GPU-Accelerated Join Selectivity Estimation using KDE Models

Paper:

Estimating Join Selectivities using Bandwidth-Optimized Kernel Density Models,

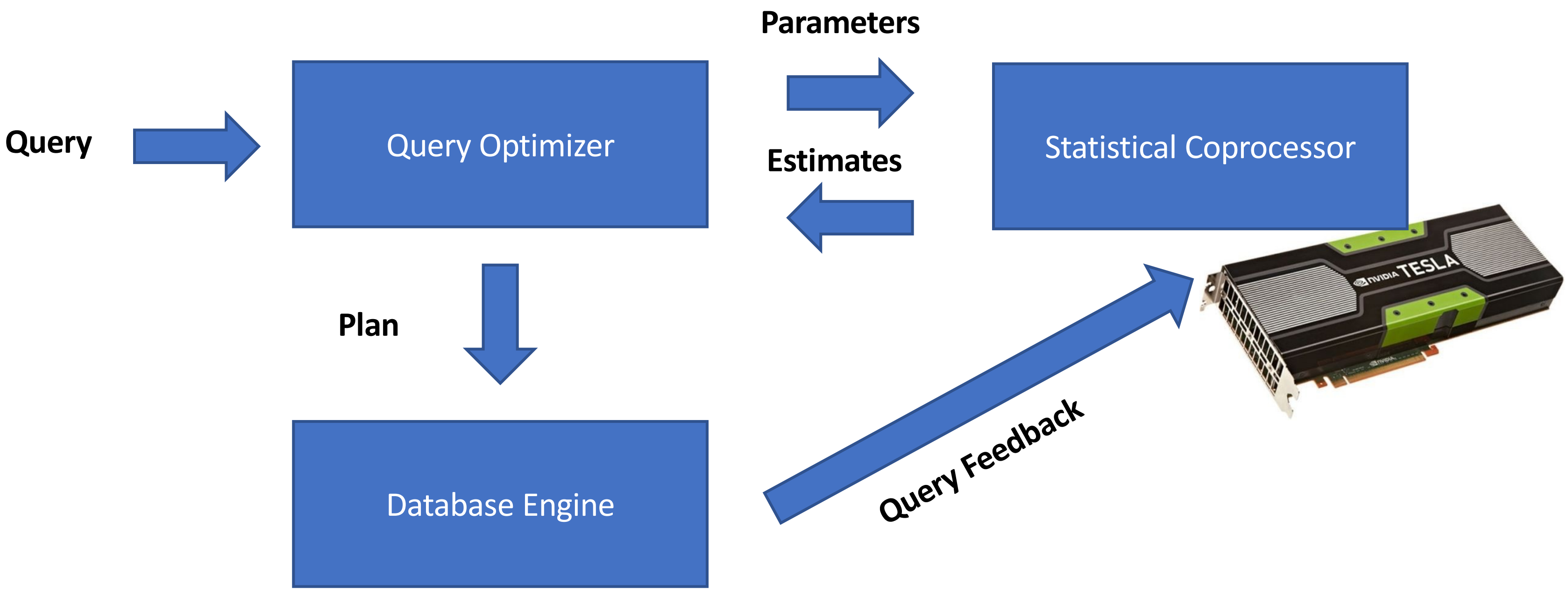
Martin Kiefer, Max Heimel, Sebastian Breß, Volker Markl

PVLDB, Volume 10 Issue 13, September 2017

Technische Universität Berlin

German Research Center for Artificial Intelligence

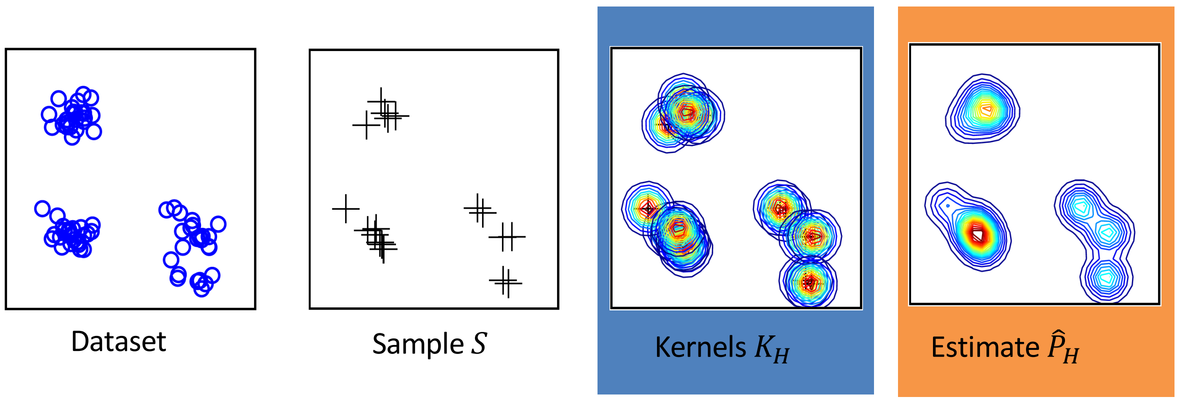
GPU-Accelerated **K**ernel **D**ensity **E**stimation for Join Selectivity Estimation



