

Sample Synopses for Approximate Answering of Group-By Queries

Philipp Rösch
Technische Universität Dresden
01062 Dresden
Dresden, Germany
philipp.roesch@tu-dresden.de

Wolfgang Lehner
Technische Universität Dresden
01062 Dresden
Dresden, Germany
wolfgang.lehner@tu-dresden.de

ABSTRACT

With the amount of data in current data warehouse databases growing steadily, random sampling is continuously gaining in importance. In particular, interactive analyses of large datasets can greatly benefit from the significantly shorter response times of approximate query processing. Typically, those analytical queries partition the data into groups and aggregate the values within the groups. Further, with the commonly used roll-up and drill-down operations a broad range of group-by queries is posed to the system, which makes the construction of highly-specialized synopses difficult.

In this paper, we propose a general-purpose sampling scheme that is biased in order to answer group-by queries with high accuracy. While existing techniques focus on the size of the group when computing its sample size, our technique is based on its standard deviation. The basic idea is that the more homogeneous a group is, the less representatives are required in order to give a good estimate. With an extensive set of experiments, we show that our approach reduces both the estimation error and the construction cost compared to existing techniques.

1. INTRODUCTION

The rapid growth of sensors, e.g. for RFID, as well as the more and more detailed gathering of customer profiles by various companies are just two of the many examples that cause the soaring of the amount of data in current data warehouse systems. In order to profit from this growth, previously unknown information that is hidden in the data, like trends or patterns, has to be extracted. A typically used technique for this extraction is statistical analyses in applications like data mining and decision support systems. However, since those applications require to read large portions of the data, their runtimes significantly increase as well. Moreover, the desired interactivity for explorative analyses or preliminary queries is hard to achieve.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the ACM. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires a fee and/or special permissions from the publisher, ACM. *EDBT 2009*, March 24–26, 2009, Saint Petersburg, Russia. Copyright 2009 ACM 978-1-60558-422-5/09/0003 ...\$5.00

One common solution is the use of small synopses, like histograms [16, 22], wavelets [7, 21] or samples [25], which reflect the characteristics of the underlying data. These synopses allow fast approximate query answers. Especially for explorative analyses and preliminary queries, those estimates of the real values are well accepted.

Among these kinds of synopses, samples have proven to be a good choice: They are easy to implement, they provide probabilistic error bounds, and they can be used for a broad range of applications, like approximate query processing [1, 4, 8, 9, 13, 14, 17, 23], query optimization [5, 6, 10, 12, 15], data analysis [20, 24] or stream processing [18].

In the scenario described above, group-by queries play an important role: The data are typically segmented into groups and aggregated within these groups. Further, the commonly used roll-up and drill-down operations trigger a multitude of group-by queries. Acharya et al.[1] and Babcock et al. [4] have shown that uniform samples of the base data are inappropriate for the approximate answering of group-by queries. The problem is that for uniform samples, the size of a group is regarded as its utility. However, this does not hold in practice, where small groups often are at least as important as large ones. As a consequence of uniform sampling, small groups are under-represented (or even missing) in the sample. The proposed solutions present biased samples for answering group-by queries with high accuracy. These sampling schemes bias the sample inclusion probability of a tuple with respect to the size of the group that tuple belongs to.

In this paper, we propose a novel sampling scheme for constructing memory-bounded group-aware sample synopses. As the proposed solutions in [1] and [4], our sampling scheme is biased in order to provide highly-accurate approximate answers of group-by queries. Similar to [1], we focus on general-purpose synopses in order to equally support arbitrary group-by queries and roll-up and drill-down operations. However, the main difference compared to the existing techniques is that we bias the sample with respect to the relative standard deviation (RSD) of the groups. The basic idea is that groups with low variations of the aggregated values can be represented by just a few tuples, whereas higher variations require larger samples; or in other words, the more homogeneous a group is, the fewer representatives are required in order to give a good estimate.

We start the presentation of our novel approach with a dis-

cussion about how to generally specify the quality of group-aware sample synopses (GASS) in Section 2. Based on this, we introduce RSD-based GASS in Section 3. We first show a hierarchical approach that is motivated by [1]. As stated above, the computation of the sample sizes is adapted to use the variation of the group instead of its size. We further propose a flat approach that significantly reduces the construction cost of the synopsis. We show how our biased sampling scheme can be used for approximate query answering in Section 4, and propose an extension of group-aware sample synopses by using the techniques of the outlier handling of [23] in order to further reduce the estimation error considerably in Section 5. With an extensive set of experiments, we evaluated the properties of our novel approaches. The results shown in Section 6 point out our contributions:

- We propose two RSD-based group-aware sampling schemes. These biased sampling schemes consistently produce approximate query results with higher accuracy compared to existing techniques.
- With the flat approach, we reduce the construction cost of comparable general-purpose group-aware sample synopses from exponential in the number of group-by columns to linear in the number of group-by columns.
- We propose an extension that further significantly reduces the estimation error by an additional outlier handling.
- In summary, we propose a heuristic approach to fastly compute memory-bounded group-aware sample synopses for approximate query answers with low estimation errors.

In Section 7, we classify RSD-based GASS in the field of database sampling and compare it with existing techniques. Finally, we conclude the paper with a summary and an outlook in Section 8.

2. PRELIMINARIES

In this section, we discuss methods to specify the quality of a group-aware sample synopsis. We show how the error of an estimate can be computed and how the estimation errors of the individual groups can be combined to an overall synopsis error. We limit our discussion to AVG and SUM aggregates, which are most commonly used, especially in OLAP settings; our techniques may apply to other aggregates as well.

2.1 Notation

We start with a brief summarization of the notation used in this paper. For a relation R and a given set of group-by attributes G , let \mathcal{G} denote the set of non-empty groups defined by a GROUP BY CUBE command with all of these attributes. Further, let a be the attribute used for aggregation; then t_{ia} is the value of a for tuple $t_i \in R$. Now, for group $g \in \mathcal{G}$,

$$L_a(g) = \sum_{t_i \in g} t_{ia}$$

denotes the linear sum of attribute a , and $\mu_a(g) = L_a(g)/|g|$ denotes its average value. Together with the quadratic sum

$$Q_a(g) = \sum_{t_i \in g} t_{ia}^2$$

the standard deviation of a in group g can be expressed as

$$\begin{aligned} \sigma_a(g) &= \sqrt{\frac{1}{|g|} \sum_{t_i \in g} (t_{ia} - \mu_a(g))^2} \\ &= \sqrt{\frac{Q_a(g)}{|g|} - \left(\frac{L_a(g)}{|g|}\right)^2}. \end{aligned}$$

Using $L_a(g)$ and $Q_a(g)$ allows for the incremental computation of $\sigma_a(g)$. This fact will be used to speed up synopsis computation as explained later. Next, let

$$RSD_a(g) = \frac{\sigma_a(g)}{|\mu_a(g)|}$$

denote the relative standard deviation of a in group g .¹

With S_g being a uniform sample from g ,

$$\hat{\mu}_a(g) = \frac{1}{|S_g|} \sum_{t_i \in S_g} t_{ia}$$

is an unbiased estimate of $\mu_a(g)$. Moreover, with $|S_g| = n_g$, the standard error of this estimate is:

$$\sigma_{\hat{\mu}_a}(g, n_g) = \sqrt{\frac{\sigma_a^2(g)}{n_g} \left(1 - \frac{n_g}{|g|}\right)} = \sigma_a(g) \sqrt{\frac{1}{n_g} - \frac{1}{|g|}}.$$

The standard error of an estimate reflects its precision: It indicates how much the estimate deviates from the exact value if sampling were to be performed multiple times. Consequently, minimizing the standard error of an estimate results in maximizing its quality.

