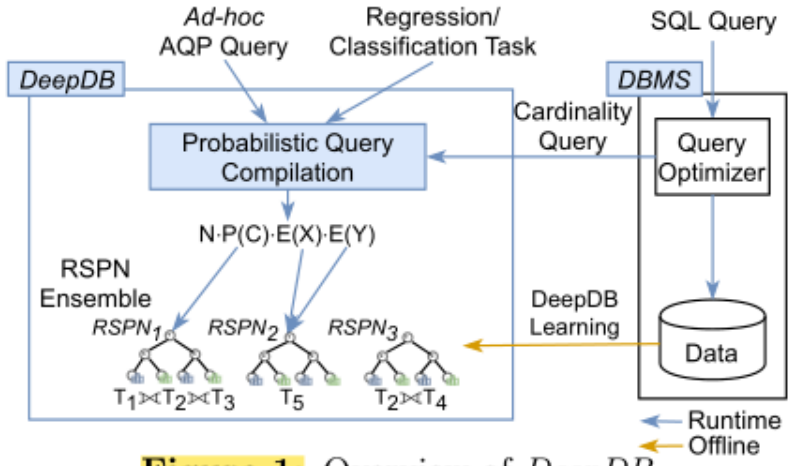


Figure 1: Overview of *DeepDB*.

Customer

c_id	c_age	c_region	$\mathcal{F}_{C\leftarrow O}$
1	20	EU	2
2	50	EU	0
3	80	ASIA	2

Order

o_id	c_id	o_channel
1	1	ONLINE
2	1	STORE
3	3	ONLINE
4	3	STORE

(a) Ensemble with Single Tables

Customer⟗Order

$\mathcal{N}_C$	c_id	c_age	c_region	$\mathcal{F}'_{C\leftarrow O}$	$\mathcal{N}_O$	o_id	o_channel
1	1	20	EU	2	1	1	ONLINE
1	1	20	EU	2	1	2	STORE
1	2	50	EU	1	0	NULL	NULL
1	3	80	ASIA	2	1	3	ONLINE
1	3	80	ASIA	2	1	4	STORE

(b) Ensemble with Full Outer Join

多表概率分布估计

DeepDB (Relational Sum-Product Networks RSPN)

- Case1:有完全匹配的RSPN

Q_1:
```
SELECT COUNT(*) FROM CUSTOMER C
    WHERE c_region='EU';
```

$$|\mathtt{C}| \cdot \mathbb{E}(\mathbf{1}_{\mathtt{c\_region='EU'}}) = 3 \cdot \tfrac{2}{3} = 2$$

Customer

c_id	c_age	c_region	$\mathcal{F}_{C \leftarrow O}$
1	20	EU	2
2	50	EU	0
3	80	ASIA	2

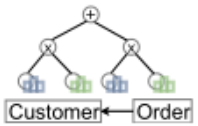
Order

o_id	c_id	o_channel
1	1	ONLINE
2	1	STORE
3	3	ONLINE
4	3	STORE

(a) Ensemble with Single Tables

Q_2:
```
SELECT COUNT(*) FROM CUSTOMER C
    NATURAL JOIN ORDER O
    WHERE c_region='EU' AND
    o_channel='ONLINE';
```

could be represented as $|\mathtt{C} \bowtie \mathtt{O}| \cdot P(\mathtt{o\_channel='ONLINE'} \cap \mathtt{c\_region='EU'})$ which is $4 \cdot \frac{1}{4} = 1$.



Customer ⟗ Order

$\mathcal{N}_C$	c_id	c_age	c_region	$\mathcal{F}'_{C \leftarrow O}$	$\mathcal{N}_O$	o_id	o_channel
1	1	20	EU	2	1	1	ONLINE
1	1	20	EU	2	1	2	STORE
1	2	50	EU	1	0	NULL	NULL
1	3	80	ASIA	2	1	3	ONLINE
1	3	80	ASIA	2	1	4	STORE

(b) Ensemble with Full Outer Join

多表概率分布估计

DeepDB (Relational Sum-Product Networks RSPN)

