Workload-Driven Antijoin Cardinality Estimation

Florin Rusu[1], Zixuan Zhuang[1], Mingxi Wu[2], Christopher Jermaine[3]

frusu@ucmerced.edu, zzhuang@ucmerced.edu, mingxi.wu@turn.com, cmj4@cs.rice.edu

[1]UC Merced, [2]Turn Inc., [3]Rice University

Introduction

Motivating example.
Consider a standard TPC-H query "What is the total supply cost for parts less expensive than $700 supplied in 2007 by suppliers who did not supply any part in 2006?" having the corresponding SQL statement:

```
SELECT SUM(ps_supplycost*l1.l_quantity)
FROM partsupp ps, lineitem l1
WHERE ps_suppkey=l1.l_suppkey AND
 ps_partkey=l1.l_partkey AND
 year(l1.l_shipdate)=2007 AND
 ps_supplycost < 700 AND NOT EXISTS
  (SELECT * FROM lineitem l2
  WHERE l2.l_suppkey=l1.l_suppkey AND
   year(l2.l_shipdate)=2006)
```
(1)

Following figure sketches three alternative let-deep evaluation plans for this query.

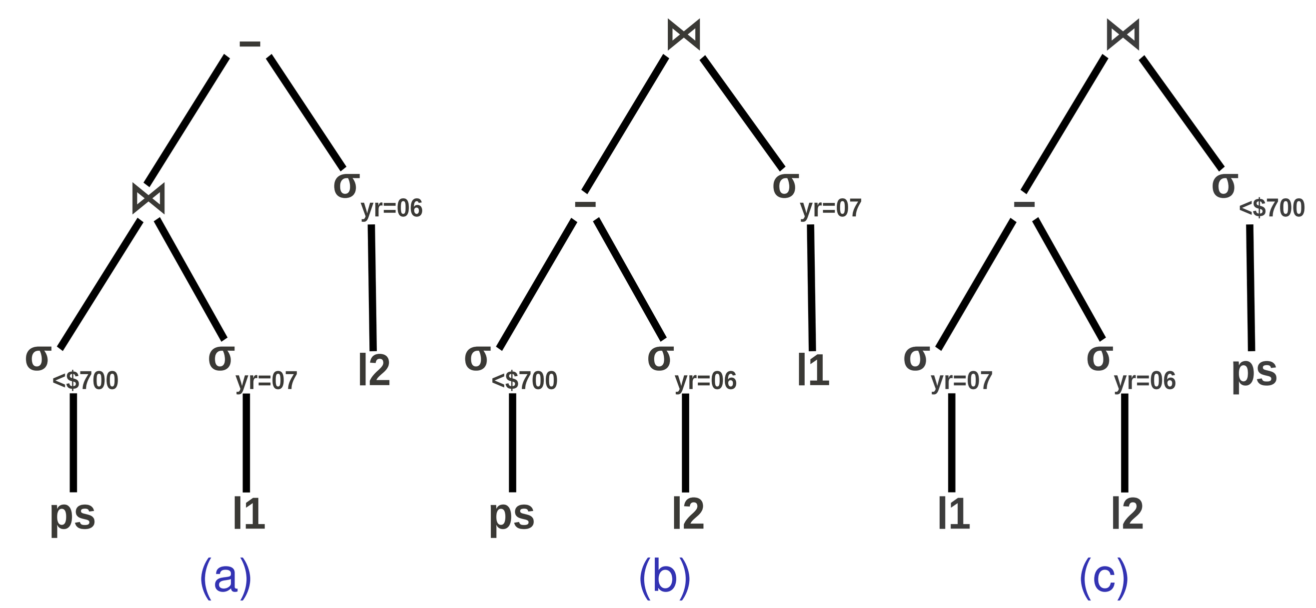
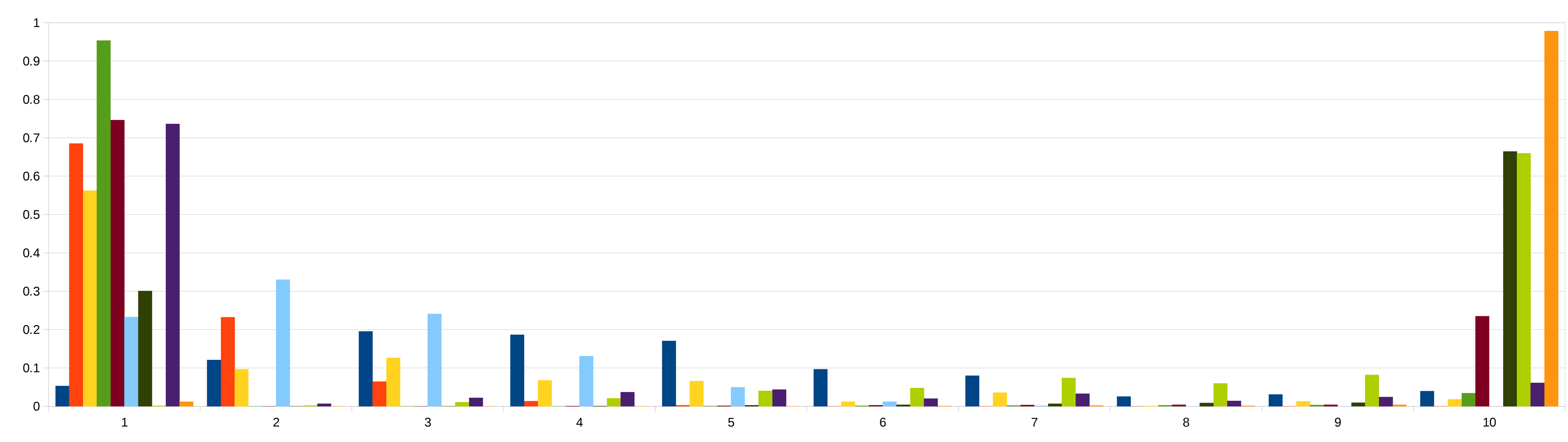


