AI4DB:AI Meets Query Optimization

Luming Sun, Cuiping Li, Tao Ji, Hong Chen @ DWBI
School of Information, Renmin University

2021/11/10

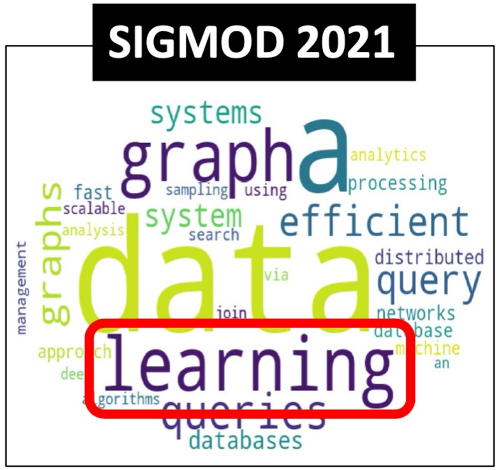
AI4DB

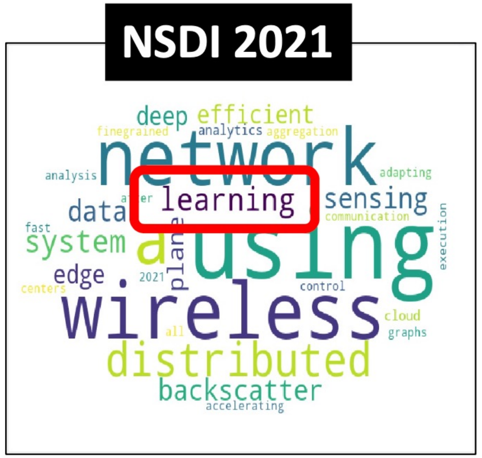
Database system & Data Management Powered by Artificial Intelligence

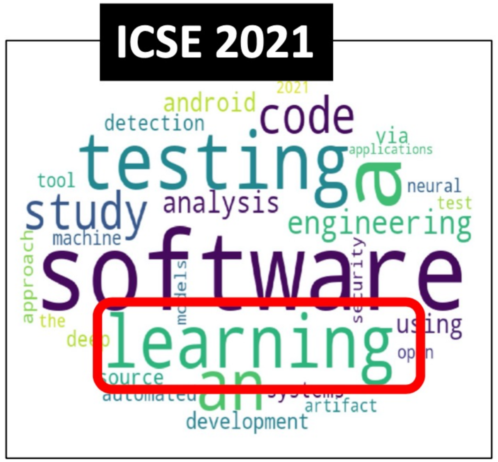
AI + DB

- AI:
  - Advance in CV, NLP, …
  - Statistic, learning, inference, planning
- DB:
  - Static
  - Data volume, sophisticated workload, hardware
- AI4DB
  - Goal: Reduce labor costs & Improve system performance
  - Query workload, data distribution, hardware features, history performance