- Case2:有涉及更多表的RSPN，需要降维
  - 如对第三个RSPN查询Q1

Q_1: `SELECT COUNT(*) FROM CUSTOMER C WHERE c_region='EU';`

$$|\mathtt{C} \bowtie \mathtt{O}| \cdot \mathbb{E}(1/\mathcal{F}'_{C \leftarrow O} \cdot \mathbf{1}_{\mathtt{c\_region='EU'}} \cdot \mathcal{N}_C)$$

$$5 \cdot \frac{1/2+1/2+1}{5} = 2$$

  - 其中 $\mathcal{F}'_{C \leftarrow O}$ 代表Order表中有几个节点和Customer表这一行连接，也就是多算了几遍

    $\mathcal{N}_C$ 代表这一列是否是补充的

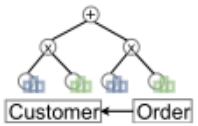
Customer

c_id	c_age	c_region	$\mathcal{F}_{C \leftarrow O}$
1	20	EU	2
2	50	EU	0
3	80	ASIA	2

Order

o_id	c_id	o_channel
1	1	ONLINE
2	1	STORE
3	3	ONLINE
4	3	STORE

(a) Ensemble with Single Tables



Customer⟗Order

$\mathcal{N}_C$	c_id	c_age	c_region	$\mathcal{F}'_{C \leftarrow O}$	$\mathcal{N}_O$	o_id	o_channel
1	1	20	EU	2	1	1	ONLINE
1	1	20	EU	2	1	2	STORE
1	2	50	EU	1	0	NULL	NULL
1	3	80	ASIA	2	1	3	ONLINE
1	3	80	ASIA	2	1	4	STORE

(b) Ensemble with Full Outer Join

多表概率分布估计

DeepDB (Relational Sum-Product Networks RSPN)

- Case3:需要有多个RSPN相组合
  - 如对第一个和第二个RSPN查询Q2

Q_2:
```
SELECT COUNT(*) FROM CUSTOMER C
    NATURAL JOIN ORDER O
    WHERE c_region='EU' AND
    o_channel='ONLINE';
```

$$|\mathrm{C}| \cdot \underbrace{\mathbb{E}(\mathbf{1}_{\texttt{c\_region='EU'}} \cdot \mathcal{F}_{C \leftarrow O})}_{Q_L} \cdot \underbrace{\mathbb{E}(\mathbf{1}_{\texttt{o\_channel='ONLINE'}})}_{Q_R}$$

$$3 \cdot \frac{2+0}{3} \cdot \frac{2}{4} = 1$$

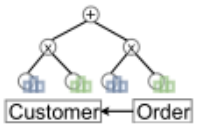
Customer

c_id	c_age	c_region	$\mathcal{F}_{C \leftarrow O}$
1	20	EU	2
2	50	EU	0
3	80	ASIA	2

Order

o_id	c_id	o_channel
1	1	ONLINE
2	1	STORE
3	3	ONLINE
4	3	STORE

(a) Ensemble with Single Tables



Customer ⟗ Order

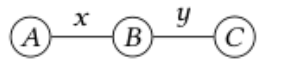
$\mathcal{N}_C$	c_id	c_age	c_region	$\mathcal{F}'_{C \leftarrow O}$	$\mathcal{N}_O$	o_id	o_channel
1	1	20	EU	2	1	1	ONLINE
1	1	20	EU	2	1	2	STORE
1	2	50	EU	1	0	NULL	NULL
1	3	80	ASIA	2	1	3	ONLINE
1	3	80	ASIA	2	1	4	STORE

(b) Ensemble with Full Outer Join

多表概率分布估计

Nerucard (AutoRegressive Models)

- Nerucard在Naru的基础上借鉴了DeepDB思想
  - 对所有表全外连接形成的单表使用自回归模型学习其概率分布
  - 全外连接数据量太大，该方法的难点在于如何公平抽样模拟
  - 在完全连接J(一个多重集)中的每一个元组必须以1/|J|的概率等概率采样
- 采样方法
  - 步骤1：自下而上的对可用于连接的属性的每一个值，计算他们的Join counts，即在其他属性中有多少行能和它进行Join
  - 步骤2：samper逐表对连接键进行采样，出现概率与连接计数成正比
  - 步骤3：按选取出的连接键来查询基表以获取非连接键的值

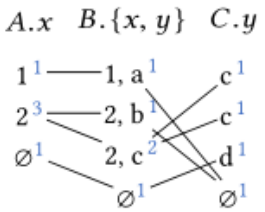


$A.x$
1
2

$B.x$	$B.y$
1	a
2	b
2	c

$C.y$
c
c
d

(a) Schema and base tables



(b) Join counts

$A.x$	$B.x$	$\mathcal{F}_{B.x}$	$B.y$	$C.y$	$\mathcal{F}_{C.y}$	$\mathbb{1}_A$	$\mathbb{1}_B$	$\mathbb{1}_C$
1	1	1	a	∅	1	1	1	0
2	2	2	b	∅	1	1	1	0
2	2	2	c	c	2	1	1	1
2	2	2	c	c	2	1	1	1
∅	∅	1	∅	d	1	0	0	1