Figure : Alternative left-deep execution plans for query (**??**). "–" represents the antijoin operator.

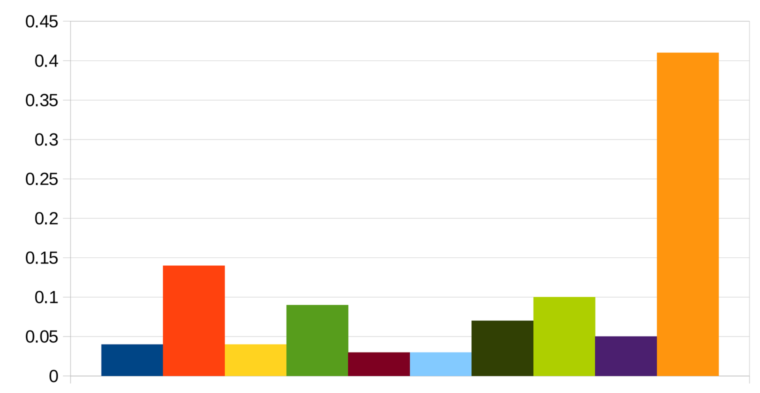
Problem statement & contributions.
The problem we consider in this paper is sampling-based antijoin cardinality estimation. We explore *how a set of prior (anti)join queries and their results can be used to devise better sampling-based antijoin cardinality estimators for query optimization.* To this end, we introduce a novel sampling-based estimator for antijoin cardinality that – unlike existent estimators – provides the required efficiency to be implemented in a query optimizer. The solution we propose is the first to integrate query workload information in the estimator definition using a Bayesian statistics framework.

Work-load Driven Estimation

we introduce the four steps of the proposed Bayesian inference framework. The ***learning phase*** uses statistical methods to build a prior histogram model composed of a number of candidate match frequency histogram patterns.



(a)



(b)

Figure : Prior histogram model, 10 components, 10 bucketss. (a) is the histograms, and (b) is the corresponding weights.

In the ***characterization phase***, a likelihood distribution of the current sample histogram h' corresponding to each learned histogram pattern is generated. The multinomial maximum likelihood estimator h' computed over a set $MC = \{h'_{(1)}, h'_{(2)}, \ldots, h'_{(MCc)}\}$ of MCc sample histograms is given by:

$$h'[j] = \frac{\sum_{i=1}^{MCc} h'_{(i)}[j]}{\sum_{j'=0}^{m}\sum_{i=1}^{MCc} h'_{(i)}[j']}, 0 \le j \le m \qquad (2)$$

The ***inference phase*** uses the likelihood distribution computed in the characterization phase and the observed sample histogram h' to infer the posterior histogram probabilities. The prior weights of the histogram model are updated based upon the evidence seen in the sample histogram h'.