Until now, we have only considered a single aggregation attribute. However, in common data warehouse scenarios, we have a multitude of aggregation attributes. The computation of the quality of an estimate can easily be adapted to handle multiple attributes by making the standard error relative to the mean:

$$RSE_{\hat{\mu}_a}(g, n_g) = \frac{\sigma_{\hat{\mu}_a}(g, n_g)}{|\mu_a(g)|} = RSD_a(g) \sqrt{\frac{1}{n_g} - \frac{1}{|g|}}. \quad (1)$$

By using the relative standard error (RSE), we become independent from units of measurement. This allows us to take the sum of the individual RSEs as the quality measure of an estimate. Let $A = \{a_1, \dots, a_k\}$ be the set of aggregation attributes. Then

$$RSE(g, n_g) = \sum_{a \in A} RSE_{\hat{\mu}_a}(g, n_g)$$

gives the estimation error of group g .

For simplification, the formulas given so far have been restricted to the AVG aggregation function. However, the

¹The relative standard deviation is not defined for $\mu_a = 0$ and may get very large for $\mu_a \approx 0$. In our implementation, we set $RSD_a(g) = \sigma_a(g)$ whenever $\mu_a \in [-1, 1]$.

estimation error for the SUM aggregation function is very similar and can be incrementally computed as well.

Finally, in order to measure the quality of a complete group-aware sample synopsis, we have to combine the individual RSEs of the groups. Therefore, we consider two error measures:

$$\mathcal{E}_{AVG} = \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} RSE(g, n_g),$$

which computes the average relative standard error over all groups, and

$$\mathcal{E}_{MAX} = \max_{g \in \mathcal{G}} (RSE(g, n_g)),$$

which returns the largest relative standard error of all groups.

Obviously, these quality measures attribute the same utility to all kinds of groupings.

2.2 Observations

Existing techniques, like [1] or [4], determine the sample sizes of the individual groups with respect to their cardinalities, that is, two groups with the same cardinality get equal sample sizes.

EXAMPLE 1. Consider a relation R with $|R| = 10,000$ and the following groups $G = \{g_1, g_2\}$:

Group	Cardinality	RSD
g_1	5,000	1%
g_2	5,000	49%

For a synopsis Ψ with $|\Psi| = 100$, cardinality-based techniques assign $n_1 = n_2 = 50$, resulting in an average error of $\mathcal{E}_{AVG} = (0.14\% + 6.89\%)/2 = 3.52\%$; the maximum error is $\mathcal{E}_{MAX} = 6.89\%$.

As can be seen in (1), the quality of an estimate, however, is also influenced by the relative standard deviation of the group. Consequently, taking the RSD of a group into account when computing its sample size promises a positive impact on the quality of a group-based sample synopsis.

After these preliminary considerations, we next show how RSD-based group-aware sample synopses can be constructed.

3. RSD-BASED GROUP-AWARE SAMPLE SYNOPSES

In this section, we describe the idea and the computation of RSD-based group-aware sample synopses in more detail. We start with a hierarchical approach and subsequently propose a flat approach that significantly reduces the computational effort.

The general idea of RSD-based synopses is to take the fact into account that groups with low variations in the aggregation attributes can be represented by just a few tuples; groups with high variations, however, require noticeably more representatives in order to provide approximate query results with high accuracy.

EXAMPLE 2. Regard the scenario of example 1. By computing the sample sizes based on the RSDs of the groups, the sample sizes change to $n_1 = 2$ and $n_2 = 98$. As a consequence, the average error of the synopsis can be reduced to $\mathcal{E}_{AVG} = (0.71\% + 4.90\%)/2 = 2.80\%$; the maximum error decreases to $\mathcal{E}_{MAX} = 4.90\%$.

3.1 Hierarchical Approach

Inspired by the results shown in Example 2, our group-aware sample synopses use the relative standard deviations of the individual groups for the computation of the sample sizes. In order to provide highly-accurate query answers for arbitrary group-by queries, the proceeding of synopsis computation of the hierarchical approach is similar to Congressional Sampling [1]. Let $\mathcal{B} \subseteq \mathcal{G}$ denote the set of non-empty groups under the grouping G . The grouping G results in the finest possible partitioning for group-bys on R . We refer to \mathcal{B} as the *base groups*. Any group h on any other grouping $T \subset G$ is the union of one or more groups g from \mathcal{B} . We denote each such g to be a *subgroup* of h .

EXAMPLE 3. For a relation R with three columns A , B and C , let $G = \{A, B\}$ be the columns used for grouping and C be the aggregation attribute. The groups and the relative standard deviation of attribute C (RSD_C) within these groups (computed from randomly generated data) are given in Table 1. For this example, the set of base groups is given by $\mathcal{B} = \{(a_1, b_1), (a_1, b_2), (a_2, b_1), (a_2, b_2)\}$, and for the grouping $T = \{A\}$, the set of tuples in the group $h = a_1$ is the union of the tuples in the subgroups (a_1, b_2) and (a_1, b_1) .

Synopsis Computation

Let M be the memory bound for the synopsis. With the hierarchical approach, the construction of our RSD-based group-aware sample synopses Ψ consists of 4 phases:

Phase 1: Initialization. During a single scan of relation R , the quantities $L_a(g)$ and $Q_a(g)$ are maintained for each group $g \in \mathcal{G}$. Based on these quantities, the individual RSDs of all the groups are computed.

Phase 2: Flat Partitioning. In the second phase, we proceed for each $T \subseteq G$ as follows: With \mathcal{T} as the set of non-empty groups under grouping T , we partition the available space M among all groups in \mathcal{T} with respect to their relative standard deviations. Let

$$SUMRSD(\mathcal{T}) = \sum_{g \in \mathcal{T}} RSD_a(g)$$

be the sum of RSDs of all groups in \mathcal{T} .² Then, the sample size $n_{g,T}$ of a group g is determined by³

$$n_{g,T} = \frac{RSD_a(g)}{SUMRSD(\mathcal{T})} M.$$

²As for the estimation error, for multiple aggregation attributes $A = \{a_1, \dots, a_k\}$ we simply use $\sum_{a \in A} RSD_a(g)$ instead of $RSD_a(g)$

³For simplicity we assume throughout this paper that each group is larger than the sample size. Handling scenarios when this is not the case is straightforward. Further, for groups with $RSD_a(g) \approx 0$, we replace $n_{g,T}$ by a constant c .

Table 1: Example scenario and the resulting group sample sizes of RSD-based group-aware sample synopses

A	B	RSD_C	$n_{g,AB}$	$n_{g,A}$	$n_{g,B}$	(unscaled)	Hierarchical	Flat
a_1	b_1	33.19%	28.20	27.54 (of 41.25)	28.35 (of 68.03)	(28.35)	28.03	28.20
a_1	b_2	16.53%	14.04	13.71 (of 41.25)	13.88 (of 31.97)	(14.04)	13.88	14.04
a_2	b_1	46.45%	39.46	40.14 (of 58.75)	39.68 (of 68.03)	(40.14)	39.69	39.46
a_2	b_2	21.54%	18.30	18.61 (of 58.75)	18.09 (of 31.97)	(18.61)	18.40	18.30
a_1	–	29.38%		41.25		(42.39)	41.91	42.24
a_2	–	41.85%		58.75		(58.75)	58.09	57.76
–	b_1	39.67%			68.03	(68.49)	67.72	67.66
–	b_2	18.64%			31.97	(32.65)	32.28	32.34
–	–	35.29%				(101.14)	100.00	100.00

Phase 3: Hierarchical Partitioning. In the next phase, we recursively consider for each group g the set of its subgroups. The procedure is similar to that from the second phase: First, we compute the quantity $SUMRSD$ over all the subgroups of g , and afterwards, we partition the sample size $n_{g,T}$ among the subgroups with respect to their RSDs.

