Improved Selectivity Estimation by Combining Knowledge from Sampling and Synopses

Magnus Müller
Database Research Group

Guido Moerkotte
Database Research Group

Oliver Kolb
Scientific Computing Group

University of Mannheim, Germany
{magnus, moerkotte, kolb}@uni-mannheim.de

ABSTRACT

Estimating selectivities remains a critical task in query processing. Optimizers rely on the accuracy of selectivities when generating execution plans and, in approximate query answering, estimated selectivities affect the quality of the result. Many systems maintain synopses, e.g., histograms, and, in addition, provide sampling facilities. In this paper, we present a novel approach to combine knowledge from synopses and sampling for the purpose of selectivity estimation for conjunctive queries. We first show how to extract information from synopses and sampling such that they are mutually consistent. In a second step, we show how to combine them and decide on an admissible selectivity estimate. We compare our approach to state-of-the-art methods and evaluate the strengths and limitations of each approach.

PVLDB Reference Format:
Magnus Müller, Guido Moerkotte, Oliver Kolb. Improved Selectivity Estimation by Combining Knowledge from Sampling and Synopses. *PVLDB*, 11(9): 1016-1028, 2018.
DOI: https://doi.org/10.14778/3213880.3213882

1. INTRODUCTION

The problem of estimating the selectivities of predicates is of interest in various fields of data processing. In approximate query processing, selectivity estimation techniques are employed in `count` queries where accuracy is traded for response time [6, 8]. In query optimization, selectivity estimates are crucial parameters to cost functions which determine the decision-making in query plan selection. Recent research indicates that query optimizers benefit greatly from improved selectivity estimates [16, 25].

Selectivities are estimated by available information. Many systems maintain synopses, e.g., histograms, and, in addition, provide sampling facilities. The crux of the matter is how to utilize this information: Consider a query with predicates $p_1 \equiv$ `A > 5`, $p_2 \equiv$ `B between 2.7 and 3.5`, and $p_3 \equiv$ `C = 'green'` over a relation R with attributes A, B and C. Suppose the system provides histograms over the single attributes that give approximate selectivities for p_1, p_2, and p_3 as well as a multi-dimensional histogram that approximates the joint distribution of B and C and, thus, gives an approximate selectivity for $p_2 \wedge p_3$. Furthermore, assume that the system maintains a sample of R. One possibility of finding estimates for the unknown selectivities of $p_1 \wedge p_2$ and $p_1 \wedge p_2 \wedge p_3$ is to simply compute the ratio of qualifying entries in the sample. However, if R is large and the selectivities of the predicates are low, the quality of this estimate is often insufficient. Another approach is to derive estimates solely based on the known synopses. Elaborate methods based on the principle of maximum entropy have been developed by Markl et al. to consistently process multi-dimensional synopses [20]. The question remains how to utilize both synopses and sampling to estimate selectivities. To the best of our knowledge, Yu, Koudas and Zuzarte are the only ones who have attempted to answer this question [34]. As the main problem, they consider selectivities obtained from sampling that are inconsistent with selectivities obtained from synopses, e.g., the selectivity of $p_1 \wedge p_2$ derived from a sample may be larger than the selectivity of p_1 as provided by a histogram, regardless of the fact that both cannot hold true simultaneously. Hence, their idea is to refine estimates derived from sampling until they comply with the selectivities known from synopses. However, as we will see later, their approach has limited capabilities.

In this paper, we introduce CSE, a novel approach to selectivity estimation for conjunctive queries for single relations, that consistently combines sampling and synopses. To this end, for each selectivity derived from a sample and each selectivity obtained from a synopses structure, we construct intervals that contain the true selectivity either guaranteed or with high probability. We then go on to produce selectivity estimates by solving an optimization problem that is constrained to these intervals. A key property of our approach is that it can incorporate multi-dimensional synopses. Moreover, our approach proves robust in our evaluation in all scenarios. In particular, our approach not only outperforms existing state-of-the-art methods in a scenario where the selectivities of some of the predicates found in a query are known precisely, but also performs best in real-world scenarios where synopses structures provide selectivities with approximation errors. This is due to our method that extracts intervals, instead of point-estimates, from sampling and synopses to overcome the issue of estimating selectivities based on inaccurate approximations.

The remainder of the paper is structured as follows: In the next section, we discuss some preliminaries and intro-

Proceedings of the VLDB Endowment, Vol. 11, No. 9

DOI: https://doi.org/10.14778/3213880.3213882

duce our notation. We then present related work in Section 3. In Section 4, we introduce our CSE approach. Section 5 contains an extensive evaluation of our approach in terms of prediction accuracy and run time in comparison to other state-of-the-art approaches under scenarios with varying parameters. To the best of our knowledge, we are the first who consider synopses with approximation errors in our evaluation, as it is the case in real-world scenarios. Finally, we draw a conclusion and discuss future work.

2. PRELIMINARIES

In this section, we introduce our notation and discuss preliminaries. A conjunctive query P, defined as a conjunction of n *simple predicates* or *boolean factors*, over a relation R represents the starting point of our discussion.

$$P := p_1 \wedge p_2 \wedge \cdots \wedge p_n$$

A predicate is simple if it compares an attribute value to a literal. We denote by $N = \{1, \ldots, n\}$ the index set of P.

2.1 Predicates and Selectivities

A selectivity is a value in the interval $[0, 1]$ and is defined as the fraction of entries in a data set or relation that satisfies some specified predicate. We distinguish selectivities induced by predicates that are defined by two formulae. For both, the argument X is a subset of the index set N of a given conjunctive query P, i.e., $X \subseteq N$. The first formula is defined as

$$F_\beta(X) := \bigwedge_{i \in X} p_i,$$

i.e., $F_\beta(X)$ is a conjunction of those predicates whose index is contained in X. In case $X = \emptyset$, we define $F_\beta(X) \equiv true$.

The second formula is defined as

$$F_\gamma(X) := \bigwedge_{i \in X} p_i \wedge \bigwedge_{i \in N \setminus X} \neg p_i,$$

i.e., $F_\gamma(X)$ defines the *minterms* of the conjunctive query P. For a boolean function of n variables, a minterm is defined as a conjunction in which each of the n variables appears exactly once, possibly in its complement form.

To illustrate the difference in the formulae and their consequence on the selectivity, consider the conjunctive query $p_1 \wedge p_2 \wedge p_3$ with index set $N = \{1, 2, 3\}$. Let $X = \{1, 3\} \subseteq N$. Then $\beta(X)$ is the selectivity of $F_\beta(X) = p_1 \wedge p_3$, which we call the β-selectivity of X. Similarly, $\gamma(X)$ is the selectivity of $F_\gamma(X) = p_1 \wedge p_3 \wedge \neg p_2$, which we call the γ-selectivity of X.

Note that for all X, the β-selectivity $\beta(X)$ is greater than or equal to the γ-selectivity $\gamma(X)$. This is because $F_\gamma(X)$ contains at least the predicates in $F_\beta(X)$, and additional predicates imply a lower or at most unchanged selectivity. For the same reason we have that $\beta(X') \geq \beta(X)$ for all $X, X' \subseteq N$ with $X \supset X'$. Furthermore note that from our definition above $F_\beta(\emptyset) \equiv true$ it follows that $\beta(\emptyset) = 1$ since every entry in a data set or relation satisfies this condition.

Observe that both F_β and F_γ depend only on $X \subseteq N$. Since all $X \subseteq N$ form the power set of N, which is known to contain 2^n elements, the number of β- and γ-selectivities is 2^n each.

Finally, observe that all $X \subseteq N$ can be numbered by bitvectors $bv(X) := (d_n, \ldots, d_1)$, where $d_i = 1$ if $i \in X$, and $d_i = 0$ otherwise, for $1 \leq i \leq n$. Therefore, without introducing ambiguity, we refer to a formula or selectivity likewise by its characteristic bitvector $bv(X)$ for some set of indices X.

2.2 Relation between β- and γ-selectivities

Every conjunctive query $F_\beta(X)$ can be expressed as the disjunction of those minterms $F_\gamma(Y)$ that positively contain at least the literals in $F_\beta(X)$. For instance, for conjunctive query $p_1 \wedge p_2$ and $F_\beta(\{1\}) = p_1$ the minterms that positively contain at least the literals in $F_\beta(\{1\})$ are $F_\gamma(\{1\}) = p_1 \wedge \neg p_2$ and $F_\gamma(\{1, 2\}) = p_1 \wedge p_2$ and, thus, we have that $p_1 \equiv (p_1 \wedge \neg p_2) \vee (p_1 \wedge p_2)$.

It follows that every β-selectivity $\beta(X), X \subseteq N$ can be computed from γ-selectivities as

$$\beta(X) = \sum_{X \subseteq Y \subseteq N} \gamma(Y), \tag{1}$$

or in words: $\beta(X)$ is composed of those $\gamma(Y)$ where at least the predicates contained in $F_\beta(X)$ occur positively in $F_\gamma(Y)$. Figure 1 illustrates the relationship between β-selectivities and γ-selectivities according to Equation 1 for the previous example $p_1 \wedge p_2$.

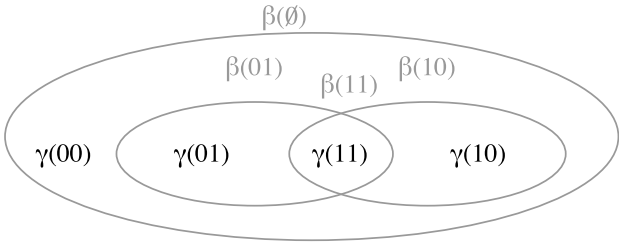


Figure 1: Grey ellipses mark the γ-selectivities that contribute to each β-selectivity for the conjunctive query $p_1 \wedge p_2$.