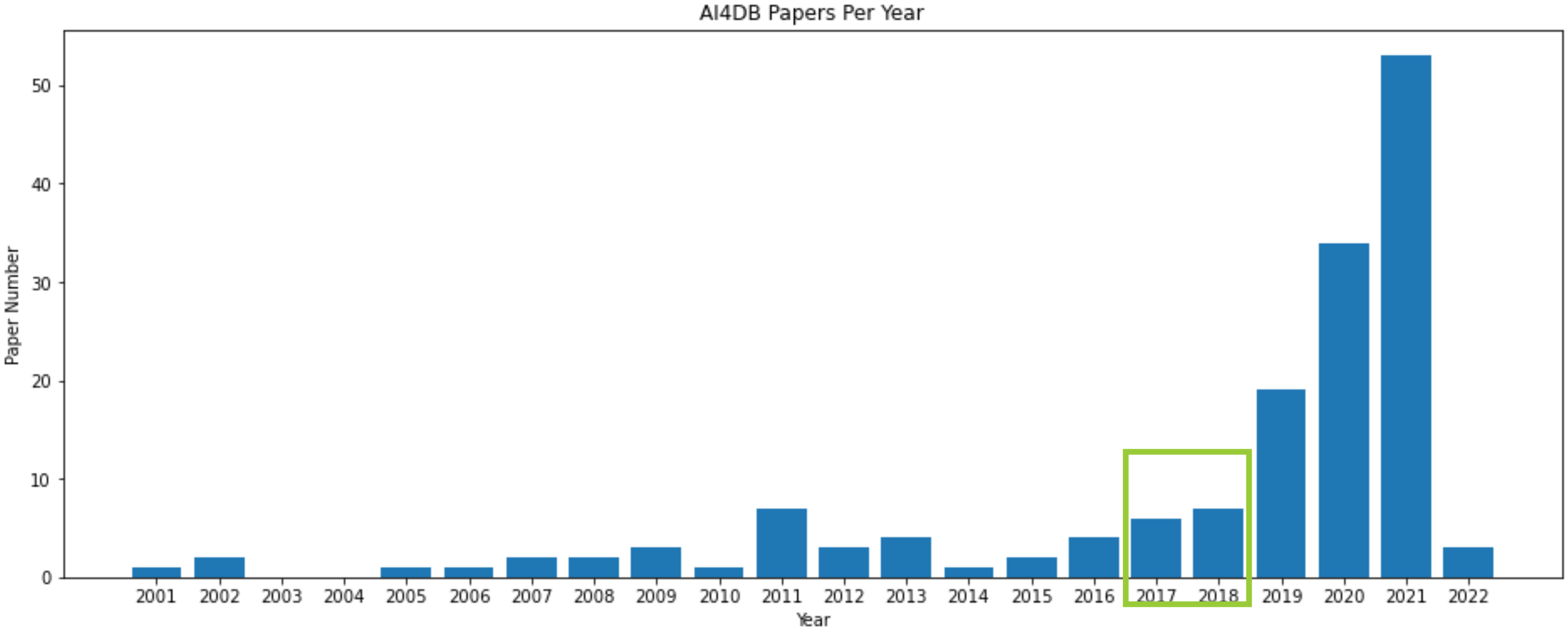
Artificial Intelligence for System



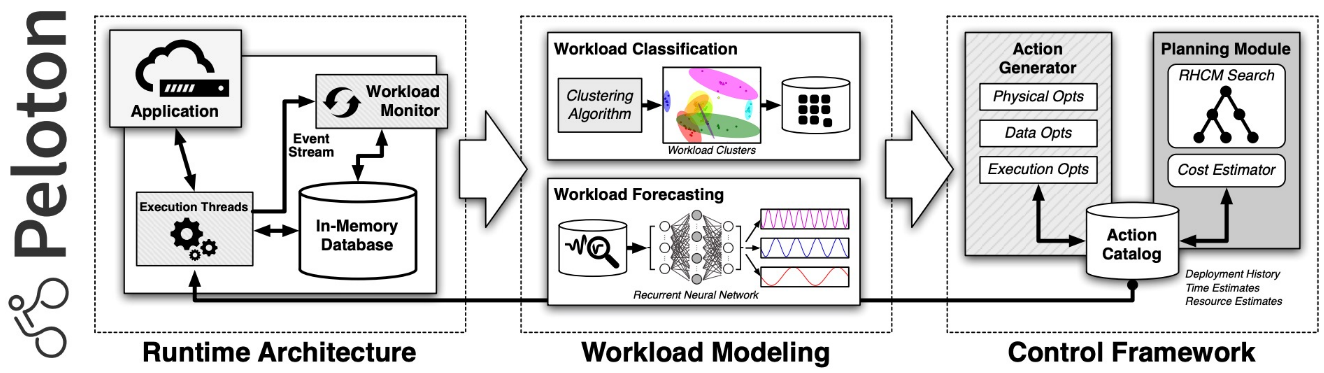




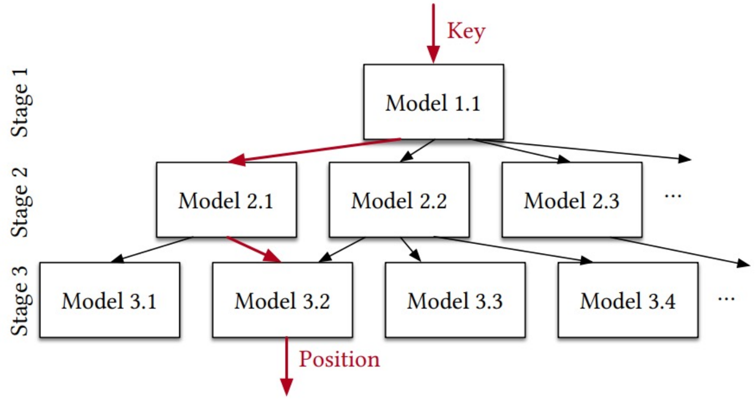
AI4DB Paper List https://github.com/LumingSun/ML4DB-paper-list



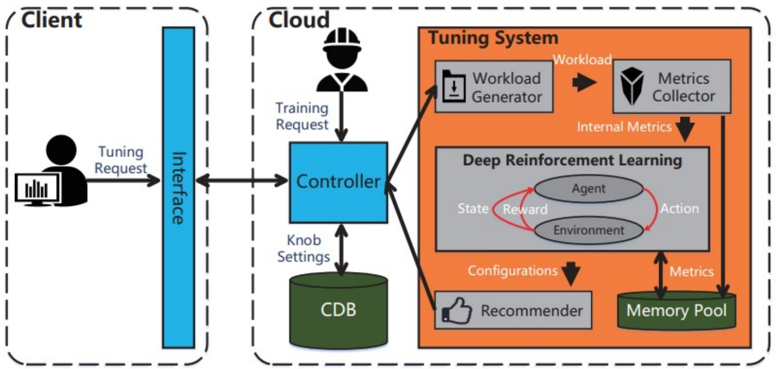
AI4DB



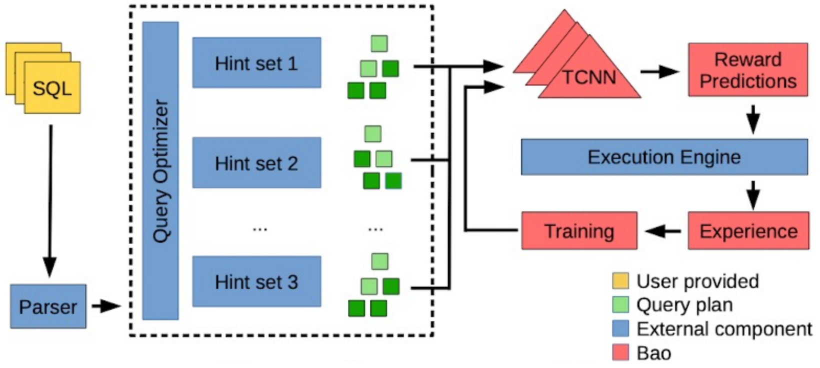
Andy Pavlo, et all. CIDR'17



Tim Kraska, et all. SIGMOD'18

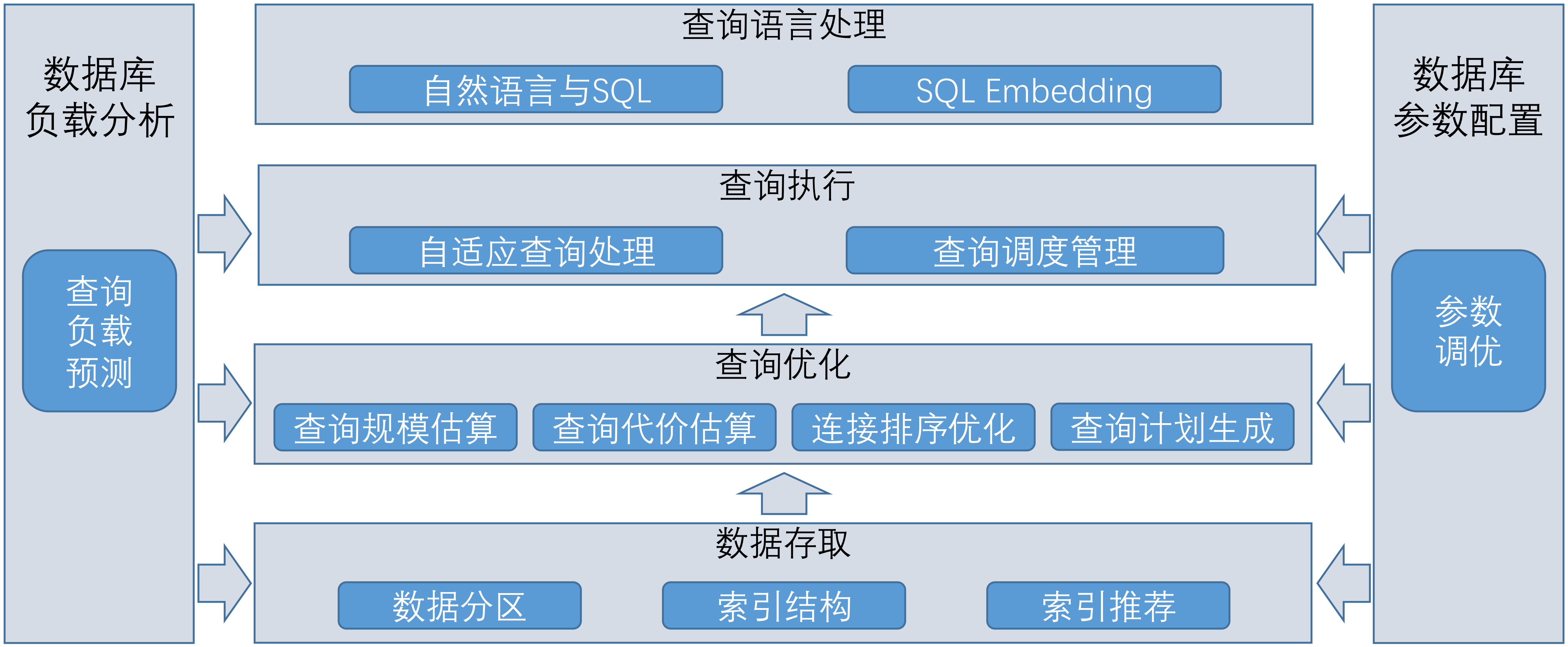


Guoliang Li, et all. SIGMOD'19



Ryan Marcus, et all. VLDB'21

AI4DB



孙路明, 张少敏, 姬涛, 李翠平, & 陈红. (2019). 人工智能赋能的数据管理技术研究. *软件学报*, *31*(3), 600-619.