Background: Kernel Density Estimators

Average… … over the kernel contributions

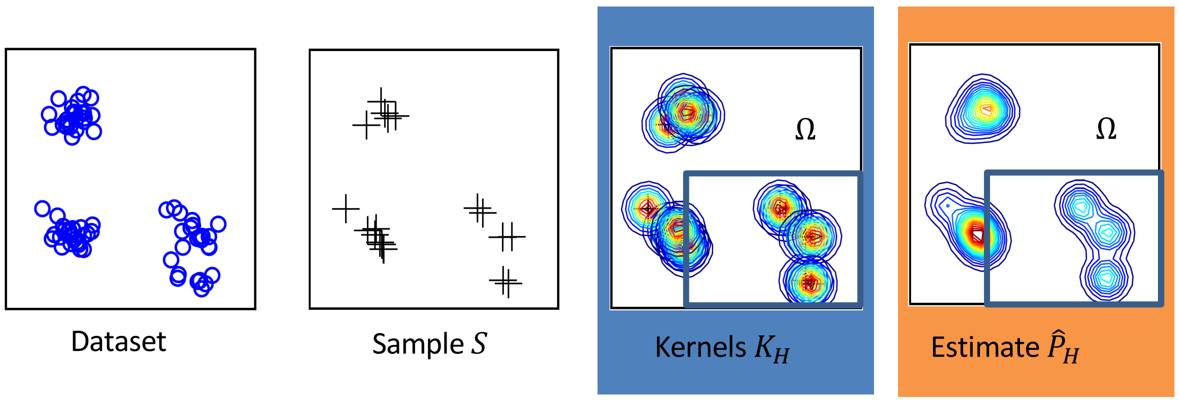
$$\hat{P}_H(\vec{x}) = \frac{1}{|S|} \sum_{i=1}^{|S|} K_H(s_i, \vec{x})$$



Background: Kernel Density Estimators

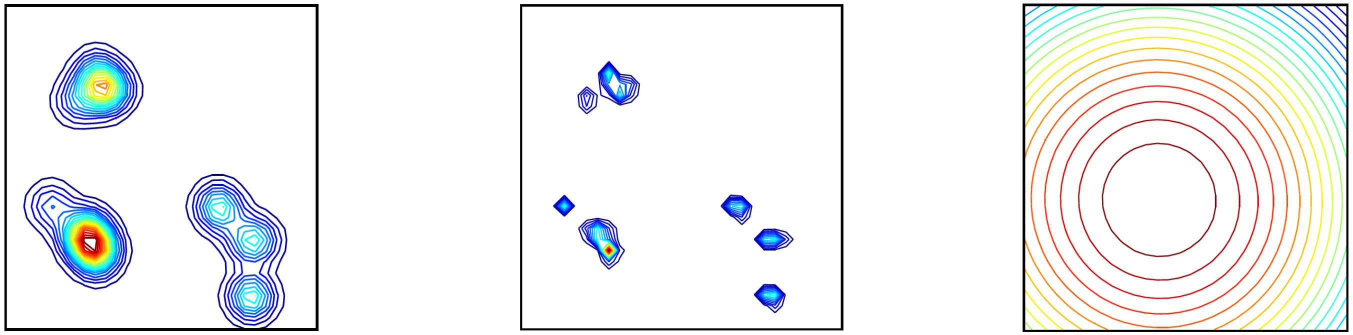
Average… … over the kernel contributions

$$\mathrm{sel}(\Omega) = \frac{1}{|S|} \sum_{i=1}^{|S|} \int_{\Omega} K_H(s_i, \vec{x})\, d\vec{x}$$



Background: Kernel Density Estimators for Multi-Dimensional Selectivity Estimation [1]

The bandwidth matrix H controls the smoothing applied on the sample



Good fit Overfit Underfit

- Range selections over base tables
- Bandwidth optimization based on the estimation error
- Easy model maintenance

[1] Self-Tuning, GPU-Accelerated Kernel Density Models for Multidimensional Selectivity Estimation, SIGMOD'15

The Problem:
Multi-Dimensional Join Selectivity Estimation

$$Q = \sigma_{c_1}(R_1) \bowtie_{R_1.A_1 = R_2.A_1} \sigma_{c_2}(R_2)$$

- and generalization to multiple joins
- **Databases:** Independence Assumption
    - Often violated
    - Introduce large errors, potentially bad query plans
- **Research:** Various Methods (e.g. Sampling, Sketches)
- **Our Approach:** Kernel Density Estimators

Why KDEs for Join Selectivities?

- Multivariate Estimator
- No independence assumption
- Hybrid between samples and histograms
  - Small bandwidth: Sample evaluation
  - Increasing bandwidth: More smoothing, increasing bucket sizes
  - Bandwidth optimization selects proper bandwidth

The Approach: Join and Base Table Models

$$Q = \sigma_{c_1}(R_1) \bowtie_{R_1.A_1 = R_2.A_1} \sigma_{c_2}(R_2)$$

Join KDE Model ($\boldsymbol{P}$)

Bandwidth H

Sample from $R_1 \bowtie_{R_1.A_1=R_2.A_1} R_2$

Compute: $P(c_1 \wedge c_2)$

Easy to evaluate, better estimates

Base Table KDE Model ($\boldsymbol{P_1}$)

Bandwidth H

Sample from R_1

Base Table KDE Model ($\boldsymbol{P_2}$)

Bandwidth H

Sample from R_2

Compute: $\sum_{v \in A} P_1(A_1 = v \wedge c_1) \cdot P_2(A_2 = v \wedge c_2)$

Support for base table and join selectivities
Easy to construct and to maintain

Table Model: Computation Components

$$Q = \sigma_{c_1}(R_1) \bowtie_{R_1.A_1 = R_2.A_1} \sigma_{c_2}(R_2)$$

Selectivity:

$$\frac{1}{s_1 \cdot s_2} \sum_{i=1,j=1}^{s_1,s_2} \hat{p}_1^{(i)}(c_1) \cdot \hat{p}_2^{(j)}(c_2) \cdot \hat{J}_{i,j}$$

Sum over cross product of two samples

Invariant Contributions: Contribution of sample points wrt. selection predicate

Cross Contribution: Distance function on join attributes of sample points

Table Model: Sample Pruning

$$\frac{1}{s_1 \cdot s_2} \sum_{i=1,j=1}^{s_1,s_2} \boxed{\hat{p}_1^{(i)}(c_1)} \cdot \hat{p}_2^{(j)}(c_2) \cdot \hat{J}_{i,j}$$