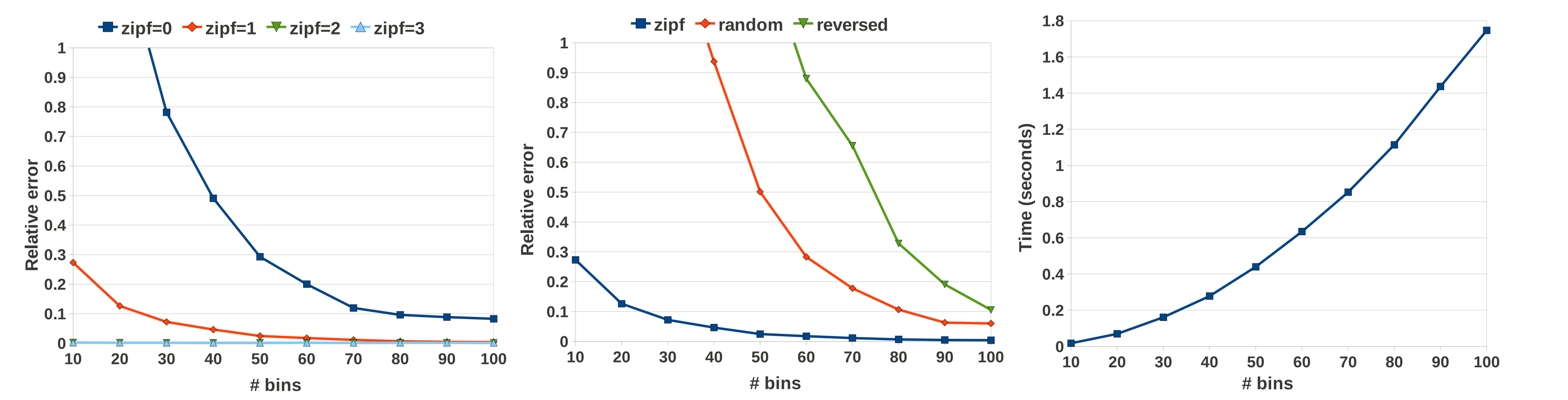
$$w'_l = \frac{w_l \cdot pdf_{mult}\left(h'_{query} \mid \vec{h}'_{(l)}\right)}{\sum_{l'=1}^{c} w_{l'} \cdot pdf_{mult}\left(h'_{query} \mid \vec{h}'_{(l')}\right)}, 0 \le l \le c \qquad (3)$$

The ***optimization phase*** generates a large number MCo of histograms from the superpopulation computed in the inference phase. The optimal estimator is computed such that it minimizes the sum-squared error (SSE):

$$SSE = \sum_{i=1}^{MCo}\left(C_{P_i}[0] - \sum_{j=0}^{m} b_j \cdot h'_i[j]\right)^2 \qquad (4)$$

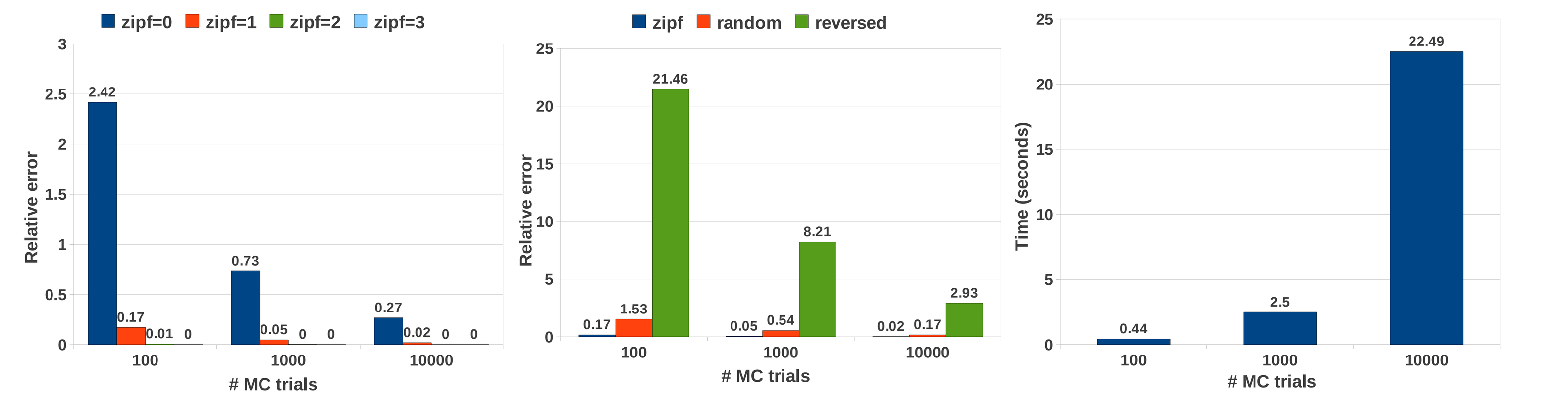
Experimental Evaluation

- Micro-Benchmarks.