Phase 4: Finalization. In the last phase, each group g is assigned the maximum of all the sample sizes assigned to this group during the second and the third phase: $n_G = \max_{T \subseteq G} n_{g,T}$. Finally, we scale down the sample sizes to limit the space used to M , i. e., the sample size n_g is computed by

$$n_g = \frac{M}{\max_{T \subseteq G} n_{g,T} \sum_{j \in \mathcal{B}} \max_{T \subseteq G} n_{j,T}}.$$

After the computation of the sample sizes, samples of all groups $g \in \mathcal{B}$ are drawn, i. e.

$$\Psi = \bigcup_{g \in \mathcal{B}} s_g.$$

This recursive approach is adapted from [1]. However, the main difference is that for [1] the sizes of the subgroups sum up to the size of the supergroup, which does not hold for the RSDs. Nevertheless, we have chosen this approach in order to evaluate the impact of changing the allocation criterion.

EXAMPLE 4. Consider again the scenario given in Table 1 for the construction of a group-aware sample synopsis with the hierarchical approach. During the initialization phase, the RSDs are computed as given in the column RSD_C .

In the second phase, the flat partitioning is executed: For $T = \{AB\}$, we have $SUMRSD(T) = 117.71\%$. Let $M = 100$, then for $g = (a_1, b_1)$, the sample size $n_{g,AB} = \frac{33.19\%}{117.71\%} \cdot 100 = 28.20$. The computation for the other groups in T as well as for the groups for $T = \{A\}$ and $T = \{B\}$ is analogous. The resulting sample sizes for $T = \{AB\}$ are given in column $n_{g,AB}$, and for $T = \{A\}$ and $T = \{B\}$ the sample sizes are given in the lower part of the table in columns $n_{g,A}$ and $n_{g,B}$, respectively.

In the third phase, the subgroups for $T = \{A\}$ and $T = \{B\}$ are considered. Taking, for example, group $g = (a_1)$ from $T = \{A\}$, the RSDs of the subgroups of g sum up to 49.72.

Now, the sample size of g ($n_g = 41.25$) is partitioned among its subgroups, i. e. for group $g_1 = (a_1, b_1)$ we get $n_{g_1,A} = \frac{33.19\%}{49.72\%} \cdot 41.25 = 27.54$, and for group $g_2 = (a_1, b_2)$ we get $n_{g_2,A} = \frac{16.53\%}{49.72\%} \cdot 41.25 = 13.71$.

In the last phase, for all groups $g \in \mathcal{B}$, the maximum of the sample sizes is computed as shown in the upper part of column (unscaled) and scaled down to fit the memory bound M as shown in the upper part of column Hierarchical of Table 1. We also present the resulting sample sizes for groupings $T \subset G$ in the lower part of the table for illustrative reasons. Also, we did not round the individual sample sizes to integers to emphasize the impact of the single steps and to show the differences between the two proposed algorithms.

Although, the construction algorithm was explicitly shown for RSD-based synopses above, our implementation is kept far more abstract.

Algorithmic Framework

The algorithmic framework that underlies the proposed approaches allows the computation of arbitrary hierarchical group-aware sampling schemes, hence, it supports Congressional Sampling [1] as well. For the construction of a synopsis, one only has to specify the weight function, e. g., the relative standard deviation for RSD-based synopses or the Congressional procedure (consisting of House- and Senate-like allocations) for Congressional samples. As a consequence, in the four steps given above, the abstract weight function is always used: The sum of weights is computed and the partitioning is based on the ratio of the weights. This allows to easily plug in new allocation strategies.

3.2 Flat Approach

Albeit the hierarchical approach seems promising, there are two facts that induced us to consider the flat approach:

- First, the construction cost using the hierarchical proceeding is exponential in the number of group-by columns. Obviously, the large number of group-by columns in current data warehouse databases imply prohibitively high system loads for synopses construction.
- Second, due to the fact that the RSDs of the subgroups do not sum up to the RSD of the supergroup, we put the recursive approach into question.

Table 2: Example scenario showing the shortcoming of the flat approach

A	B	RSD_C	$n_{g,AB}$	$n_{g,A}$	$n_{g,B}$	(unscaled)	Hierarchical	Flat
a_1	b_1	1.57%	1.94	10.12 (of 55.68)	1.88 (of 88.34)	(10.12)	6.98	1.94
a_1	b_2	7.07%	8.76	45.56 (of 55.68)	11.66 (of 11.66)	(45.56)	31.43	8.76
a_2	b_1	72.11%	89.30	44.42 (of 44.42)	86.46 (of 88.34)	(89.30)	61.59	89.30
a_1	–	90.59%		55.68		(55.68)	38.41	10.70
a_2	–	72.11%		44.42		(89.30)	61.59	89.30
–	b_1	53.55%			88.34	(99.42)	68.57	91.24
–	b_2	7.07%			11.66	(45.56)	31.43	8.76
–	–	90.14%				(144.98)	100.00	100.00

Resulting from these considerations, the construction of a group-aware sample synopsis Ψ with the flat approach looks as follows:

Phase 1: Initialization. The initialization phase is not changed. Thus, the individual RSDs of all the groups are computed.

Phase 2: Flat Partitioning. In the second phase, we restrict the computation of the sample sizes to the base groups \mathcal{B} . The general procedure of the sample size computation remains unchanged. However, since we only consider base groups, there are no subgroups available for the hierarchical partitioning. Hence, phase 3 from the hierarchical approach can be omitted.

Phase 3: Finalization. The finalization phase is simplified and now only includes the drawing of the samples of all groups $g \in \mathcal{B}$ and the composition of the synopsis

$$\Psi = \bigcup_{g \in \mathcal{B}} s_g.$$

EXAMPLE 5. For the scenario given in Table 1, the flat approach proceeds as follows: The first phase is unchanged, that is, we compute the RSDs as given in column RSD_C . Now, in the second phase, we only compute the (flat) partitioning for $T = \{AB\}$, and draw the samples in the final phase. The resulting sample sizes are given in the last column, named Flat. As can be seen easily, these results equal the intermediate results of the hierarchical approach for $T = \{AB\}$ given in column $n_{g,AB}$. Further, the results of the hierarchical and the flat approach are very similar, while the construction cost of the flat approach is significantly smaller.

Obviously, this proceeding simplifies the construction algorithm of the hierarchical group-aware sample synopsis: It reduces the complexity from exponential in the number of group-by columns to linear in the number of group-by columns. However, this simplification has its price: If there is a group g with two homogeneous subgroups, the sample size of g will be small. If, however, the union of the tuples of both subgroups has a large variation, the estimation error for group g will be large as well.

EXAMPLE 6. Consider the scenario shown in Table 2. Now, the underlying data are generated carefully: Group

(a_1, b_1) consists of values of about 100, the values of group (a_1, b_2) are about 5, and group (a_2, b_1) consists of values equally distributed between 1 and 100. The point is that both (a_1, b_1) and (a_1, b_2) have low RSDs, while the RSD of (a_1) is large. In the second phase (flat partitioning) of both approaches, both (a_1, b_1) and (a_1, b_2) get small sample sizes. As a consequence, after the second phase, the sample size of (a_1) is small as well, which does not reflect its RSD. This problem cannot be captured by the flat approach, hence, group (a_1) is under-represented in the resulting synopsis. The hierarchical approach, however, is insensitive to those data formations, as can be seen in the table.

The last example shows the weakness of the flat approach. However, those pathological data formations will not be common in real-world datasets. In general, the difference between the RSDs of two groups and their union will not be that large, and further, the effect usually is even smaller when the respective group has a larger number of subgroups. In particular, in all our experiments this effect was not visible.

