Approximating Multidimensional Range Counts with Maximum Error Guarantees

Michael Shekelyan
University of Warwick
Coventry, United Kingdom
michael.shekelyan@warwick.ac.uk

Anton Dignös, Johann Gamper
Free University of Bozen-Bolzano
Bozen-Bolzano, Italy
{*dignoes, gamper*}*@inf.unibz.it*

Minos Garofalakis
ATHENA Research Center, Greece
Technical University of Crete, Greece
minos@athenarc.gr

***Abstract*—We address the problem of compactly approximating multidimensional range counts with a guaranteed maximum error and propose a novel histogram-based summary structure, termed SliceHist. The key idea is to operate a grid histogram in an approximately rank-transformed space, where the data points are more uniformly distributed and each grid slice contains only a small number of points. Then, the points of each slice are summarised again using the same technique. As each query box partially intersects only few slices and each grid slice has few data points, the summary is able to achieve tight error guarantees. In experiments and through analysis of non-asymptotic formulas we show that SliceHist is not only competitive with existing heuristics in terms of performance, but additionally offers tight error guarantees.**

1. Introduction

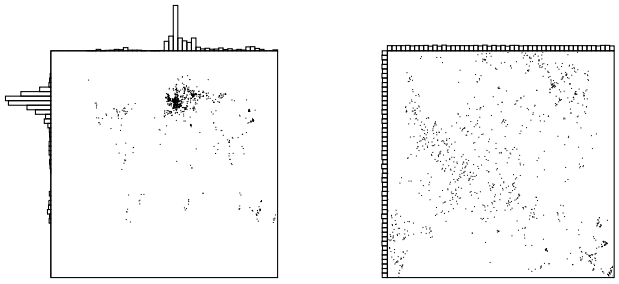
Motivation. Producing accurate selectivity estimates for multidimensional range queries based on compact summary structures (or synopses) plays a crucial role in relational query optimization, query profiling, and approximate query processing [1]. Cost-based query optimizers employ such synopses to obtain accurate estimates of intermediate result sizes. Similarly, query profilers and approximate query processors require compact data synopses in order to provide users with fast and useful feedback on their original query. Such query feedback allows OLAP and data-mining applications to identify the truly interesting regions of the data and to focus their explorations quickly and effectively.

Since it is difficult to determine how well a summary technique works for arbitrary queries and datasets, we focus on multidimensional summary approaches with ε-error guarantees. Such approaches can construct for any dataset of size n and dimensionality d, a predictably small summary that can bound the true count T of any query box to a range $L \leq T \leq L + \varepsilon n$ for some lower bound $L \leq T$. While it is defined as an additive error guarantee, it implies a multiplicative error (q-error [2]) guarantee for larger counts $T \geq 2\varepsilon n$, in which case $L \geq \varepsilon n$ and $(L + \varepsilon n)/L \leq 2$. This means that summaries with ε-error guarantees have at most multiplicative error 2 for all query boxes with at least $2\varepsilon n$ points.

One can derive from the Chernoff-Hoeffding bounds (see also [3], [4]) that a simple random sample of size $\frac{2}{\varepsilon^2}(2d \ln n + \ln \frac{2}{1-p})$ of a d-dimensional multiset of n points will satisfy the aforementioned ε-error guarantees with probability p. While confidence intervals for the correct count can be derived from a random sample, it does not yield tight *deterministic* bounds and the question remains which summaries we should construct to predictably obtain such bounds. More sophisticated and deterministic sampling techniques have been studied (from a more theoretical point of view) in the computational geometry [5], [6], streaming [7], [8] and discrepancy theory [9], [10], [11], [12] literature. In these works, sampling-based summary approaches, termed *ε-approximations*, with the lowest asymptotic sample size are sought after. The best asymptotic space bound of $O_d(\frac{1}{\varepsilon}\log^{2d-2}(\frac{1}{\varepsilon})\, poly(\log\log(\frac{1}{\varepsilon}))$ is currently achieved by [7], [5], where the subscript d in $O_d(\ldots)$ indicates that d is treated as a small constant. While the approaches scale theoretically very well with ε, practical efficiency of deterministic sample construction remains a major concern: existing techniques [7], [6], [8], [11], [12], [5] need at least 10^{24} arithmetic operations to construct samples with ε-error guarantees ($\varepsilon \leq 1\%$), rendering them astronomically expensive even for simple problem instances.