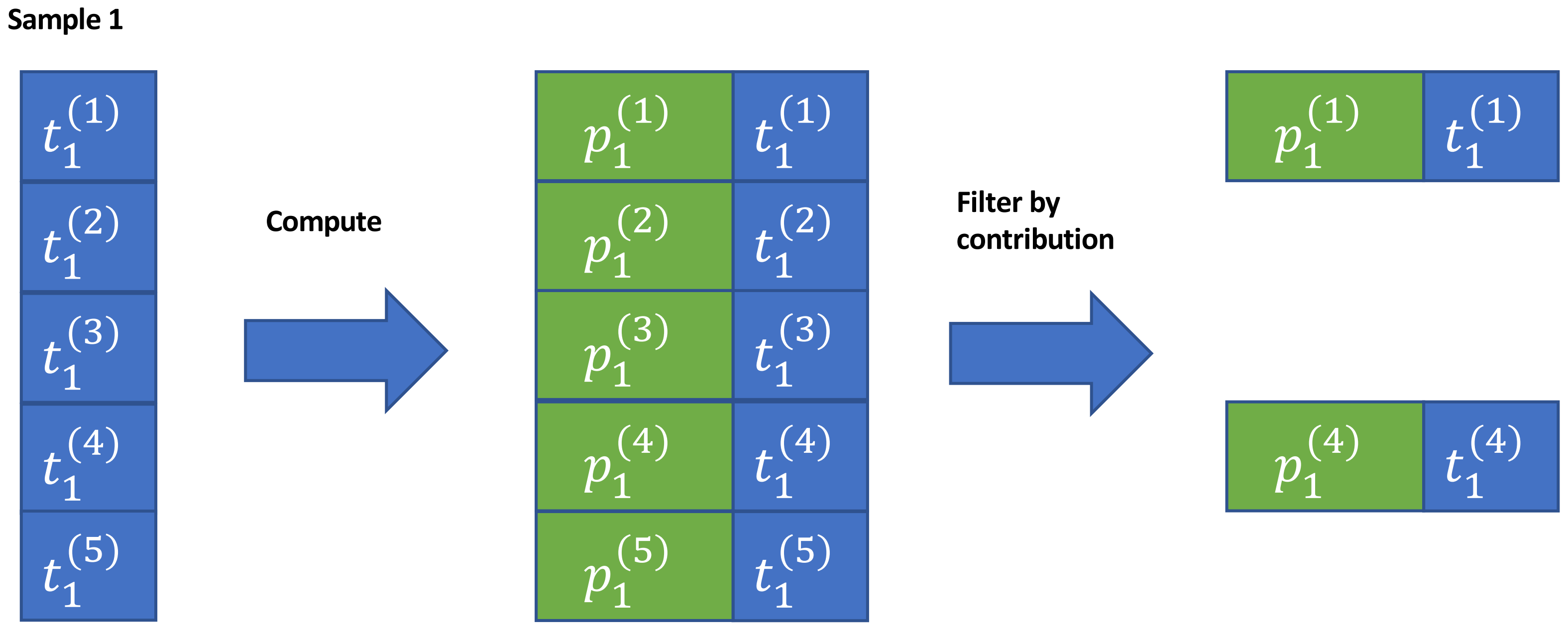
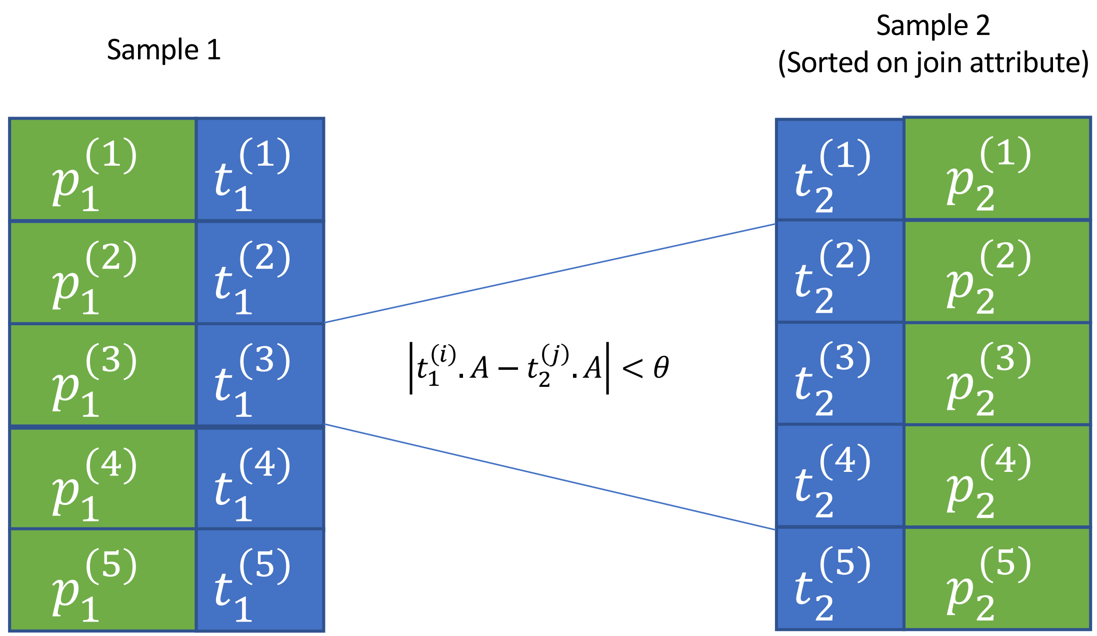


