Join Size Estimation Subject to Filter Conditions

David Vengerov
Oracle Labs
529 Winterberry Way
San Jose, CA 95129
1-650-520-7157
david.vengerov@oracle.com

Andre Cavalheiro Menck
Oracle Corp.
1689 Sierra Street
Redwood City, CA 94061
1-425-691-9882
andre.menck@hotmail.com

Mohamed Zait
Sunil P. Chakkappen
Oracle Corp.
mohamed.zait@oracle.com
sunil.chakkappen@oracle.com

ABSTRACT

In this paper, we present a new algorithm for estimating the size of equality join of multiple database tables. The proposed algorithm, Correlated Sampling, constructs a small space synopsis for each table, which can then be used to provide a quick estimate of the join size of this table with other tables subject to dynamically specified predicate filter conditions, possibly specified over multiple columns (attributes) of each table. This algorithm makes a single pass over the data and is thus suitable for streaming scenarios. We compare this algorithm analytically to two other previously known sampling approaches (independent Bernoulli Sampling and End-Biased Sampling) and to a novel sketch-based approach. We also compare these four algorithms experimentally and show that results fully correspond to our analytical predictions based on derived expressions for the estimator variances, with Correlated Sampling giving the best estimates in a large range of situations.

1. INTRODUCTION

Accurate cardinality estimation of database queries is the most important problem in database query optimization. In 2014, a database expert with many decades of experience, Guy Lohman, pointed out in a blog post on wp.sigmod.org that the query cost model can introduce errors of at most 30%, while the query cardinality model can easily introduce errors of many orders of magnitude. He also pointed out that the grossest cardinality misestimates occur for join queries, but that regrettably little work has been done on accurate join size estimation. In this paper we contribute to this very important area of investigation.

In order to estimate the size of a join query very quickly, the database optimizer has to use a small synopsis of each table, which was constructed ahead of time. In some scenarios, there are only a few join graphs that are expected to be used most frequently, while the predicate filters that select the rows to be joined are specified at run time, uniquely for each query. For example, given a table of customer information and a table of product sales information with a column of customer ID who purchased each product, one may wish to repeatedly join these two tables on the customer ID attribute while considering only those customers that reside in each major city (i.e., applying a filter Customer.City = X).

Another requirement that we assume in this paper is that the synopsis of a table needs to be constructed in one pass. The size of database tables is growing very rapidly, and it would be a huge waste of resources to use a multi-pass method for constructing a synopsis of a table that holds many terabytes or petabytes of data. There are also some domains where the input data to joins consists of data streams and the synopsis needs to be constructed while the data is streaming by, so that it could be queried at any time about the latest estimate of the join size of two streams. While some researchers did address estimation of join sizes based on small-space synopses (as discussed in Section 2), we are not aware of any published one-pass methods that can work with filter conditions that are specified after the synopses are constructed.

In this paper, we present two new algorithms capable of solving this problem: Correlated Sampling and a novel sketch-based approach (whose details are given in Section 3.4.3). Given a join graph of interest, the Correlated Sampling algorithm constructs a synopsis of each table T_i by including each row from that table into the synopsis with probability p_i, which depends on the number of join attributes in T_i. Random variables that are used to make inclusion decisions are shared among the tables, as will be explained in Section 3.3. After the samples of all tables have been built, an unbiased join size estimate can be computed subject to dynamically specified filter conditions, without modifying the samples. This is achieved by simply selecting those rows from each sample that satisfy the filter condition for that table.

The rest of the paper is organized as follows. In Section 2 we discuss the related work. In Section 3 we present the competing algorithms for equijoin size estimation (Bernoulli Sampling, End-Biased Sampling, Correlated Sampling and a novel sketch-based approach) for the case of two tables and derive the variances of their estimates. In Section 4, we extend Correlated Sampling to multiple tables and complex join conditions. In Section 5, we present an experimental comparison of the approaches described in Section 3 and show that results fully correspond to our analytical predictions based on estimator variances that we derived for them. Section 6 concludes the paper.

2. RELATED WORK

An early review of existing sampling methods for estimating the selectivity of a join is given in [7]. The "cross-product" sampling scheme described in that work was shown to give the best estimates out of the simple sampling schemes considered. The "cross-product" scheme is equivalent to the one described in Section 3.1 if the sampling is performed at the level of rows rather than blocks.

The work in [5] refines the cross-product sampling scheme by introducing a separate treatment for tuples that have high

Proceedings of the VLDB Endowment, Vol. 8, No. 12
Copyright 2015 VLDB Endowment 2150-8097/15/08.

frequencies versus those that have low frequencies in the original tables. The resulting "Bifocal" sampling algorithm for join size estimation is proven to estimate the join size within an error factor close to 2 with high probability. Unfortunately, this algorithm relies on having an index of join attribute values for one of the tables in order to estimate the number of tuples in that table that join with each low-frequency tuple from the other table.

The "End-Biased" sampling algorithm [4] extends the work in [5] by eliminating the need for having an index of join values on any of the tables. It also uses the idea of performing "correlated" sampling (using a common hash function) of the tables to be joined. However, End-Biased Sampling requires making an extra pass through the data in order to compute the frequencies of the join attribute values.

The Correlated Sampling algorithm introduced in this paper extends the End-Biased Sampling algorithm by eliminating the need for knowing the frequencies of the join attribute values. Thus, it is suitable for streaming contexts or for very large databases where one-pass processing is required. It is also presented and analyzed for the case of multiple tables and complex join conditions, while the End-Biased sampling algorithm was only presented and analyzed for the case of two tables with a single join attribute.

The phrase "correlated sampling" has also been used as a part of the CS2 algorithm in [12], which is designed to estimate the join size of multiple tables subject to arbitrary predicate filter conditions. Two ways of implementing CS2 are suggested in [12]: either requiring multiple passes through the data or requiring an unpredictably large amount of space to store the samples. This makes CS2 not suitable for the problem we consider.

A good review of sketch-based methods for join size estimation is presented in [10]. That work, however, limited its scope to the case of two tables (or data streams), and no filter conditions were considered. A sketch-based approach for estimating the join size of multiple tables was presented in [3]. Even though, as we show in Section 3.4.3, the presence of dynamically specified filter conditions can be modeled as addition of special "imaginary" tables, the authors in [3] did not make this connection.

The work in [6] applies the central idea of Bifocal sampling (separately treating high-frequency and low-frequency values) to the sketching domain. The resulting "skimmed sketches" approach can estimate the join size more accurately than the basic techniques described in [10]. However, it still requires knowing frequencies for the most frequent join attribute values (so that they could be "skimmed away" and processed separately).

An approach to approximate query processing using nonstandard multi-dimensional wavelet decomposition has been presented in [2], which can estimate join sizes subject to dynamically specified predicate filter conditions. Unfortunately, this approach cannot be used for streaming data, since access to old data is required for constructing the wavelet synopsis. More importantly, this method fails to create small-space synopses when the data is very sparse in its multi-dimensional representation, since in this case the highest-resolution coefficients have the largest magnitude, and there are almost as many of them as the original data points. Thus, the standard practice of keeping wavelet coefficients with the largest magnitude will result in very large synopses.

3. TWO-TABLE JOINS

3.1 Independent Bernoulli Sampling

3.1.1 Algorithm Description

The simplest join size estimation algorithm is to form independent Bernoulli samples S_1 and S_2 (with sampling probabilities p_1 and p_2) of tables T_1 and T_2 that are being joined, compute the join size J' of the two samples, and then scale it appropriately.

To derive the required scaling factor, let J be the true join size of the two tables. Also, let X_{ij} be a Bernoulli random variable (r.v.) that is equal to 1 if row i from table T_1 and row j from table T_2 are included in the sample and is equal to 0 otherwise. There will be exactly J such r.v.'s for which both rows will have the same join attribute value. Then, $E[J'] = J \cdot p_1 p_2$ because each of the J r.v.'s evaluates to 1 with probability $p_1 p_2$. Therefore, if the final join size estimate is computed as $\hat{J} = J'/p_1 p_2$, then $E[\hat{J}] = E[J'/p_1 p_2] = J$, and thus $1/p_1 p_2$ is the required scaling factor that makes the final estimate unbiased.

3.1.2 Variance Analysis

As was observed in the previous section, the estimate $\hat{J}$ can be viewed as a sum of J Bernoulli r.v.'s for which $P(1) = p_1 p_2$, $P(0) = 1 - p_1 p_2$, with each r.v. being scaled by $1/p_1 p_2$. Unfortunately, the variance of $\hat{J}$ is not equal to the sum of the variances of the individual Bernoulli r.v.'s because many of them are dependent. For example, X_{11} and X_{21} are dependent (if row 1 from table T_2 is not included, then both of these r.v.'s must be 0).