MOSE: A Monotonic Selectivity Estimator Using Learned CDF

What is Cardinality/Selectivity

Q: SELECT *
FROM $Student$ WHERE age > 15
AND $gender$ = ‘Male’;

Card(Q) = 4
Sel(Q) = Card(Q) / #row
=4/9 = 0.444

age	gender	GPA
21	Female	3.42
20	Male	2.58
18	Female	2.79
20	Female	3.98
24	Female	3.71
20	Male	3.50
21	Male	4.0
23	Female	3.66
22	Male	3.12

Why Cardinality/Selectivity Estimation

2014

IS QUERY OPTIMIZATION A "SOLVED" PROBLEM?

Databases

Guy Lohman, IBM DB2 (40 years' experience)

"The root of all evil, the Achilles Heel of query optimization, is the estimation of the size of intermediate results, known as cardinalities."

2015

How Good Are Query Optimizers, Really?

"We have also shown that relational database systems produce large estimation errors that quickly grow as the number of joins increases, and that these errors are usually the reason for bad plans."

2018 - 2021

Multiple research groups consistently working on learned selectivity estimators

Massachusetts Institute of Technology

Berkeley UNIVERSITY OF CALIFORNIA

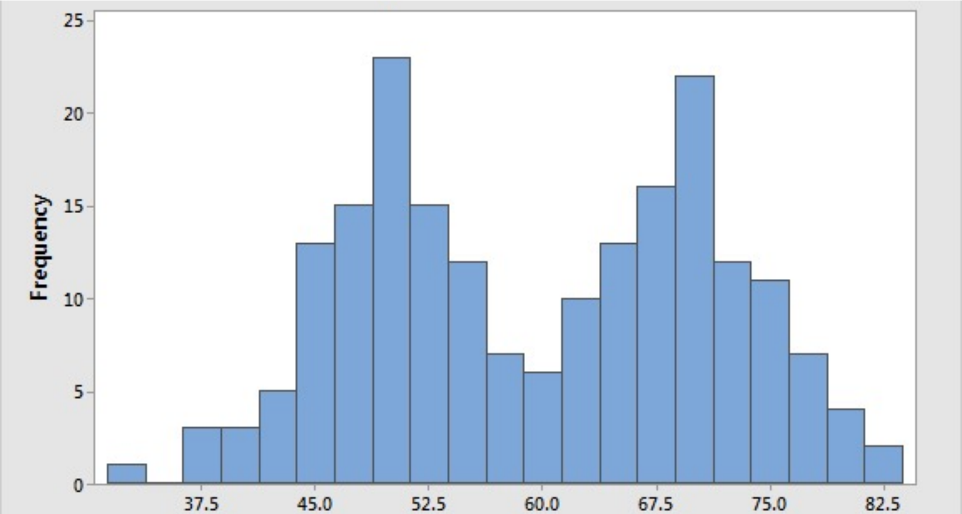
Technical University of Munich

Microsoft

Alibaba Group

Traditional Selectivity Estimation Methods

- Histograms



- Sampling
- Most Common Values (MVC)

How Learned Selectivity Estimators Work

- Methodology 1: **Query-driven**
  - Key Idea: Model as a Regression problem

Query $\xrightarrow{\text{Feature Engineering}}$ Feature Vector $\xrightarrow{\text{ML Models}}$ Selectivity

- Methodology 2: **Data-driven**
  - Key Idea**:** Model as a Joint Distribution Estimation problem

A_1	A_2	…	A_n

$\Rightarrow$ $\mathbf{P(A_{1,} A_{1,} \ldots , A_n)}$

Methodology 1: Query-Driven

Training

Query Pool | Labels

$Q1$: SELECT *FROM *Student* WHERE *age* > 20; 4
$Q2$: SELECT *FROM *Student* WHERE *GPA* < 3.5 AND *GPA* > 3.0; 2
$Q3$: SELECT *FROM *Student* WHERE *gender* = ‘Female’; 5
… …

Featurize

$Q1$: <0.8, 1.0, 0.0, 0.0, 0.0, 1.0> 4
$Q2$: <0.0, 1.0, 0.0, 1.0, 0.3, 0.6> 2
$Q3$: <0.0, 1.0, 1.0, 1.0, 0.0, 1.0> 5
…

Train

Regression Model

Inference

Q: SELECT * FROM *Student*
WHERE *age* > 15 AND gender = “Male”

Featurize

Q: <0.0, 0.9, 0.0, 1.0, 0.8, 1.0>

Inference

Estimation: 4!

Methodology 1: Query-Driven

- **MSCN** [Kipf, A et all. CIDR 19]

  Neural Network + Sampling
- **LW-XGB** [Dutt, A et all. VLDB 19]

  XGBoost+ Histogram
- **LW-NN** [Dutt, A et all. VLDB 19]

  Neural Network + Histogram
- **QuickSel** [Yongjoo, P et all. SIGMOD 20]

  Mixture Model

Shortcomings:

- Require large amount of training data
- Violate basic rule of selectivity estimation

  - Monotonicity: $sel(1 < x < 2) \leq sel(1 < x < 3)$

  - Validity: $sel(1 < x < 0) = 0$

  - Consistency: $sel(1 < x \leq 2) + sel(2 < x < 3)$

  $= sel(1 < x < 3)$

How Learned Selectivity Estimators Work

- Methodology 1: **Query-driven**
  - Key Idea: Model as a Regression problem

$$\text{Query} \xrightarrow{\text{Feature Engineering}} \text{Feature Vector} \xrightarrow{\text{ML Models}} \text{Selectivity}$$

- Methodology 2: **Data-driven**
  - Key Idea**:** Model as a Joint Distribution Estimation problem

A_1	A_2	…	A_n

$$\Rightarrow \quad \mathbf{P(A_{1,} A_{1,} \ldots , A_n)}$$

Methodology 2: Data-Driven

Training

age	gender	GPA
21	Female	3.42
20	Male	2.58
18	Female	2.79
20	Female	3.98
24	Female	3.71
20	Male	3.50
21	Male	4.0
23	Female	3.66
22	Male	3.12

Train

P(age, gender, GPA)

Joint Distribution Estimation Model

Inference

Q: SELECT * FROM $Student$
WHERE age > 15 AND gender = “Male”

$$P(age > 20, \text{gender} = \text{“Male”})$$

Inference

Estimation: 4!