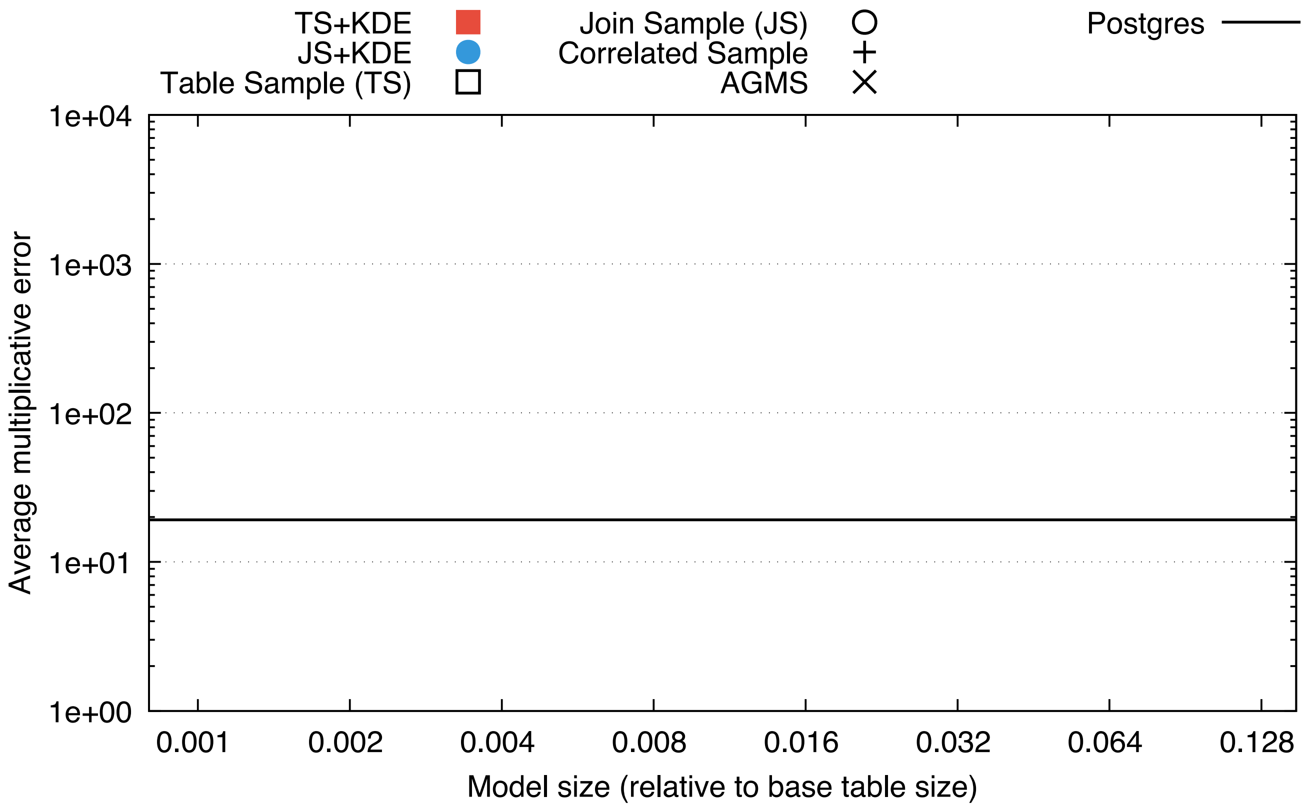
Table Model: Cross Pruning

$$\frac{1}{s_1 \cdot s_2} \sum_{i=1,j=1}^{s_1,s_2} \hat{p}_1^{(i)}(c_1) \cdot \hat{p}_2^{(j)}(c_2) \cdot \boxed{\hat{J}_{i,j}}$$



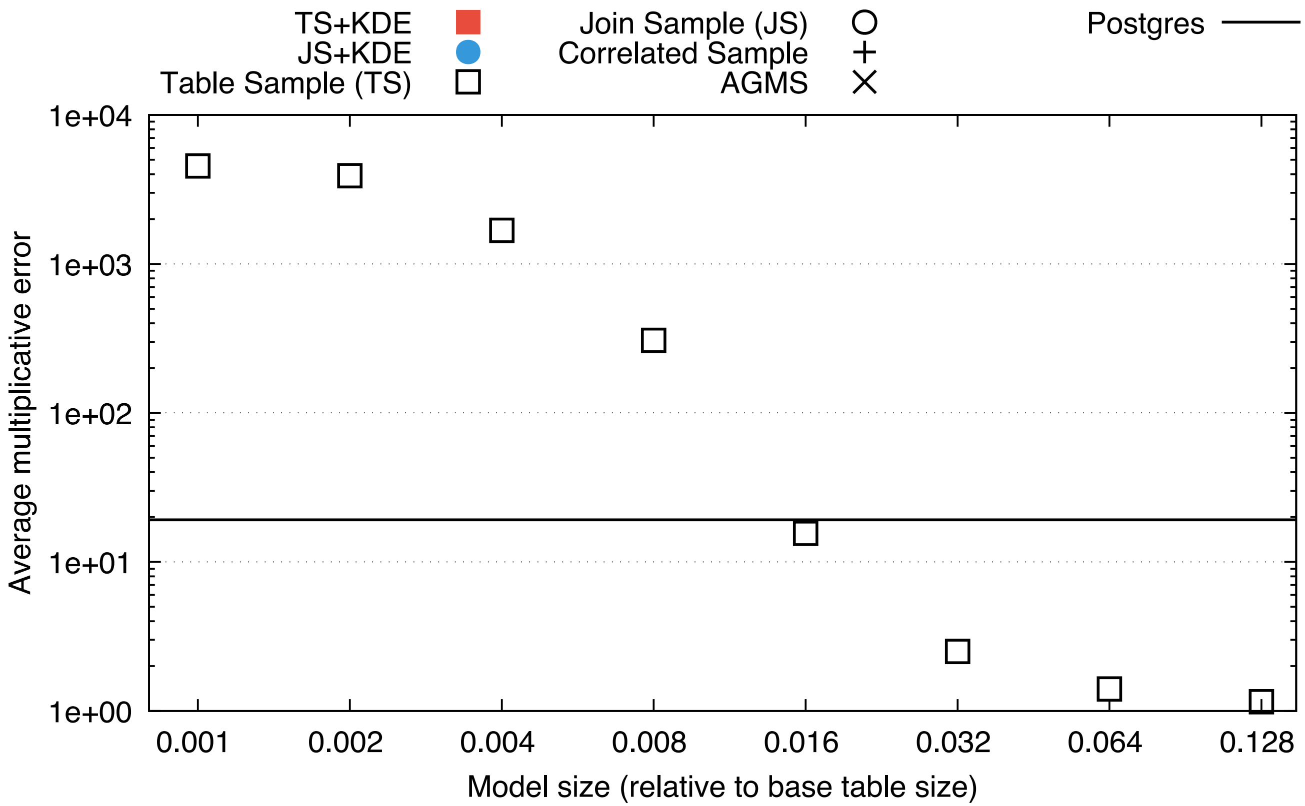
Evaluation: Scaling the Model Size (Postgres)