Therefore, in order to compute the variance of $\hat{J}$, we need to use the fact that the variance of the sum of random variables is equal to the sum of their variances plus the sum of all ordered pairwise covariances (i.e., counting both $Cov(X, Y)$ and $Cov(Y, X)$). Since $Cov(X, Y) = E[XY] - E[X]E[Y]$, it follows that $Cov(X_{ij}, Y_{kj}) = p_1^2 p_2 - (p_1 p_2)^2$ and $Cov(X_{ij}, Y_{ik}) = p_1 p_2^2 - (p_1 p_2)^2$. Let $F_i(v)$ be the frequency with which the join attribute value v appears in table T_i. For any r.v. X_{ij} that corresponds to a pair of rows with a common join attribute v, there are $F_1(v) - 1$ other r.v.'s of the form X_{kj} and $F_2(v) - 1$ other r.v.'s of the form X_{ik} that refer to rows that both have join attribute value equal to v. The sum of covariances of all such r.v.'s with X_{ij} is equal to

$$(F_1(v) - 1)(p_1^2 p_2 - (p_1 p_2)^2) + (F_2(v) - 1)(p_1 p_2^2 - (p_1 p_2)^2),$$

and since there are $F_1(v)F_2(v)$ r.v.'s that correspond to different pairs of rows with a common join attribute value v, and the variance of each such r.v. is $p_1 p_2(1 - p_1 p_2)$, we have:

$$\begin{aligned} Var(J') = \sum_v F_1(v)F_2(v)[p_1 p_2(1 - p_1 p_2) \\ + (F_1(v) - 1)(p_1^2 p_2 - p_1^2 p_2^2) \\ + (F_2(v) - 1)(p_1 p_2^2 - p_1^2 p_2^2)] \end{aligned}$$

where the sum is taken over all join attribute values that occur in both tables. Since $\hat{J} = J'/p_1 p_2$, it follows that $Var(\hat{J}) = \frac{Var(J')}{(p_1 p_2)^2}$, and thus we arrive at:

$$\begin{aligned} Var(\hat{J}) = \sum_v F_1(v)F_2(v) \left[\left(\frac{1}{p_1 p_2} - 1\right)\right. \\ \left. + \ (F_1(v) - 1)\left(\frac{1}{p_2} - 1\right) + (F_2(v) - 1)\left(\frac{1}{p_1} - 1\right)\right] \end{aligned} \quad (1)$$

3.1.3 Extension to dynamic filter conditions

If a filter condition c_i is specified for table T_i after the samples have been created, then the join size of T_1 and T_2 subject to the filter conditions can be estimated by first computing the join size J' of the samples S_1 and S_2 by joining only those rows from the samples that satisfy the corresponding filter conditions and then forming the final join size estimate as $\hat{J} = J'/p_1p_2$. Exactly the same variance analysis as presented above still works for this case, provided that $F_i(v)$ is replaced by $F_i^{(c_i)}(v)$, which denotes the number of rows in table T_i that have a value v for the join attribute and that satisfy the predicate filter condition c_i (which can possibly be specified over other attributes in table T_i). This implies that the variance of Bernoulli sampling decreases as the filter condition becomes more selective, since $F_i^{(c_i)}(v)$ can only decrease in this case.

3.1.4 Discussion

The drawback of the Bernoulli Sampling approach is that the probability of an infrequent join attribute value being included in both samples is very small when individual sampling probabilities p_i are small. Thus, if such infrequent values dominate the two tables, then the variance of the join size estimate will be very high. This intuitive argument is made formal by equation (1) above, which shows that for small $F_1(v)$ and $F_2(v)$, the variance of the join size estimate is dominated by $1/p_1p_2$.

3.2 End-Biased Sampling

3.2.1 Algorithm Description

The End-Biased Sampling algorithm [4] addresses the shortcoming of the Bernoulli Sampling algorithm discussed in Section 3.1.4. This algorithm has a tunable parameter K_i for each table T_i. This parameter controls the trade-off between estimation accuracy and the sample size, and should be set through manual experimentation, according to the authors. The parameter K_i is used to compute the sampling probability p_v for each join attribute value v, which is given by $p_v = F_i(v)/K_i$. Values for which $F_i(v) > K_i$ are included in the sample with probability 1. In order to better account for infrequent join attribute values appearing in both tables, the sampling process is performed in a coordinated fashion between the two tables. This is achieved by selecting a hash function $h()$, which maps the domain of the join attribute uniformly into the range [0,1]. Then, a row with a join attribute value v is included into the sample only if $h(v) \le p_v$.

The End-Biased Sampling algorithm estimates the join size as $\hat{J} = \sum_v c_v$, where for each join attribute value v that occurs in both tables:

$$c_v = \begin{cases} F_1(v)F_2(v) & \text{if } F_1(v) \ge K_1 \text{ and } F_2(v) \ge K_2 \\ K_1F_2(v) & \text{if } F_1(v) < K_1 \text{ and } F_2(v) \ge K_2 \\ F_1(v)K_2 & \text{if } F_1(v) \ge K_1 \text{ and } F_2(v) < K_2 \\ F_1(v)F_2(v) \cdot \max(\frac{K_1}{F_1(v)}, \frac{K_2}{F_2(v)}) & \text{if } F_1(v) < K_1 \text{ and } F_2(v) < K_2 \end{cases}$$

3.2.2 Variance Analysis

It is proven in [4] that the join size estimate $\hat{J}$ computed using the above method is unbiased. It is also shown that

$$Var(\hat{J}) = \sum_v \Delta_v \qquad (2)$$

where

$$\Delta_v = \\ = \begin{cases} 0 & \text{if } F_1(v) \ge K_1 \text{ and } F_2(v) \ge K_2 \\ \left(\frac{K_1}{F_1(v)} - 1\right) F_1^2(v)F_2^2(v) & \text{if } F_1(v) < K_1 \text{ and } F_2(v) \ge K_2 \\ \left(\frac{K_2}{F_2(v)} - 1\right) F_1^2(v)F_2^2(v) & \text{if } F_1(v) \ge K_1 \text{ and } F_2(v) < K_2 \\ (\max\left(\frac{K_1}{F_1(v)}, \frac{K_2}{F_2(v)}\right) - 1)F_1^2(v)F_2^2(v) & \text{if } F_1(v) < K_1 \text{ and } F_2(v) < K_2 \end{cases}$$

3.2.3 Extension to dynamic filter conditions

The End-Biased Sampling can be extended to handle the case when a filter condition c_i is specified for the table T_i after the samples have been created. In this case, the following value of c_v would be used when computing the join size estimate $\hat{J} = \sum_v c_v$:

$$c_v = \\ = \begin{cases} F_1^{c_1}(v)F_2^{c_2}(v) & \text{if } F_1(v) \ge K_1 \text{ and } F_2(v) \ge K_2 \\ K_1F_2^{c_2}(v) & \text{if } F_1(v) < K_1 \text{ and } F_2(v) \ge K_2 \\ F_1^{c_1}(v)K_2 & \text{if } F_1(v) \ge K_1 \text{ and } F_2(v) < K_2 \\ F_1^{c_1}(v)F_2^{c_2}(v) \cdot \max\left(\frac{K_1}{F_1(v)}, \frac{K_2}{F_2(v)}\right) & \text{if } F_1(v) < K_1 \text{ and } F_2(v) < K_2 \end{cases}$$

3.2.4 Discussion

The drawback of the End-Biased Sampling algorithm is that its computation of the join size estimate requires the prior knowledge of the frequencies of all join attribute values, making this method unsuitable for the streaming context. Also, it is not possible to set the parameter K_i ahead of time so as limit the sample size to a certain value, which is often required in real-world contexts.

3.3 Correlated Sampling

3.3.1 Algorithm Description

In this section we present a novel Correlated Sampling algorithm, which addresses the shortcomings of End-Biased Sampling. We first present this algorithm for the simple case of two tables and a single equijoin condition between them: $u_{11} = u_{21}$, where the first attribute in table T_1 is supposed to be equal to the first attribute in table T_2. Then, in Section 4, we present an extension of this algorithm to the case of multiple tables and arbitrary join conditions.

Let n_1 be the desired sample size for table T_1 and n_2 be the desired sample size for table T_2. Such sample sizes can be achieved, in expectation, if each row from table T_i is selected with probability $p_i = n_i/|T_i|$, where $|T_i|$ denotes the size of table T_i. The selection of rows is performed by first selecting a hash function $h()$, which maps the domain of the join attribute uniformly into the range [0,1]. A row r in table T_i in which the join attribute u_{i1} takes the value v is then included in the sample S_i if $h(v) < p_i$. Alternatively, this process can be viewed as generating for each attribute value v a uniform random number between 0 and 1, with the seed of the random generator being set to v. If the generated random number is less than p_i, then row *r* is included in the sample S_i.

Let $p_{min} = \min(p_1, p_2)$. The Correlated Sampling algorithm first computes the join size J' of samples S_1 and S_2 and then divides the result by p_{min} in order to arrive at the final estimate $\hat{J} = J'/p_{\min}$ of the join size of T_1 and T_2.