Methodology 2: Data-Driven

- **Naru** [Yang, Z et all. VLDB 20]

  Auto-regressive Model

- **DeepDB** [Hilprecht, B et all. VLDB 20]

  Sum Product Network

- **FLAT** [Rong, Z et all. VLDB 2021]

  Graphical Model

Shortcomings:

- Heavy costs on model training and inference
- Violate basic rule of selectivity estimation

  - Monotonicity: $sel(1 < x < 2) \leq sel(1 < x < 3)$

  - Validity: $sel(1 < x < 0) = 0$

  - Consistency: $sel(1 < x \leq 2) + sel(2 < x < 3)$

    $= sel(1 < x < 3)$

  - **Stability**: $sel(1 < x < 2) = sel(1 < x < 2)$

MOSE: A Monotonic Selectivity Estimator Using Learned CDF

- **Aim**

  Reliable, accurate and efficient learned selectivity estimator

- **Problem Settings**

  Multi-dimensional predicates on single table

- **Methodology**

  Query-based

- **Key observation**

  The joint cumulative distribution function (CDF) of the data in a table can be used to compute the selectivity for query range predicates

CDF to Selectivity

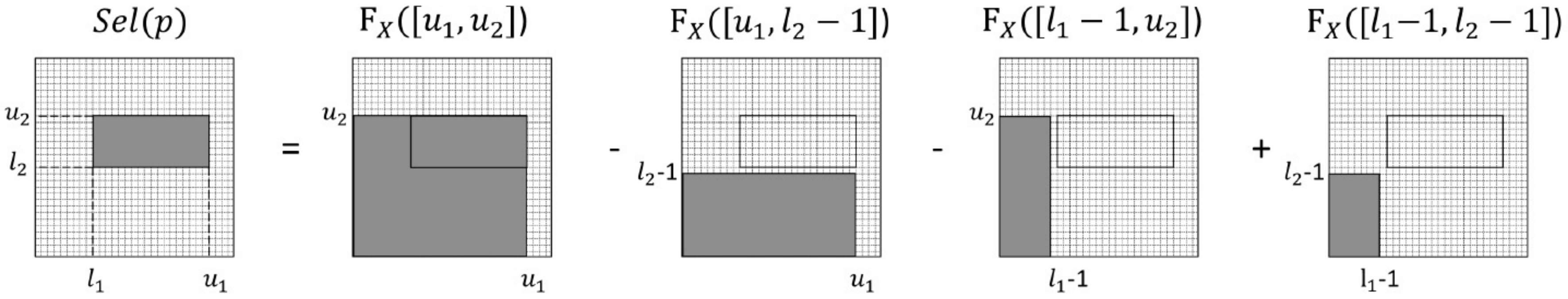
- Multi-dimensional CDF:

  Random variable $X = (X_1, X_2, \dots, X_d)$,

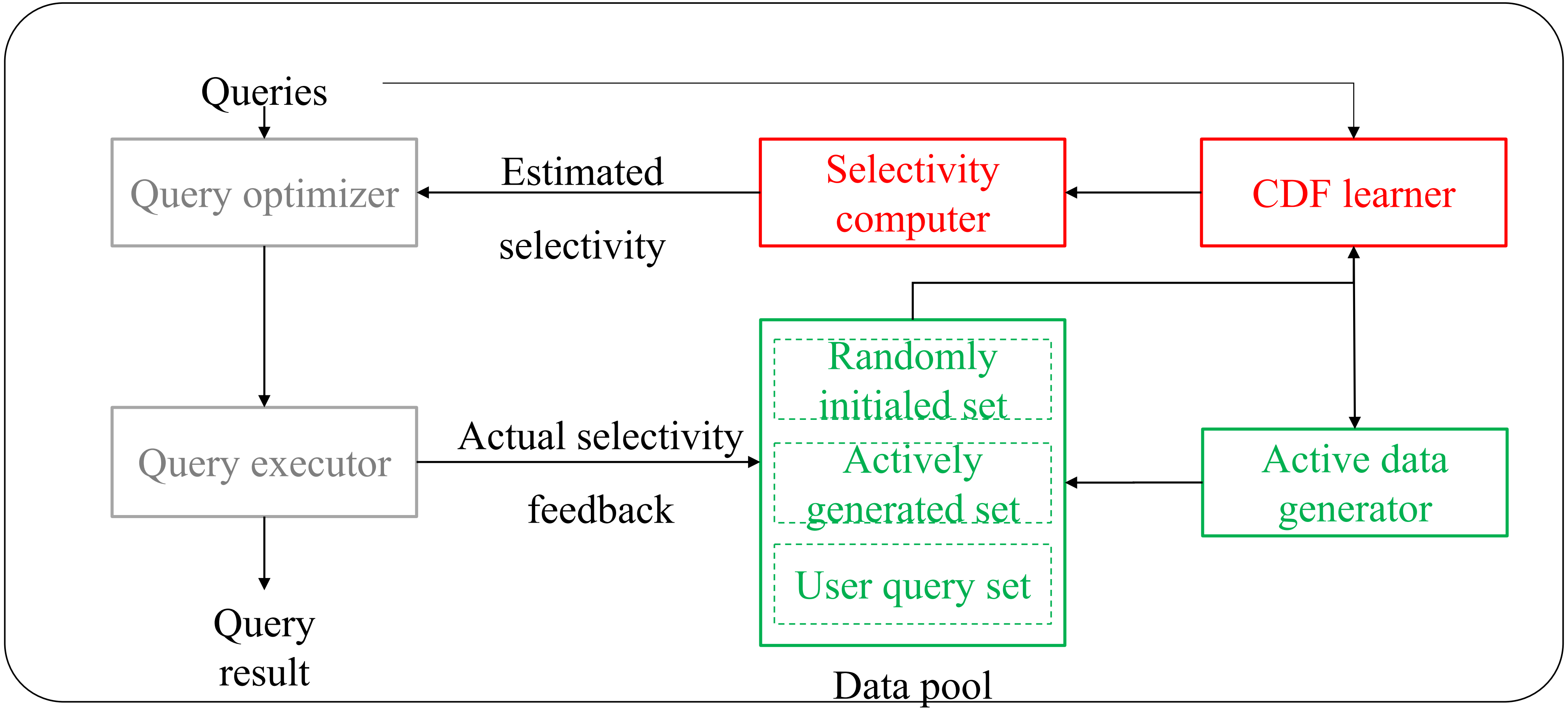
  CDF: $F_X(x) = \Pr(X_1 \le x_1, X_2 \le x_2, \dots, X_d \le x_d)$
- CDF to selectivity

$$sel(p) = \sum_{\forall x_i \in \{l_i - 1, u_i\}} \left\{ \left(\prod_{i=1}^{d} s(i) \right) F_X(x) \right\}$$

$$\begin{aligned} sel(p) = &F_X([u_1, u_2]) - F_X([u_1, l_2 - 1]) \\ &- F_X([l_1 - 1, u_2]) + F_X([l_1 - 1, l_2 - 1]) \end{aligned}$$

