Constructing Join Histograms from Histograms with q-error Guarantees

Kaleb Alway*
David R. Cheriton School of Computer Science
Faculty of Mathematics, University of Waterloo
kpalway@uwaterloo.ca

Anisoara Nica
SAP SE
Waterloo, Ontario, Canada
Anisoara.Nica@sap.com

ABSTRACT

Histograms are implemented and used in any database system, usually defined on a single-column of a database table. However, one of the most desired statistical data in such systems are statistics on the correlation among columns. In this paper we present a novel construction algorithm for building a *join histogram* that accepts two single-column histograms over different attributes, each with q-error guarantees, and produces a histogram over the result of the join operation on these attributes. The join histogram is built only from the input histograms without accessing the base data or computing the join relation. Under certain restrictions, a q-error guarantee can be placed on the produced join histogram.

It is possible to construct adversarial input histograms that produce arbitrarily large q-error in the resulting join histogram, but across several experiments, this type of input does not occur in either randomly generated data or real-world data. Our construction algorithm runs in linear time with respect to the size of the input histograms, and produces a join histogram that is at most as large as the sum of the sizes of the input histograms. These join histograms can be used to efficiently and accurately estimate the cardinality of join queries.

Keywords

Query Optimization; Histogram; Cardinality Estimation

1. INTRODUCTION

In this paper we use a general notion of q-error. As defined in Moerkotte et al. [1], an estimate $\hat{f}$ of a value f is said to be θ, q-acceptable if either $f \leq \theta \wedge \hat{f} \leq \theta$ or $||\hat{f}/f||_Q \leq q$, where $||x||_Q = max(x, 1/x)$. Note that when the parameter θ is zero, the notion of θ, q-acceptability is the same as a plain q-error bound. We consider the statistical question of cardinality estimation of a join query of the form:

$Join(J) = \Pi_{R.A}(\sigma_{(R.A=T.B)\wedge p_1(R.A)\wedge p_2(T.B)}(R \times T))$
where:

- The column $R.A$ has a θ_1, q_1-acceptable histogram H_1
- The column $T.B$ has a θ_2, q_2-acceptable histogram H_2
- $p_1(R.A)$ and $p_2(T.B)$ are *histogram-SARGable* predicates on columns $R.A$ and $T.B$, i.e., predicates for which the selectivity estimation can be computed using the histograms H_1 and H_2, respectively. For example, a *histogram-SARGable* predicate on the column $R.A$ is $(R.A\varphi c)$ with $\varphi \in \{<, \leq, =, \geq, >\}$ and c a constant.

We assume that the histograms H_1 and H_2 satisfy some general properties, namely, the histograms must be able to provide a θ, q-acceptable estimate $\hat{f}_{v_i}$ for each value v_i in the column domain, which approximates the frequency f_{v_i}, i.e. the number of times v_i occurs in the column.

Our objective is to construct a histogram H for the column J of the join relation $Join(J)$, using the input histograms H_1 and H_2, without computing the join query or accessing the base data of the relations R and T. We study under what conditions the join histogram is also θ, q-acceptable.

2. θ, q-ACCEPTABLE JOIN HISTOGRAM

The first step in our algorithm is to restrict both input histograms by applying the *histogram-SARGabale* predicates $p_1(R.A)$ and $p_2(T.B)$, as well as restricting their domains to the ranges present in the active domains of both columns.

Therefore we can theoretically assume a common domain $V = \{v_1, \cdots, v_k\}$, with $v_1 < \cdots < v_k$, such that each v_i is in the range of the active domain of both $R.A$ and $T.B$, and satisfies the filter predicates $p_1(R.A)$ and $p_2(T.B)$. The frequency of any individual element v_i in $R.A$, $\hat{f}_1(v_i)$, can be estimated using H_1. Similarly, we can attain a frequency estimate for v_i in $T.B$, $\hat{f}_2(v_i)$, using H_2. Let $f_1(v_i)$ and $f_2(v_i)$ denote the corresponding precise frequencies.