To prove that $\hat{J}$ is an unbiased estimate of the join size, notice that rows where the join attribute (u_{i1} or u_{i2}) is equal to v will appear in both samples if and only if $h(v) < p_{min}$. Viewing $h(v)$ as a uniform random variable (as was previously suggested), this will happen with probability p_{min}. Then, a value v that appears in both tables is expected to contribute $p_{min}F_1(v)F_2(v)$ to the expected join size computed over the samples S_1 and S_2. Overall, the expected size of this join is equal to $\sum_v p_{min}F_1(v)F_2(v)$. Thus, if the join size of samples S_1 and S_2 is divided by p_{min}, then in expectation the result will be equal to $\sum_v F_1(v)F_2(v)$, which is exactly the true join size of T_1 and T_2.

3.3.2 Variance Analysis

The join size estimate computed above is a summation over all join attribute values v of Bernoulli random variables (for which $P(1) = p_{min}$, $P(0) = 1 - p_{min}$), each of which is scaled by $F_1(v)F_2(v)/\,p_{min}$. The variance of each such random variable is equal to $p_{min}(1 - p_{min})(F_1(v)F_2(v)/\,p_{min})^2$ and they are independent. Thus, the variance of the final estimate $\hat{J}$ is given by

$$Var(\hat{J}) = \left(\frac{1}{p_{\min}} - 1\right)\sum_v F_1^2(v)F_2^2(v) \qquad (3)$$

where the sum is taken over the values v that occur in both tables.

3.3.3 Extension to dynamic specified filter conditions

Let's say that we are only interested in joining those rows in table T_i that satisfy a predicate filter condition c_i. As before, each join attribute value v will appear in both samples with probability p_{min}, and if selected, it will contribute $F_1^{(c_1)}(v)F_2^{(c_2)}(v)$ to the join size computed over samples S_1 and S_2. Thus, the expected contribution of v to this join size is $p_{min}F_1^{(c_1)}(v)F_2^{(c_2)}(v)$, and hence the expected overall join size is $\sum_v p_{min}F_1^{(c_1)}(v)F_2^{(c_2)}(v)$, which when divided by p_{min} gives the desired join size subject to the specified predicate filter condition.

The variance of Correlated Sampling subject to a filter condition c_i for table T_i can be derived in exactly the same way as before, where $F_i(v)$ is replaced by $F_i^{(c_i)}(v)$. Thus, the variance of the estimate is reduced as the filter condition becomes more strict.

3.3.4 Discussion

The Correlated Sampling algorithm presented above can be summarized in the following steps:

1. Choose randomly a hash function $h()$ from a strongly universal family of hash functions that map the domain of the join attribute into [0,1). There are many well-known algorithms for constructing such a hash function – see [9] and references therein. The most well-known and robust one is $h(v) = \big((av + b) \bmod p\big)/\mathrm{p}$, where $a, b \in [1, \mathrm{p})$ are randomly chosen integers and p is a large prime number.

2. Scan the table T_1, observe the value v taken by the join attribute in each row, and then select that row into the sample S_1 if $h(v) < p$ (where p can be set to 0.01 or something smaller if a smaller sample is desired). Do the same for table T_2, constructing the sample S_2.

3. Estimate the join size of T_1 and T_2 as $\hat{J} = (S_1 \bowtie S_2)/p$

The Correlated Sampling algorithm addresses the major problem that independent Bernoulli sampling has with join attribute values that repeat infrequently in both tables. Such values are sampled in a correlated fashion and their contribution to the variance of the estimator scales as $\frac{1}{\min(p_1, p_2)}$, which is much smaller than $\frac{1}{p_1 p_2}$ scaling factor of Bernoulli sampling if both p_1 and p_2 are small.

The Correlated Sampling algorithm also addresses all the shortcomings of the End-Biased Sampling algorithm presented in Section 3.2: it does not require the prior knowledge of the frequencies of join attribute values and it does not have any parameters that need to be set through manual experimentation. This makes Correlated Sampling suitable for the streaming context or for processing very large tables, where only single-pass processing is feasible.

Both End-Biased and Correlated Sampling reduce the join size estimation variance relative to Bernoulli sampling when the tables are dominated by infrequent values. However, a careful comparison of the variances (which we perform in Section 5.4) shows that join attribute values that occur frequently in both tables make a larger contribution to the variance of Correlated Sampling than to that of Bernoulli Sampling (especially if the sampling probabilities are large). This suggests that the Correlated Sampling algorithm can be improved if such values are detected ahead of time, their frequencies are accurately estimated, and their contribution to the join size is computed directly using the estimated frequencies, as is done in [4] and [5].

The best algorithm we found so far (according to our separate tests not included in this paper due to space constraints) for detecting the most frequent values in a data stream is the Filtered Space-Saving (FSS) algorithm [8]. This algorithm makes a single pass over the data (performing O(1) operations for each tuple) and can thus be ran in parallel with the sampling phase of Correlated Sampling. The FSS algorithm creates a list of suggested most frequent values, and for each value it also gives its estimated frequency and the maximum estimation error. The values that appear in the candidate lists for both T_1 and T_2 and have the maximum percentage frequency estimation error less than a certain threshold can be considered to be "frequent values" and their contribution to the join size of T_1 and T_2 can be computed by a direct multiplication of their estimated frequencies. By choosing the percentage error threshold to be small enough, one can incur a small bias in the join size estimate (since FSS, by design, overestimates the frequencies) but not incur the large variance, since the most frequent values will be absent from the summation in equation (3). The remaining "non-frequent" values in the samples S_1 and S_2 can then be joined together, and if the resulting join size is divided by p_{min}, then one will obtain an unbiased contribution of all non-frequent values to the join size of T_1 and T_2 (because the expected contribution to the join size of S_1 and S_2 of each "non-frequent value" v appearing in both tables is $p_{min}F_1(v)F_2(v)$).

3.4 A Novel Sketch-based Approach

3.4.1 Algorithm Description

The first sketch-based approach to join size estimation was presented in [1], and the sketch structure proposed in that paper later received an acronym AGMS sketch, composed of the first letters of the last names of the authors of [1]. Consider a data stream $A = \{v_1, v_3, v_2, v_3, v_4, v_5, v_1, \dots\}$, which might be the output of a dynamically computed complex SQL query, which then needs to be joined with another similar stream as a part of a yet more complex query. Alternatively, A can represent the sequence of join attribute values of sequential rows read from a large database table, which is so large that it can only be processed using a one-pass method, as a stream. The atomic AGMS sketch corresponding to the stream A is constructed as

$$sk(A) = \xi_1 + \xi_3 + \xi_2 + \xi_3 + \xi_4 + \xi_5 + \xi_1 + \cdots,$$

where ξ_i are 2-wise independent random variables that are equal to -1 or +1 with equal probability (for which $E[\xi_i \xi_j] = 0$ if $i \neq j$). That is, whenever the value v_i is observed in the stream, the value of ξ_i is added to the AGMS sketch.

We will now show that the product $\hat{J}$ of atomic AGMS sketches $sk(T_1)$ and $sk(T_2)$ for two tables T_1 and T_2 is an unbiased estimate of the true join size of T_1 and T_2: $E[\hat{J}] = E[sk(T_1) \cdot sk(T_2)] = |T_1 \bowtie T_2|$. The expected contribution to $E[\hat{J}]$ of each row with an attribute value v in table T_1 is:

$$E\left[\sum_{i \in T_2} \xi_v \xi_i\right] = E\left[\sum_{i=v} \xi_v \xi_i\right] + E\left[\sum_{i \neq v} \xi_v \xi_i\right] = E\left[\sum_{i=v} \xi_v \xi_i\right] = F_2(v),$$

where we have used the fact that $E[\xi_v \xi_i] = 0$ if $i \neq v$, while $E[\xi_v \xi_v] = 1$. Since there are $F_1(v)$ rows in table T_1 that have join attribute value equal to v, the total contribution to $E[\hat{J}]$ of each value v is $F_1(v)F_2(v)$, which implies that $E[\hat{J}] = \sum_v F_1(v)F_2(v) = |A \bowtie B|$.

3.4.2 *Variance Analysis*

Since $\hat{J} = sk(T_1) \cdot sk(T_2)$ is an unbiased estimate of the true join size, the variance of $\hat{J}$ can be reduced if k independent copies of this estimator are averaged. It was shown in [1] that in this case:

$$Var(\hat{J}) = \frac{1}{k}\left(\left(\sum_v F_1^2(v)\right)\left(\sum_v F_2^2(v)\right) + \left(\sum_v F_1(v)F_2(v)\right)^2 - \sum_v F_1^2(v)F_2^2(v)\right) \tag{4}$$

where the above sums are taken over all attribute values that occur in at least one of the tables. We will use this convention in the next section as well.

3.4.3 *Extension to dynamic filter conditions*

While the sketch-based approach for join size estimation described above has been widely cited in the literature, we have not seen any attempts to extend it to the case of dynamically specified predicate filters. We present such an extension below.

The basic idea is to view each filter condition on attribute j in table i (which we denote by u_{ij}) as an "imaginary" table containing a single column of all attribute values for u_{ij} that satisfy this condition (not all of these values might appear in table i). When a query with a particular predicate filter condition is submitted, an imaginary table is constructed and its sketch is computed, which when multiplied with the sketches of original tables (constructed offline based on the expected join graph) gives an unbiased join size estimate.