(c) Full outer join, with virtual columns in blue

```
-- In full join, |A.x=2|=3.
-- Q1. True answer is 2.
SELECT COUNT(*)
FROM A JOIN B ON x
       JOIN C ON y
WHERE A.x = 2;
-- Q2. True answer is 1.
SELECT COUNT(*)
FROM A WHERE A.x = 2;
```

(d) Schema subsetting

多表概率分布估计

不同方法的评估质量

- Data-Driven 优于 Workload-Driven
  - NeuroCard ~= DeepDB > GACE > E2E > MSCN
- NeuroCard和DeepDB各有优势
  - NeuroCard整体效果更好，尤其边缘误差
  - DeepDB 中位数更好

Table 2: JOB-light, estimation errors. Lowest errors are bolded.

Estimator	Size	Median	95th	99th	Max
Postgres	70 KB	7.97	797	$3 \cdot 10^3$	10^3
IBJS	–	1.48	10^3	10^3	10^4
MSCN	2.7 MB	3.01	136	$1 \cdot 10^3$	10^3
E2E (quoting [38])	N/A	3.51	139	244	272
DeepDB	3.7 MB	1.32	4.90	33.7	72.0
DeepDB-large	32 MB	**1.19**	**4.66**	35.0	39.5
NeuroCard	3.8 MB	1.57	5.91	**8.48**	**8.51**

GACE 3.xMB 2.11 19.12 103.13 219.59

多表概率分布估计

多表概率分布估计总结

- 在单表模型的基础上，对可能存在连接的表先进行全外连接

- DeepDB
  - 采样检测不同属性对之间的关联性，存在关联则建立其RSPN
  - 完全匹配的RSPN
  - 存在涉及更多表的RSPN，降维估计（F和N指标）
  - 使用多个RSPN组合估计

- Nerucard
  - 对所有表进行全外连接
  - 计算Join count公平采样减小误差

基于数据概率分布基数估计方法

总结

- 基数估计与概率分布关系
  - 基数估计->选择率估计->数据概率分布估计
- 单属性概率分布估计
  - 直方图，KDE
- 单表多属性概率分布估计
  - 多维直方图，多维KDE
  - Sum-Product Networks
  - Naru(MADE 自回归模型)
- 多表概率估计
  - DeepDB
  - Nerucard
- Data-Driven和Workload-Driven相结合
  - 如将Data-Driven的概率密度作为后者的输入

基于数据概率分布基数估计方法

总结

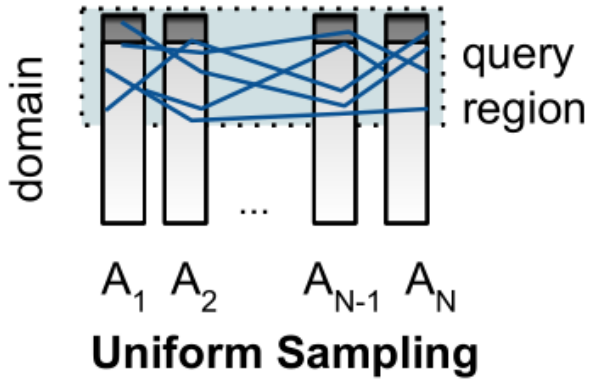
- Data-Driven 与Workload-driven的对比
  - 仍在研究阶段，应用少
- Data-Driven 与Workload-driven的结合可能性?
  - 仍在研究阶段，应用少

However, we do not argue that data-driven models are a silver bullet to solve all possible tasks in a DBMS. Instead, we think that data-driven models should be combined with workload-driven models when it makes sense. For example, a workload-driven model for a learned query optimizer might use the cardinally estimates of our model as input features. This combination of data-driven and workload-driven models provides an interesting avenue for future work but is beyond the scope of this paper.