After the presentation of the idea and the algorithm of our RSD-based group-aware sample synopsis, we will show how to answer queries from biased samples in the next section.

4. APPROXIMATE QUERY ANSWERING WITH BIASED SAMPLES

In this section, we show how group-aware sample synopsis can be used for approximate query answering. Using a biased sampling scheme as proposed in this paper requires some more sophisticated estimation techniques than in the uniform case. In the uniform sampling case, each tuple has the same probability of being selected and estimation is mostly simple. For the SUM operator, for example, the estimate is computed by simply multiplying the sum of the values in the sample by the inverse of the sampling fraction. Applying this procedure to biased samples would lead to biased estimates.

Fortunately, there is a seamless solution: As biased samples can be regarded as a union of uniform random samples with different sampling fractions, they also can be treated as stratified samples. Hence, using standard techniques for estimators based on stratified sampling schemes, we can generate unbiased answers with the help of all the tuples in the biased sample [11]. With this technique, each stratum has the inverse of its sampling fraction as an associated scaling factor. In order to estimate a SUM operator, the sum of

each stratum is scaled up by the associated scaling factor, and these sums are added up to the final estimate. An estimate for the COUNT operator is computed by summing up the individual scaling factors of each tuple relevant for the current query. Finally, for an estimate of an AVG operator, we compute the scaled SUM divided by the scaled COUNT.

As mentioned in [1], the key issue in scaling is the possibility to be able to efficiently associate each tuple with its corresponding scaling factor. The authors mention two approaches: a) store the scaling factor for each tuple within the sample, i. e., augment the sample with an additional column for the scaling factor, and b) use a separate table to store grouping identifiers together with the scaling factors. For details, advantages, and drawbacks see [1]. While these approaches take the AQUA system [2] as a basis, we focus on the Derby/S [19] system. The key differences of these two systems are the following:

- While the AQUA system is a middleware software tool sitting on top of a DBMS, Derby/S itself is a full-fledged DBMS with samples integrated as first-class citizens. In more detail, Derby/S is an extension of the open-source database system Derby⁴ for approximate query processing.
- In the AQUA system, all queries to the base tables are automatically rewritten in order to use the available synopsis structures. In contrast, with Derby/S, users can specify whether they want to get an approximate or an exact result by using SQL/S, an extension of SQL, for approximate query answering [19]. Exact queries are executed on the base tables while approximate queries are automatically rewritten in order to use the synopses instead.

Derby/S supports several sampling designs, including stratified samples, associated with appropriate rewrite strategies and estimation operators. A sample of our proposed sampling scheme is split into several tables, one for each group. The scaling factors are stored in the respective system catalog tables. Incoming approximate queries are automatically rewritten and aggregation operators are substituted by estimation operators. The group-wise intermediate results are scaled and merged as described above.

EXAMPLE 7. Consider a group-aware sample synopsis for the `lineitem` table of the TPC-D schema with $G = \{1_returnflag\}$. Since the `1_returnflag` attribute has three different values, the resulting synopsis consists of three sample tables named `lineitem_smpl_1`, `lineitem_smpl_2`, and `lineitem_smpl_3`. An approximate group-by query with `1_returnflag` as the group-by attribute is executed as shown in the execution plan given in Figure 1. The nodes in this execution plan show the individual operations together with the query costs so far. First, the sample tables are scanned. Afterwards, the group-wise estimates are computed within the `GROUPBY` nodes and subsequently merged by the `UNION` nodes. The proximate `GROUPBY` node is only needed for queries where the group-by attributes are not identical with

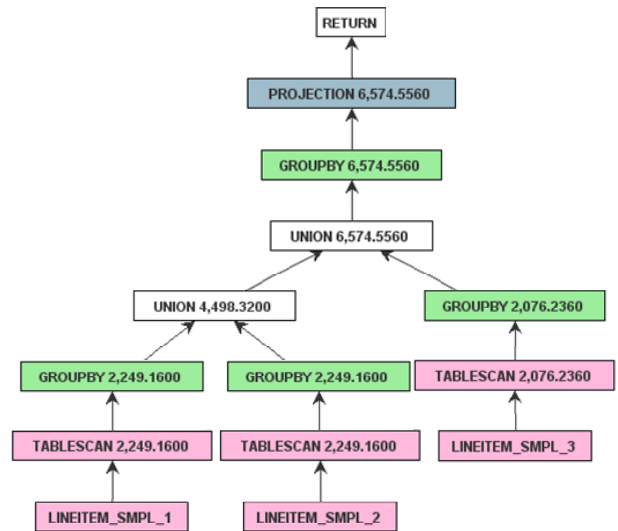


Figure 1: Query execution plan for an approximate group-by query on a group-aware sample synopsis using Derby/S

the set G of the group-aware sample synopsis. As can be seen, no costs are caused by this operator for the current query. Finally, unnecessary attributes are eliminated by the `PROJECTION` node and the approximate query result is returned.

After this sketch of approximate query answering with biased samples in Derby/S, we will use the next section to show how the algorithms can be extended.

5. EXTENSION

This section describes an extension of group-aware sample synopses that further allows for large reductions of the relative standard errors of the samples, and hence, of the overall synopsis error.

When estimating an aggregate from a sample, *outliers* in the data may lead to large estimation errors [8, 23]. With outliers, we denote values that are significantly different from the rest of the data, but that are important for the result of the aggregation function. Take, for example, a hypothetical company with a revenue in 2008 of \$1 billion and a single purchase order of \$100 million. This order is represented as a single tuple in the database, but it constitutes 10% of the total revenue. When random sampling is applied prior to aggregation, outliers in the data typically lead to underestimation (if they are not present in the sample) or overestimation (if present). Thus, outliers should be well represented by a synopsis of the underlying dataset.

Chaudhuri et al. proposed a sampling scheme called *outlier indexing* [8], which significantly reduces the estimation error by storing outliers separately and sampling from the remaining part of the data set. In [23], we extended this approach to handle multiple aggregation attributes by introducing three measures for the efficient identification of outliers for multiple attributes. Among these measures, the \mathcal{M}_{AVG} measure turned out to be the most promising. With this measure,

⁴<http://db.apache.org/derby/>

outliers are identified by their impact on the average relative standard error of the aggregation attributes, i.e., the larger the reduction of the RSE of the data *without* the current tuple compared to the RSE of the data *with* the current tuple, the more likely this tuple is an outlier; for more details see [23]. With an additionally proposed heuristic approach, the outlier handling runs in $O(|R| \log M)$ worst-case time and requires $O(M)$ space. Thus, it has only low impact on the performance of the synopsis computation.

As mentioned above, our proposed biased sampling schemes can also be regarded as stratified sampling schemes, where the stratum boundaries are given by the base groups \mathcal{B} . In such a stratified sampling scheme, the individual samples are independent from each other. Hence, we can simply apply the *multi-column outlier index* (MCOI) from [23] to each stratum. The synopsis design changes in the way that now, the sample of a group may consist of separately stored outliers and a sample from the remaining part of the data of that group. The extension is plugged in the finalization phase, i.e., the construction algorithm is only slightly altered. With the extension, the finalization phase changes as follows: After scaling down the sample sizes, MCOI is applied in each group; a modification of MCOI is not required. Since MCOI computes both the outliers and the samples, the terminal draw of the samples of the unextended approaches can be omitted.

Obviously, this procedure can be applied to all stratified sampling schemes; thus, it is also applicable to [1]. After this introduction of the extension, we evaluate our approaches in the next section.