For illustrative purposes, we present a simple case where the predicate filter condition c_i is specified over the second attribute u_{i2} in table T_i and is of the form $u_{i2} \in R_i$, where R_i is some range (or more generally, is a union of disjoint ranges). Let $sk_{u_{i2}}(R_i)$ be AGMS sketch of R_i (i.e., of the table containing a single column of all possible values of $u_{i2} \in R_i$) constructed with a family ψ^i of random variables (of the type described in Section 3.4.1) by adding ψ_y^i to the sketch whenever a row r from T_i is observed for which $u_{i2} = y$. Let $sk_{u_{i1}u_{i2}}(T_i)$ denote a "dual" AGMS sketch of T_i constructed by adding $\xi_v \psi_y^i$ to the sketch whenever a row r from T_i is observed for which $u_{i1} = v$ and $u_{i2} = y$, where ξ is a family of random variables that corresponds to the join attribute. Then, using a similar analysis to the one conducted in Section 3.4.1, it is not difficult to show that

$$E\left[sk_{u_{12}}(R_1) \cdot sk_{u_{11}u_{12}}(T_1) \cdot sk_{u_{21}u_{22}}(T_2) \cdot sk_{u_{22}}(R_2)\right]$$

gives us precisely the join size of T_1 and T_2 subject to the filter conditions described above. Informally, this is so because $sk_{u_{i2}}(R_i) \cdot sk_{u_{i1}u_{i2}}(T_i)$ is a sum of products of the form $\xi_v \psi_y^i \psi_z^i$, for which $E[\xi_v \psi_y^i \psi_z^i] = 1$ only if $y = z$, and is equal to 0 otherwise. For any particular join attribute value v, there will be $F_i^{(c_i)}(v)$ products that involve ξ_v and that have an expected value of 1. Thus, each join attribute value v will contribute, in expectation, $F_1^{c_1}(v)\, F_2^{c_2}(v)$ to the join size estimate, which when summed up over all join attribute values gives us precisely the join size of T_1 and T_2 subject to the filter conditions described above.

Following a similar procedure to the one used in the proof of Lemma 4.4 in [1], it is possible to show that:

$$\begin{aligned} &Var\left[sk_{u_{12}}(R_1) \cdot sk_{u_{11}u_{12}}(T_1) \cdot sk_{u_{21}u_{22}}(T_2) \cdot sk_{u_{22}}(R_2)\right] \\ &\quad \leq 2 \sum_v \left(|R_1| \sum_{v_{12}} F_1^2(v, v_{12}) + \left(F_1^{(c_1)}(v)\right)^2\right) \\ &\quad \times \sum_v \left(|R_2| \sum_{v_{22}} F_2^2(v, v_{22}) + \left(F_2^{(c_2)}(v)\right)^2\right) \end{aligned} \tag{5}$$

where $|R_i|$ is the number of rows in T_i that satisfy condition c_i, $F_i(v, v_{i2})$ is the number of rows in table T_i for which the value of the join attribute is equal to v and the value of the other attribute u_{i2} (over which the filter condition is specified) is equal to v_{i2}, and $F_i^{(c_i)}(v)$ is the number of rows in table T_i with the value of the join attribute being equal to v that satisfy the filter condition $c_i: u_{i2} \in R_i$.

It is instructive to compare the variance bound in equation (5) with the one in equation (4), when a single pair of atomic sketches is used (i.e.: $k = 1$). We can bound the variance in equation (4) as follows:

$$Var(\hat{J}) \leq \left(\sum_v F_1^2(v)\right)\left(\sum_v F_2^2(v)\right) + \left(\sum_v F_1(v)F_2(v)\right)^2.$$

The second term on the right is less than or equal to the first term by Cauchy-Schwarz, and thus:

$$Var(\hat{J}) \leq 2\left(\sum_v F_1^2(v)\right)\left(\sum_v F_2^2(v)\right). \tag{6}$$

Comparing equation (6) with equation (5) we see that introduction of predicate filter conditions on tables T_1 and T_2 resulted in $F_i^2(v)$ being changed into $|R_i|\sum_{v_{i2}} F_i^2(v, v_{i2}) + \left(F_i^{(c_i)}(v)\right)^2$. Let's consider two extreme cases:

1. Almost all values of the attribute u_{i2} pass the filter condition c_i
2. Only a single value $v_{i2}^{c_i}$ of the attribute u_{i2} passes the filter condition c_i.

The Cauchy-Schwarz inequality implies that $F_i^2(v) \leq |u_{i2}|\sum_{v_{i2}} F_i^2(v, v_{i2})$, where $|u_{i2}|$ is the number of different values that the attribute u_{i2} can take in table T_i. Therefore, in case 1, $F_i^{(c_i)}(v) \approx F_i(v)$ and $|R_i|\sum_{v_{i2}} F_i^2(v, v_{i2}) \approx |u_{i2}|\sum_{v_{i2}} F_i^2(v, v_{i2}) \geq F_i^2(v)$, implying that $Var(\hat{J})$ will increase at least by a factor of 4. If very few different values of u_{i2} are present for each distinct value of the join attribute, then $Var(\hat{J})$ can increase by *much* more than a factor of 4 because the term $|R_i|$ will dominate. This runs in a stark contrast to the sampling methods presented in Section 3.1 – 3.3, for which the introduction of a predicate filter condition *reduces* the variance of the join size estimate.

In case 2, we need to compare $F_i^2(v)$ vs. $\sum_{v_{i2}} F_i^2(v, v_{i2}) + F_i^2\left(v, v_{i2}^{c_i}\right)$. If many different values of u_{i2} are present for each distinct value of the join attribute, then the former will be larger than the latter. However, if only a single value of u_{i2} is present for each distinct value of the join attribute, then $\sum_{v_{i2}} F_i^2(v, v_{i2}) = F_i^2\left(v, v_{i2}^{c_i}\right) = F_i^2(v)$, implying that $Var(\hat{J})$ can increase by a factor of 4.

Join size estimation, in general, involves estimating the selectivity of a natural join from a cross product of all tables involved in the join. Therefore, it is not surprising that the absolute value of the estimator variance will increase proportionally to the size of this cross product (as shown in [1]). In the case of sketch estimators, the introduction of filter predicates is equivalent to introducing a new table into the join. This will increase the cross product by a factor of the size of the filtering range, which can potentially increase the variance by a very large factor.

3.4.4 *Discussion*

The sketch-based approach presented above is well-suited for the streaming context, since it works directly on the data stream (it simply increments a single counter per atomic sketch of a table T_i). Once a sketch of the data streams of interest is computed over the attributes of interest, this approach allows for estimating the join sizes over different join attributes and different predicate conditions (specified independently for each attribute) without having to access the data again.

In order to reduce the variance of the join size estimate using this approach, many atomic sketches are required per table, as was explained in Section 3.4.2. It may seem that the space requirement of such an approach with *N* atomic sketches per table (which requires storage of *N* floating point numbers) is the same as the one for a sampling approach that uses a sample of size *N* (which requires the storage of *N* sampled join attribute values). This is not the case, however, because in order to dynamically generate the 2-wise independent random variables ξ_i for each atomic sketch, one would need to store the seeds for such a generator. The required seed space is not large – it is logarithmic in the number of distinct values that we expect the join attribute can have in the stream [11]. If predicate filter conditions are expected on attributes $u_{i2}, \dots u_{im}$, then a sampling-based approach would need to store $m - 1$ additional values per sampling point (the values of the attributes $u_{i2}, \dots u_{im}$), while a sketch-based approach would need to store $m - 1$ additional atomic sketches, corresponding to the $m - 1$ conditional ranges for the attributes $u_{i2}, \dots u_{im}$.

The main drawback of the sketch-based join size estimator presented in this section is its very large variance in the case when filter condition with a large range $|R_i|$ is specified or when the number of different values of u_{i2} that are present for each distinct value of the join attribute is small. Another drawback is a large increase in the time required to compute the sketch of the conditional range R_i if $|R_i|$ is large. If R_i is a contiguous range, then it can be sketched quickly using range-summable random variables [11]. However, if the filter condition c_i is complex and requires evaluation of each possible value of u_{i2}, then a large sketching time cannot be avoided. Yet another drawback of this sketch-based approach relative to Correlated Sampling is that the latter method can be used with predicated filters over multiple columns, such as $u_{ik} < u_{jl}$, while the sketch-based approach cannot handle such predicate conditions.

4. CORRELATED SAMPLING FOR MULTIPLE TABLES

In order to analyze Correlated Sampling for the case of multiple tables and complex join conditions, we need to introduce the notion of equivalence classes. For two tables T_i and T_j and join attributes u_{ik}, u_{jl}, we denote $u_{ik} \sim u_{jl}$ whenever the join condition $u_{ik} = u_{jl}$ is present. For any attribute u_{ik}, we denote by $\psi(u_{ik})$ its equivalence class under the relation $\sim$, which includes all other attributes that have to be equal to u_{ik} under the considered join query. We will assume that the complex query has *K* equivalence classes: $\Psi_1, \dots, \Psi_K$.