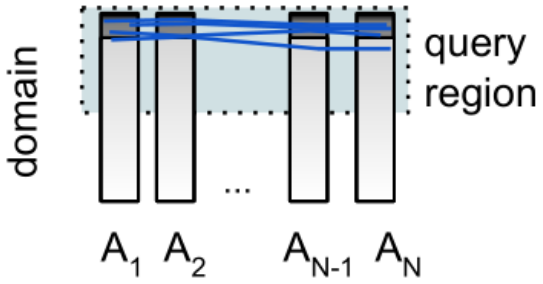
支持更新

DeepDB

- 如何支持更新?
  - SPN
- Progressive Sampling
  - SPN



Uniform Sampling



Progressive Sampling

谢谢

基于数据概率分布基数估计方法

总结

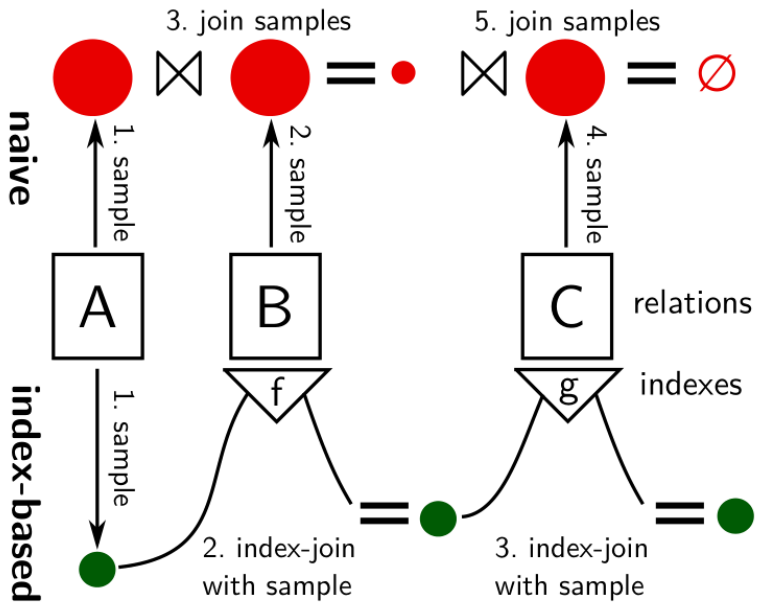


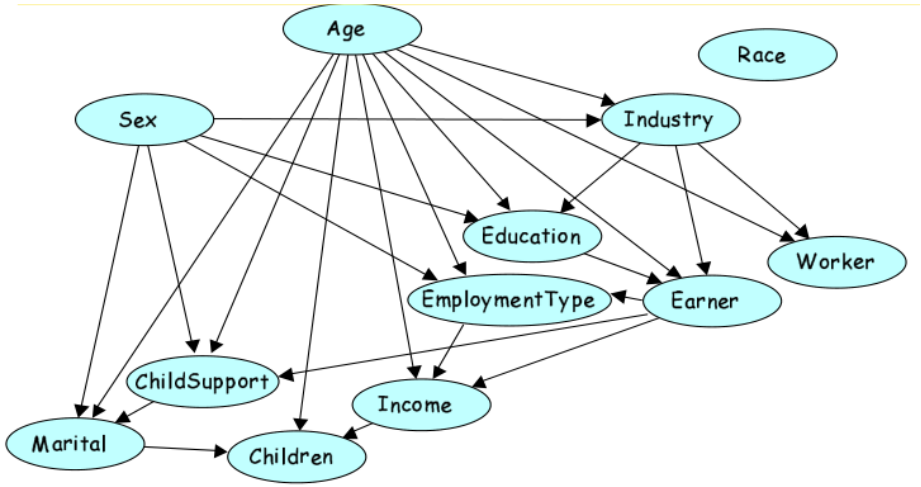
Figure 1: Naive (top) vs. index-based (bottom) join sampling

概率分布估计

概率图模型-贝叶斯网络

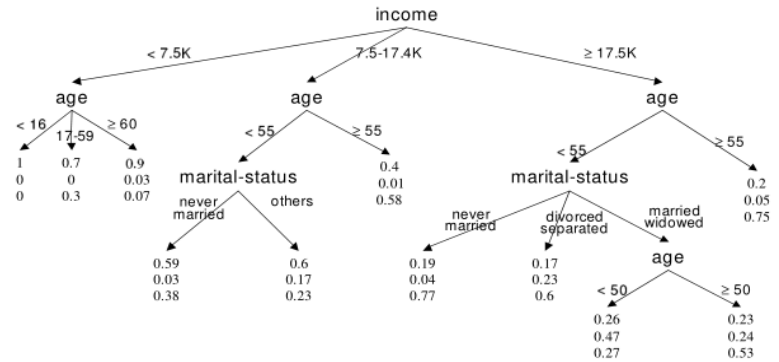
- 概率图模型
  - 在图的基础上表示概率分布的模型
- 箭头指向代表依赖关系

- CPD：条件概率分布
  - 模型



(a)

Figure 2: (a) A Bayesian network for the census domain. *Age* and *Marital-Status*.