In contrast, the database literature [1], [13], [14], [15], [16], [17], [18], [19] focused more on multidimensional summaries that empirically appear to work well, but do not possess any formal error guarantees. One exception are multidimensional equi-depth histograms [20], which with $O_d(\frac{1}{\varepsilon^d})$ buckets can be shown to achieve ε-error guarantees. A better histogram-based technique, termed *dyadic histograms*, can be found in the streaming literature [7]. A dyadic histogram is basically an approximate range tree [21] which does not store any points and omits the lower levels of the tree. It achieves ε-error guarantees in $O_d(\frac{1}{\varepsilon}\log^{2d-2}\frac{\log(\frac{1}{\varepsilon})}{\varepsilon})$ space. Similar to range trees, dyadic histograms work best for two-dimensional data and begin to become impractically large and slow to construct in even just three dimensions.

Our Contributions. To improve ε-approximation performance in higher dimensions, we propose a histogram-based summary structure, termed *SliceHist*. The basic idea is to operate multidimensional grid (i.e., equi-width) histograms in a transformed space, where each slice of a regular grid contains a bounded number of points. A SliceHist summary



(a) original data (b) rank transformation

Figure 1: Rank transformation for data summarization.

transforms both data points and queries using the same count-preserving data transformation, which is a variant of the *rank transformation*, a well-known tool in statistics and computational geometry. In statistics (cf. Sklar's theorem), the rank transformation allows *any* multivariate distribution to be captured by a multivariate distribution (copula) with *uniform marginals* [22]. A direct implication of uniform marginals is that the data space cannot be arbitrarily skewed, e.g., with arbitrary dense or empty areas. This is illustrated in Figure 1, where data points are clearly spread more uniformly in the rank-transformed space. Another interesting property is that independently-distributed dimensions translate to (multidimensional) uniformity in the rank-transformed space. For instance, a data set's mutual information (an independence measure) can be computed from the entropy (a uniformity measure) of the copula space (statistical equivalent to rank space) [23]; that is, independence between data dimensions is quantified through uniformity in the rank-transformed space.

In this work, we employ the rank transformation as a building block for ε-approximate data summarization. A 1-level SliceHist is simply an *equi-width histogram* of the rank-transformed data, i.e., a uniform-grid histogram defined by splitting each dimension of the rank space into intervals of equal size (depending on the desired error ε). A *k-level SliceHist* is defined recursively, and comprises a 1-level SliceHist G along with a $(k-1)$-level SliceHist for *each slice* (i.e., "row", "column", etc.) of the grid histogram G in *all dimensions*. As our analysis demonstrates, this simple recursive ε-approximation construct can achieve $O(\frac{1}{\varepsilon^\lambda}\log(\frac{1}{\varepsilon}))$ storage costs for any constant $\lambda > 1$. However, this requires a large number of levels k, which also implies large constants (independent of ε) hidden in the O-notation. While this is a useful reminder that low asymptotic complexity is not necessarily indicative of performance in practice, we feel that it also provides a very interesting theoretical result offering new insights into the complexity of the ε-approximation problem. Fortunately, for $k = 2$, SliceHist offers a near-linear storage complexity of $O(\varepsilon^{-d/(2-\frac{1}{d})})$ in low to medium dimensionalities, outperforms existing quantile-based ε-approximations in practice, and can even compete with state-of-the-art heuristic solutions, such as DigitHist histograms [24] (which, of course, cannot offer any ε-approximation guarantees).