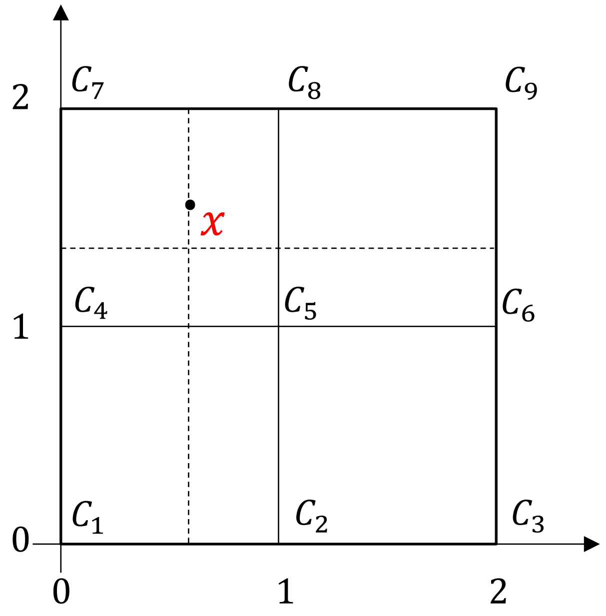


MOSE Overview



CDF Learner

- **Monotonic Lattice Regression Model**



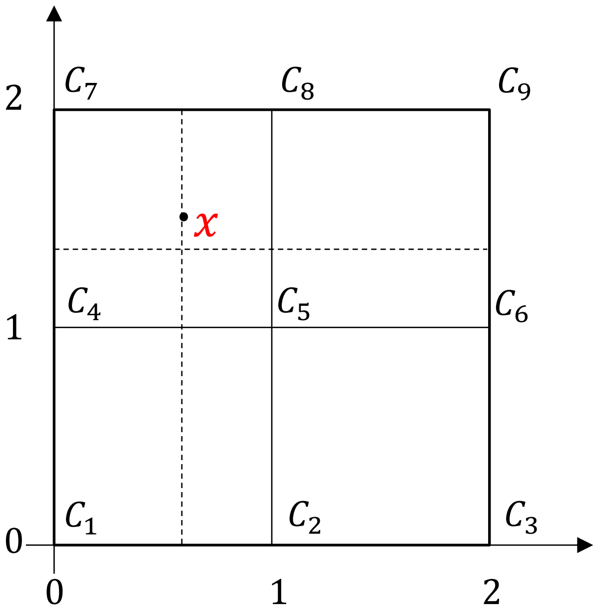
C: Lattice Node
Theta: Lattice Parameter

$$\hat{y} = F_X(x) = \sum_{j=1}^{m} \phi(x)_j \theta_j, \quad \sum_{j=1}^{m} \phi(x)_j C_j = x, \quad \sum_{j=1}^{m} \phi(x)_j = 1.$$

$$\begin{aligned} \theta &= \arg\min_{\theta \in \mathbb{R}^m} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2 \\ &= \arg\min_{\theta \in \mathbb{R}^m} \sum_{i=1}^{n} \left(\left(\sum_{j=1}^{m} \phi(x)_{ij} \theta_j \right) - y_i \right)^2. \end{aligned}$$

$$\theta = \arg\min_{\theta \in \mathbb{R}^m} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2, \, s.t. \; A\theta^T \leq 0.$$

Monotonic Constrain

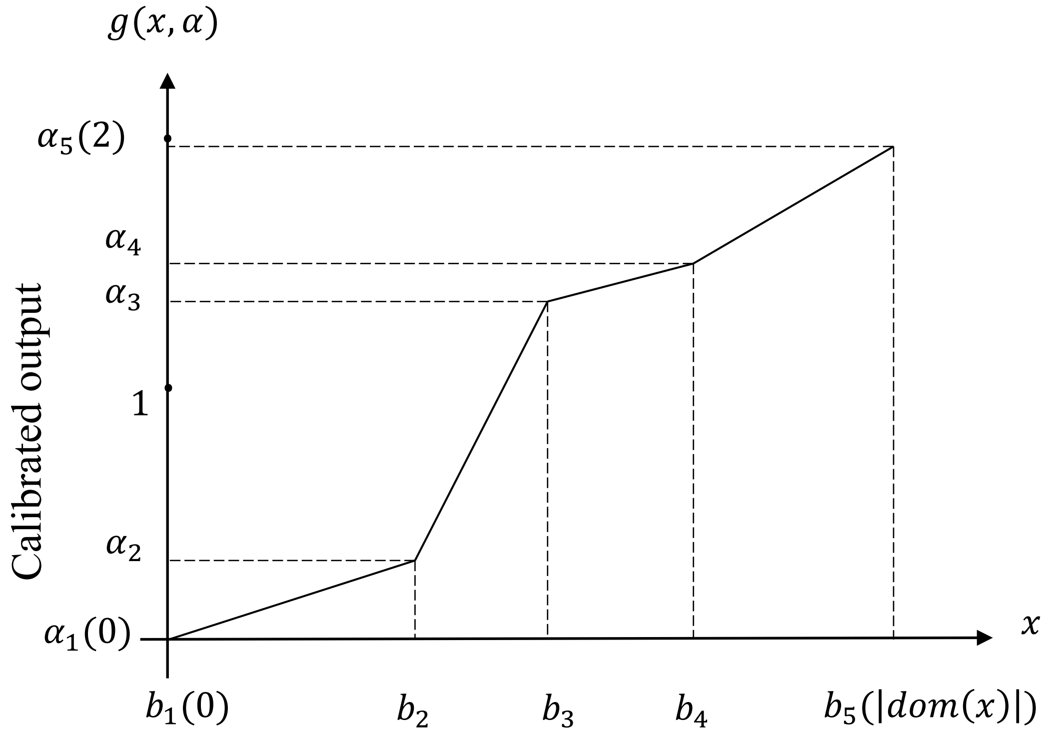


C: Lattice Node
Theta: Lattice Parameter

$$\theta = \arg\min_{\theta \in \mathbb{R}^m} \sum_{i=1}^{n} \left(\hat{y}_i - y_i\right)^2, s.t.\ A\theta^T \le 0.$$

$$\begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \\ \theta_7 \\ \theta_8 \\ \theta_9 \end{bmatrix} \le \mathbf{0} \Leftrightarrow \begin{matrix} \theta_1 \le \theta_2 \\ \theta_2 \le \theta_3 \\ \theta_4 \le \theta_5 \\ \theta_5 \le \theta_6 \\ \theta_7 \le \theta_8 \\ \theta_8 \le \theta_9 \\ \theta_1 \le \theta_4 \\ \theta_2 \le \theta_5 \\ \theta_3 \le \theta_6 \\ \theta_4 \le \theta_7 \\ \theta_5 \le \theta_8 \\ \theta_6 \le \theta_9 \end{matrix}$$