Note that we can compute all $X \subseteq Y \subseteq N$ efficiently by considering bitvector $bv(X)$ and enumerating all bitvectors $bv(Y)$ that contain a 1 at least at those positions where $bv(X)$ contains a 1 since

$$\{Y | N \supseteq Y \supseteq X\} \iff \{Y | bv(X) = bv(X) \& bv(Y)\},$$

where & denotes *bitwise AND*. Observe that $\beta(\emptyset)$ is the sum of all complete conjuncts, since all $Y \supseteq \emptyset$.

As an example, consider the conjunctive query $p_1 \wedge p_2$ and let $X = \{1\} \widehat{=} 01$. Then the set of all $Y \supseteq X$ is $\{\{1\}, \{1, 2\}\} \widehat{=} \{01, 11\}$ and $\beta(\{1\}) = \gamma(01) + \gamma(11)$.

2.3 Matrix Representation

Since for all $X \subseteq N$ Equation 1 gives one linear equation, together these equations form a system of linear equations $b = Cx$ with vectors $b = (\beta(\emptyset), \ldots, \beta(N))^T$ and $x = (\gamma(\emptyset), \ldots, \gamma(N))^T$, where T as superscript denotes *transposed*, and design matrix C, to which we refer as the *complete design matrix*. For zero-based indexing, the definition of the $2^n \times 2^n$ matrix C follows directly from the enumeration of summands in Equation 1:

$$C_{i,j} = \begin{cases} 1 & \text{if } bv(i) \subseteq bv(j) \\ 0 & \text{else} \end{cases} \tag{2}$$

Note that C is Boolean and each row indicates which γ-selectivities contribute to a β-selectivity. Furthermore note that we assume the $\beta(X)$ in b and the $\gamma(X)$ in x to be sorted in ascending order of their bitvector-value $bv(X)$.

Consider the example system $Cx = b$ for $p_1 \wedge p_2$:

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \gamma(00) \\ \gamma(01) \\ \gamma(10) \\ \gamma(11) \end{pmatrix} = \begin{pmatrix} \beta(00) \\ \beta(01) \\ \beta(10) \\ \beta(11) \end{pmatrix}.$$

Given all γ-selectivities, compute all β-selectivities as Cx. In case we are given all β-selectivities, we (conceptually) compute all γ-selectivities by inverting C and computing $C^{-1}b$.

Note that it is easy to prove that the matrix C is indeed invertible by showing that it is upper-triangular with no zeros on the main diagonal.

2.4 A Linear System Induced by Synopses

Synopses structures, like histograms, sketches or wavelets, approximate the distribution of single attributes or attribute groups. In terms of our notation, synopses structures provide β-selectivities. Ideally, a database system would maintain synopses over all possible combinations of attributes in the database. Selectivity estimation would be easy then since for every conjunctive query P with index set N, we could simply obtain the selectivity of P, i.e., $\beta(N)$, from the synopses structures. Additionally, we could obtain the selectivity $\beta(X)$ for all $X \subseteq N$ and formulate the linear system $b = Cx$ that we saw in the previous section. Unfortunately though, since the number of attribute combinations grows exponentially in the number of attributes, it is infeasible to maintain synopses for all attribute groups. In reality, synopses are only available for low-dimensional attribute groups. For instance, a database system may maintain single attribute statistics, referred to as 1D synopses, and statistics for attribute combinations of two attributes, referred to as 2D synopses. In this section, we derive a system of equations similar to the one in the previous section, but this time induced by the synopses maintained in a database system.

As before, let N be the index set of a given conjunctive query P. Suppose we know, due to synopses, the β-selectivities $\beta(X)$ for some but not all $X \subseteq N$. We collect these X in a set G, to which we refer as the *knowledge set*, since it specifies the β-selectivities that are known from synopses. Denote by b the corresponding $|G|$-dimensional vector of β-selectivities $\beta(X), X \in G$ to which we refer as the known selectivities. Each β-selectivity in b induces a linear equation defined by Equation 1. Together these equations form the linear system $b = Ax$ where $x = (\gamma(\emptyset), \ldots, \gamma(N))^T$ holds all γ-selectivities and A is a $|G| \times 2^n$ design matrix defined as follows

$$A_{i,j} = \begin{cases} 1 & \text{if } bv(G_i) \subseteq bv(j), \\ 0 & \text{else.} \end{cases}$$

We refer to A as the *partial design matrix*. Note that every partial design matrix A is simply the selection of those rows in the complete design matrix C that correspond to equations for which the $\beta(X)$ is known. Furthermore note that for the linear system $b = Ax$ we assume the $\beta(X)$ in b, the $\gamma(X)$ in x, and the sets of indices X in the knowledge set G to be sorted in ascending order by their bitvector-value $bv(X)$. Finally note that, unless all $X \subseteq N$ are part of the knowledge set G, the linear system $Ax = b$ is underdetermined. Thus, assuming $Ax = b$ is solvable, there exist infinitely many solutions for x.

As an example, consider the conjunctive query $p_1 \wedge p_2 \wedge p_3$ with index set $N = \{1, 2, 3\}$ over the attributes A, B, C of some relation R. Let $p_1 \equiv$ `A between 1 and 10` with a selectivity of 0.1, $p_2 \equiv$ `B` $\leq$ `100` with a selectivity of 0.2 and $p_3 \equiv$ `C = 5` with a selectivity of 0.01. Assume the database system maintains statistics for each individual attribute and multivariate statistics for the attribute group A, B. Then, the knowledge set is $G = \{\{\emptyset\}, \{1\}, \{2\}, \{1, 2\}, \{3\}\} \widehat{=} \{000, 001, 010, 011, 100\}$. The β-selectivity of each simple predicate is already given above. For $p_1 \wedge p_2$ suppose a selectivity of 0.05; note that it would be 0.02 if p_1 and p_2 were independent. Then, the system of linear equations $Ax = b$ is

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} \gamma(000) \\ \gamma(001) \\ \gamma(010) \\ \gamma(011) \\ \gamma(100) \\ \gamma(101) \\ \gamma(110) \\ \gamma(111) \end{pmatrix} = \begin{pmatrix} 1 \\ 0.1 \\ 0.2 \\ 0.05 \\ 0.01 \end{pmatrix},$$

where solutions for vector x, i.e., $(\gamma(000), \gamma(001), \ldots, \gamma(111))^T$ are admissible if all γ-selectivities lay in $[0, 1]$.

In general, the linear system $b = Ax$ captures all knowledge that is available due to synopses in a database system. In [20] the linear system $b = Ax$ is given in implicit form, where Markl et al. substitute the design matrix A by so-called *components* induced by the elements of the knowledge set.

Note that by now we assumed that synopses structures allow us to *know* some β-selectivity. This is not correct. Synopses structures *approximate* β-selectivities, and we will see the impact of this distinction in later sections.

3. RELATED WORK

Many research areas relate to selectivity estimation. Literally all estimation techniques rely on some background information in the form of a sample or synopses. For an extensive survey on sampling and synopses, see the book by Cormode, Garofalakis, Haas and Jermaine [8]. Recently Shekelyan et al. presented a novel approach to construct histograms for multi-dimensional data that give error bounds for estimates. Here we review selected selectivity estimation techniques. Some of them will serve as competitors in our evaluation. We first present techniques that rely on synopses only. Then, we mention techniques that exploit sampling only. Finally, we show how Yu, Koudas and Zuzarte combine synopses and sampling [34].

3.1 Synopses-based Selectivity Estimation

As discussed in Section 2.4, synopses structures provide β-selectivities for certain single attributes or attribute groups. Synopses-based estimators use this information to derive estimates for the unknown selectivities.

Traditionally, synopses-based estimators assume independence. Thus, given a query with index set N and two approximately known β-selectivities $\beta(X)$ and $\beta(Y)$ with $X, Y \subseteq N$ and $X \cap Y = \emptyset$, the joint selectivity $\beta(X \cup Y)$ is simply computed as $\beta(X) * \beta(Y)$.

A novel heuristic by Microsoft [5] takes statistical relationships into account by assuming that there naturally exist similar statistical relationships among attributes. Since

database systems usually store statistics for at least all single columns, the β-selectivity for each of the n simple predicates in a query can be extracted. In a next step, these selectivities are sorted in ascending order, i.e., $\beta(1) \leq \beta(2) \leq \cdots \leq \beta(n)$. The core idea is then to take the first k selectivities and compute an estimate as $\prod_{i=1}^{k} \beta(i)^{0.5^{(i-1)}}$. Note that this implies that $n-k$ selectivities do not contribute to the estimate. However, since $\beta(i+1) \geq \beta(i)$ and due to the *exponential back-off*, the factor $\beta(i)^{0.5^{(i-1)}}$ converges to the limit 1, where it does not change to the product and, therefore, to the estimate anyway. The Microsoft SQL Server Team choses $k=4$ and justifies their choice with the rapid speed of convergence. To give some intuition for this choice of k, note that for an estimate computed with $k=4$ to differ from an estimate computed with $k=5$ by 50%, i.e., $s_5^{1/16} = 0.5$, the fifth most selective selectivity s_5 must have a value of around 0.00002.