For any reasonable space budget (smaller than an Exabyte), SliceHist achieves the best ε-error guarantees, which can be seen from the numerical analysis of the exact formulas (cf. Table 1). If the space budget was allowed to be astronomically large, dyadic histograms could achieve better ε-error guarantees. Both dyadic histograms and SliceHist can be constructed in a small number of data passes using approximate quantile summaries, but dyadic histograms need to perform $O_d(\log^{d-1}\frac{1}{\varepsilon})$ more insertions and binary searches per point, which makes their construction vastly more costly. Unlike dyadic histograms, SliceHist offers a logarithmic query time for any number of dimensions.

Our technical contributions can be summarized as:

- We propose SliceHist, a novel histogram summary for multidimensional data with ε-error guarantees, based on δ-rank transformations and equi-width histograms.
- We formally analyze SliceHist, demonstrating that it offers logarithmic query times and log-linear construction times in any number of dimensions.
- We show that SliceHist achieves the tightest ε-error guarantees for a given space budget and experimentally competes head-to-head with state-of-the-art heuristic approaches that lack ε-error guarantees.

2. Count-Preserving δ-rank Transformation

At the core of SliceHist is to transform the data into a more convenient space for data summarisation. As an exact rank transformation would require space linear in the size of the data points, we propose to utilize an δ-approximate rank transformation that can be obtained with the help of quantile summaries [25], lexicographic ordering and interpolating between available quantiles.

Definition 1 (δ-rank transformation)**.** *A function $\phi : \mathbb{R}^d \to [0,1]^d$ is a δ-rank transformation of a d-dimensional dataset $\mathcal{D}$ if and only if it satisfies the following conditions:*

i.) *for any point $p \in \mathcal{D}$ and any box $(s \times t) \in \mathbb{R}^d \times \mathbb{R}^d$ it holds that $p \in (s \times t) \Leftrightarrow \phi(p) \in (\phi(s) \times \phi(t))$;*

ii.) *for any dimension i and interval $[a,b]$ along that dimension it holds that $(b-a) - \frac{\delta}{2} \leq \frac{|\{p \in \mathcal{D} \mid a \leq \phi(p)_i \leq b\}|}{|\mathcal{D}|} \leq (b-a) + \frac{\delta}{2}$.*

A δ-rank transformation maps any point to a point in a unit space where each dimension is a normalized rank. By transforming any two opposite corners of boxes, one can also obtain transformed boxes in the normalized rank space. The first property of a δ-rank transformation ensures that for any point inside (resp. outside) a box, the transformed point is also inside (resp. outside) the transformed box. The second property quantifies the precision of the approximate ranks in each dimension. Ideally, an interval covering half of the space should contain exactly $\frac{1}{2}$ of the ranks, but instead it is allowed to contain between $\frac{1-\delta}{2}$ and $\frac{1+\delta}{2}$ of the ranks, where $\delta \in [0,1]$ is the approximation precision.

Next, we describe how a δ-rank transformation storing $O(\frac{d}{\delta})$ values can be obtained using quantile summaries [25].

Definition 2 (quantile-based δ-rank transformation)**.** *Let $\mathcal{D}$ be a d-dimensional dataset and $\mathcal{S}_i$ denote a sequence of points $q_1, \ldots, q_m$ representing the approximate quantiles in dimension i for $1 \leq i \leq d$, where $m = \lceil \frac{3}{\delta} \rceil$ and q_j is the $\frac{\delta}{6}$-approximate quantile of the rank $\frac{j}{m}$ along dimension i. Then the transformation function is defined as $\phi(p) = (\phi(p)_1, \ldots, \phi(p)_d)$ with*