(b)

(b) A tree-structured CPD for the *Children* node given its parents *Income*,

单表多属性概率分布估计

AutoRegressive 自回归模型 Naru

- Naru
  - 在自编码器的基础上利用MLP作为Mask层

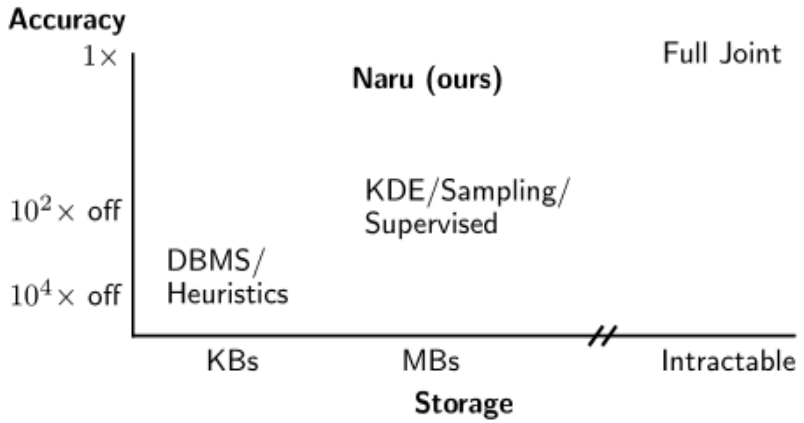
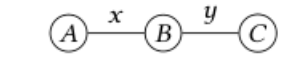


Figure 1: Approximating the joint data distribution in full, Naru enjoys high estimation accuracy and space efficiency.

多表概率分布估计

Nerucard (AutoRegressive Models)

- 查询中涉及所有表格（Q1）
  - DeepDB 找关联性，按需求学习SPN

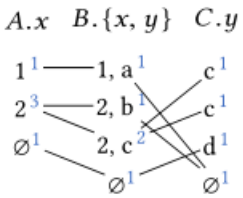


$A.x$
1
2

$B.x$	$B.y$
1	a
2	b
2	c

$C.y$
c
c
d

(a) Schema and base tables



(b) Join counts

$A.x$	$B.x$	$\mathcal{F}_{B.x}$	$B.y$	$C.y$	$\mathcal{F}_{C.y}$	$\mathbb{1}_A$	$\mathbb{1}_B$	$\mathbb{1}_C$
1	1	1	a	∅	1	1	1	0
2	2	2	b	∅	1	1	1	0
2	2	2	c	c	2	1	1	1
2	2	2	c	c	2	1	1	1
∅	∅	1	∅	d	1	0	0	1

(c) Full outer join, with virtual columns in blue

```
-- In full join, |A.x=2|=3.
-- Q1. True answer is 2.
SELECT COUNT(*)
FROM A JOIN B ON x
       JOIN C ON y
WHERE A.x = 2;
-- Q2. True answer is 1.
SELECT COUNT(*)
FROM A WHERE A.x = 2;
```

(d) Schema subsetting

- 查询中缺失部分表格（Q2）
  - （a）由三个表及其连接键组成的连接模式。

多表概率分布估计

Nerucard (AutoRegressive Models)

- 评估质量
  - DeepDB 找关联性，按需求学习SPN

- 实例
  - (a) 由三个表及其连接键组成的连接模式。
    (非连接列省略)
  - (b)
  - (c)
  - (d)

Table 2: JOB-light, estimation errors. Lowest errors are bolded.

Estimator	Size	Median	95th	99th	Max
Postgres	70 KB	7.97	797	$3 \cdot 10^3$	10^3
IBJS	–	1.48	10^3	10^3	10^4
MSCN	2.7 MB	3.01	136	$1 \cdot 10^3$	10^3
E2E (quoting [38])	N/A	3.51	139	244	272
DeepDB	3.7 MB	1.32	4.90	33.7	72.0
DeepDB-large	32 MB	**1.19**	**4.66**	35.0	39.5
NeuroCard	3.8 MB	1.57	5.91	**8.48**	**8.51**