6. EXPERIMENTS

We ran a variety of experiments in order to evaluate the effectiveness and the efficiency of our RSD-based group-aware sample synopses. We compared the hierarchical approach (H-GASS) and the flat approach (F-GASS) with Congressional Sampling (CS) [1]; we have chosen CS since this approach has the same focus of general-purpose group-aware sample synopses as our approaches. We experimented with well-defined synthetic datasets in order to discover the impact of certain “data formations” on the quality of the synopsis. Finally, we ran experiments on a large real-world dataset consisting of retail data.

Note, that the considered algorithms are deterministic with respect to the resulting sample sizes. Hence, the quantities that measure the error of a synopsis, that is, \mathcal{E}_{AVG} and \mathcal{E}_{MAX} , can analytically be computed.

Summary of Results

- Both approaches of RSD-based group-aware sample synopses result in lower estimation errors (\mathcal{E}_{AVG} as well as \mathcal{E}_{MAX}) than CS does in most cases.
- The flat approach of the RSD-based group-aware sample synopses (F-GASS) consistently causes lower estimation errors (\mathcal{E}_{AVG} as well as \mathcal{E}_{MAX}) than CS does.
- The hierarchical approach of RSD-based group-aware sample synopses (H-GASS) produces synopses with the lowest maximum estimation error (\mathcal{E}_{MAX})

- The construction of F-GASS is several orders of magnitude faster than that of H-GASS and CS.
- The extension for outlier handling enables significantly smaller estimation errors.

6.1 Experimental Setup

We implemented the RSD-based GASS approaches on top of DB2 using Java 1.6. The experiments were conducted on a Dual Core AMD Opteron (2 GHz) system running Linux with 9 GB of main memory.

In order to provide comparable results, we conducted our experiments on generated data as described in [1]: As base data, we used the `lineitem` table from TPC-D schema with a size of 1 million tuples. For the grouping, we considered the columns `l_returnflag`, `l_linestatus` and `l_shipdate`; as aggregation attribute, we used `l_extendedprice`. We further introduced skew in the group sizes and in the data of the aggregation attribute. As in [1], this was done by using the Zipf distribution with varying values for the z -parameter ranging from 0 (uniform) to 1.5. (highly skewed). We also varied the number of groups from 10 to 100,000. Table 3 summarizes the parameters and their ranges of values. Unless stated otherwise, the parameters take the value given in the last column (Default value); again, these values are chosen according to [1].

Our real-world dataset consists of market research data and is made up of 13,223,779 tuples. The dataset has four columns used for grouping and several columns used for aggregation. The group-by columns are `project`, `date`, `country`, and `sales_channel`; group sizes of the base groups \mathcal{B} vary between 1 and 6,081 tuples. The information used for aggregation is, for example, about sales units or stock units.

6.2 Results for Synthetic Data

After the description of the experimental setup, we now present our experimental results.

Number of groups. Our first experiment evaluates the impact of the number of groups on the estimation error and on the number of missing groups. We computed synopses with CS, H-GASS and F-GASS on datasets with the number of groups ranging from 10 to 100,000. For each synopsis, we computed \mathcal{E}_{AVG} as well as the fraction of missing groups. The results are given in Figure 2. As can be seen, for small (10) and large (100,000) numbers of groups, the three approaches are similar. In the former case, all groups are easily well represented independent from the approach; in the latter case, the missing groups dominate the result, i.e. the small groups are missing, and the large groups are well represented. However, the difference between the approaches can be seen in the remaining group sizes: For 1,000 groups, none of the approaches misses a group, but F-GASS can produce the synopsis of the highest quality, followed by H-GASS and CS. For 10,000 groups, CS misses only 0.05% of the groups, whereas F-GASS misses 4.7% and H-GASS misses 6.6% of all groups. However, the quality of the groups contained in the approximate query answer is noticeably higher for F-GASS. So, it is up to the user to decide which result is better, that of F-GASS or that of CS. Anyway, H-GASS is a

Table 3: Parameters for the experiments

Parameter	Range of values	Default value
Skew of group sizes	0 – 1.5	0.86
Skew of aggregation column	0 – 1.5	0.86
Number of groups	10 – 100,000	1,000
Number of group-by columns	1 – 5	3
Memory bound	1% – 10%	5%

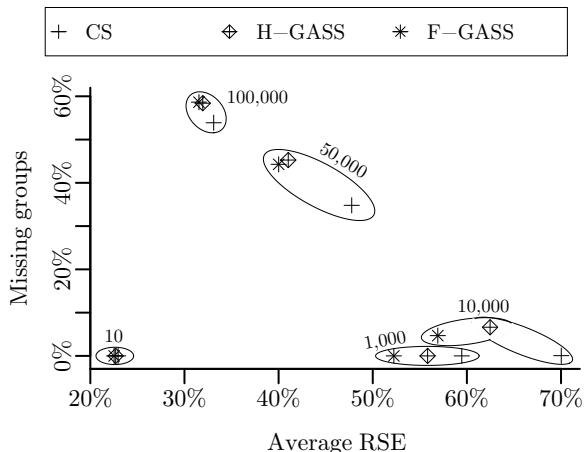


Figure 2: Impact of the number of groups

bad choice for this setting, since it is worse than F-GASS in both dimensions. Finally, for 50,000 groups, H-GASS and F-GASS are very similar, whereas CS produces approximate answers with more groups but larger errors.

Number of group-by attributes. A related experiment considers the number of group-by attributes. For the dataset with the default parameter values, we computed synopses with CS, H-GASS and F-GASS for an increasing number of group-by attributes. The number of group-by attributes varied from 1 to 5. Note that for a given dataset the increase of the number of group-by attributes also implies an increase of the number of groups. In all settings, all three approaches produce synopses with no missing groups; hence, we can restrict the discussion on the estimation error. The results are shown in Figure 3. First, for all approaches, the estimation error increases with increasing number of group-by columns. The reason is the simultaneous increase of number of groups for an increasing number of group-by columns. Comparing the approaches, one can see that the RSD-based synopses consistently produce synopses with smaller \mathcal{E}_{AVG} (average RSE). In detail, using F-GASS results in the synopses with the lowest error, whereas H-GASS lies in between F-GASS and CS.

Skew of group sizes. In the next experiment, we varied the skew of the group sizes from $z = 0$ (uniform) to $z = 1.5$ (highly skewed). Note that the default value of $z = 0.86$ results in a 90 – 10 distribution and is commonly used. Again, in all settings, all three approaches produce synopses with

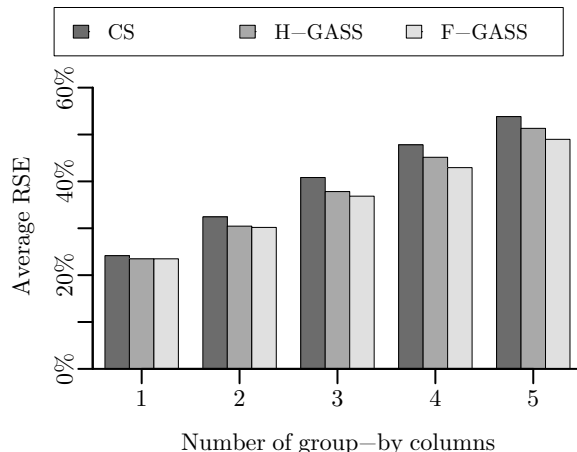


Figure 3: Impact of number of group-by attributes

no missing groups and we can restrict the discussion on the estimation error. Figure 4 depicts the results of the experiments. As in the previous plots, Figure 4a shows the impact of the group-size skew on \mathcal{E}_{AVG} . Here, for all approaches, the estimation error decreases with increasing skew of the group sizes: The larger the skew, the more small groups exist and as long as no group is missing, the more of these small groups have a lot of representatives resulting in a small estimation error. In all cases, F-GASS produces the synopses with the lowest \mathcal{E}_{AVG} ; for H-GASS, only the case of $z = 0$ results in a slightly larger average error than for CS; otherwise, the average error is smaller.