$$\phi(p)_i = \begin{cases} \frac{j}{m} & p = q_j \in \mathcal{S}_i \\ \frac{j}{m} + (\frac{k}{m} - \frac{j}{m}) \frac{p_i - ((q_j)_i - \Delta)}{((q_k)_i + \Delta) - ((q_j)_i - \Delta)} & p \notin \mathcal{S}_i \end{cases}$$

where $0 < \Delta < 1$ is a real-valued constant and q_j, q_k are the closest neighbours of p in the total order over $\mathcal{D} \cup \{p\}$ in dimension i s.t. $q_1 <_i \ldots <_i q_j <_i p <_i q_k <_i \ldots <_i q_m$ where $<_i$ orders points first by dimension i and then lexicographically by the remaining dimensions.

A quantile-based transformation builds a $\frac{\delta}{6}$-approximate quantile summary for each dimension to obtain the quantiles $[q_1, \ldots, q_m]$ at the ranks $\{\frac{1}{m}, \ldots, \frac{m}{m}\}$ with $m = \lceil \frac{3}{\delta} \rceil$. It then (approximately) answers a rank query for a point p by finding the neighbouring quantiles in each dimension and reporting the interpolated rank between those two quantiles, except when the point lies on a quantile, in which case the quantile's rank is reported instead.

Such a transformation satisfies the δ-approximation property, because an interval $[a, b]$ in the exact rank space contains around $(b-a)$ ranks and the quantile summaries can displace at most a fraction of $\frac{\delta}{6} + \frac{\delta}{3} = \frac{\delta}{2}$ ranks into (or out of) the interval $[a, b]$. It also satisfies the count preservation property of δ-rank approximations, because the interpolation preserves the order in each dimension and does not allow values to coincide with ranks of neighbouring quantiles. The latter property is achieved through a constant Δ close to 0, e.g., $\Delta = 10^{-6}$ (choosing a small Δ is advantageous due to finite numerical precision, but in principle any value greater than 0 and smaller than 1 is valid).

3. SliceHist: ε-Approximate Grid Histogram

The core idea of SliceHist is to first transform the data (using an approximate rank transformation) into a space where the data is more uniformly distributed, and then to construct a grid histogram in the transformed space (1A and 1B in Figure 2). This requires at least two scans over the data. In the first pass an approximate quantile summary for each dimension is constructed for the approximate rank transformation. In the second pass the points are rank-transformed before they are passed on to the grid histogram. To improve precision, each grid slice of the grid histogram can be summarised again (2A and 2B in Figure 2), which will require further passes over the data, one of which can be done in parallel with the second pass.

To query the SliceHist summary, the corners of the query rectangle Q are first transformed using the same rank transformation, yielding a transformed query box R (1B in Figure 3). Then, the precise count for the grid cells that are completely covered by the transformed box R can be retrieved, whereas the edges of the query are passed to the next level of SliceHist (2B in Figure 3).

3.1. Definition

We start by introducing 1-*level SliceHist*, a simple histogram summary that combines a δ-rank transformation with a multi-dimensional equi-width partitioning. It achieves ε-error guarantees by limiting the number of points in each slice of a grid. A slice is a generalization of rows and columns to more dimensions, i.e., all the grid cells that share the same coordinate in some dimension.

Definition 3 (1-level ε_1-SliceHist)**.** *A* 1-level ε_1-SliceHist*, $SH(\mathcal{D}) = (\phi, G)$, of a multiset of points $\mathcal{D}$ is a δ-rank transformation ϕ of $\mathcal{D}$, paired with a d-dimensional regular grid histogram G over the transformed points $\mathcal{D}_\phi$. The parameter ε_1 determines that the transformation function ϕ has approximation parameter $\delta = (\frac{\alpha-1}{\alpha}\varepsilon_1)$ and G has a grid resolution of $\lceil \frac{\alpha}{\varepsilon_1} \rceil$ slices per dimension. The real $1 < \alpha \leq 2$ is chosen such that the total number of summary values $d^2 \lceil (\frac{\alpha}{\alpha-1}) \frac{3}{\varepsilon_1} \rceil + \lceil \frac{\alpha}{\varepsilon_1} \rceil^d$ is minimized.*