The theoretical estimate for the frequency of v_i in the join result is $\hat{f}(v_i) = \hat{f}_1(v_i)\hat{f}_2(v_i)$. The following result allows us to place a loose θ, q-acceptability bound on this estimate:

Lemma 2.1. *If $\hat{f}_1, \hat{f}_2$ are estimates for f_1, f_2, θ_1, q_1-acceptable and θ_2, q_2-acceptable, respectively, then $\hat{f}_1\hat{f}_2$ is a $\theta_1\theta_2, q'$-acceptable estimate for $f_1 f_2$, $q' = \max(\theta_1 q_2, \theta_2 q_1, q_1 q_2)$.*

We define $f^i \stackrel{def}{=} f(v_i)$, $f^+(v_m, v_n) \stackrel{def}{=} f^+(m, n) \stackrel{def}{=} \sum_{m \leq i \leq n} f^i$,

and similarly, $\hat{f}^+(v_m, v_n) \stackrel{def}{=} \hat{f}^+(m, n) \stackrel{def}{=} \sum_{m \leq i \leq n} \hat{f}^i$.

Our join histogram algorithm is based on the construction algorithm described in [1], with the following important adjustment which makes possible the building of the join his-

*The student author and the adviser Dr. Anisoara Nica worked on this project during Fall 2015 co-op term at SAP Waterloo, Canada.

SIGMOD/PODS'16 June 26 - July 01, 2016, San Francisco, CA, USA

ACM ISBN 978-1-4503-3531-7/16/06.
DOI: http://dx.doi.org/10.1145/2882903.2914828

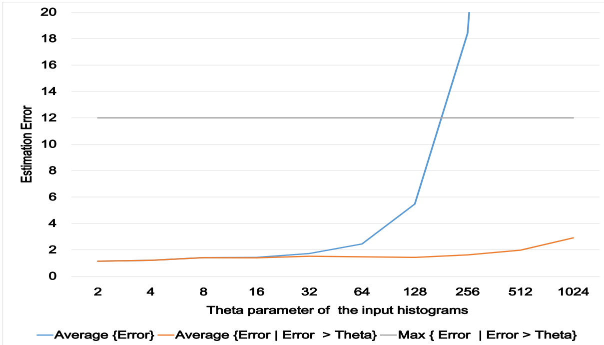

Figure 1: Estimation error using join histograms built from $\theta, 2$-acceptable histograms ($\theta \in \{2, 4, \ldots, 1024\}$); Query: $\sigma_{c_1 \le J \le c_2} Join(J)$; $e = \dfrac{\text{real number of rows}}{\text{estimated number of rows}}$: Estimation Error = $\max(\ e,\ \frac{1}{e}\ )$

togram without accessing the base data: instead of using the true frequency value f^+ when checking our error bounds, we treat $\hat{f}^+$ as the true frequency as these can be obtained only from the input histograms. This algorithm yields a sequence of θ, q-acceptable buckets, each with a different θ, q, having an "estimate of an estimate", as the frequencies used to construct these buckets are the estimated frequencies computed from the input histograms.

The following lemma allows us to place a bound on the resulting join histogram's error.

LEMMA 2.2. *If $\hat{f}$ is a θ, q-acceptable estimate for f and $\bar{f}$ is a θ', q'-acceptable estimate for $\hat{f}$, then $\bar{f}$ is a $\theta\theta', q''$-acceptable estimate for f, with $q'' = \max(qq', q'\theta, q\theta')$*

Lemma 2.2 shows that it is possible to construct a join histogram that is θ, q-acceptable for finite values of θ and q.

We now give some interesting details of our construction algorithm when join columns are dictionary-encoded[1]. Consider the scenario where the columns $R.A$ and $T.B$ are dictionary-encoded by ordered dictionaries and the histograms H_1 and H_2 are built on the dictionary values as described in [1]. For any range of actual domain values $[l, u]$, there are corresponding ranges $[i_1, j_1]$ for $R.A$ and $[i_2, j_2]$ for $T.B$ which are the minimum ranges containing dictionary entries mapping to an element in the range $[l, u]$. Let's assume that $j_1 - i_1 > j_2 - i_2$. In this discussion, we make the following two theoretical assumptions:

1 For every $n \in [i_2, j_2]$, $\exists m \in [i_1, j_1]$ such that $D_2(n) = D_1(m)$, where D_1 and D_2 are the dictionaries for $R.A$ and $T.B$ respectively. In other words, every distinct value of $T.B$ in $[l, u]$ also occurs in $R.A$[2].