For example, consider tables T_1, T_2 and T_3 with join attributes u_{11} and u_{12} in T_1, u_{21} in T_2, and u_{31} and u_{32} in T_3. Then, the set of join conditions $\{u_{11} = u_{31}, u_{12} = u_{32}, u_{11} = u_{21}\}$ implies two equivalence classes: Ψ_1 in which $u_{11} \sim u_{21} \sim u_{31}$ and Ψ_2 in which $u_{12} \sim u_{32}$. Pictorially, these equivalence classes are strings of connected edges on the join graph.

For each equivalence class Ψ_k, we define a uniform hash function h_k from the attribute domain to [0,1]. We require, for $h_k \neq h_j$, the hash values of these two functions be completely independent, even on correlated inputs. We also define the function $\phi: u_{ij} \mapsto k$ to map an attribute to its equivalence class index k (that is, the value of k so that $\psi(u_{ij}) = \Psi_k$).

Let U_i be the set of all join attributes for table T_i and let $|U_i|$ denote the size of this set (the number of join attributes in table

T_i). For any join attribute $u_{ij} \in U_i$ and the value v_{ij} it takes, consider the following event:

$$h_{\phi(u_{ij})}(v_{ij}) < (p_i)^{\frac{1}{|U_i|}} \tag{7}$$

We will refer to this as the inclusion condition for u_{ij}. A given row r will be included in the sample S_i if the inclusion condition is satisfied for all join attributes u_{ij} in that row. For a truly random hash function, the probability that each one of the inclusion conditions for that row is satisfied is simply $(p_i)^{\frac{1}{|U_i|}}$. Since the chosen hash functions generate independent uniform random variables (regardless of correlations among the inputs), the event that any given attribute in row r satisfies its inclusion condition is independent of all the other attributes satisfying the inclusion condition. Mathematically, this can be expressed as:

$$P\left(h_{\phi(u_{i1})}(v_{i1}) < (p_i)^{\frac{1}{|U_i|}} \wedge \dots \wedge h_{\phi\left(u_{i|U_i|}\right)}\left(v_{i|U_i|}\right) < (p_i)^{\frac{1}{|U_i|}}\right) =$$

$$= P\left(h_{\phi(u_{i1})}(v_{i1}) < (p_i)^{\frac{1}{|U_i|}}\right) \dots P\left(h_{\phi\left(u_{i|U_i|}\right)}\left(v_{i|U_i|}\right) < (p_i)^{\frac{1}{|U_i|}}\right)$$

$$= \left((p_i)^{\frac{1}{|U_i|}}\right)^{|U_i|} = p_i$$

This allows us to ignore correlations among the attributes of each table when performing sampling and when determining the sampling probability p_i.

Now assume that we have a complex join query that includes equijoin conditions for tables $T_1, \dots, T_N$. Consider a particular row r_{out} from the output of that join and let's break it up into rows $r_1, \dots, r_N$ from the individual tables that were joined in order to create that output row. Recall that the probability of a row r_i being included into the sample S_i is p_i and let's compute P_{inc}, the probability that all rows $r_1, \dots, r_N$ get included into the corresponding samples, which is also the probability that the row r_{out} appears in the output of the join. Using the definition of inclusion condition in equation (7) and the notion of equivalence classes we introduced earlier, this probability can be expressed as:

$$P_{\text{inc}} = P\left(\bigwedge_{i=1}^{N} \bigwedge_{j=1}^{|U_i|} h_{\phi(u_{ij})}(v_{ij}) < (p_i)^{\frac{1}{|U_i|}}\right)$$

$$= P\left(\bigwedge_{k=1}^{K} \bigwedge_{u_{ij} \in \Psi_k} h_{\phi(u_{ij})}(v_{ij}) < (p_i)^{\frac{1}{|U_i|}}\right) \tag{8}$$

where we have used the fact that each attribute appears in exactly one equivalence class. Recall the two following facts, which are true by construction:

– For every attribute $u_{ij} \in \Psi_k$, we know $\phi(u_{ij}) = k$

– Any two attributes u_{ij} and u_{kl} in Ψ_k satisfy the join condition $u_{ij} = u_{kl}$. In particular, this means that $v_{ij} = v_{kl}$.

This allows us to define a single attribute value, call it α_k, for the attributes in Ψ_k, so that for any attribute $u_{ij} \in \Psi_k$ it must be the case that $v_{ij} = \alpha_k$. With these two notes, we can re-write the expression in (8) as:

$$P_{\text{inc}} = P\left(\bigwedge_{k=1}^{K} \bigwedge_{u_{ij} \in \Psi_k} h_{\phi(u_{ij})}(v_{ij}) < (p_i)^{\frac{1}{|U_i|}}\right)$$

$$= P\left(\bigwedge_{k=1}^{K} \bigwedge_{u_{ij} \in \Psi_k} h_k(\alpha_k) < (p_i)^{\frac{1}{|U_i|}}\right)$$

$$= P\left(\bigwedge_{k=1}^{K} h_k(\alpha_k) < \min_{i \in \Psi_k}\left((p_i)^{\frac{1}{|U_i|}}\right)\right) \tag{9}$$

where we have used the fact that $h_k(\alpha_k) < (p_i)^{\frac{1}{|U_i|}}$ for all $u_{ij} \in \Psi_k$ if and only if this is true for the smallest value of $(p_i)^{\frac{1}{|U_i|}}$. For the sake of simplicity, we have also used a slight abuse of notation in equation (9) above, with $\min_{i \in \Psi_k}$ denoting a minimum taken over all tables that appear in the equivalence class k.

Noting that the hash functions h_k produce independent uniform random variables, we have:

$$P_{\text{inc}} = P\left(\bigwedge_{k=1}^{K} h_k(\alpha_k) < \min_{i \in \Psi_k}\left((p_i)^{\frac{1}{|U_i|}}\right)\right)$$

$$= \prod_{k=1}^{K} P\left(h_k(\alpha_k) < \min_{i \in \Psi_k}\left((p_i)^{\frac{1}{|U_i|}}\right)\right) = \prod_{k=1}^{K} \min_{i \in \Psi_k}\left((p_i)^{\frac{1}{|U_i|}}\right) \tag{10}$$

This is a quantity which can be readily computed. It is the same for all rows that appear in the output of the join, implying a very simple way of estimating the cardinality of the join: compute the join size over samples $S_1, \dots, S_N$ and divide it by the inclusion probability P_{inc} computed above.

In order to prove that the above estimation procedure is unbiased, denote by $\vec{v}$ the vector of join attributes that define a particular row in the output of the join. The true join size J can now be expressed as: $J = \sum_{\vec{v}} F(\vec{v})$, where $F(\vec{v})$ is the number of rows in the output of the join that have a combination of join attributes specified by $\vec{v}$. Let $I(\vec{v})$ be the indicator variable which is equal to 1 if and only if the row with attributes specified by $\vec{v}$ is included into the join output. Let J' be the join size of the samples, so that:

$$E[J'] = E\left[\sum_{\vec{v}} F(\vec{v}) I(\vec{v})\right] = \sum_{\vec{v}} E[I(v)] F(\vec{v})$$

$$= P_{\text{inc}} \sum_{\vec{v}} F(\vec{v}) = P_{\text{inc}} \cdot J \tag{11}$$

Therefore, $J = E[J']/P_{\text{inc}}$, as was previously claimed. For the variance of this estimator, we use a similar technique:

$$\text{Var}[J'] = \sum_{\vec{v}} \text{Var}[I(\vec{v})] F^2(\vec{v}) = P_{\text{inc}}(1 - P_{\text{inc}}) \sum_{\vec{v}} F^2(\vec{v}) \tag{12}$$

Since the final estimate of the join size is $\hat{J} = J'/P_{\text{inc}}$, we have:

$$\text{Var}(\hat{J}) = \left(\frac{1}{P_{\text{inc}}} - 1\right) \sum_{\vec{v}} F^2(\vec{v}) \tag{13}$$

which agrees with the result that was derived in Section 3.3 for the case of two-table joins. If a predicate filter condition c_i is used for table T_i, then equations (11), (12) and (13) still hold with $F_i^{c_i}(v)$ used instead of $F_i(v)$.

4.1 Choosing Sampling Probabilities

The choice of sampling probabilities is a very practical concern, since they determine the variance of the join size estimate through equation (13). Given a particular join graph and sampling probability for any one table, one can use equation (10) in order to determine the optimal sampling probability for all other tables.

Assume that p_1 is given and that tables T_1 and T_2 both appear in the same equivalence class Ψ. Consider the possibility that $p_1^{\frac{1}{|U_1|}} > p_2^{\frac{1}{|U_2|}}$: in this case, the multiplicative factor in P_{inc} due to Ψ will be at most $p_2^{\frac{1}{|U_2|}}$. In other words, one could have taken a smaller sampling probability p_1 and still have obtained the same accuracy. A symmetric argument can be applied to the case of $p_1^{\frac{1}{|U_1|}} < p_2^{\frac{1}{|U_2|}}$, implying that the most efficient sampling probability (i.e.: the choice that does not "waste" any of the resulting sample) occurs when $p_2 = p_1^{\frac{|U_2|}{|U_1|}}$. Note that this is true for any other sampling probability p_i where $i \in \Psi$. This reasoning defines a method for determining the sampling probability for all the tables in the join graph, since every table is connected to another via some join condition (i.e.: equivalence class). Starting with the sampling probability for a single table in the join graph, one can traverse the join graph in the manner described, determining the sampling probabilities for all the tables in the join graph from a single parameter.