A 1-level SliceHist operates a grid histogram in a δ-rank transformed space. It transforms each data point, determines in which grid cell the transformed point is contained, and increments the count of that cell. In order to answer queries over the original space, it transforms the end points of the query box and then uses the grid histogram to answer the query. By choosing the right values for δ and the grid resolution, it can be guaranteed that at most $\varepsilon_1 n$ transformed points fall into each slice of the grid. The variable $\alpha > 1$ allows to balance the precision of the grid histogram and the δ-rank transformation. While it is difficult to express the optimal value for α analytically, for a given data dimensionality d and ε_1 it is very easy to numerically find the optimal value that minimises the number of summary values comprised of $d \lceil (\frac{\alpha}{\alpha-1}) \frac{3}{\varepsilon_1} \rceil$ quantile points (from ϕ) and $\lceil \frac{\alpha}{\varepsilon_1} \rceil^d$ grid counts (from G).

Example 1. *In Figure 2, the gray parts of 1A and 1B form the 1-level ε-SliceHist (ϕ, G) for a dataset $\mathcal{D}$. The transformation function ϕ is represented by two one-dimensional histograms with 20 buckets in 1A (acting as stand-ins for quantile summaries that are difficult to visualize), and the two-dimensional grid histogram G in 1B with 5 slices along each dimension. Notice that the transformed data is more uniformly spread. To better illustrate the transformation, three random points are depicted as a triangle, diamond and square. For instance, the triangle point is transformed from $(0.47, 0.48)$ to its approximate rank $(0.26, 0.41)$, i.e., according to the histograms 26% of the points have lower x-coordinates and 41% lower y-coordinates.*

Lemma 1. *Let G be the grid histogram of a 1-level ε_1-SliceHist summarising n points. Each grid slice of G counts at most $\varepsilon_1 n$ points.*

Proof. Due to the second property in Definition 1 and $\delta = \frac{\alpha-1}{\alpha}\varepsilon_1$, each slice of width w in the transformed space

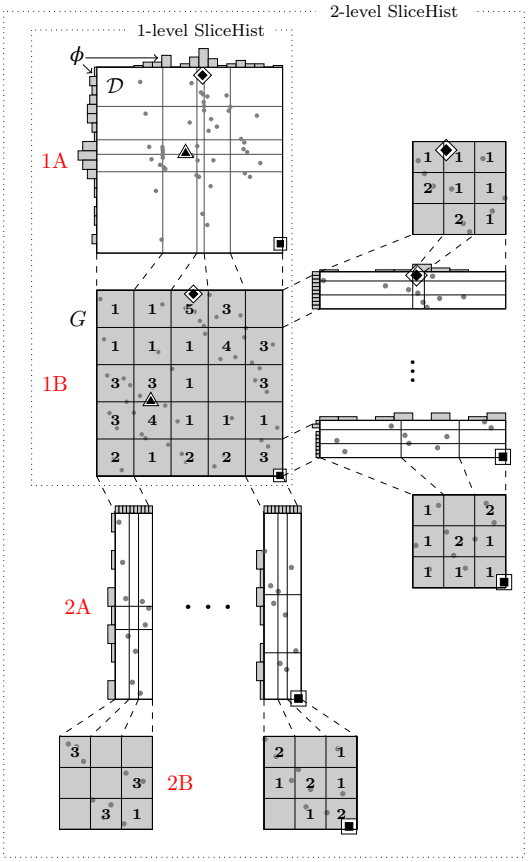


Figure 2: 2-Level SliceHist summary comprised of eleven 1-level SliceHist summaries (only five are depicted): the 1-level SliceHist of the original data $\mathcal{D}$ (1A and 1B) and a 1-level SliceHist for each slice (row and column) in the grid G of the ϕ-transformed data.