In their work on synopses-based *consistent selectivity estimation via maximum entropy* [20], Markl et al. show how to exploit all available synopses to obtain more accurate estimates. Their method is based on the principle of maximum entropy. Thus, in absence of knowledge, their estimator assumes independence. One benefit of maximum entropy is its interpretability: When estimates are bad, more information on attribute combinations not satisfying the independence assumption is needed. To find the desired selectivities, Markl et al. formulate an optimization problem. The objective is to find the γ-selectivities that maximize the entropy subject to the constraints given by the system of equations $Ax = b$ that we introduced in Section 2.4. However, it may be the case that $Ax = b$ is unsolvable, in which case there exists no solution to the optimization problem. $Ax = b$ is unsolvable if the β-selectivities in b are mutually inconsistent, meaning that not all selectivities in b can hold true at the same time. Such inconsistencies occur because synopses structures usually only yield approximations for selectivities. To ensure that $Ax = b$ is solvable, Markl et al. adjust the β-selectivities in b when necessary. The adjustments are computed in a minimal way with respect to a metric. Different approaches exist in the literature to compute minimal adjustments of b, based on l_1-norm [20] and based on l_q-norm [24]. Once a solution for the vector of γ-selectivities x is found, Equation 1 is applied to obtain estimates for the unknown β-selectivities.

3.2 Sampling-based Selectivity Estimation

A different approach to selectivity estimation is to rely on sampling. Many sampling-based only approaches can be found in the literature [19, 17, 10]. Oracle recently presented a sampling approach that adapts the sample size to the confidence intervals of the obtained selectivities [7].

3.3 HASE

To the best of our knowledge, Yu et al., in their *HASE* paper [34], were the first and only ones who combine selectivities from sampling and synopses. The core idea is to introduce variables that compensate for differences between sampling and synopses. Then, find the smallest compensation factors possible.

In the following, we describe in a nutshell how HASE works. Note that we have reformulated their problem specification to make it consistent with our notation: As before, let N denote the index set of a given conjunctive query P. Assume, due to some synopses, we are given a $1+|N|$-dimensional vector of β-selectivities containing the β-selectivity when no predicate is applied, i.e., 1, and the β-selectivity of each simple predicate in the given conjunctive query. In addition, suppose, due to sampling, we are given a $2^{|N|}$-dimensional vector x that contains an estimate for each γ-selectivity.

HASE(b, x)

Input: known selectivities b, sampling results x
Output: estimated β-selectivities
1 let w be a variable weight vector
2 let c be a constant dampening vector
3 $\underset{w}{\text{minimize}} \quad cD(w)$
subject to $\quad A(x \circ w) = b$
4 **return** $C(x \circ w)$

That is, synopses give us some β-selectivities and sampling gives us γ-selectivities. However, due to the imprecise nature of sampling, we expect that the sampling-selectivities are not consistent with the synopses-selectivities, i.e., $Ax \neq b$. Here, A is a $(1+|N|) \times 2^{|N|}$ partial design matrix, as described in Section 2.4. Hence, Yu et al. introduce a weight vector w for compensation. Then, an admissible solution satisfies $A(x \circ w) = b$, where $\circ$ denotes entry-wise multiplication. In general, there exist infinitely many of such assignments for the weights. The objective is to find the one with a minimal sum of (mapped) weights. The weight vector w is mapped using a distance function D and a dampening vector c, that associates a user-defined dampening factor with each component in w. A codification of this process is given in Algorithm HASE.

The limitation of HASE is that they can handle 1-dimensional synopses only. A generalization to multi-dimensional synopses introduces potential mutual inconsistencies, however they do not consider methods to overcome inconsistencies in the known selectivities. In addition, as we will see in our evaluation, HASE fails at exploiting the potential of combining sampling and synopses in terms of accuracy.

4. COMBINED SELECTIVITY ESTIMATION

In this section, we present CSE, a novel technique to estimate the selectivities for some conjunctive query P. Section 4.1 demonstrates how to construct sampling bounds by deriving confidence intervals for all γ-selectivities associated with P. Section 4.2 shows how to derive bounds on β-selectivities from synopses. In Section 4.3 we show how to formulate a constrained optimization problem where the constraints are given by the bounds obtained in Sections 4.1 and 4.2. The optimal solution to this optimization problem serves as the selectivity estimate. One approach to compute the optimal solution is presented in Section 4.4.

4.1 Sampling Bounds

Sampling allows one to estimate the selectivity of any type of predicate. It is well-known that an unbiased estimate of the selectivity of some conjunctive query can be obtained by counting the number of qualifying samples and dividing this number by the sample size. However, if the number of

qualifying samples is low, the quality of the estimate is often insufficient. Here we want to investigate a method that uses a sample to construct confidence intervals for γ-selectivities such that the true γ-selectivity is contained in the interval with high probability.

Assume we draw a random sample of size m from a population of size M. Say we observe that k_X items in the sample satisfy the predicate $F_\gamma(X)$ corresponding to some $X \subseteq N$, where N refers to the index set of some conjunctive query. The goal is to determine K_X, the number of items in the population that qualify, since $K_X/M = \gamma(X)$. Clearly, using M, m, k_X, we cannot find an approximation for K_X that is guaranteed to be correct.

However, we can bring certainty, to an arbitrary degree, to sampling by constructing confidence intervals $[\gamma^l(X), \gamma^u(X)]$ with high confidence levels. Later, we use these confidence intervals to estimate unknown β-selectivities.

We model sampling as an *urn problem* with the following characteristics: (1) each item either qualifies or does not qualify, (2) we draw without replacement. This urn problem induces the *hypergeometric distribution* [15, Chapt. 3.2]

$$\Pr(Z = k_X) = \frac{\binom{K_X}{k_X}\binom{M-K_X}{m-k_X}}{\binom{M}{m}},$$

where $\Pr(Z = k_X)$ denotes the probability of drawing exactly k_X items that qualify in m draws.

The goal is to determine the pair of random variables lower-K_X, denoted by K_X^l, and upper-K_X, denoted by K_X^u, such that

$$\Pr(K_X^l \leq K_X \leq K_X^u) = 1 - \alpha,$$

where $\alpha \in (0, 1)$. Then, $[K_X^l, K_X^u]$ is a confidence interval for K_X with confidence level $1-\alpha$. We experimentally found $\alpha = 10^{-3}$ to be a good value. Therefore, we are quite certain the true cardinality K_X lays in the computed bounds.

Unfortunately, exact methods are computationally expensive [30]. However, assuming $m \ll M$, the hypergeometric distribution coincides with the binomial distribution[1]. Then the *Wilson Score interval* method with continuity correction [26] provides us with an approximation for

$$[\frac{K_X^l}{M}, \frac{K_X^u}{M}] =: [\gamma^l(X), \gamma^u(X)].$$

The method is derived by the Yate's chi-squared test, used to test how likely it is that differences in observations occur by chance. The interval boundaries are efficiently computed as

$$[\frac{2k_X + z^2 - T}{2(m+z^2)}, \frac{2k_X + z^2 + T}{2(m+z^2)}], \tag{3}$$

where $T = z\sqrt{z^2 - \frac{1}{m} + 4k_X(1 - \frac{k_X}{m}) + (4\frac{k_X}{m} - 2)} + 1$, and z denotes the $1 - \frac{\alpha}{2}$ quantile of a standard normal distribution. However, if no sample items qualify, the lower bound must be taken as 0. Similarly, if all items qualify, the upper bound must be taken as 1.

Note that there exist several other methods to construct (approximate) confidence intervals for the given distribution [31].

[1]which refers to sampling with replacement

GetSamplingBounds(P, n, N, S, m, α)

```
1  let counts be an associative array
2  for each t ∈ S
3       let X be a bitvector of size n
4       for i = 0 to n
5            if p_i(t)
6                 X[i] = 1
7            else
8                 X[i] = 0
9       counts[X] = counts[X] + 1
10 z = QuantileStandardNormalDist(1 − α/2)
11 let x^l and x^u be associative arrays of size 2^n
12 for each X ⊆ N
13      k_X = counts[bv(X)]
14      T = z√(z^2 − 1/m + 4k_X(1 − k_X/m) + (4k_X/m − 2)) + 1
15      x^l[X] = (2k_X + z^2 − T)/(2(m + z^2))
16      x^u[X] = (2k_X + z^2 + T)/(2(m + z^2))
17 return (x^l, x^u)
```

Algorithm GetSamplingBounds describes how to compute $[\gamma^l(X), \gamma^u(X)]$ for all $X \subseteq N$. The result is stored in two vectors x^l and x^u for which it holds that

$$x^l \leq x \leq x^u, \tag{4}$$

with high probability. The algorithm first computes k_X for all $X \subseteq N$ in a single pass over the sample. Then, Equation 3 is applied to compute lower and upper bounds given a significance level α. The input is the conjunctive query P with n simple predicates and index set N, a sample S of size m and significance level α.

A notable property is that if the confidence interval is wide for some X it must be tighter for others, since each item in the sample must qualify the γ-predicate corresponding to some $X \subseteq N$. To see that, recall the discussion of minterms in Section 2.

4.2 Synopses Bounds

We have seen in Section 2.4 that synopses structures provide β-selectivities for certain single attributes or attribute groups. As mentioned in the related work section, provided β-selectivities are usually approximations of the true β-selectivity of some predicate. As such, they are subject to approximation errors. Approximation errors occur since, e.g., histograms, approximate selectivities based on frequencies and boundaries of buckets and make assumptions regarding the distribution of values in histogram buckets. This can cause a system of equations induced by synopses $Ax = b$ to be unsolvable. As mentioned in the related work section, in such cases Markl et al. propose adjustments of the β-selectivities in the vector b to make $Ax = b$ solvable. However, adjusting selectivities adds a time-consuming step to the selectivity estimation process, cf. [20] where adjusting took longer than estimation.