5. EXPERIMENTAL COMPARISON

5.1 Dataset Description

In order to confirm the theoretical analysis performed in Sections 3 and 4, we have implemented Bernoulli, Correlated and End-Biased sampling, as well as AGMS sketches. These techniques were then used to estimate join sizes on a schema containing one fact table (a collection of sales records) and two dimension tables (one with customer data and the other with product data). The fact (sales) table contained 92052 entries, with 7059 distinct values of customer_id and 72 distinct values of product_id). The most common value of customer_id occurs 41 times in the table fact table, while the most common value of product_id occurs 2911 times. The table of customer data contained 11052 rows, each with a distinct customer_id. The products table contained exactly 72, each with a unique product_id.

Experiments in Sections 5.2 – 5.6 focus on joining the sales table with the customer table along the customer_id column while making different types of modifications to each table. In section 5.7, we present the results from joining all three tables together.

For all of the experiments conducted, the sampling probability shown refers to the sampling probability of the fact (sales) table. For the dimension tables, the technique described in Section 4.1 was used to determine the sampling probabilities. When performing Bernoulli sampling, the same sampling probabilities were used for each one of the tables as in Correlated Sampling. For End-Biased Sampling, threshold values were chosen such that the resulting samples contained the same number of elements as the Bernoulli and Correlated samples. For sketches, the sampling probabilities were first used to compute the number of samples that would have been taken by a sampling method, and this number was then divided by the number of tables in the join graph, resulting in the number of atomic sketches produced for each table. As a result, both sketches and sampling methods used a similar amount of final data based on which the join size estimates were computed.

5.2 Basic Experiments: Two Tables, no Filters

Figure 1 shows frequency histograms of the join size estimates produced by each technique, over multiple trials, when joining the fact (sales) table with the customer data table along the customer_id column without using any predicate filter conditions. A sampling probability of 0.01 was used.

The observed estimator variances (largest for Bernoulli sampling and smallest for sketches) fully correspond with variance formulas derived in Section 3. To see this, notice that since each record in the customer table has a unique customer_id, it follows that $F_2(v) = 1$ for all v in this experiment. In this case, assuming $p_1 = p_2 = p \ll 1$, the variance of Bernoulli sampling is approximately given by

$$Var_{Ber}(\hat{J}) \approx \frac{1}{p^2}\sum_v F_1(v) + \frac{1}{p}\sum_v F_1^2(v)$$

while the variance of Correlated Sampling becomes

$$Var_{Corr}(\hat{J}) = \frac{1}{p}\sum_v F_1^2(v)$$

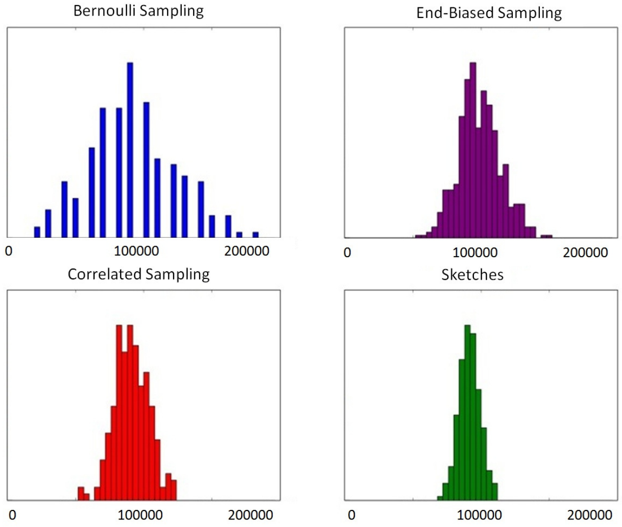

Figure 1. Frequency histograms of join size estimates.

This shows that Bernoulli sampling will always have a larger variance than Correlated Sampling if $F_2(v) = 1$. For End-Biased Sampling (assuming that the sampling probabilities are small enough that all attribute values are sampled probabilistically) the variance becomes:

$$Var_{End}(\hat{J}) = (\mathrm{K}_2 - 1)\sum_v F_1^2(v)$$

If, in one of the tables, all attribute values occur with the same frequency, then in order for the expected sizes of the End-Biased and Correlated samples to be equal, we must have the sampling threshold K equal to $1/p$. If the frequencies of some values are increased and of others are decreased while the total number of rows in a table is kept unchanged, then the expected End-Biased sample size will increase, as more frequent values will be sampled with a larger probability. Thus, in order for the expected sizes of End-Biased and Correlated samples to be the same, we must have $K > 1/p$, and in this case the equations derived above show that the variance of End-Biased sampling will be larger than that of Correlated Sampling. This fact can be observed in Figure 1.

Extending equation (6) to k atomic sketches, we get the following upper bound for the sketch-based estimator when $F_2(v) = 1$:

$$Var(\hat{J}) \leq \frac{2N_2}{k}\left(\sum_{v} F_1^2(v)\right)$$

where N_2 is the number of distinct join attribute values that appear in table T_2. If $2N_2/k < 1/p$, then the variance of the sketch-based estimator will be smaller than that of Correlated Sampling, which is what we observe in Figure 1. An opposite case is shown in Figure 4, which we will describe later.

It is interesting to examine closer the histogram for Bernoulli sampling. Even though the estimate is unbiased, it takes discrete values. Each discrete step corresponds to the simultaneous inclusion of a pair of rows from both tables with the same customer_id. Since the event of sampling a row is not dependent on the value of the join attribute in that row, the contribution to the join size estimate will be $1/p_1p_2$ for every such matching pair of rows included (as was shown in Section 3.1.2). With the sampling probability of 0.01 for both tables, each additional matching pair of rows will increase the join size estimate by $1/0.01^2 = 10000$. This is exactly the size of the smallest discrete steps observed along the x-axis, confirming our theoretical analysis. The similar discrete step behavior was seen in all the experiments performed with Bernoulli sampling.

For Correlated Sampling, contributions to the join size happen at the level of join attribute values that appear in both tables (rather than individual rows, as in Bernoulli Sampling), and each such value contributes $p_{min}F_1(v)F_2(v)$ to the join size. Therefore, unlike in the case of Bernoulli sampling, the join size estimate obtained by Correlated Sampling is not restricted to multiples of any specific value and thus can potentially to be more accurate.

5.3 Dependence on Memory Usage

The next experiment computed the relative errors (defined as observed standard deviation of the estimator divided by the true join size) after 500 trials of each technique as a function of the memory usage (sampling probability). The results are displayed in Figure 2 and follow the theoretical analysis performed in the previous section for the case when $F_2(v) = 1$.

Furthermore, we can observe how each method's variance changes with p:

$$\frac{d\,Var_{Ber}(\hat{J})}{d\,p} \approx -\left(\frac{2}{p^3}\sum_{v} F_1(v) + \frac{1}{p^2}\sum_{v} F_1^2(v)\right)$$

$$\frac{d\,Var_{Corr}(\hat{J})}{d\,p} \approx -\frac{1}{p^2}\sum_{v} F_1^2(v)$$

In particular, this implies:

$$\left|\frac{d\,Var_{Corr}(\hat{J})}{d\,p}\right| < \left|\frac{d\,Var_{Ber}(\hat{J})}{d\,p}\right|$$

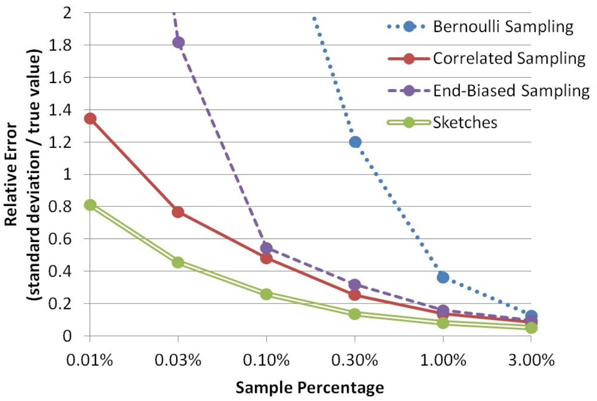

Figure 2. Relative errors for $F_2(v) = 1$.

which shows that the variance for the Bernoulli estimate has a stronger dependence on the sampling probability, especially forsmall values of p. This matches the behavior in Figure 2, which shows a much steeper reduction of the relative error for Bernoulli sampling than for the other techniques as p increases.

5.4 Dependence on Data Replication

To illustrate how each estimator's variance changes depending on the distribution of the data in the dimension tables, the experiments in the previous section were repeated using the following modified versions of the customer table (Table 2):

Version 1: each row in the table was replicated a uniformly random number of times (between 0 and 100).