Attribute-Aware Calibration



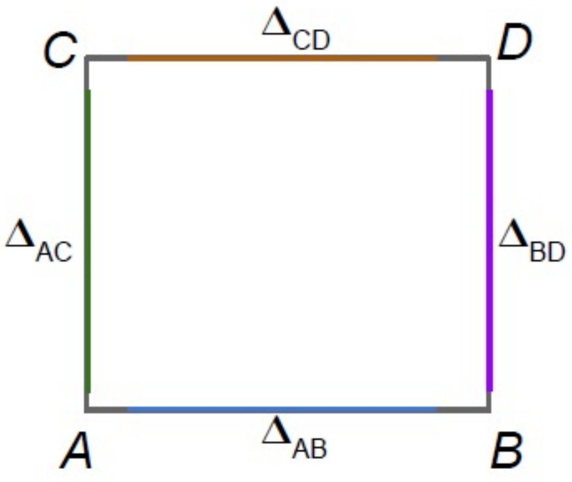
$$\theta, \alpha = \arg\min_{\theta,\alpha} \sum_{i=1}^{n} \left(\left(\sum_{j=1}^{m} \phi(g(x,\alpha))_{ij}\theta_j \right) - y_i \right)^2 + \lambda R(\theta)$$

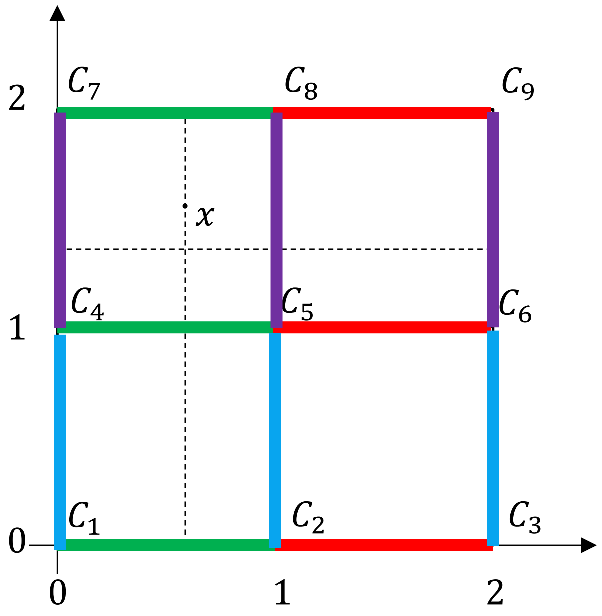
$$s.t.\ \ A\theta^T \leq 0 \ and\ B\alpha^T \leq 0\ ,$$

Cell-Wise Regularizer

Graph Laplacian:
flatter function

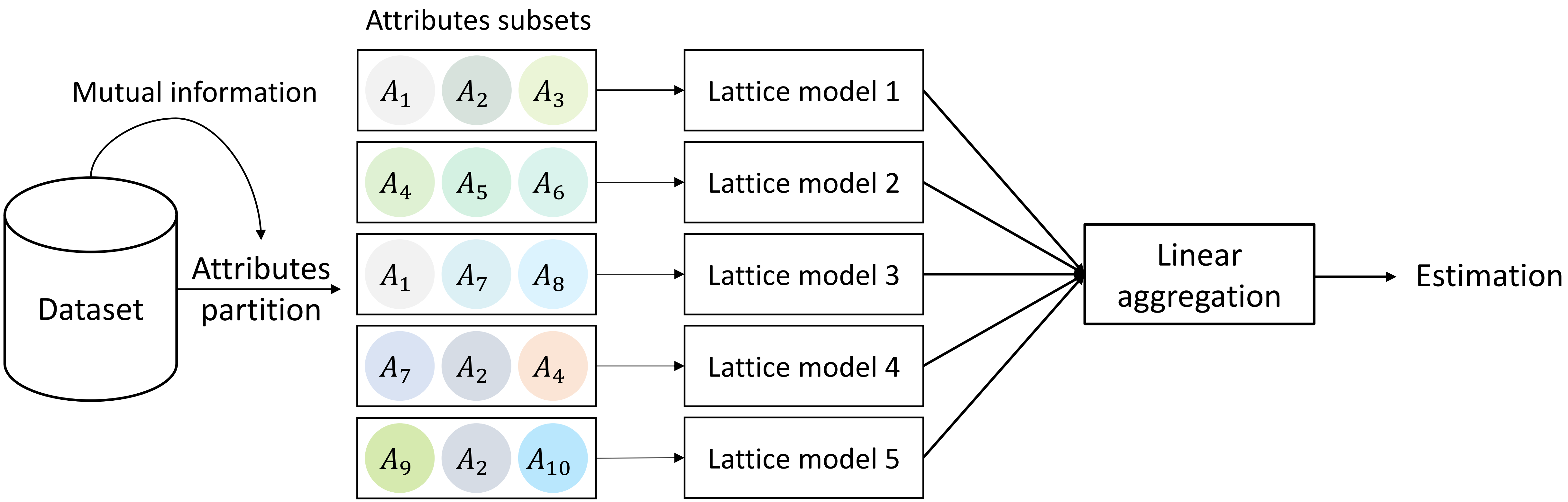
Penalizes:
$(A\text{-}C)^2 + (A\text{-}B)^2 + (C\text{-}D)^2 + (B\text{-}D)^2$
$= \Delta_{AC}^{\,2} + \Delta_{AB}^{\,2} + \Delta_{CD}^{\,2} + \Delta_{BD}^{\,2}$





$$R(\theta) = \sum_{i=1}^{d} \sum_{\substack{C_r, C_s \text{ such that} \\ C_r \text{ and } C_s \text{ adjacent on dimension } i}} H_i(C_r, C_s)(\theta_r - \theta_s)^2$$

Lattice Ensemble



ACTIVE DATA GENERATOR

- Challenges:
  - Infinite query space (NOT pool based active learning)
  - Regression problem (NO model uncertainty)
- Solution
  - Picking the lattice cells that are most valuable or necessary to optimize
  - Two factors: (1) cell accuracy; (2) cell density
  - Weighted lattice sampling

Weighted Lattice Sampling

Algorithm 2: Active Data Generator

input : X_L is the initial labeled data,
θ is model weight of CDF learner,
$\mathcal{T}$ is the cost threshold of data collection,
ϵ is the cost function to label a data instance,
B is the number of data selected in one batch,
$\mathcal{P}$ is a function to calculate cell sampling weight

output: X is the labeled training data

$X \leftarrow X_L$ `// total training set`
$t \leftarrow 0$ `// initialize total cost`
while $t < \mathcal{T}$ **do**
  $\theta \leftarrow$ `TrainModelWith`(X)
  $Error \leftarrow$ `Evaluate`(θ, X)
  $P_{TopError} \leftarrow$ `Top`$(X, Error, K)$
  $\mathcal{P}_c \leftarrow X, P_{TopError}$
  $cells \leftarrow$ `WeightedSampling`$(\mathcal{P}_c, B)$
  $X_A \leftarrow$ `RandomPointGenerate`$(cells)$
  $X_{AL} \leftarrow$ `ExecuteQuery`(X_A)
  $t = t + \epsilon(X_A)$
  $X = X \cup X_{AL}$
return X

$$\mathcal{P}_c = \frac{1 + \omega k_c}{1 + M_c},$$

M_c: points in cell C

k_c: k points of C are in the TOP-K worst estimation

Accuracy

TABLE 2: Selectivity estimation accuracy on DMV

Estimator	Training data size				
	200	400	600	800	1000
LWM	0.03474	0.02576	0.01817	0.01758	0.01607
NN	0.04787	0.03082	0.02153	0.01803	0.01577
QuickSel	0.02151	0.01421	0.01296	0.01125	0.01027
MOSE	**0.00674**	**0.00543**	**0.00463**	**0.00429**	**0.00393**