Dataset: DMV
Query: Q1U



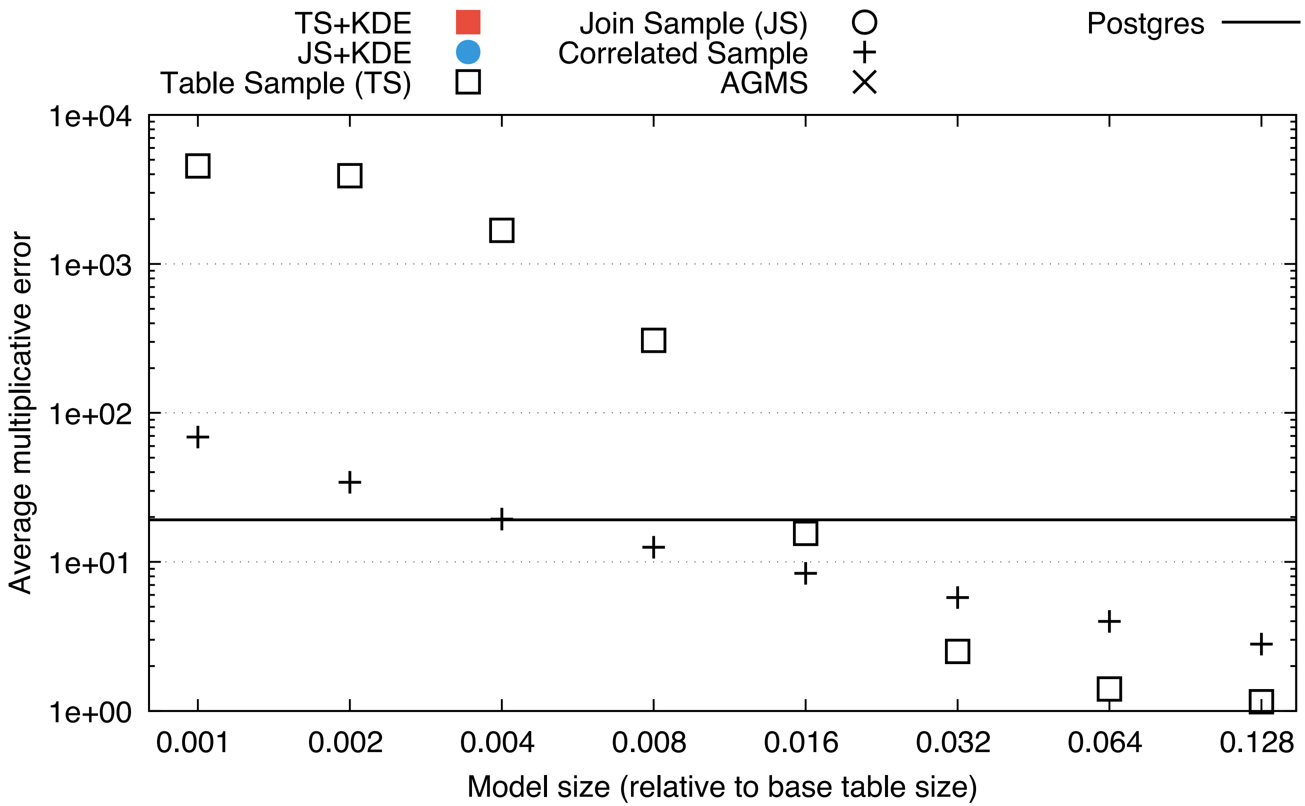
Evaluation: Scaling the Model Size (Table Sample)