(a) (b) (c)

Figure : Accuracy ((a) and (b)) and execution time (c) as a function of the number of bins in the sample match frequency histogram.



(a) (b) (c)

Figure : Accuracy ((a) and (b)) and execution time (c) as a function of the number of Monte Carlo trials in the optimization phase.

- TPC-H Data

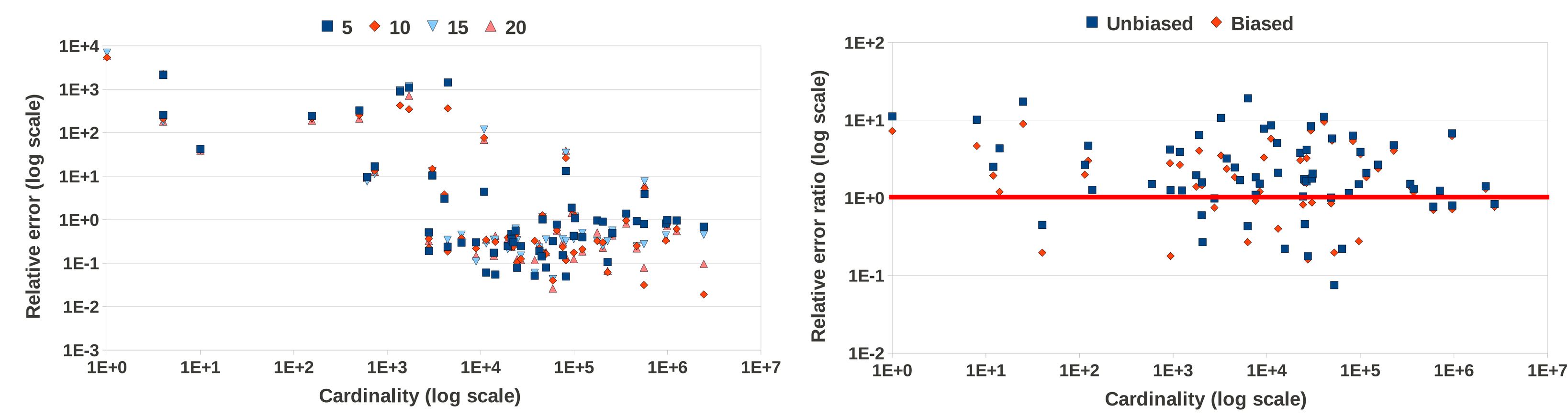
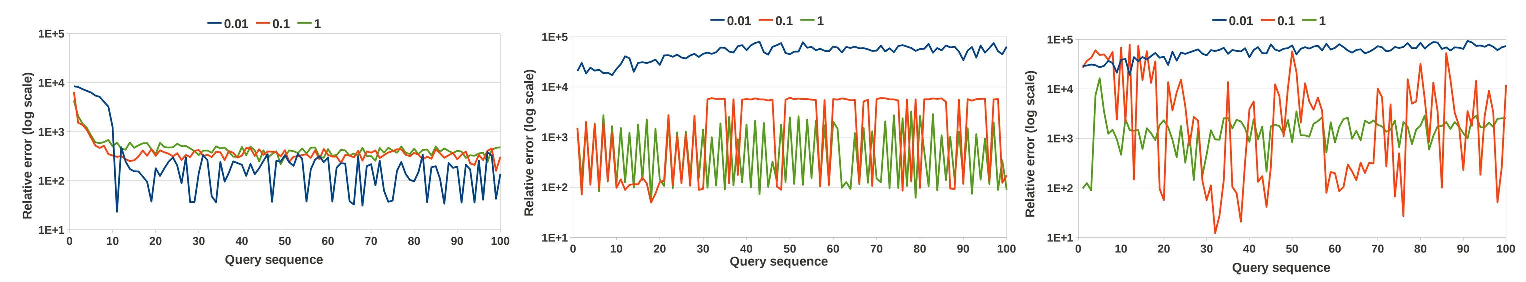


Figure : Accuracy as a function of the number of mixtures.

Figure : Comparison with previous estimators.



(a) (b) (c)

Figure : Accuracy for a query executed in sequences of 100(a) 10(b), and 1(c) in a workload of 1000 queries of 10 different classes.

Conclusion

In this paper, we present a novel sampling-based estimator for antijoin cardinality that provides the required efficiency to be implemented in a query optimizer. We plan to extend the proposed estimator to parallel settings where samples are distributed across computing nodes in future work.