TABLE 3: Selectivity estimation accuracy on Forest

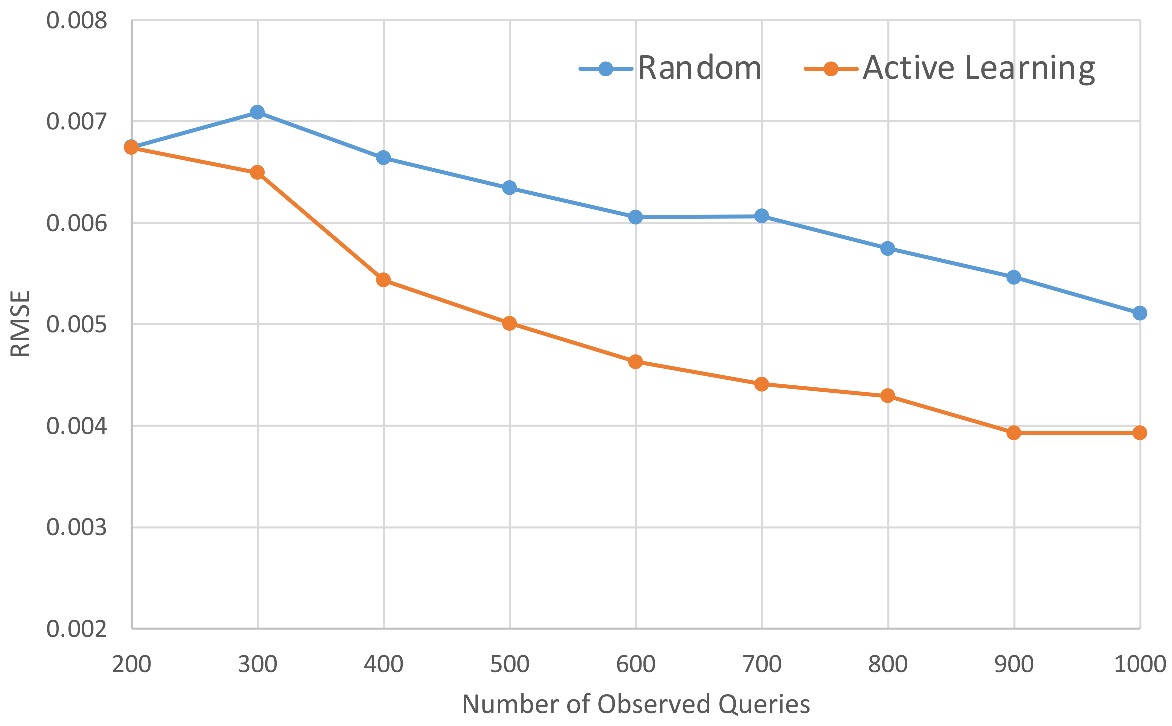
Estimator	Range predicates dimension				
	2D	4D	6D	8D	10D
AVI	0.23020	0.06069	0.01060	0.00240	0.000582
Sampling	0.00642	0.01164	0.00452	0.00946	0.000718
Naru	0.20113	0.59320	0.56103	0.10131	0.308497
LWM	0.03125	0.01573	0.00729	0.00229	0.000574
NN	0.00638	0.01226	0.00943	0.00240	0.000582
QuickSel	0.00470	0.00773	0.00382	0.83949	0.000590
MOSE	**0.00419**	**0.00544**	**0.00274**	**0.00223**	**0.000555**