Here, we investigate an approach that relies on boundaries of histogram buckets. Figure 2 shows an example histogram that approximates the distribution of an attribute `Age`. The bar ranging from the first to the third bucket represents a range predicate. Clearly, since we have a histogram for `Age`, the index of this predicate would be part

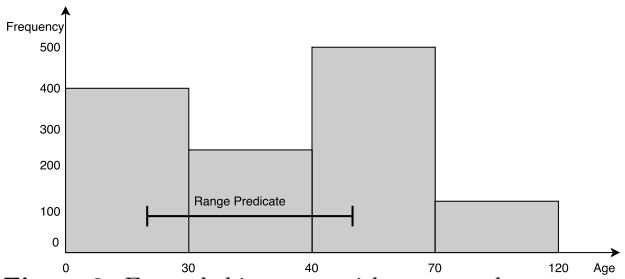


Figure 2: Example histogram with an example range predicate.

of the knowledge set G. However, the only thing we actually know is that the frequency of the second bucket is part of the result frequency, since the range of the predicate spans further than the bucket. The frequencies of the first and third bucket can only be approximated via an intra bucket approximation scheme, that is, e.g., simply assuming an uniform distribution. Such assumptions can translate to wrong approximations, which ultimately lead to unsolvable systems of equations induced by synopses $Ax = b$.

Hence, we use bucket boundaries to derive an interval that contains the true frequency. In our example, this interval ranges from the frequency of the second bucket to the cumulated frequencies of the first three buckets. Note that a selectivity is simply the relative counterpart of a frequency. Therefore, for a given conjunctive query P with index set N, we can derive a lower bound $\beta^l(X)$ and an upper bound $\beta^u(X)$ for each X in the knowledge set G.

The bounds depend on the bucketing scheme of a histogram. In commercial database systems, histograms with different bucketing schemes can be found. Oracle uses top frequency histograms and equi-depth histograms. By default, the maximum number of buckets is 254 [1]. In IBM DB2, they employ equi-depth histograms with a maximum of 100 buckets [2]. In Microsoft SQL Server or Azure SQL Database, respectively, maxdiff histograms with a maximum of 200 buckets are used [4]. For multi-dimensional synopses, we recommend histograms with tight bounds, e.g., [29].

SAP HANA uses maxdiff histograms and q-optimal histograms, i.e., histograms with a maximum multiplicative error for estimates [3]. Q-optimal histograms give intra bucket guarantees that allow one to specify the width of the bounds upon histogram construction. Hence, given an obtained frequency, the bounds are already known.

The obtained lower bound $\beta^l(X)$ and upper bound $\beta^u(X)$ for each X in G then allow us to formulate the system of inequalities

$$b^l \leq Ax \leq b^u, \tag{5}$$

where $b^l = (\beta^l(\emptyset), \ldots, \beta^l(N))^T$ is the vector of known lower bounds and $b^u = (\beta^u(\emptyset), \ldots, \beta^u(N))^T$ the vector of known upper bounds. Note that we always have that $\beta^l(\emptyset) = \beta^u(\emptyset) = 1$. This system of inequalities is consistent and solvable. Thus, we can find solutions for x, the vector of γ-selectivities.

4.3 Estimating Selectivities: The Optimization Problem

Suppose, we are given sampling bounds x^l and x^u as stated in Inequality 3 and, in addition, synopses bounds b^l and b^u as stated in Inequality 5. In this section, we show how to constrain an optimization problem to these bounds for the purpose of selectivity estimation.

For the objective function, we adopt the maximum entropy principle [20], since we consider it reasonable to assume independence in absence of knowledge. The entropy is maximized by the most uniform admissible solution. In vector form, the entropy function is given by $-x^T \log(x)$ and can be maximized by minimizing its negative form. Note that in principle, every convex objective function allows one to find a global optimum.

Then, using the negated entropy function as objective function and the bounds from sampling and synopses as constraints, we formulate the constrained optimization problem

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & x^T \log(x) \\ \text{subject to} \quad & b^l \leq Ax \leq b^u, \\ & x^l \leq x \leq x^u. \end{aligned} \tag{6}$$

The solution vector x serves as an estimate for all γ-selectivities. By applying Equation 1 to all $X \subseteq N$, we obtain estimates for all β-selectivities.

Note that the optimization problem is subject to two types of constraints: (1) bounds on variables, often referred to as *box constraints*, and (2) linear inequality constraints. Furthermore note that the objective function is strictly convex, which allows one to find the global minimum by searching for a local one. Strict convexity of a multidimensional function can be shown by proving that its Hessian matrix is positive definite.

Further note that in rare cases it happens that the solution space of Problem 6 is empty. One solution to this problem is to widen the bounds and try to solve the problem again. Another possibility is to find a solution that minimizes the constraint violation, e.g., in l_1-norm. We apply a solver that implements the latter, as discussed in Section 4.4.4.

Example: Suppose we are to estimate the selectivity of a conjunctive query that only contains one predicate p with index set $\{1\}$. In the preliminaries we observed that the selectivity of no predicate being applied $\beta(\emptyset)$ is always 1. However, assume for the sake of the graphical illustration of the optimization problem in Figure 3 that we only know $b^l = (\beta^l(\emptyset)) = (0.3)$ and $b^u = (\beta^u(\emptyset)) = (1)$ for the lower and upper bounds on β-selectivities, respectively. In addition, suppose that sampling, as described in Section 4.1, gives us $x^l = (\gamma^l(0), \gamma^l(1))^T = (0.1, 0.05)^T$ for the lower bounds and $x^u = (\gamma^u(0), \gamma^u(1))^T = (0.6, 0.7)^T$ for the upper bounds on γ-selectivities, where $\gamma(0)$ denotes the selectivity of $\neg p$ and $\gamma(1)$ denotes the selectivity of p. Then our knowledge induces the following optimization problem

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & x^T \log(x) \\ \text{subject to} \quad & (0.3) \leq \begin{pmatrix} 1 & 1 \end{pmatrix} x \leq (1), \\ & \begin{pmatrix} 0.1 \\ 0.05 \end{pmatrix} \leq x \leq \begin{pmatrix} 0.6 \\ 0.7 \end{pmatrix}, \end{aligned}$$

where $\begin{pmatrix} 1 & 1 \end{pmatrix}$ defines the 1×2 design matrix that corresponds to matrix A in optimization problem 6.

The optimal solution is $x = (0.367, 0.367)$ and is marked as the point of *Max Entropy* in Figure 3. Note that this point is no vertex. Furthermore note that the sum over all estimated γ-selectivities given in vector x does not add up to 1, since we did not set $\beta^l(\emptyset) = \beta^u(\emptyset) = 1$ in this example.

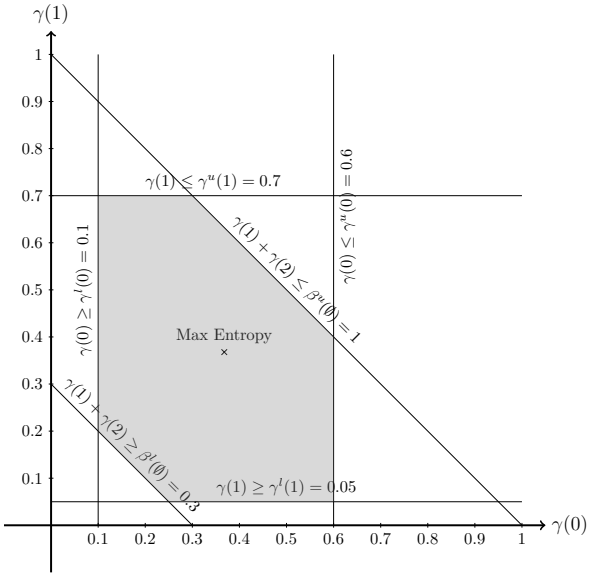


Figure 3: Graphical representation of an optimization problem example for a single predicate.

The final estimate for the selectivity of p, denoted by $\beta(1)$, is obtained by applying Equation 1, which yields $\beta(1) = \sum_{\{1\} \subseteq Y \subseteq N} \gamma(Y) = 0.367$.

4.4 Solving the Optimization Problem

In this section, we describe Mehrotra's predictor-corrector algorithm, which can be applied to Problem 6. Note that this section is complementary to the previous one: Those who are not interested in how to solve optimization problems may skip this section.

Mehrotra's algorithm is well-established and implemented in optimization libraries such as IPOPT [33], which we apply. The algorithm belongs to the class of interior point methods, which find an optimal solution by following a path through the feasible region.

4.4.1 Rewrite

We rewrite the constraints in Problem 6 to simplify the discussion of Mehrotra's predictor-corrector algorithm. Note that no rewrite is required to apply IPOPT. After the rewrite of the constraints, the optimization problem is subject to greater-equals inequality constraints only. To this end, we leave $Ax \geq b^l$ unchanged and write $Ax \leq b^u \iff -Ax \geq -b^u$. For the variable constraints, we make use of the $2^n \times 2^n$ identity matrix I and rewrite $x \geq x^l$ to $Ix \geq x^l$ and $x \leq x^u$ to $-Ix \geq -x^u$.

Then, the rewritten problem is given by

$$\begin{aligned} &\text{minimize} && x^T \log(x) \\ &\text{subject to} && \begin{bmatrix} A \\ -A \\ I \\ -I \end{bmatrix} x \geq \begin{bmatrix} b^l \\ -b^u \\ x^l \\ -x^u \end{bmatrix}. \end{aligned} \tag{7}$$

We write $Mx \geq c$ as the short version of the constraints and denote by m the number of rows in M or c.