contains at most $(w + \frac{\alpha-1}{\alpha}\varepsilon_1)n$ points. A grid resolution of $\lceil \frac{\alpha}{\varepsilon_1} \rceil$ slices per dimension ensures that each slice has at most width $w = \frac{\varepsilon_1}{\alpha}$. Taken together it follows that each slice contains at most $\frac{1}{\alpha}\varepsilon_1 n + \frac{\alpha-1}{\alpha}\varepsilon_1 n = \varepsilon_1 n$ points. □

Lemma 2. *A 1-level $(\frac{\varepsilon_1}{2d})$-SliceHist of a point multiset $\mathcal{D}$ can determine the count of any box query with additive error $\leq \varepsilon_1|\mathcal{D}|$.*

Proof. In order to summarise a multiset of points, it first transforms each point using ϕ and then counts the number of transformed points along a regular grid. Box queries of the shape $(s \times t)$ can then be answered by posing the query $(\phi(s) \times \phi(t))$ to the grid histogram. Each box query partially intersects at most 2 slices of the grid histogram per dimension, and each slice of the grid contains at most $\varepsilon_1|\mathcal{D}_\phi| = \varepsilon_1|\mathcal{D}|$ points (cf. Lemma 1). □

Lemma 3. *Any 1-level $(\frac{\varepsilon_1}{2d})$-SliceHist has $O(\frac{d^3+d^d}{\varepsilon_1^d})$ summary values.*

Proof. As α is chosen such that it minimises the number of summary values, an arbitrary choice of α is an upper bound for the actual summary size. Thus, picking $\alpha = 2$, we obtain that the number of summary values is at most $d^2\lceil\frac{6d}{\varepsilon_1}\rceil + \lceil\frac{2d}{\varepsilon_1}\rceil^d = O(\frac{d^3}{\varepsilon_1} + \frac{d^d}{\varepsilon_1^d})$. □

If we create a 1-level ε_1-SliceHist of the original data and summarise each slice of its grid histogram again with a 1-level ε_2-SliceHist, we obtain a summary with ε-error of $(2d)(2d-1)\varepsilon_1\varepsilon_2 n$. This is because each slice at the lowest level contains at most $\varepsilon_1\varepsilon_2 n$ points and at most $(2d)(2d-1)$ such slices are partially intersected by a query. The number $(2d)(2d-1)$ of partially intersected slices is derived as follows: at the top-level the query partially intersects $2d$ slices, and each of the $2d$ sub-queries along each slice is passed on to the bottom-level, where each sub-query intersects at most $2d-1$ slices (it is one slice less because the sub-queries always completely contain the slice on one side, as the sub-query "hugs" that side). For the choice $\varepsilon_1 = \varepsilon_2 = \frac{\sqrt{\varepsilon}}{(2d)(2d-1)}$ we only have to store on the top level one 1-level ε_1-SliceHist with $O(\frac{1}{\sqrt{\varepsilon^d}})$ values and on the lower level a 1-level ε_2-SliceHist with $O(\frac{1}{\sqrt{\varepsilon^d}})$ values for each of the $O_d(\frac{1}{\sqrt{\varepsilon}})$ top-level slices, which totals to $O(\frac{1}{\sqrt{\varepsilon^{d+1}}})$ values.

If we would recursively continue this idea, we would obtain for k levels and $\varepsilon_1 = \ldots = \varepsilon_k = \frac{\sqrt[k]{\varepsilon}}{(2d)(2d)^{k-1}}$ a summary with ε-error and size $O_d(\frac{1}{\sqrt[k]{\varepsilon^{d+k-1}}})$. If we would like to achieve $O_d(\frac{1}{\varepsilon^\lambda})$ for an arbitrarily small λ, we could pick the smallest k such that $\frac{d+k-1}{k} \leq \lambda$, i.e., $k = \lceil \frac{d-1}{\lambda-1} \rceil$.