In order to demonstrate the impact on \mathcal{E}_{MAX} , we also plotted this quantity in Figure 4b. Again, the RSD-based approaches consistently result in lower errors, which is now even more significant. Additionally, for \mathcal{E}_{MAX} , H-GASS is better than F-GASS.

Skew in aggregation values. Another experiment analyzes the impact of the skew of the aggregation column. Since the skew of the aggregation column influences the RSDs of the groups, this parameter is expected to have the largest influence on the RSD-based approaches. As in the previous experiments, we computed synopses with CS, H-GASS and F-GASS. We varied the skew of the aggregation column from $z = 0.5$ to $z = 1.5$. Again, we plotted the results for both \mathcal{E}_{AVG} (see Figure 5a) and \mathcal{E}_{MAX} (see Figure 5b). First, unlike in the previous experiments, now the resulting synopses miss some groups: For $z = 1.5$, H-GASS misses 4.2% and F-GASS misses 5.4% of the groups, whereas

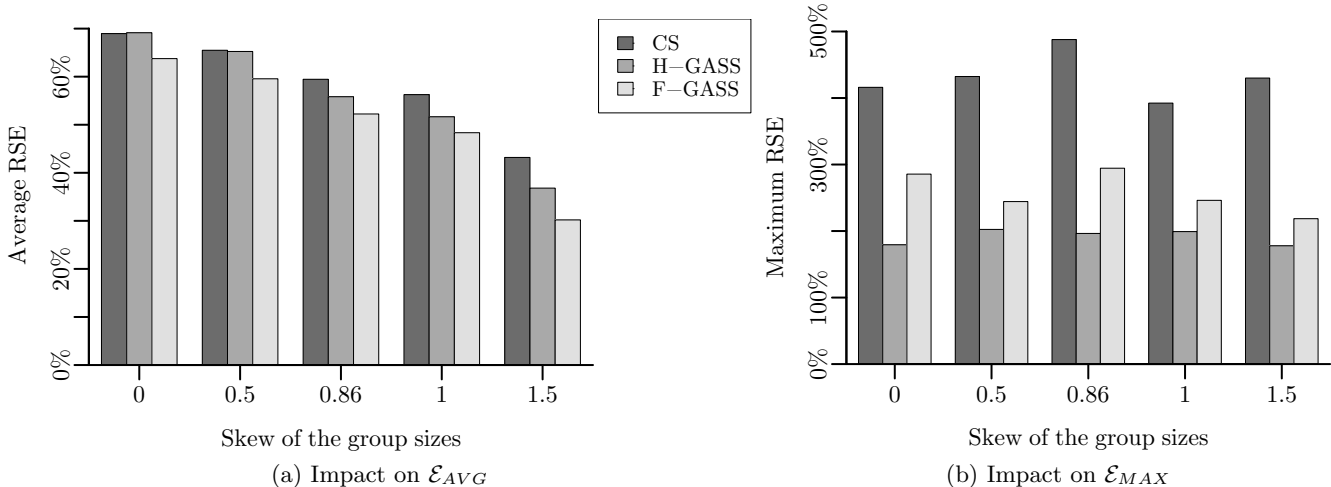


Figure 4: Impact of skew of the group sizes

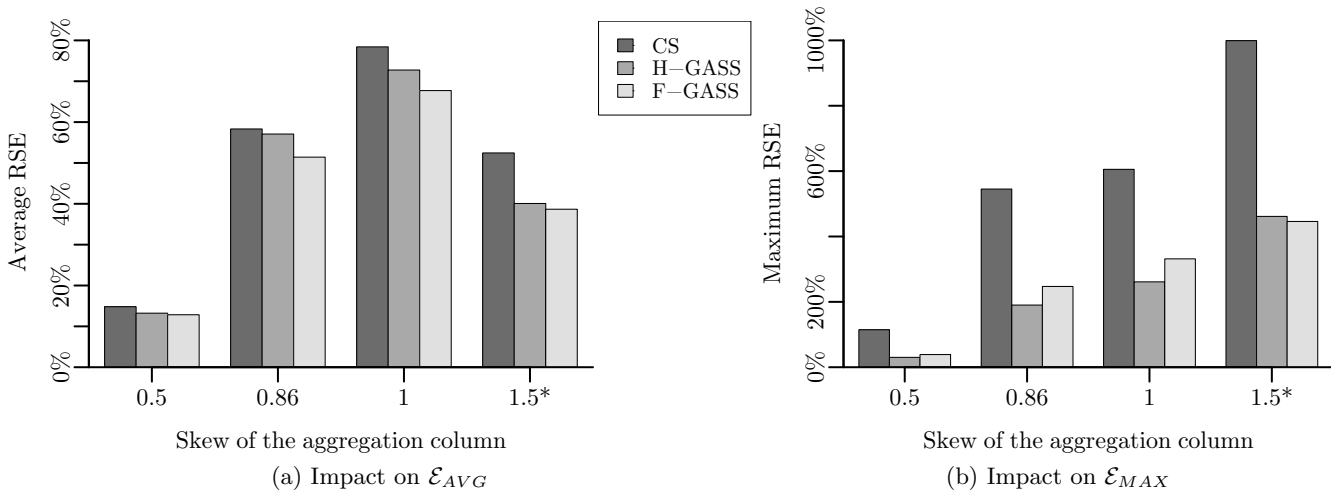


Figure 5: Impact of skew of the aggregation attribute

CS does not miss any group.

For \mathcal{E}_{AVG} , the results look similar to the previous experiments: Both H-GASS and F-GASS produce synopses with lower errors, while the error with F-GASS is even lower than with H-GASS. More interesting are the results for \mathcal{E}_{MAX} : Although the error increases for all three approaches with increasing skew, the error of CS increases much faster than for the RSD-based approaches. This shows the impact of the variation of a group on the error of the synopsis.

Even though the impact on \mathcal{E}_{MAX} is larger than on \mathcal{E}_{AVG} in all our experiments, we further on use \mathcal{E}_{AVG} for evaluation since we think that this value is more intuitive and representative for determining a synopsis' quality.

Memory bound. We also evaluated the impact of the memory bound on the synopsis quality. With the default parameter values, we computed CS, H-GASS and F-GASS. The

memory bound was varied from 1% to 10% of the size of the base data. As expected, the error decreases for increasing memory bounds. Again, the error of F-GASS is consistently lower compared to H-GASS and CS. For small synopses, the error of H-GASS and CS is similar; for increasing memory bounds, the error of H-GASS converges to the error of F-GASS. The results are plotted in Figure 6.