4.4.2 Optimality Conditions

An optimal solution to optimization problem 7 satisfies the Karush-Kuhn-Tucker conditions. In our case we have

$$\log(x) + e - M^T\lambda = 0 \tag{8a}$$
$$Mx - y - c = 0 \tag{8b}$$
$$y_i\lambda_i = 0 \qquad i = 1, 2, \ldots, m \tag{8c}$$
$$y, \lambda \geq 0 \tag{8d}$$

where 8a states that the gradient with respect to x of the Lagrangian for Problem 7 $\nabla_x \mathcal{L}(x, y, \lambda)$ must be zero. $\log(x)+e$ with $e = (1, 1, \ldots, 1)^T$ is the gradient of the objective function $x^T \log(x)$. Condition 8b states that the constraints in Problem 7 must hold. Here, $y \in R^{|G|+|G|+2^n+2^n}$ is a slack vector that compensates for the inequalities $Mx - c \geq 0$. The conditions 8c state that either (1) constraint i is *active*, meaning its slack variable y_i is zero and constraint i effectively imposes an equality constraint at this point, or (2) constraint i is inactive at this point, then its Lagrange multiplier λ_i must be zero. Hence, an equivalent way of writing conditions 8c is $(Mx - c)_i\lambda_i = 0$ for $i = 1, 2, \ldots, m$.

Since our objective function is strictly convex and all our constraints are linear, the aforementioned necessary conditions are also sufficient.

Mehrotra's predictor-corrector algorithm finds a solution that satisfies the optimality conditions in 8 in an iterative process. In each iteration, a new iterate is computed as

$$(x, y, \lambda) + \alpha(\Delta x, \Delta y, \Delta\lambda),$$

where (x, y, λ) is the current iterate, $(\Delta x, \Delta y, \Delta\lambda)$ is the search direction and α is the step size.

Conceptually speaking, while iterating, the search direction handles optimality conditions 8a-8c, while the step size selection provides an α that respects condition 8d and the *sufficient decrease* condition, which will be discussed later.

4.4.3 Search Direction

When determining the search direction we do two things. (1) We ignore 8d. (2) We do not force $y_i\lambda_i$ to be zero in 8c, but to be a pre-defined fraction $\sigma \in [0, 1]$ of the average value of the pairwise products in y and λ, i.e., $\mu := 1/m \sum_{i=1}^{m} y_i\lambda_i = y^T\lambda/m$. A choice of $\sigma > 0$ tends to allow for larger step sizes, since y and λ are biased towards positivity. The system we obtain for the search direction is

$$F(x, y, \lambda) := \begin{bmatrix} \log(x) + e - M^T\lambda \\ Mx - y - c \\ Y\Lambda e - \sigma\mu e \end{bmatrix} = 0, \tag{9}$$

where $Y := \text{diag}(y_1, y_2, \ldots, y_m)$ and $\Lambda := \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_m)$.

Those solutions to Equation 9 that additionally satisfy $y, \lambda > 0$ represent the so-called *central path* that leads to an optimal solution as $\sigma\mu$ approaches zero, since then the optimality conditions, stated in 8, are satisfied.

We then formulate the following linear approximation $\hat{F}$ for F to predict a search direction

$$\hat{F}(x+\Delta x^p, y+\Delta y^p, \lambda+\Delta\lambda^p) = F(x, y, \lambda) + J(x, y, \lambda) \begin{bmatrix} \Delta x^p \\ \Delta y^p \\ \Delta\lambda^p \end{bmatrix},$$

where $J(x, y, \lambda)$ denotes the Jacobian of F.

Equations 9 tell us to find a root of F, and hence, we set $\hat{F}(x+\Delta x^p, y+\Delta y^p, \lambda+\Delta\lambda^p)=0$. Computing $-F(x,y,\lambda)$ and plugging in the values for F and J we get

$$\begin{bmatrix} L' & 0 & -M^T \\ M & -I & 0 \\ 0 & \Lambda & Y \end{bmatrix} \begin{bmatrix} \Delta x^p \\ \Delta y^p \\ \Delta\lambda^p \end{bmatrix} = \begin{bmatrix} -L+M^T\lambda \\ -Mx+y+c \\ -Y\Lambda e+\sigma\mu e \end{bmatrix}, \quad (10)$$

where $L := \log(x)+e$ and $L' := \frac{\partial L}{\partial x_i} = \text{diag}(1/x)$ denotes the Hessian of the objective function. Solving this system for $(\Delta x^p, \Delta y^p, \Delta\lambda^p)^T$ is called the predictor step. Its result can be used as a search direction. Note that for $\sigma = 0$ the search direction is the same as in Newton's method in optimization when solving 8a - 8c.

However, additionally performing a so-called correction step, defined by the following system, tends to reduce the number of iterations until convergence [27, Chapt. 14.2, 16.6]

$$\begin{bmatrix} L' & 0 & -M^T \\ M & -I & 0 \\ 0 & \Lambda & Y \end{bmatrix} \begin{bmatrix} \Delta x^c \\ \Delta y^c \\ \Delta\lambda^c \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\Delta X^p \Delta S^p e \end{bmatrix}, \quad (11)$$

where we solve for $(\Delta x^c, \Delta y^c, \Delta\lambda^c)^T$. Finally, the search direction for the current iteration becomes $(\Delta x^p, \Delta y^p, \Delta\lambda^p)+(\Delta x^c, \Delta y^c, \Delta\lambda^c)$.

4.4.4 Step Size

In each iteration, a step size α is selected such that the by now ignored condition 8d, stating $y, \lambda \geq 0$, is satisfied. The maximum step size we consider is $\alpha^{max} = \max\{\alpha \in (0,1] : (y,\lambda)+\alpha(\Delta y, \Delta\lambda) \geq \tau(y,\lambda)\}$, which satisfies the condition $(y,\lambda)+\alpha(\Delta y,\Delta\lambda) \geq 0$ with a buffer controlled by $\tau \in (0,1)$. In addition, a step size α must lead to sufficient decrease of a so-called merit function. A merit function combines the two goals, reducing the objective function and satisfying the constraints, in one function. For our problem, a valid merit function is

$$\phi(x,y) = x^T log(x) - v||Mx-y-c||, \quad (12)$$

where v is a penalty parameter and can be chosen to be the largest Lagrange multiplier λ_i in λ, but many other choices exist [27, Chapt. 15.4, 18.3], and $||\cdot||$ can be chosen to be the l_1-norm. With regard to solvability of Problem 6, the merit function ϕ gives interesting insights. For a solvable problem, the second term of ϕ eventually vanishes and the merit function coincides with the objective function. While, given an unsolvable problem, as ϕ is decreased, we find a solution that minimizes the constraint violation.

Then, to find an $\alpha \in (0, \alpha^{max}]$ that provides sufficient decrease of the merit function ϕ, we perform a backtracking line search, where we start with $\alpha = \alpha^{max}$ and iteratively decrease α until

$$\phi(x+\alpha\Delta x, y+\alpha\Delta y) \leq \phi(x,y) + \eta\alpha D_{\Delta x,\Delta y}\phi(x,y), \quad (13)$$

where $\eta \in (0,1)$ and $D_{\Delta x,\Delta y}\phi(x,y)$ denotes the directional derivative of $\phi(x,y)$ in the direction Δx and Δy, see [27, Chapt. A.2] for details.

4.4.5 Starting Point

A starting point has to satisfy only the positivity constraints $x, y, \lambda > 0$. In particular, it is not required to lay in the feasible region. However, the choice of the starting point impacts how fast the algorithm converges. Various heuristics exist [27, Chapt. 14.2, 16.6]. In our case, we choose

CSE($b^l, b^u, x^l, x^u, \epsilon, maxDuration$)

```
1  let Mx ≥ c be the constraints as written in Eq. 7
2  let y be a slack vector
3  Choose (x^(0), y^(0), λ^(0)) > 0
4  k = 0
5  maxTimePoint = TIME() + maxDuration
6  repeat
7      (x, y, λ) = (x^(k), y^(k), λ^(k))
8      μ = y^T * λ/m
9      Choose σ ∈ [0, 1]
10     Solve eq. 10 to obtain (Δx^p, Δy^p, Δλ^p)
11     Solve eq. 11 to obtain (Δx^c, Δy^c, Δλ^c)
12     (Δx, Δy, Δλ) = (Δx^p, Δy^p, Δλ^p)
                      +(Δx^c, Δy^c, Δλ^c)
13     Choose τ ∈ (0, 1)
14     α = max{α ∈ (0, 1] : (y, λ)+α(Δy, Δλ) ≥ τ(y, λ)}
15     while Eq. 13 not satisfied
16         α = α/2
17     (x^(k+1), y^(k+1), λ^(k+1)) = (x, y, λ)+α(Δx, Δy, Δλ)
18     k = k + 1
19 until ||(x^(k), y^(k), λ^(k))−(x^(k−1), y^(k−1), λ^(k−1))|| < ε
       or TIME() > maxTimePoint
20 return Cx_k
```

$x_0 = (x^l + x^u)/2$ and leave the initialization of y and λ to IPOPT [33].

4.4.6 Exit Condition and Codification

Ideally, we iterate until the optimality conditions, stated in 8, are satisfied. However, a practical convergence criterion is to terminate when the distance between consecutive iterates $||(x^{(k)}, y^{(k)}, \lambda^{(k)}) - (x^{(k-1)}, y^{(k-1)}, \lambda^{(k-1)})||$ is smaller than some small value ϵ, since we cannot expect to make significant progress beyond this point. Here, $||\cdot||$ denotes l_2-norm. In many applications, though, it is critical to obtain a fast estimate. We account for that by a maximum time span. Of course, time constraints provide no guarantees with respect to optimality.