This asymptotic analysis omits constants that grow exponentially with k, such that in practice a 1-level or 2-level SliceHist has generally the smallest size (and in some rarer cases a 3-level SliceHist).

Definition 4 (k-level $[\varepsilon_1, \ldots, \varepsilon_k]$-SliceHist)**.** *A k-level $[\varepsilon_1, \ldots, \varepsilon_k]$-SliceHist of a data set $\mathcal{D}$ with $k > 1$ is a recursive data structure comprised of a 1-level ε_1-SliceHist $SH(\mathcal{D}) = (r, G)$ and, for each slice of the grid histogram G, a $(k-1)$-level $[\varepsilon_2, \ldots, \varepsilon_k]$-SliceHist of the points in the slice.*

The $\varepsilon_1, \ldots, \varepsilon_k$ values specify the precision for each level of the hierarchical structure, where ε_1 specifies the precision of the 1-level SliceHist at the top level. Lemma 7 shows how to choose the optimal values for $\varepsilon_1, \ldots, \varepsilon_k$, given an error guarantee ε.

Adding a recursive structure to SliceHist benefits it more than simply increasing the grid resolution and thereby enables more precise and compact summaries.

Example 2. *Figure 2 shows a 2-level SliceHist (gray parts). It is comprised of a 1-level SliceHist on the original data (gray parts in 1A and 1B) and, for each slice in the grid histogram G, another 1-level SliceHist that summarizes the data points in the corresponding slice. For instance, the gray parts in 2A and 2B represent the 1-level SliceHist of the first vertical slice. In total, this yields ten sub-summaries,*

five in each dimension. For the ε-rank transformations, one-dimensional histograms are used with 20 buckets in 1A and 10 buckets in 2A.

3.2. Constructing SliceHist

The construction of a SliceHist is outlined in Algorithm 1. In the initialization step, the optimal parameters are determined to minimize the summary size formula from Theorem 1. From a numerical evaluation of Theorem 1, we know that the optimal choice of k is between 1 and 4 for summary sizes below one exabyte. We therefore try out all four values and pick the best one. Once we fix a $k \in \{1,2,3,4\}$ we get specific values for $\varepsilon_1,\ldots,\varepsilon_k$ according to Lemma 7 and can determine the arbitrarily close to optimal $\alpha_1,\ldots,\alpha_k$ parameters. As the optimal choice of $\alpha_j,\ldots,\alpha_k$ is independent of $\alpha_1,\ldots,\alpha_{j-1}$, first α_k is determined, then α_{k-1} and so on, until α_1 is computed. The next step is to construct a 1-level SliceHist. For this, line 6 constructs a δ_1-rank transformation for the multiset of points $\mathcal{D}$. In line 7, the grid histogram G_1 is constructed by transforming the data points and counting the transformed points (1B in Figure 2). Finally, the algorithm iterates in a breadth-first manner through all levels from 2 to k and constructs a $(k-1)$-level SliceHist for each slice in each dimension of the grid histogram G.

Algorithm 1: SLICEHIST CONSTRUCTION

Input: Data set $\mathcal{D}$, precision ε
Result: k-level $[\varepsilon_1,\ldots,\varepsilon_k]$-SliceHist for $\mathcal{D}$