2 If $[i, j]$ is covered by a single bucket in a histogram H, then for any $m \in [i, j]$, $\hat{f}(m) = \dfrac{\hat{f}^+(i, j)}{j - i}$.

By the Lemmas 2.1 and 2.2, we can compute an error-bounded frequency for the join result on the range $[l, u]$ with the sum $\displaystyle\sum_{n=i_2; D_2(n)=D_1(m)}^{j_2} \hat{f}_2(n)\hat{f}_1(m)$. Using the second assumption, when $[l, u]$ is contained in a single bucket, this simplifies to:

$$\hat{f}(l, u) = \sum_{n=i_2}^{j_2} \frac{\hat{f}_1^+(i_1, j_1)}{j_1 - i_1} \frac{\hat{f}_2^+(i_2, j_2)}{j_2 - i_2} = \frac{\hat{f}_1^+(i_1, j_1)\hat{f}_2^+(i_2, j_2)}{j_1 - i_1}$$

This tight bound motivates our construction algorithm **buildJoinHistogram** to generate buckets in H_J corresponding to maximum intervals $[l, u]$ contained in exactly one bucket in both histograms H_1 and H_2.

buildJoinHistogram(H_1, H_2, D_1, D_2)
Output: Join histogram H_J for the column J in $Join(J)$
index i_1, i_2, j_1, j_2, value l, u, vector *buckets*, bucket b
$initJoinIteration(i_1, j_1, i_2, j_2)$
while $i_1 < H_1.maxValue() \wedge i_2 < H_2.maxValue()$ **do**
  $findNextBoundaries(i_1, j_1, i_2, j_2, l, u)$ // find next $[l, u]$
  $b.begin = l$, $b.end = u$
  $b.count = H_1.getCardinality([i_1, j_1]) \cdot \frac{H_2.getCardinality([i_2, j_2])}{\max(j_1 - i_1, j_2 - i_2)}$
  $b.distinct = \min(j_1 - i_1, j_2 - i_2)$
  $buckets.push(b)$
end while
$H_J.initBuckets(buckets)$
return H_J

The main idea of the algorithm is to iterate over the buckets of H_1 and H_2 by looking up the dictionary values stored at the next bucket boundary and comparing them to find the largest valid range for the next bucket to add to H_J.

Our final algorithm prototypes a set of improvements such as dropping the assumption 1., and applying a post-construction optimization where adjacent buckets in H_J are merged as long as the q-error is maintained.

Figure 1 depicts the experimental results for estimation errors using join histograms built by our algorithm. We use a combination of real-world data and randomly generated data. The input histograms were constructed with $q = 2$, and θ varied in the range 2 to 1024. A join histogram was built for the join relation $Join(J) = \sigma_{R.A=T.B}(R \times B)$ (i.e., without filter predicates). This means that each bucket has $\theta^2, 2\theta$-acceptable upper bounds (by Lemma 2.1). We run a set of range queries of the form $Q(c_1, c_2) = \sigma_{c_1 \le J \le c_2} Join(J)$ and record the actual number of rows, and the estimated number of rows computed using the join histogram. The estimation error is computed as $\max(\ e,\ \frac{1}{e}\ )$ where $e = \dfrac{\text{real number of rows}}{\text{estimated number of rows}}$.

For large θ, when the actual number of rows in $Q(c_1, c_2)$ is less than θ^2, the average q-error is large as well, as the estimation from the join histogram is bound by θ^2. As θ grows larger, most range queries contain fewer than θ^2 elements, so the average q-error of all queries rises rapidly. However, as shown by these experiments, the real q-error observed using the join histograms is often much lower than the theoretical q-error bounds established on the join histogram's estimates.

3. REFERENCES

[1] G. Moerkotte, D. DeHaan, N. May, A. Nica, and A. Boehm. Exploiting ordered dictionaries to efficiently construct histograms with q-error guarantees. In *ACM SIGMOD*, pages 361–372, 2014.

[1]We have also extended this algorithm to work on histograms built directly on the domain values, but leave its details out of this discussion.

[2]This assumption was relaxed in our implementation.