Version 2: for each customer_id already in the original table, another 15 distinct ones were added that do not appear in the sales table. Thus, this version contained 176832 distinct customer_id's.

The relative errors of the considered techniques for version 1 of the modified customer table are shown in Figure 3.

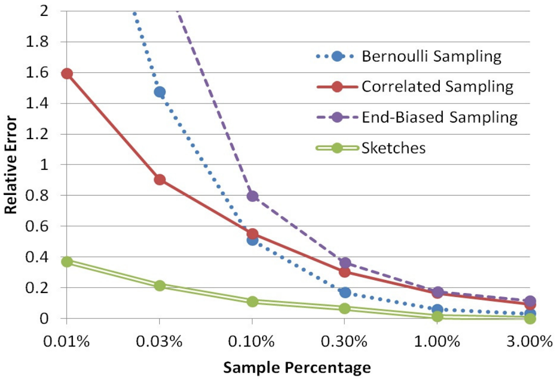

Figure 3. Relative errors when $F_2(v)$ is a uniform random variable between 1 and 100

As before, we observe that the accuracy of Bernoulli sampling has a stronger dependence on the sampling probability than the other techniques. In this case, however, Bernoulli sampling is able to outperform Correlated Sampling for large sampling percentages.

To understand this, note that the variance of Bernoulli sampling given in equation (1) can approximated as follows:

$$Var_{Ber}(\hat{J}) \approx \sum_v \left(\frac{1}{p} F_1^2(v)F_2(v) + \frac{1}{p^2} F_1(v)F_2(v) + \frac{1}{p} F_1(v)F_2^2(v)\right)$$

while the variance of Correlated Sampling as:

$$Var_{Corr}(\hat{J}) \approx \sum_v \frac{1}{p} F_1^2(v)F_2^2(v)$$

As $F_1(v)$ and $F_2(v)$ both start to increase, the term $\frac{1}{p}F_1^2(v)F_2^2(v)$ in the above expression begins to dominate every term in the expression for $Var_{Ber}(\hat{J})$. If this happens for a sufficiently large fraction of join attribute values, the variance of the Correlated Sampling estimate can increase beyond that of Bernoulli sampling, unless p is so small that the term $\frac{1}{p^2}F_1(v)F_2(v)$ in the expression for $Var_{Ber}(\hat{J})$ dominates all other terms. This behavior is captured in Figure 3, which shows that the variance of Bernoulli sampling is smaller than that of Correlated Sampling for the case when $F_2(v)$ is, on average, equal to 50 and $p > 0.001$.

It is also interesting to note that Correlated Sampling outperforms End-Biased sampling in this experiment. To understand this, observe that when the sampling threshold K in End-Biased sampling is large and all values are sampled probabilistically:

$$Var_{End}(\hat{J}) = \left(\max\left(\frac{K_1}{F_1(v)}, \frac{K_2}{F_2(v)}\right) - 1\right) F_1^2(v)F_2^2(v)$$

In the above formula, the term of the form $F_1^2(v)F_2^2(v)$ will always be multiplied by the maximum of $\frac{K_1}{F_1(v)}$ and $\frac{K_2}{F_2(v)}$. Thus, when $F_1(v)$ and $F_2(v)$ vary independently (as is the case in this experiment), this factor will not work efficiently to reduce the variance of the join size estimate. Quite on the contrary: in this case, the multiplicative factor will (on average) be slightly greater than $1/p$ (as was noted in Section 5.2), explaining the poor performance of End-Biased sampling. In fact, this suggests that End-Biased sampling will only perform better than Correlated sampling when both tables being joined have similar frequency distributions $F_i(v)$. We have observed this experimentally: the only situation in which End-Biased sampling outperformed Correlated Sampling for small sampling probabilities was for self-joins of tables with non-constant frequency distributions.

The results of joining the sales table with version 2 of the customer table are presented in Figure 4.

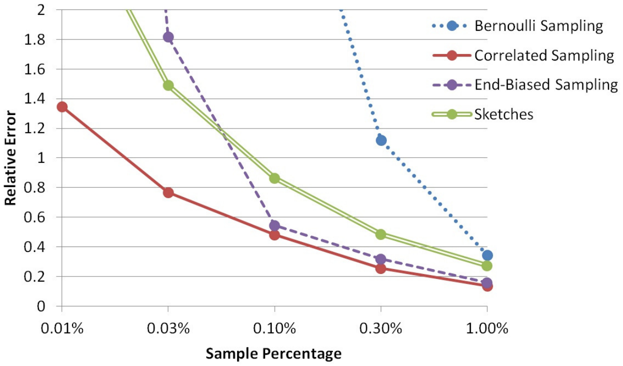

Figure 4. Relative errors for 16X larger customer table

These results show that while the accuracy of Correlated Sampling has not changed relative to the "base case" in Figure 2, the accuracy of the sketch-based approach has noticeably decreased. This can be explained by noting that in equation (4), the sums extend over all attribute values in each of the tables. By increasing the number of unique entries in the customer table by a factor of 16, many more terms were added into this sum. However, for Correlated Sampling, as seen in equation (13), the sum runs only over attribute values which appear in both tables being joined. Since the additional customer_id's introduced were not added to the sales table in this experiment, the variance of the Correlated Sampling estimator was not affected. This experiment shows that the variance of sketches depends on the data which does not appear in the join, while sampling methods are robust to such changes.

5.5 Dependence on Data Skew

In this experiment, we estimated join size of customer table with two modified versions of the sales table (Table 1):

1. No skew: all duplicate versions of customer id were removed in this version, so that each of the 7059 distinct customer_id values has exactly one transaction.

2. High skew: one particular value of cust_id was set to be very popular. This version still contains 92052 entries with 7059 distinct customer_id's, but with the most common value occurring 9205 times.

The relative errors (observed standard deviations divided by the true join size) for all considered techniques when estimating the join size of the modified versions of the sales table with the customer table are shown in Figure 5. We refer to the original sales table as the "medium-skew" dataset.

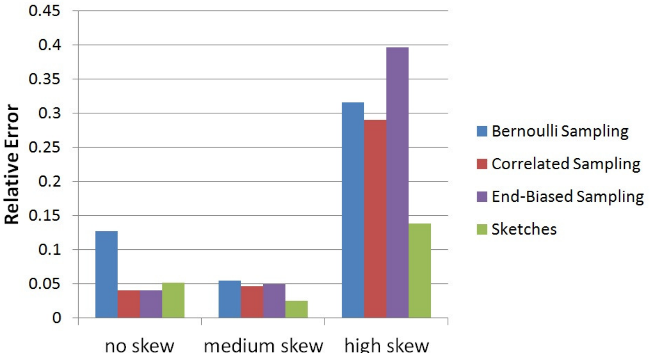

Figure 5. Relative errors as a function of skew in the sales table

As can be seen in Figure 5, the presence of skew affects all methods negatively. This is due to the presence of terms of the form $\sum_v a_v F_1^2(v)$ for a set of constants a_v in all of the variance expressions that were derived earlier. A high value of $F_1(v)$ for any v generates a term that dominates all other terms, thereby quickly increasing the variance. For Correlated Sampling in particular, the sharp variance increase for the high-skew case can be explained as follows. Since the decision to include a row is based on the value of the join attribute in that row, the join size estimate begins to depend heavily on whether or not the highly popular value of v is included in the sample. This creates a binary decision, with a relatively low probability, which has a large effect on the estimator's result, thereby increasing the variance.

The variance of Correlated Sampling, however, can be reduced if the values that occur frequently in both tables are treated separately, as was discussed in Section 3.3.4.

5.6 Dependence on Filter Conditions

Finally, we examine how the presence of filter conditions can affect the variance of each method. The first experiment involves filtering the value of the customer_city_id attribute, requiring that it is less than some limit x. A sampling probability of 0.01 was used in all experiments. The relative errors of the considered techniques are presented in Figure 6 as a function of x.

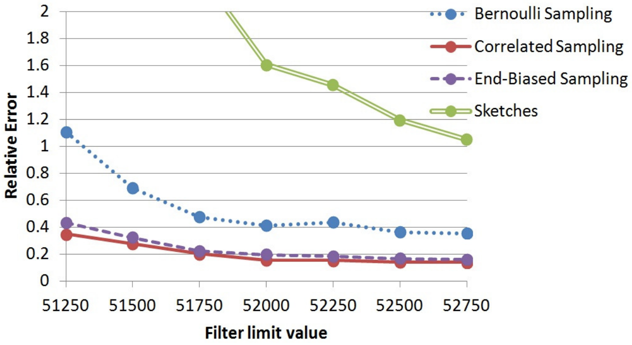

Figure 6. Relative errors as a function of filter selectivity on customer_city_id

A similar experiment was conducted with the value of customer_credit_limit being filtered, and its results are shown in Figure 7.