Dataset: DMV
Query: Q1U



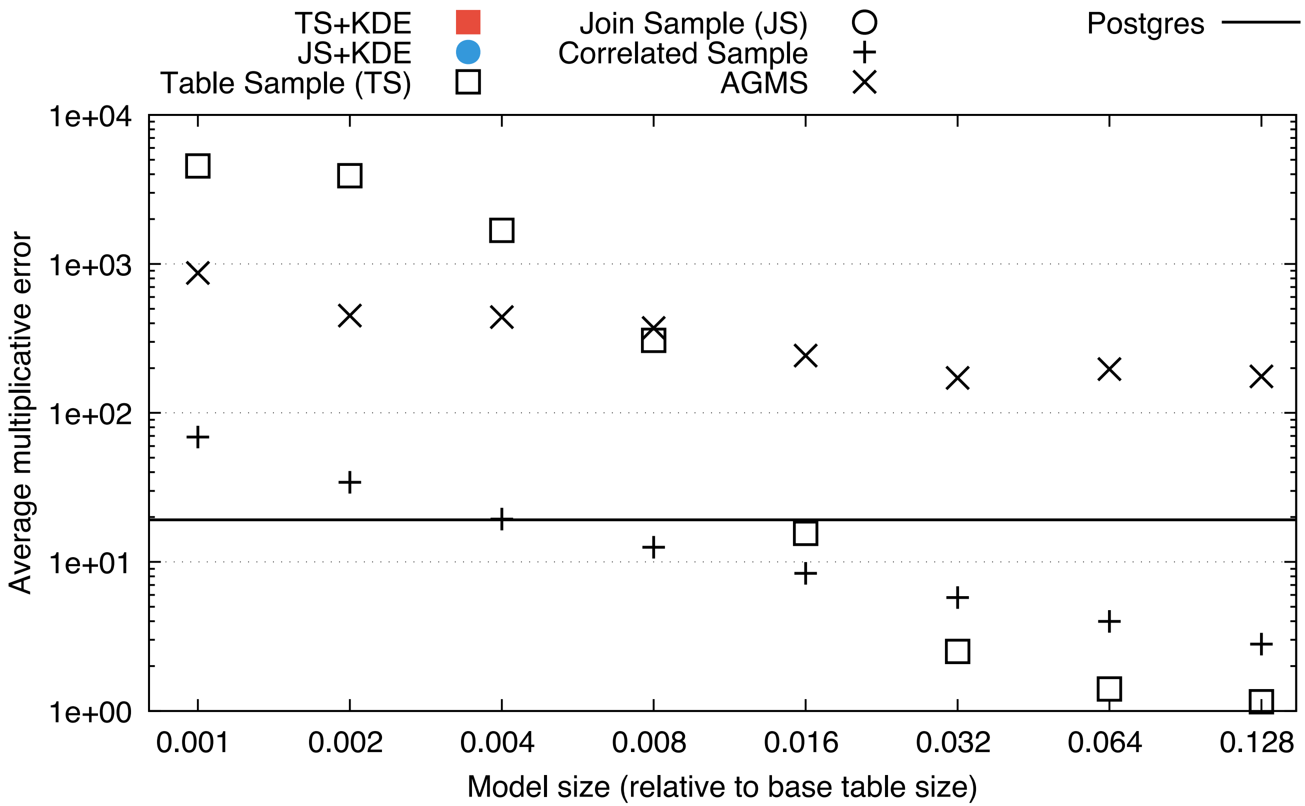
Evaluation: Scaling the Model Size (Correlated Sample)

Dataset: DMV
Query: Q1U



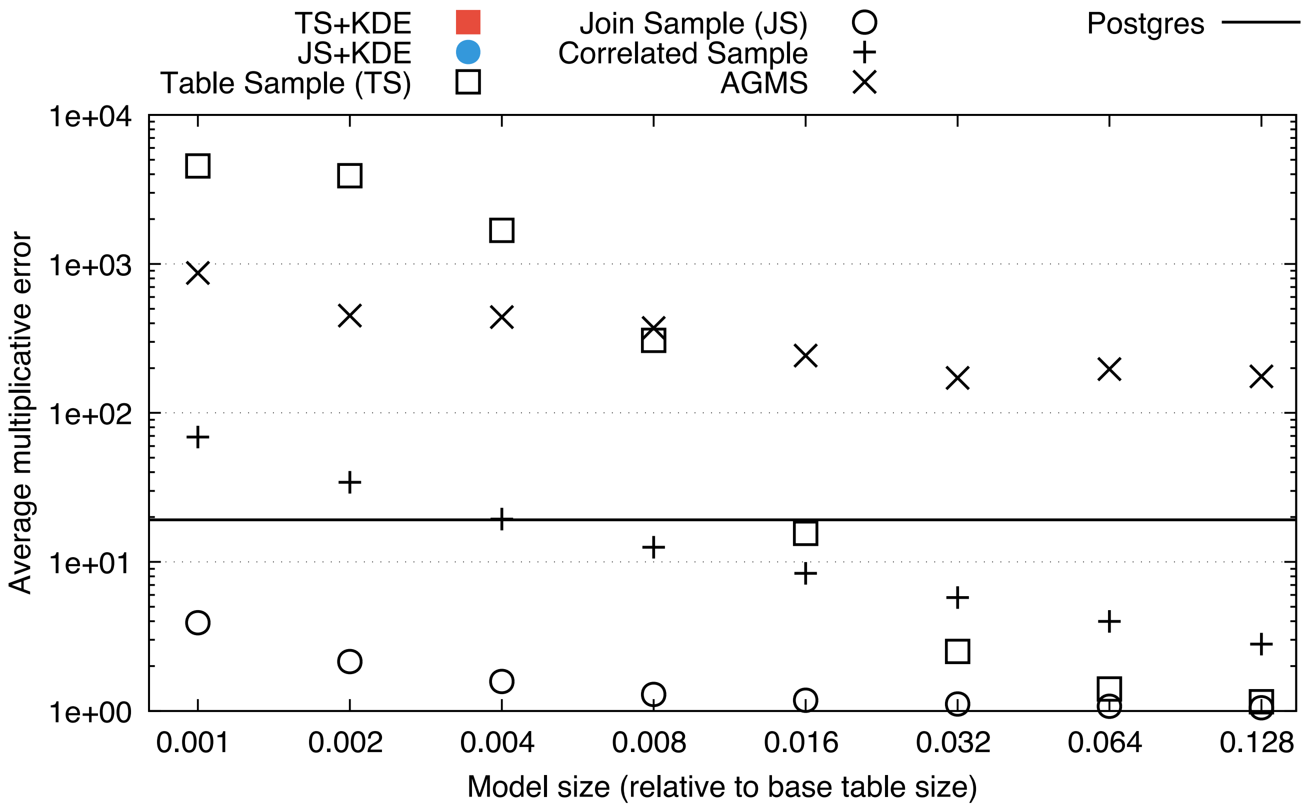
Evaluation: Scaling the Model Size (AGMS Sketch)