Computational effort. As stated above, the hierarchical procedure of H-GASS and CS is very expensive. We compared the hierarchical and the flat approach for a varying number of levels in the hierarchy of the groups. We used the dataset from the second experiment, that is, the experiment on the number of group-by attributes. Thus, the number of levels varied from 1 to 5. In order to be independent from object creation overhead or variations of the measurements, we simply used the number of loops as indicator of the computational effort. In Figure 7, the results are shown. As expected, the effort for the hierarchical computa-

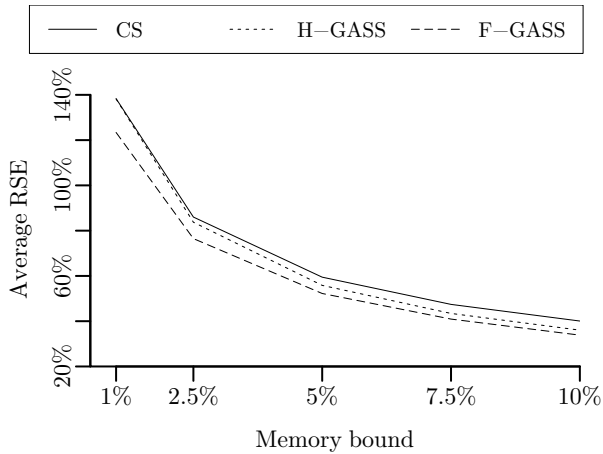


Figure 6: Impact of the memory bound

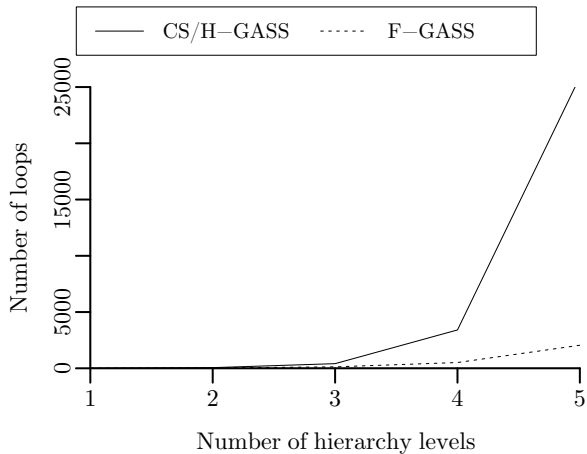


Figure 7: Computational effort

tion significantly increases with the number of levels in the hierarchy. Unlike expected, the effort of F-GASS increases slightly more than linear in the number of hierarchy levels. The reason, however, is just the simultaneously growing number of groups. Consequently, the results slightly intermix two parameters: the number of levels in the hierarchy, and the number of groups. However, the impact of the latter is of less significance. Summarizing, this experiment shows that the hierarchical approaches fastly get prohibitively expensive while the computational effort of the flat approach increases very slowly.

Outlier handling. In a final experiment, we evaluated the effectiveness of the outlier extension. As in the fourth experiment, we varied the skew of the aggregation attribute from $z = 0.5$ to $z = 1.5$ in order to have different kinds of extreme values. We computed H-GASS and F-GASS with and without the MCOI extension⁵ and compared the average estimation error. The results, shown in Figure 8, clearly show that the additional outlier handling is able to signifi-

⁵For the outlier computation, we used the \mathcal{M}_{AVG} measure proposed in [23]

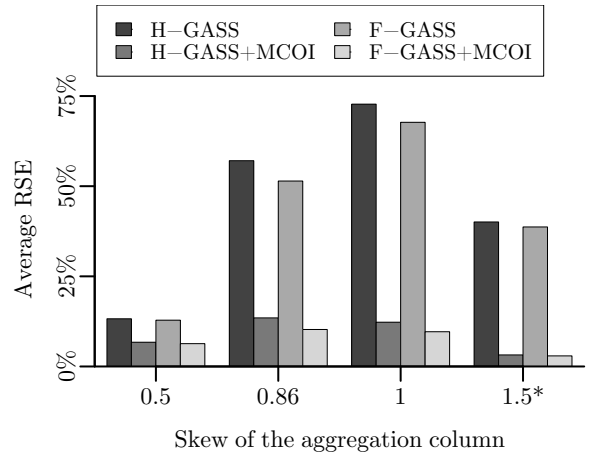


Figure 8: Impact of outlier handling

cantly reduce the average estimation error for both H-GASS and F-GASS.

Summary. With this extensive set of experiments on well-defined synthetic datasets, we demonstrated that the RSD-based approach is promising in the field of approximate answering of group-by queries for a broad range of data formations. With consistently lower estimation errors and—if using F-GASS—significantly lower construction cost, users of applications based on approximate query processing will greatly benefit from our proposed techniques.

6.3 Results for Real-World Data

We conducted a variety of experiments on the sales dataset described above. We evaluated the estimation error for the average of the column containing the sales units.

Number of group-by attributes / Number of groups. First, we considered the number of group-by attributes. As stated above, in this experiment, the simultaneously varying number of groups influences the results as well. For clarification, we show the number of groups for the respective number of group-by columns in Table 4.

Again, we computed group-aware sample synopses with CS, H-GASS, and F-GASS and measured the overall synopsis error using \mathcal{E}_{AVG} . The results for a memory bound of 5% of the data size can be seen in Figure 9. As for the synthetic dataset, for a small number of groups, the three approaches produce almost equal results: $\mathcal{E}_{AVG} = 3.85\%$ for CS, and $\mathcal{E}_{AVG} = 3.82\%$ for the RSD-based approaches, respectively. For a larger number of groups (or group-by attributes), however, the differences get clearer. For all settings, both H-GASS and F-GASS cause smaller estimation errors than CS; moreover, the average estimation error of F-GASS is even lower than for H-GASS.

The fraction of missing groups was at most 7% for CS, 11% for H-GASS and 12% for F-GASS, which is considerably small for the large number of groups.

Table 4: Number of groups in the real-world dataset for different numbers of group-by columns

Group-by columns	Groups
1	77
2	1,994
3	16,627
4	79,597

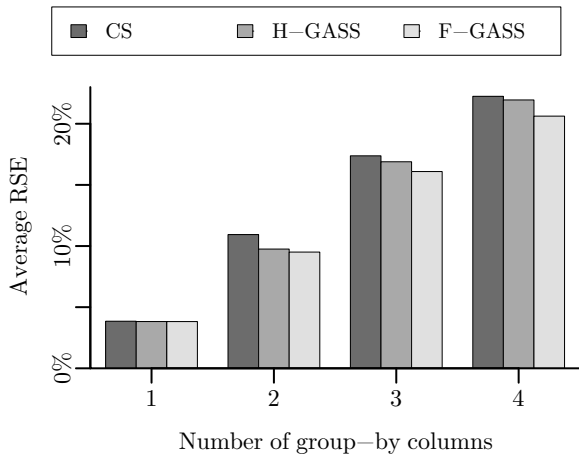


Figure 9: Impact of number of group-by attributes

Memory bound. As for the synthetic dataset, we evaluated the impact of the memory bound on the synopsis quality for the real-world dataset. We computed synopses with CS, H-GASS and F-GASS for memory bounds between 1% and 10%. The results for the \mathcal{E}_{AVG} measure are shown in Figure 10. The general result is as in the previous experiments. One difference compared to the synthetic dataset (see Figure 6) is that the difference in the average RSE increases for decreasing sample sizes.

We also conducted experiments with the outlier extension. However, for this real-world dataset, the outlier extension was not able to reduce the overall estimation error. In order to find the reason, we have to analyze the distribution of the aggregation attribute values for the individual groups.

Summary. Our experiments on the real-world data set of a market research company verified the results from the experiments on the synthetic datasets. Again, the RSD-based techniques allowed low estimation errors for the quickly constructible F-GASS synopses.

Summing up, the presented results on both synthetic and real-world datasets emphasized the benefit of using RSD-based group-aware sample synopses. For fast construction times and low average estimation errors, F-GASS is the technique of choice, whereas systems with sufficient resources for complex pre-processing of the synopses may benefit from the low maximum estimation errors of H-GASS.