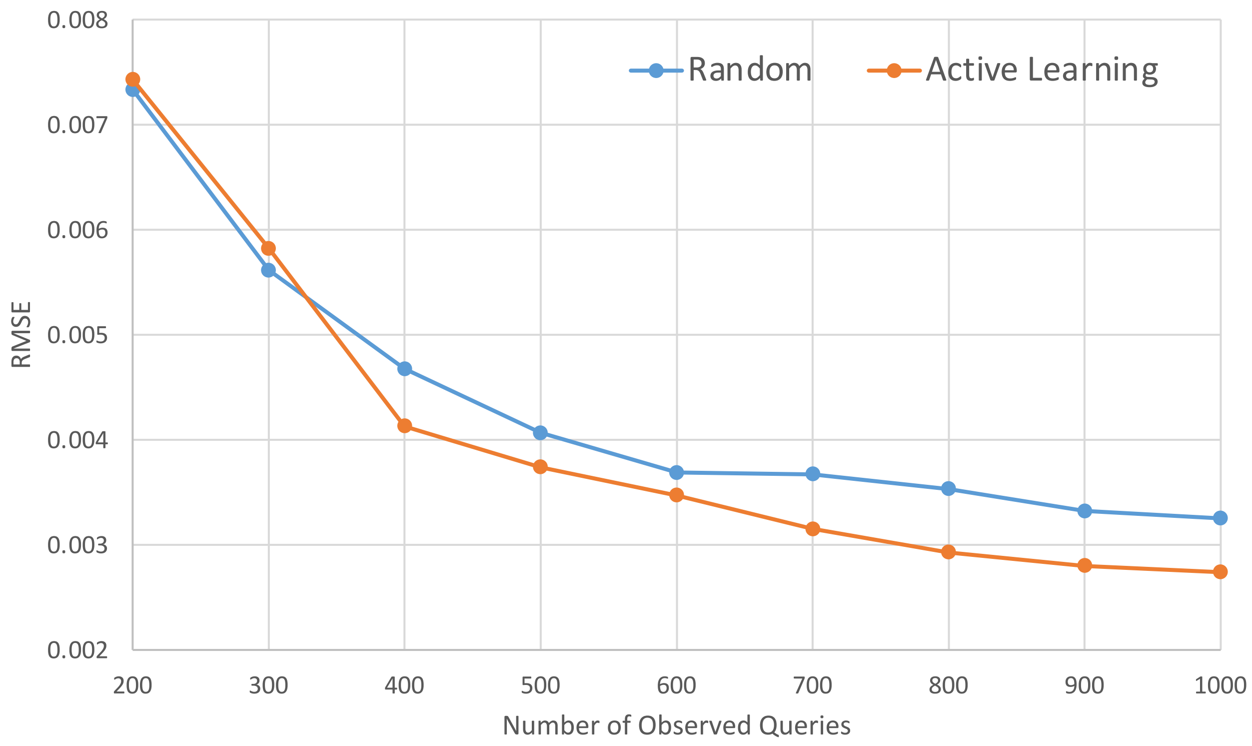
Ablation Experiments

TABLE 4: Combination of calibration and regularizer

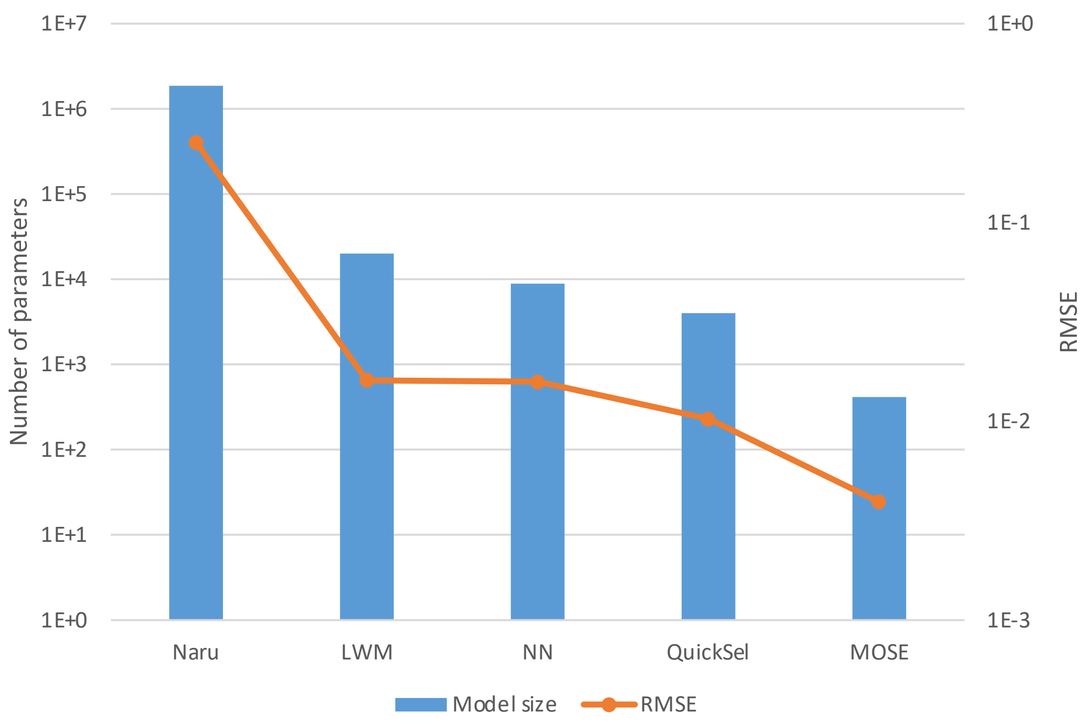
Combination method	RMSE
Laplacian regularizer + Uniform calibration	0.00713
Laplacian regularizer + A-A calibration	0.00540
C-W regularizer + Uniform calibration	0.00530
C-W regularizer + A-A calibration	**0.00393**

Active Learning

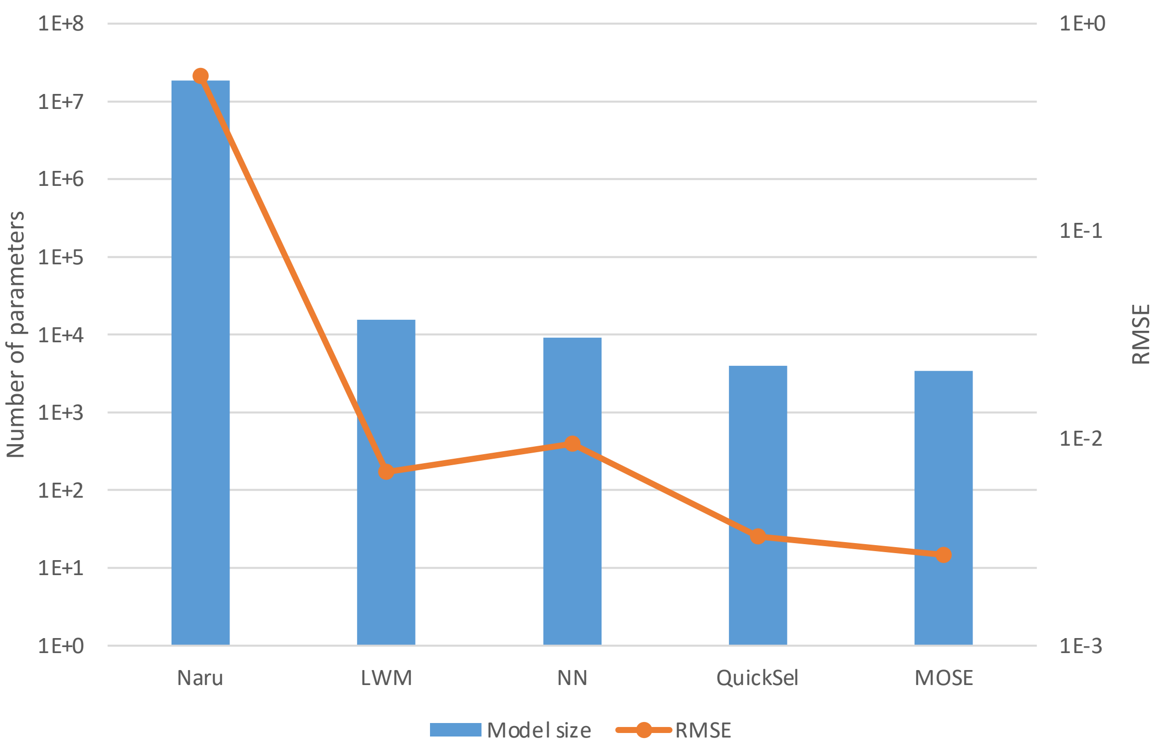




Model Size



1E+7
1E+6
1E+5
1E+4
1E+3
1E+2
1E+1
1E+0
Number of parameters
1E+0
1E-1
1E-2
1E-3
RMSE
Naru
LWM
NN
QuickSel
MOSE
Model size
RMSE



1E+8
1E+7
1E+6
1E+5
1E+4
1E+3
1E+2
1E+1
1E+0
Number of parameters
1E+0
1E-1
1E-2
1E-3
RMSE
Naru
LWM
NN
QuickSel
MOSE
Model size
RMSE

Summary

- Reliability: CDF --> selectivity: reliable
- Cell-wise regularizer + attribute-aware calibration: accurate
- Lattice ensemble based on mutual-information: efficient (model training)
- Active data generator: efficient (data collecting)
- Results:
  - Up to 62% less error
  - 1/15 number of parameters
  - 3.29x speedup

Thanks!

sunluming@ruc.edu.cn
https://lumingsun.github.io/