Dataset: DMV
Query: Q1U



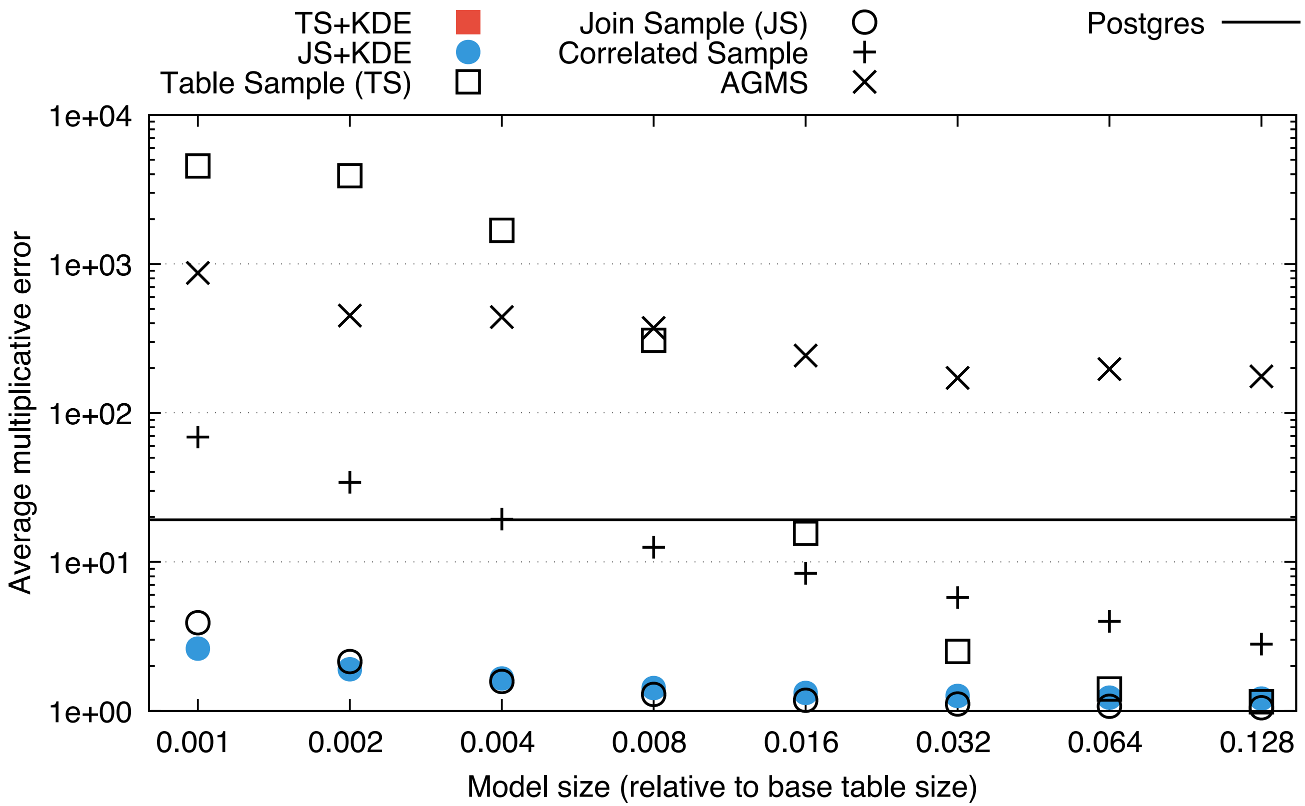
Evaluation: Scaling the Model Size (Join Sample)

Dataset: DMV
Query: Q1U



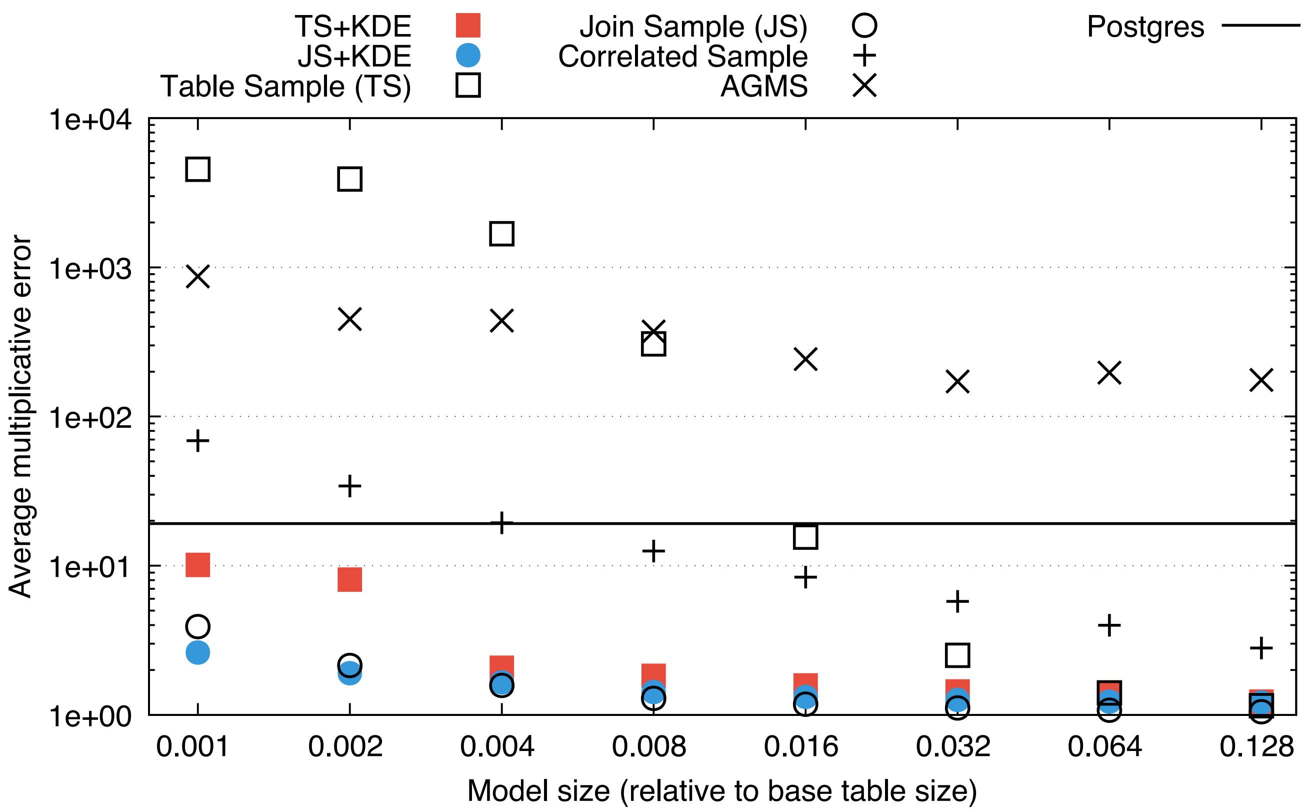
Evaluation: Scaling the Model Size (Join Sample + KDE)