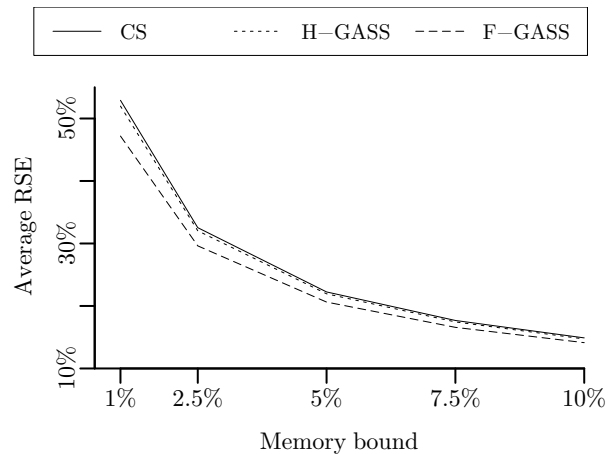


Figure 10: Impact of the memory bound

7. RELATED WORK

In the field of databases, there are a multitude of sampling techniques for approximate query processing. Especially, queries typical in OLAP environments are in the focus of optimization. Relevant query types are aggregation queries [8, 23], queries with foreign-key joins [3, 14], or group-by queries.

For group-by queries, Babcock et al. proposed *Small Group Sampling* [4]: With Small Group Sampling (SGS), a simple random sample of the base data is drawn. Additionally, all single-attribute groups that are smaller than a specific threshold are stored in their entirety (small group tables). During query execution, the approximate result is dynamically combined from these samples to produce highly-accurate approximate query answers. However, the proposed synopsis design induces high space consumption: It may build a lot of small group tables. Furthermore, tuples can be stored multiple times within the synopsis leading to low space effectiveness and processing overhead. Moreover, SGS under-represents groups that are slightly larger than the threshold as well as all small multi-attribute groups whose single-attribute subgroups are large, e.g. sells of snowblowers in California. Another solution proposed by Archarya et al. is *Congressional Sampling* [1], where the space assignment is based on the Senate and House of the American Congress. Congressional Sampling (CS) considers all groups in the data and, thus, provides general-purpose synopses. We adopted the idea of CS for the hierarchical approach of our group-aware sample synopses. Finally, contrary to our solution, both techniques, [4] and [1], focus on the size of the group but not on the variation (i.e. relative standard deviation) within the group.

In the sampling literature in general, biased sampling, and thus, stratified sampling, has been studied under many contexts [11]. Our proposed techniques can be compared with stratified sampling or subpopulation sampling, where the population is segmented into strata or subsets which correspond to the groups in our scenario.

8. SUMMARY

In this paper, we introduced RSD-based group-aware sample synopses. The goal was to propose general-purpose memory-bounded sample synopses that represent all groups of all possible group-by queries preferably well. The result is a biased sampling scheme that computes the sample sizes of the individual groups based on their variations. The hierarchical algorithm considers all possible groupings for a given set of group-by attributes. We further proposed a simplified (flat) algorithm that significantly reduces the construction cost. Both approaches can decrease the estimation error compared to existing techniques; the former has a larger impact on the maximum error over all groups, the latter on the average error. An additionally proposed extension considers extreme values (outliers) within the groups and allows for significantly lower estimation errors by special outlier handling.

Our next steps include the considerations of alternative weight functions to plug into the algorithmic framework, like a hybrid approach that considers both the variation and the size of the group. The goal is to reduce both the estimation error and the number of missing groups. Additionally, we want to examine how our approaches can be combined with Linked Bernoulli Synopses [14] in order to provide schema-level group-aware sample synopses. And further, we want to integrate our approaches into Derby/S [19]—a database system that extends Apache Derby with approximate query processing techniques—in order to evaluate their effectiveness in more realistic environments.

9. REFERENCES

- [1] S. Acharya, P. Gibbons, and V. Poosala. Congressional Samples for Approximate Answering of Group-By Queries. In *SIGMOD*, pages 487–498, 2000.
- [2] S. Acharya, P. B. Gibbons, V. Poosala, and S. Ramaswamy. The aqua approximate query answering system. In *SIGMOD*, pages 574–576, 1999.
- [3] S. Acharya, P. B. Gibbons, V. Poosala, and S. Ramaswamy. Join Synopses for Approximate Query Answering. In *SIGMOD*, pages 275–286, 1999.
- [4] B. Babcock, S. Chaudhuri, and G. Das. Dynamic Sample Selection for Approximate Query Processing. In *SIGMOD*, pages 539–550, 2003.
- [5] P. Brown and P. Haas. BHUNT: Automatic Discovery of Fuzzy Algebraic Constraints in Relational Data. In *VLDB*, pages 668–679, 2003.
- [6] J. Brutlag and T. Richardson. A block sampling approach to distinct value estimation. Technical report, University of Washington, Department of Statistics, 2000.
- [7] K. Chakrabarti, M. N. Garofalakis, R. Rastogi, and K. Shim. Approximate Query Processing Using Wavelets. In *VLDB*, pages 111–122, 2000.
- [8] S. Chaudhuri, G. Das, M. Datar, and R. M. V. Narasayya. Overcoming Limitations of Sampling for Aggregation Queries. In *ICDE*, pages 534–544, 2001.
- [9] S. Chaudhuri, G. Das, and V. Narasayya. A Robust, Optimization-Based Approach for Approximate Answering of Aggregate Queries. In *SIGMOD*, pages 295–306, 2001.
- [10] S. Chaudhuri, G. Das, and U. Srivastava. Effective Use of Block-level Sampling in Statistics Estimation. In *SIGMOD*, pages 287–298, 2004.
- [11] W. Cochran. *Sampling Techniques*. Wiley Series in Probability & Mathematical Statistics. John Wiley & Sons, 3rd edition, 1977.
- [12] D. DeWitt, J. Naughton, D. Schneider, and S. Seshadri. Practical Skew Handling in Parallel Joins. In *VLDB*, 1992.
- [13] V. Ganti, M. Lee, and R. Ramakrishnan. ICICLES: Self-Tuning Samples for Approximate Query Answering. In *The VLDB Journal*, pages 176–187, 2000.
- [14] R. Gemulla, P. Rösch, and W. Lehner. Linked Bernoulli Synopses: Sampling Along Foreign-Keys. In *SSDBM*, pages 6–23, 2008.
- [15] I. Ilyas, V. Markl, P. Haas, P. Brown, and A. Aboulnaga. CORDS: Automatic Discovery of Correlations and Soft Functional Dependencies. In *SIGMOD*, pages 647–658, 2004.
- [16] Y. E. Ioannidis and V. Poosala. Histogram-Based Approximation of Set-Valued Query-Answers. In *VLDB*, pages 174–185, 1999.
- [17] C. Jermaine. Robust Estimation With Sampling and Approximate Pre-Aggregation. In *VLDB*, pages 886–897, 2003.
- [18] T. Johnson, S. Muthukrishnan, and I. Rozenbaum. Sampling Algorithms in a Stream Operator. In *SIGMOD*, pages 1–12, 2005.
- [19] A. Klein, R. Gemulla, P. Rösch, and W. Lehner. Derby/S: A DBMS for Sample-Based Query Answering (Demo). In *SIGMOD*, pages 757–759, 2006.
- [20] G. Kollios, D. Gunopulos, N. Koudas, and S. Berchtold. An Efficient Approximation Scheme for Data Mining Tasks. In *ICDE*, pages 453–462, 2001.
- [21] Y. Matias, J. S. Vitter, and M. Wang. Wavelet-Based Histograms for Selectivity Estimation. In *SIGMOD*, pages 448–459, 1998.
- [22] V. Poosala, Y. E. Ioannidis, P. J. Haas, and E. J. Shekita. Improved Histograms for Selectivity Estimation of Range Predicates. In *SIGMOD*, pages 294–305, 1996.
- [23] P. Rösch, R. Gemulla, and W. Lehner. Designing Random Sample Synopses with Outliers. In *ICDE*, pages 1400–1402, 2008.
- [24] H. Toivonen. Sampling Large Databases for Association Rules. In *VLDB*, pages 134–145, 1996.
- [25] J. Vitter. Random Sampling with a Reservoir. *ACM Trans. Mathematical Software*, 11(1):37–57, 1985.