Algorithm CSE shows the complete pseudo code. The parameters are synopses bounds b^l and b^u, sampling bounds x^l and x^u, as well as arguments to test the exit condition as described above. In line 1, the system $Mx \geq c$ is formulated as in Equation 7. Then, a slack vector y is introduced. In line 5, the latest time point for another iteration is determined. Then, in each iteration, μ and σ are computed to determine how much optimality condition 8c is relaxed. A simple strategy is to always choose $\sigma = 0$. For a description of adaptive choices, as used in IPOPT, see [27, Chapt. 14.2, 19.3]. Next, predictor and corrector steps are performed by solving Equation 10 and 11, respectively, to obtain a search direction. To determine a step size α, we first find the maximum step size α that preserves the positivity condition 8d with some specified buffer $\tau \in (0,1)$, e.g., $\tau = 0.005$. Then, we iteratively halve α until the sufficient decrease condition, given by Inequality 13, is satisfied. The current iterate plus a step of length α in the search direction gives the new iterate. After the last iteration, the vector of all estimated β-selectivities Cx_k is returned, where C denotes the complete design matrix introduced in Section 2.3.

5. EXPERIMENTAL EVALUATION

We evaluate our approach (*CSE*) and compare it to the accuracy and run time of several existing estimation techniques that we have seen in the related work section. To this end, we consider the sophisticated methods by Yu et al. (*HASE*) [34] and Markl et al. (*MaxEntropy*) [20]. In addition, we include Microsoft's exponential back-off estimator (*MsExpBackOff*) [5] as an up-to-date industry approach as well as simple random sampling (*Sampling*), i.e., the number of qualifying samples divided by the sample size, as a sampling-only estimator in our evaluation. Lastly, in the first part of our evaluation, we consider the estimates obtained by applying the independence assumption (*Ind. Ass.*). The following table shows for each estimator the type of information it processes:

Estimator	Sampling	Synopses
CSE	✓	✓
HASE	✓	✓
MaxEntropy		✓
MsExpBackOff		✓
Sampling	✓	
Ind. Ass.		✓

Note that HASE, MsExpBackOff and Ind. Ass. all process only one-dimensional synopses, i.e., single column statistics. Furthermore note that MaxEntropy requires adjustment steps when processing multi-dimensional synopses as described in Section 3. For further details, recall the description of each model given in the related work section. If nothing else is mentioned, we model a data management system that provides estimators with a 1% sample of the data, as in related work [8, Chapt. 2], and one-dimensional histograms as well as two-dimensional histograms that capture correlations present in the data.

We use two real-world data sets in our experiments. (1) The *forest cover type* (forest) data set [18], which is popular in the machine learning community and contains more than 580k entries with 55 attributes. (2) The second data set represents joined data from the *daily global historical climatology network* (weather) [22, 21]. It comprises daily observations of climate records and contains about 3.4M entries with 7 attributes.

In our experiments, we consider different test scenarios that are defined by a data set and a number of predicates. For each test scenario, we run 10'000 conjunctive queries as in

$$\textbf{SELECT} * \textbf{FROM } dataset \\ \textbf{WHERE } p_1 \textbf{ and } \ldots \textbf{ and } p_n$$

where each p_i represents a range predicate over an attribute A with constants c_1, c_2

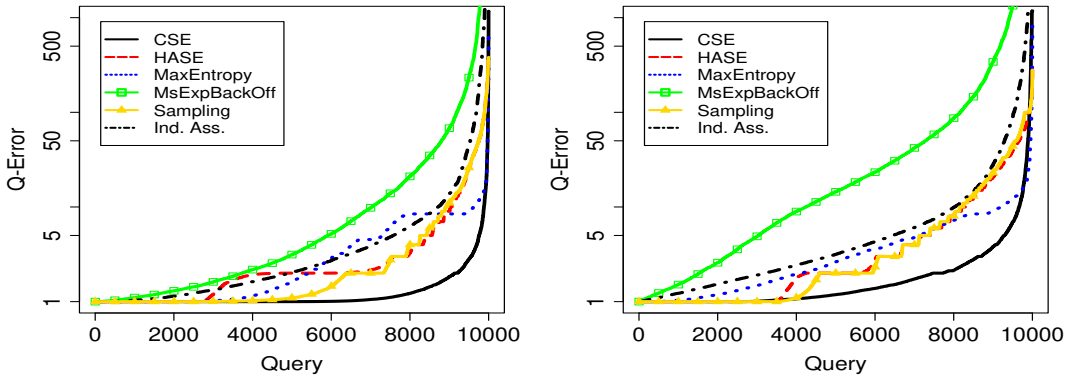
$$A \textbf{ between } c_1 \textbf{ and } c_2$$

The range predicates p_i are generated by drawing a random unused attribute A and choosing two random values from A's domain. The smaller value is used for c_1 and the larger value is used for c_2. In case A's domain has only two values, e.g., 0 and 1, we set $c_1 = c_2$. This effectively creates a point predicate.

5.1 Accuracy

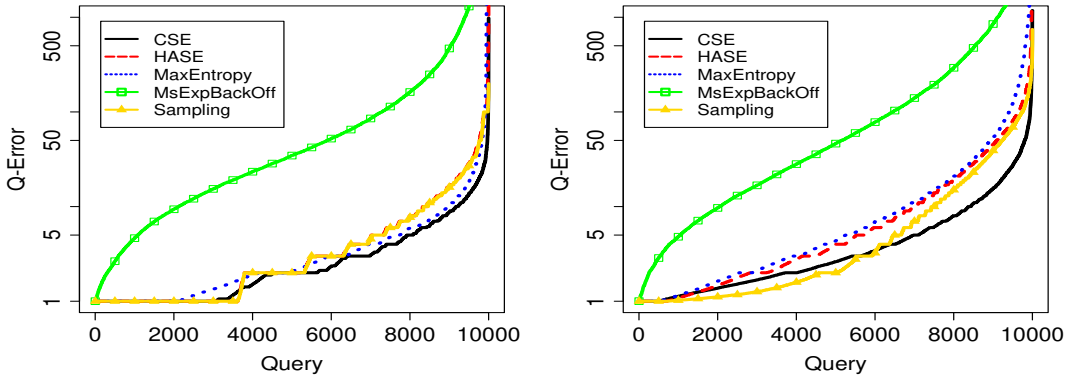
In this subsection, we look at the accuracy of estimates. The error metric used to measure the deviation between a selectivity estimate $\hat{s}$ and the true selectivity s is the *q-error* [14] defined as $max\{\frac{s}{\hat{s}}, \frac{\hat{s}}{s}\}$. Observe that this is a relative and symmetric metric. In query optimization, the q-error directly relates to query plan quality [25]. Choosing a suitable error metric is essential in model selection. While it may be obvious to see why an absolute error metric is a bad choice for our domain, it is harder to see that the commonly used relative error $(s - \hat{s})/s$ is insufficient for its asymmetry. To get an intuition, observe that in case of underestimation the worst possible error is 1, whereas in case of overestimation the error is unbounded. This causes an unacceptable systematic preference for estimators that underestimate the selectivity [32].

Notice that the q-error is undefined in case either the estimated or the true selectivity is zero. We configured all queries to return a non-empty result, hence, we do not have to worry about the true selectivity being zero. In addition, we programmed all estimators to estimate that at least one entry qualifies. Therefore, we do not have to worry about the estimated selectivity being zero. In a query optimization context, this makes sense to prevent faulty prunings in query plans.



(a) 3 predicates **(b)** 5 predicates

Figure 4: Sorted q-errors for the weather data set given one and two-dimensional synopses without approximation errors and a 1% sample.



(a) 7 predicates, weather data set **(b)** 7 predicates, forest data set

Figure 5: Sorted q-errors given one and two-dimensional synopses with approximation errors and a 1% sample.

5.1.1 *The Idealistic Case: Synopses Without Approximation Errors*

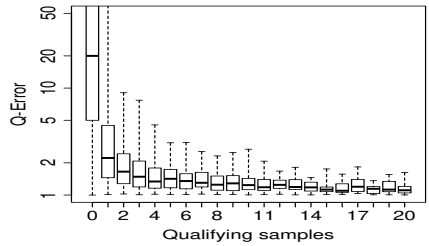
We first consider the idealistic case where histograms yield perfectly accurate selectivities. Note that in reality no database system has access to such histograms. We include this case to pick up the scenario that was considered in the related works on MaxEntropy [20] and HASE [34], where approximation errors were not considered.

In [20] queries with 3 predicates find special attention. Figure 4a shows the q-errors sorted in ascending order for each estimator for queries with 3 predicates over the weather data set. Note that the longer a curve remains flat, the better the corresponding estimator. Further note that the largest q-errors tend to occur in cases where either the estimate or the true selectivity is very small. We make the following observations in Figure 4a. (1) CSE and HASE tend to be the best-performing models, supporting the idea of combining synopses and sampling. (2) HASE mostly resembles Sampling. (3) For some queries MaxEntropy outperforms Sampling, while for others, it is the other way around. (4) The difference between CSE and MaxEntropy or Sampling, respectively, illustrates the benefit of combining synopses and sampling as we propose it. (5) Independence assumption and Microsoft's exponential back-off estimator tend to be less accurate than the other estimators.