Dataset: DMV
Query: Q1U

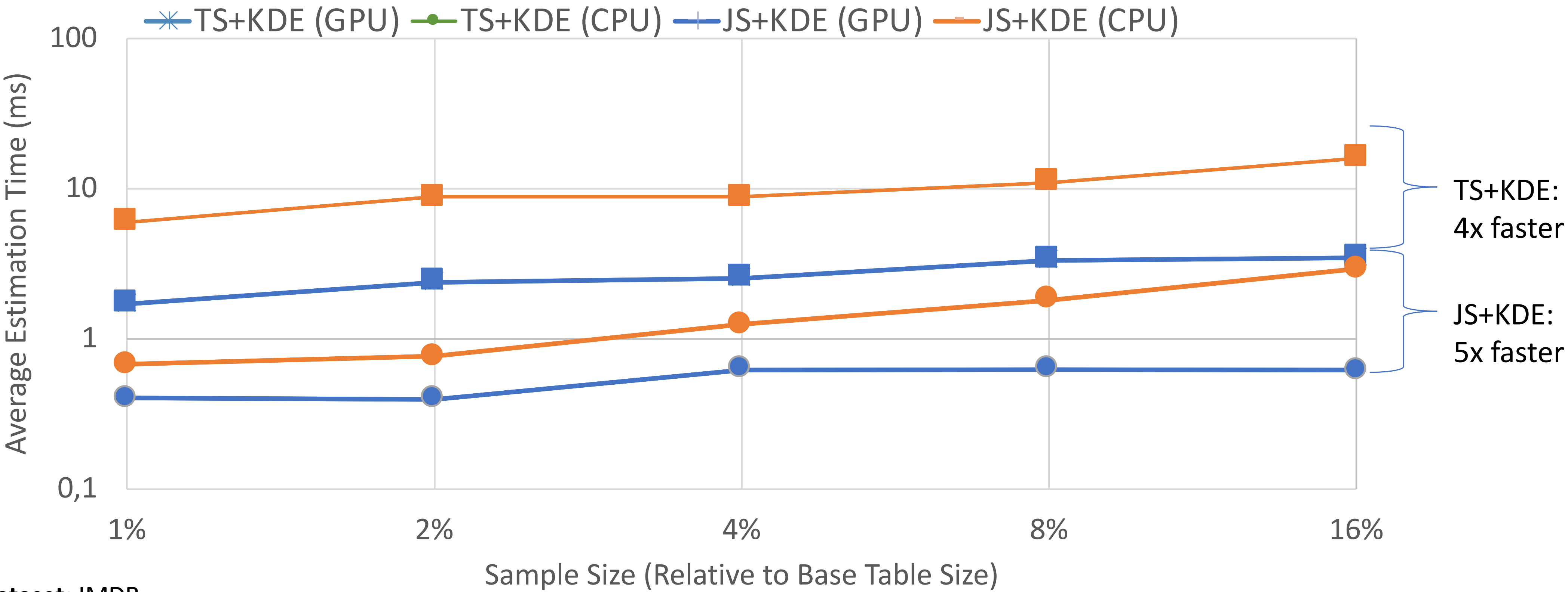


Evaluation: Scaling the Model Size (Table Sample + KDE)

Dataset: DMV
Query: Q1U



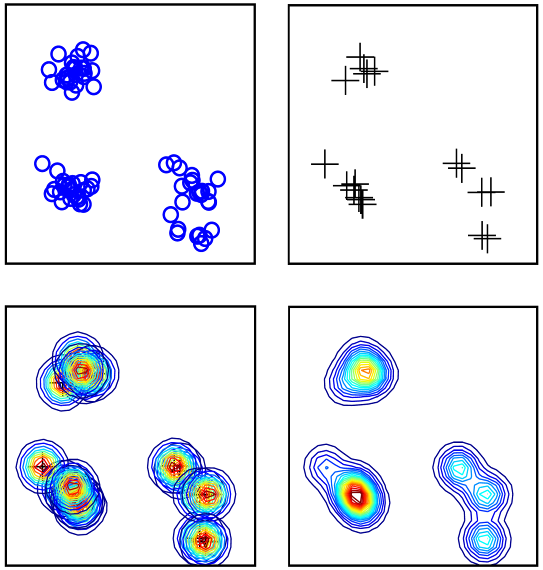
Runtime: CPU vs GPU



Dataset: IMDB
Workload: Q1U
GPU: Tesla V100
CPU: Intel Xeon Gold 5115

Conclusion

- KDE models for join selectivity estimation
- “Getting most out of your sample”
- Based on join or base table KDE models
- Learning hybrid between histograms and samples
- GPU-acceleration possible
- Experiments, data, and code online



github.com/martinkiefer/join-kde

“Estimating Join Selectivities using Bandwidth-Optimized Kernel Density Models”, PVLDB 17

Estimating Join Selectivities using Bandwidth-Optimized Kernel Density Models

Martin Kiefer, Max Heimel, Sebastian Breß, Volker Markl

Proceedings of the VLDB Endowment, 10(13), 2017

Further Publications on GPU-Accelerated Kernel Density Estimation:

Self-Tuning, GPU-Accelerated Kernel Density Models for Multidimensional Selectivity Estimation

SIGMOD 2015

Demonstrating Transfer-Efficient Sample Maintenance on Graphics Cards

EDBT 2015

Technische Universität Berlin

German Research Center for Artificial Intelligence