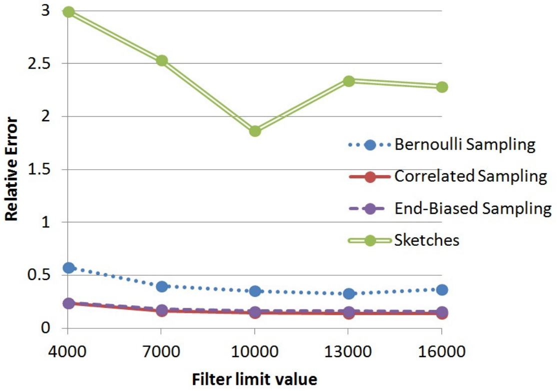

Figure 7. Relative errors as a function of filter selectivity on customer_credit_limit

Note that as the filter becomes less selective (covers a larger data range), the variance of the sketch-based estimator increases, as was shown in equation (5). However, the true join size also increases in this case, and thus the relative error of the estimator (ratio of estimator's standard deviation to the true join size) can either increase or decrease, as is the case in Figure 8. The relative error of the sampling-based methods has a similar behavior, since the variance of these methods decreases as the filter becomes more selective (as was pointed out in Sections 3.1.3 and 3.3.3), but the true join size also decreases. The fact that the relative errors of Correlated and End-Biased sampling are very similar in these experiments is due to the sampling probability of 0.01 being used, for which these methods just happen to give similar results when joining customer and sales tables, as was shown in Figure 2 (with End-Biased sampling becoming noticeably worse as the sampling probability decreases).

Table 1 shows the measured standard deviation for the sketch-based method with and without additional predicate filters.

Table 1. Standard deviation of the sketch-based method

no filters	city_ID < 52000 filter	credit_limit < 10000 filter
7558	99460	158997

The data in Table 1 conforms to the theoretical expectation that the sketch variance increases by at least a factor of 4 when filter conditions are added.

While the above experiment consisted of filtering only one attribute in the sales table (either city_ID or credit_limit), we have also tried filtering these two attributes simultaneously. The result was a complete deterioration of the accuracy for the sketch-based estimator, with the variance exceeding 50 times the true join size estimate value. This highlights the main strength of the sampling methods when compared to sketch-based methods, since the variances of the former methods actually *decrease* when filter conditions are introduced.

5.7 Three-Table Joins

In order to confirm that the two novel join size estimation methods presented in this paper (Correlated Sampling and the sketch-based approach suitable for dynamic filter conditions) can be applied to more than two tables, below we present experiments for three-table joins. As a demonstration, a natural join was performed between the sales (fact) table and the two dimension (customer and product) tables. The End-Biased sampling method was not included in this experiment because it was designed to work only for two tables and its extension to a larger number of tables is not obvious. The results are shown in Figure 8.

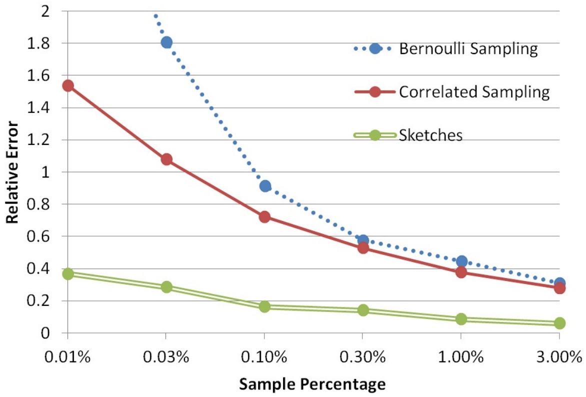

Figure 8. Relative errors of 3-table join size estimates

We have also performed this same experiment using filter conditions on the customer_credit_limit attribute (see Figure 9). This caused the variance of the sketch-based method to increase many-fold, just as was observed in Section 5.6

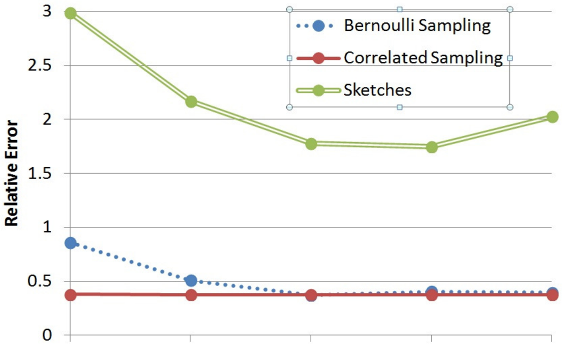

Figure 9. Relative errors of 3-table join size estimates with filter conditions

6. CONCLUSIONS

In this paper we presented a novel Correlated Sampling algorithm for performing one-pass join size estimation, which is applicable to very large database tables and to streaming scenarios. We performed detailed analytical and experimental comparisons between Correlated Sampling and the competing techniques of End-Biased Sampling, Bernoulli Sampling and a method based on AGMS sketches.

Our analysis showed that if dynamically specified filter conditions are allowed, then the variance of the sketch-based estimator greatly increases, while that of the sampling-based methods decreases. Also, as the number of attribute values that do not appear in all tables increases, the variance of the sketch-based estimator increases while that of the sampling methods remains unchanged. Thus, while the sketch-based method satisfies the one-pass requirement, its large variance in many situations makes it impractical as a general purpose join size estimation method.

We also showed that the Correlated Sampling and independent Bernoulli sampling methods are suitable for one-pass scenarios with predicate filter conditions, but the End-Biased sampling method is not suitable as it requires the prior knowledge of frequencies of all join attribute values.

Our analysis showed that Correlated Sampling will in most cases have a smaller estimation variance than Bernoulli Sampling, but it might have a larger variance if there are many join attribute values that occur with a large frequency in all tables to be joined. Thus, Correlated Sampling and independent Bernoulli Sampling can be viewed as complementary join size estimation methods, each with its own set of conditions when it performs the best. In practice, as was suggested in Section 3.3.4, one can run a frequency estimation algorithm such as FSS [8] to identify the values that frequently occur in both tables and then treat them separately. As a result, the variance of Correlated Sampling can become acceptable even for highly skewed data distributions, making it the preferred method to use if one desires to use a single sample to estimate join sizes of different queries that have the same join graph but different filter conditions. If, however, one expects join queries with different join graphs, then independent Bernoulli Sampling is the only technique out of the ones considered in this paper that will provide unbiased estimates with a single sample constructed ahead of time.

7. ACKNOWLEDGMENTS

We thank Dr. Florin Rusu from UC Merced for suggesting how the sketch-based approach developed in [3] for estimating the join size of multiple tables can be used for estimating the join size of two tables subject to dynamically specified predicate filter conditions (by creating an additional table for each filter condition, as we described in Section 3.4.3).

8. REFERENCES

[1] Alon, N., Gibbons, P.B., Matias, Y., Szegedy, M. Tracking Join and Self-Join Sizes in Limited Storage. In *Proceedings of the eighteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems* (*PODS '99),* ACM Press, New York, NY, 1999, 10-20.

[2] Chakrabarti, K., Garofalakis, M.N., Rastogi, R., and Shim, K. Approximate Query Processing Using Wavelets. In *Proceedings of the 26th International Conference on Very Large Data Bases (VLDB '00)*, Morgan Kaufmann Publishers Inc. San Francisco, CA, 2000, 111-122.

[3] Dobra, A., Garofalakis, M., Gehrke, J., and Rastogi, R. Processing Complex Aggregate Queries over Data Streams. In *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data*. ACM Press, New York, NY, 2002, 61-72.

[4] Estan C. and Naughton, J.F. End-biased Samples for Join Cardinality Estimation. In *Proceeding of the 22nd International Conference on Data Engineering (ICDE '06)*. IEEE Computer Society Washington, DC, 2006.

[5] Ganguly, S., Gibbons, P.B., Matias, Y., Silberschatz, A. Bifocal sampling for skew-resistant join size estimation. Proceeding. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data (SIGMOD '96)*, ACM Press, New York, NY, 1996, 271-281.

[6] Ganguly, S., Garofalakis, M., and Rastogi, R. Processing Data-Stream Join Aggregates Using Skimmed Sketches. In *Proceedings of International Conference on Extending Database Technology (EDBT' 2004)*. Lecture Notes in Computer Science, Volume 2992. 569-586.

[7] Haas, P.J., Naughton, J.F., Seshadri, S., and Swami, A.N. Fixed-Precision Estimation of Join Selectivity. *In Proceedings of the twelfth ACM Symposium on Principles of Database Systems*, ACM Press, NY, NY, 1993, 190-201.

[8] Homem, N. and Carvalho, J. P. Finding top-k elements in data streams, *Information Sciences,* 180, 24, (2010).

[9] Lemire, D., Kaser, O. Strongly universal string hashing is fast. *Computer Journal*, 57, 11, (2014), 1624-1638.

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

[11] Rusu, F. and Dobra, A. Pseudo-Random Number Generation for Sketch-Based Estimations. *ACM Transactions on Database Systems*, 32, 2, (2007)

[12] Yu, F., Hou, W-C., Luo, C., Che, D., Zhu, M. CS2: A New Database Synopsis for Query Estimation. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data (SIGMOD '13)*, ACM Press, New York, NY, 2013, 469-480.