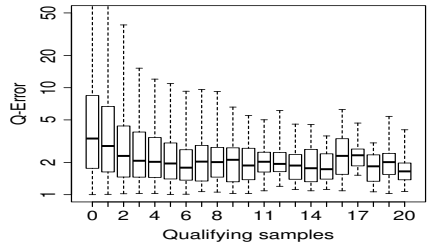
The largest queries considered in [34] contain 5 predicates. Figure 4b shows the accuracy of each estimator for queries of this size for the weather data set. Observe that (1) the curves of all estimators are shifted to the upper left as the number of predicates is increased from 3 to 5. This shift captures how selectivity estimation gets harder in the number of predicates in the conjunctive query. (2) The accuracy of HASE aligns with the accuracy of Sampling. (3) MaxEntropy worsens less than HASE or Sampling. (4) Independence assumption and Microsoft's exponential back-off estimator perform worst. Neither assuming independence nor a certain fixed degree of correlation seems to be the key to success.

5.1.2 *The Realistic Case: Synopses With Approximation Errors*

In this subsection, we consider the realistic case where synopses structures, such as histograms, yield selectivities



(a) Sampling

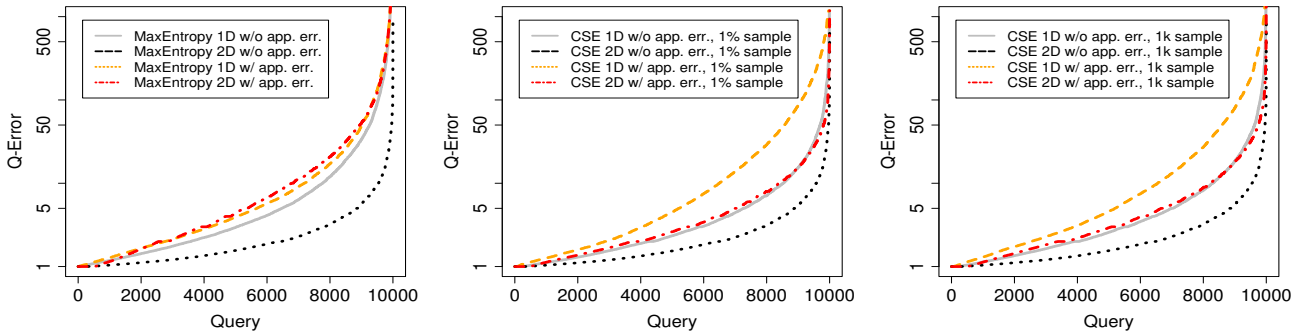


(b) CSE, synopses with approximation errors

Figure 6: q-errors in the number of qualifying samples for 7 predicate queries and a sample size of 1.000.

with approximation errors. We model q-optimal histograms [23], as can be found in SAP HANA [3], that guarantee a user-specified maximum multiplicative error for estimates, for which we choose a value of 2. Therefore, for each β-selectivity we want to provide to an estimator, we take the true β-selectivity, multiply it with a uniformly distributed multiplicative error in the range $[0.5, 2]$, and provide the product to the estimator.

Under this scenario, MaxEntropy requires adjustments of the provided selectivities as discussed in Section 3. We compute adjustments that are optimal under l_q, as described in [24]. CSE instead sets the lower bounds to $0.5 \times \beta(X)$ and the upper bounds to $2 \times \beta(X)$ for each provided selectivity $\beta(X)$ and is guaranteed to yield a feasible optimization problem. HASE, MsExpBackOff and Sampling operate as before, since they do not process multi-dimensional synopses.



(a) Max Entropy **(b)** CSE with a sample rate of 1% **(c)** CSE with a sample of size 1000

Figure 7: Sorted q-errors for queries with 7 predicates for the forest data set.

Figure 5 shows the sorted q-errors for queries with 7 predicates for both the weather and the forest data set. Taking a closer look, we make the following observations. (1) Depending on the data set, synopses-based estimators like MaxEntropy or sampling-based estimators yield more accurate estimates, cf. Figure 5a, where MaxEntropy outperforms Sampling, and Figure 5b, where Sampling outperforms MaxEntropy. (2) Generally, CSE does not perform (significantly) worse than the best of MaxEntropy and Sampling. This makes it a robust estimator. In addition, in most cases CSE clearly produces the best estimates. (3) HASE performs no better than Sampling. (4) MsExpBackOff only considers the four most selective one-dimensional predicates, hence, it ignores three given selectivities. As a result, MsExpBackOff lags far behind the other estimators for both data sets.

5.1.3 *Common Aspects in Sampling and CSE*

We now further investigate the impact of applying sampling, in addition to exploiting synopses, as we do it. Figure 6a shows a boxplot representation of the errors in the number of qualifying sample tuples. As usual, the bottom and top of the box are the first and third quartiles and the band in between represents the median. The lower and upper end of the whiskers represent the 1% and 99% percentile respectively. The last box aggregates the errors of all queries with more than 19 qualifying samples.

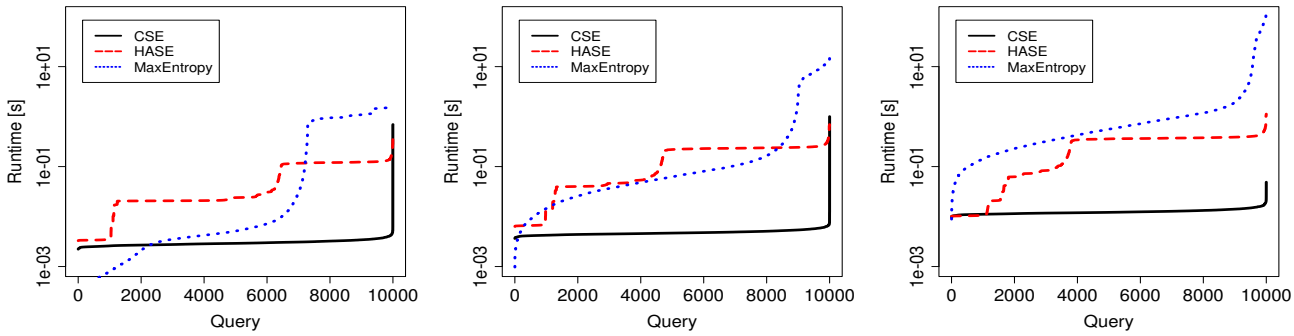
Observe how the errors of Sampling in Figure 6a decrease as more samples qualify, i.e., the estimates become more accurate. A practical way to see why the errors decrease is to observe how the confidence intervals close in in the number of qualifying samples. Since our approach incorporates sampling, it roughly follows this desirable trend, however, starting from a much lower error-level for few qualifying samples, cf. the y-axis-scales in Figures 6a and 6b. This is a competitive edge our approach gains over approaches that do not incorporate sampling.

One might get the impression that sampling is the method of choice in almost all cases, since the errors become acceptable quickly in the number of qualifying samples. This is a misperception, since the case where very few samples qualify are the dominant ones. For instance, in the experiments conducted to produce the graphs in Figure 6, in more than 7,000 out of 10,000 queries with 7 simple predicates the sample had zero qualifying tuples.

5.2 Dimensionality of the Given Synopses and Sample Size

Multi-attribute statistics are well-established in the research community. However, many database systems still maintain only single attribute statistics. In this subsection, we show how the accuracy of MaxEntropy and CSE changes as we provide one- or two-dimensional synopses. Ideally, more information results in more accurate estimates. Indeed, we observe that for each estimator in each test scenario the estimates improve as we go from one-dimensional synopses to two-dimensional synopses if the selectivities provided by the synopses structures have no approximation errors. As discussed earlier, though, this is an unrealistic assumption. In the realistic case, where synopses are subject to approximation errors, we observe in some test scenarios that the estimates become worse as more approximated selectivities were provided. Figure 7a compares the sorted q-errors of MaxEntropy for one-dimensional (1D) and one- and two-dimensional (2D) synopses. Note how the estimates improve in the idealistic case of synopses without approximation errors (w/o app. err.) but how they worsen in the case of synopses with approximation errors (w/ app. err.). This means that an estimator might perform better using less than all available information. However, this effect is data-dependent. For the weather data set, we do not observe this effect: more synopses always results in better estimates in all test scenarios for the weather data set.

Figure 7b shows the same graphs for CSE. This time in both cases, synopses with or without approximation errors, the estimates improve as we go from one-dimensional to one- and two-dimensional synopses. Since the same holds for the weather data set and different numbers of predicates, we conclude that CSE is robust in the sense that additional synopses has no negative impact on the estimates. Note that it is irrelevant how CSE 2D inacc performs in comparison to CSE 1D acc since CSE 1D acc represents a hypothetical case and we do not get to choose between synopses with or without approximation errors. Figure 7c illustrates that the graphs look similar even if the sample used in CSE is only of size 1.000. Compared to the 1% sample a sample of size 1.000 is more than a factor of 5 smaller for the forest data set. We observed that the results with a sample of size 1.000 are similar in many cases. From an industry point of view, that is good news since many database systems use such small samples.



(a) 3 predicates **(b)** 5 predicates **(c)** 7 predicates

Figure 8: Sorted runtimes given one and two-dimensional synopses with approximation errors.

Finally, note that CSE is designed for multi-dimensional synopses in combination with a sample, and we advise its application primarily in that context. In principle, CSE can be even applied to a sample only. However, as we experimentally confirmed, the most independent solution subject to sampling bounds is a terribly wrong estimate! Due to space constraints, we cannot include graphs.

5.3 Runtime

We performed single-threaded runtime measurements on a machine with Intel Skylake i5-6500 CPU with a clock rate of 3.20GHz with 16GB RAM. The machine is operated by a 64-bit linux. We employ the IPOPT library [33] to solve the optimization problem underlying our approach.

We only look at the run times of CSE, HASE and MaxEntropy. Note that Sampling and MsExpBackOff run orders of magnitude faster since they perform only simple arithmetics. Furthermore, we do not consider the cost of drawing a sample or extracting information from synopses structures but only the time it takes to produce an estimate given some information.

Figure 8 shows the run times of CSE, HASE and MaxEntropy for several problem sizes, i.e., various numbers of simple predicates. Note that we consider the realistic case where synopses are subject to approximation errors.

Looking at Figure 8a, we note that (1) MaxEntropy, which applies iterative scaling to solve the underlying optimization problem [20], works best in about 20% of all queries containing three simple predicates. That is, the overhead to get the algorithm started is low. (2) Furthermore the graph of MaxEntropy suggests that the runtime heavily depends on the known selectivities that define the constraints. (3) HASE and our approach CSE both seem to be much more independent from the values of the selectivities in the optimization problem. (4) HASE runs slower than our approach.

To see how the algorithms scale in the problem size, we analyze Figures 8a, 8b and 8c. Looking at the graphs of MaxEntropy, we observe that the runtime grows exponentially in the number of predicates which the authors stated themselves [20]. In their future work section they suspect an algorithm based on Newton's method to be faster. We have strong indication that this is true. After all, note that our approach solves the maximum entropy approach, as Markl et al. have stated it, when setting all lower and upper bounds on the variables to 0 and 1, respectively, and imposing equality constraints on the β-selectivities.

HASE and CSE both scale well. Looking at the changes in the underlying optimization problems, we find that the complexity of HASE grows slower due to fewer constraints: As we increase the number of simple predicates from n to $n+2$, the number of variables in the optimization problem quadruples since the number of γ-selectivities grows from 2^n to $2^{n+2} = 4 \times 2^n$ for both HASE and CSE. Only in CSE, a box constraint is associated with each variable, cf. Problem 6. For both approaches, the number of constraints induced by one-dimensional synopses grows from n to $n+2$. Additionally, CSE has constraints due to two-dimensional synopses which grow by a factor of $4n+2$ from $\frac{n(n-1)}{2}$ to $\frac{(n+2)((n+2)-1)}{2} = (4n+2)\frac{n(n-1)}{2}$.

6. CONCLUSION

We proposed CSE, a novel approach to combine sampling with synopses for the purpose of estimating the selectivity of conjunctive queries. The results of our experiments suggest that CSE indeed leads to more accurate selectivity estimates. Using two real-world data sets and a large number of queries, we showed the strengths and limitations of our and various other state-of-the-art approaches. Depending on the patterns in data set, a purely sampling-based estimator or a purely synopses-based estimator yields better selectivity estimates. Our approach, however, yields estimates that are at least as accurate as the estimates of the best competing estimator. This makes CSE a robust estimator, whose applicability does not depend on the data set.

For future work, we aim to enhance our method to address join selectivity estimation. Query optimizers are very sensitive to the accuracy of join selectivity estimates since errors propagate exponentially through joins [13]. There exist methods based on sampling [11, 12] as well as based on synopses [9, 28]. However, so far, methods that combine sampling and synopses for the purpose of join selectivity estimation seem to be missing.

7. REFERENCES

[1] Database SQL Tuning Guide: Histograms. https://docs.oracle.com/database/121/TGSQL/tgsql_histo.htm. Accessed: 2018-02-15.

[2] IBM Knowledge Center: Managing DB2 performance. https://www.ibm.com/support/knowledgecenter/en/SSEPEK_10.0.0/perf/src/tpc/db2z_histogramstatistics.html. Accessed: 2018-02-15.

[3] SAP HANA SQL and System Views Reference: Create statistics. https://help.sap.com/viewer/4fe29514fd584807ac9f2a04f6754767/2.0.02/en-US/20d5252d7519101493f5e662a6cda4d4.html. Accessed: 2018-02-15.

[4] Statistics in Microsoft's SQL Server and Azure SQL Database. https://docs.microsoft.com/en-us/sql/relational-databases/statistics/statistics. Accessed: 2018-02-15.

[5] The New and Improved Cardinality Estimator in SQL Server 2014. https://blogs.technet.microsoft.com/dataplatforminsider/2014/03/17/ the-new-and-improved-cardinality-estimator-in-sql-server-2014. Accessed: 2018-02-15.

[6] S. Agarwal, B. Mozafari, A. Panda, H. Milner, S. Madden, and I. Stoica. Blinkdb: queries with bounded errors and bounded response times on very large data. In *Proceedings of the 8th ACM European Conference on Computer Systems*, pages 29–42. ACM, 2013.

[7] S. Chakkappen, S. Budalakoti, R. Krishnamachari, S. R. Valluri, A. Wood, and M. Zait. Adaptive statistics in oracle 12c. *PVLDB*, 10(12):1813–1824, 2017.

[8] G. Cormode, M. Garofalakis, P. J. Haas, and C. Jermaine. Synopses for massive data: Samples, histograms, wavelets, sketches. *Foundations and Trends in Databases*, 4(1–3):1–294, 2012.

[9] A. Dobra. Histograms revisited: When are histograms the best approximation method for aggregates over joins? In *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 228–237. ACM, 2005.

[10] R. Gemulla. Sampling algorithms for evolving datasets. 2008.

[11] P. J. Haas, J. F. Naughton, S. Seshadri, and A. N. Swami. Fixed-precision estimation of join selectivity. In *Proceedings of the twelfth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 190–201. ACM, 1993.

[12] P. J. Haas, J. F. Naughton, S. Seshadri, and A. N. Swami. Selectivity and cost estimation for joins based on random sampling. *Journal of Computer and System Sciences*, 52(3):550–569, 1996.

[13] Y. E. Ioannidis and S. Christodoulakis. *On the propagation of errors in the size of join results*, volume 20. ACM, 1991.

[14] C.-C. Kanne and G. Moerkotte. Histograms reloaded: The merits of bucket diversity. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 663–674. ACM, 2010.

[15] R. J. Larsen, M. L. Marx, et al. *An introduction to mathematical statistics and its applications*, volume 5. Pearson, 2017.

[16] V. Leis, A. Gubichev, A. Mirchev, P. Boncz, A. Kemper, and T. Neumann. How good are query optimizers, really? *PVLDB*, 9(3):204–215, 2015.

[17] V. Leis, B. Radke, A. Gubichev, A. Kemper, and T. Neumann. Cardinality estimation done right: Index-based join sampling.

[18] M. Lichman. UCI machine learning repository, 2013.

[19] R. J. Lipton, J. F. Naughton, and D. A. Schneider. *Practical selectivity estimation through adaptive sampling*, volume 19. ACM, 1990.

[20] V. Markl, P. J. Haas, M. Kutsch, N. Megiddo, U. Srivastava, and T. M. Tran. Consistent selectivity estimation via maximum entropy. *The VLDB Journal*, 16(1):55–76, 2007.

[21] M. Menne, I. Durre, B. Korzeniewski, S. McNeal, K. Thomas, X. Yin, S. Anthony, R. Ray, R. Vose, B. Gleason, et al. Global historical climatology network-daily (ghcn-daily), version 3. *NOAA National Climatic Data Center*, 2012.

[22] M. J. Menne, I. Durre, R. S. Vose, B. E. Gleason, and T. G. Houston. An overview of the global historical climatology network-daily database. *Journal of Atmospheric and Oceanic Technology*, 29(7):897–910, 2012.

[23] G. Moerkotte, D. DeHaan, N. May, A. Nica, and A. Boehm. Exploiting ordered dictionaries to efficiently construct histograms with q-error guarantees in sap hana. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 361–372. ACM, 2014.

[24] G. Moerkotte, M. Montag, A. Repetti, and G. Steidl. Proximal operator of quotient functions with application to a feasibility problem in query optimization. *Journal of computational and applied mathematics*, 285:243–255, 2015.

[25] G. Moerkotte, T. Neumann, and G. Steidl. Preventing bad plans by bounding the impact of cardinality estimation errors. *PVLDB*, 2(1):982–993, 2009.

[26] R. G. Newcombe. Two-sided confidence intervals for the single proportion: comparison of seven methods. *Statistics in medicine*, 17(8):857–872, 1998.

[27] J. Nocedal and S. Wright. Numerical optimization, series in operations research and financial engineering. *Springer, New York, USA, 2006*, 2006.

[28] F. Rusu and A. Dobra. Sketches for size of join estimation. *ACM Transactions on Database Systems (TODS)*, 33(3):15, 2008.

[29] M. Shekelyan, A. Dignös, and J. Gamper. Digithist: a histogram-based data summary with tight error bounds. *PVLDB*, 10(11):1514–1525, 2017.

[30] S. K. Thompson. *Estimating Proportions, Ratios, and Subpopulation Means*, pages 57–66. John Wiley and Sons, Inc., 2012.

[31] M. Thulin et al. The cost of using exact confidence intervals for a binomial proportion. *Electronic Journal of Statistics*, 8(1):817–840, 2014.

[32] C. Tofallis. A better measure of relative prediction accuracy for model selection and model estimation. *Journal of the Operational Research Society*, 66(8):1352–1362, 2015.

[33] A. Wächter and L. Biegler. IPOPT-an interior point OPTimizer, 2009.

[34] X. Yu, N. Koudas, and C. Zuzarte. Hase: a hybrid approach to selectivity estimation for conjunctive predicates. In *International Conference on Extending Database Technology*, pages 460–477. Springer, 2006.