1 Repeat lines 2–4 for $k \in \{1,\ldots,4\}$ and use k, s.t., the expression in line 4 is minimized.
2 Compute $[\varepsilon_1,\ldots,\varepsilon_k]$ from ε and k according to Lemma 7;
3 **foreach** $\ell \in \{k,\ldots,1\}$ **do**
4 Find $1 < \alpha_\ell \leq 2$ minimising $\sum_{i=\ell}^{k} \left(\prod_{j=1}^{i-1} d\left\lceil \frac{\alpha_i}{\varepsilon_j}\right\rceil\right)\left(\left\lceil\frac{\alpha_i}{\varepsilon_i}\right\rceil^d + d^2\left\lceil\frac{\frac{\alpha_i}{\alpha_i-1}^3}{\varepsilon_i}\right\rceil\right)$
5 Let $\delta_1 = \frac{\alpha_1-1}{\alpha_1}\varepsilon_1$;
6 $\phi_1 \leftarrow \delta_1$-rank transformation of $\mathcal{D}$;
7 $G_1 \leftarrow$ grid histogram of $\mathcal{D}_{\phi_1}$ with $\lceil\frac{\alpha_1}{\varepsilon_1}\rceil^d$ cells;
8 $SH \leftarrow (\phi_1, G_1)$;
9 **foreach** *level* $\ell \in \{2,\ldots,k\}$ **do**
10 Let $\delta_\ell = \frac{\alpha_\ell-1}{\alpha_\ell}\varepsilon_\ell$;
11 **foreach** *grid histogram G on level $\ell-1$* **do**
12 **foreach** *slice S of G* **do**
13 Let X be the transformed data points in S;
14 $\phi \leftarrow \delta_\ell$-rank transformation of X;
15 $G \leftarrow$ grid histogram of X_ϕ with $\lceil\frac{\alpha_\ell}{\varepsilon_\ell}\rceil^d$ cells;
16 $SH \leftarrow SH \circ (r, G)$;
17 **return** SH;

Lemma 4. *The SliceHist summary structure with k levels of a data set with n points is constructed in $O(d^{k-1}n\log(n))$ time in $(k+1)$ passes over the data.*

Proof. In the first data scan, each point is added to the rank transformation of the 1-level grid histogram. In the j-th data scan, each point is rank transformed $(j-1)$ times and counted by d^{j-2} different $(j-1)$-level grid histograms, and if $j < k$ it is additionally added to d^{j-1} rank transformations of j-level grid histograms. Let $A = O(\log(n))$ be the cost of adding a point to a rank transformation, $R = O(\log(n))$ be the cost of rank transforming a point and $C = O(1)$ be the cost of adding the point to the (equi-width) grid histogram. Then, the construction costs are equal to $n(\sum_{j=2}^{k}(j-1)R) + n(\sum_{j=1}^{k-1} d^{j-1}A) + n(\sum_{j=2}^{k} d^{j-2}C)$, which is in $O(d^{k-1}n\log(n))$. □

3.3. Querying SliceHist

The SliceHist summary structure for a d-dimensional data set is queried by first rank transforming the query and then splitting the rank-transformed query rectangle into $(2d+1)$ parts. This is illustrated in Figure 3. Hatching indicates different parts of the query range, which are processed by different parts of SliceHist. The query in the original data space is shown on the bottom right, the rank-transformed query in the rank space on the top left. One part of the rank-transformed query range is grid-aligned (union of grid cells) in the grid histogram G of the current level (gray area on the top left). The remaining $2d$ parts are passed on to the next level, i.e., the twice rank-transformed space (2B).

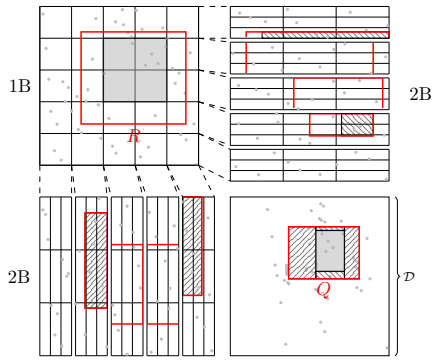


Figure 3: Query split and transformation.

Querying SliceHist is outlined in Algorithm 2. First, the query box is rank transformed by transforming its two corner points (line 2). Then, the query range R in the rank-transformed space is conceptually split into $2d+1$ parts. One part is grid-aligned with the higher-level histogram (processed in line 12). The other $2d$ parts are processed in the loop at line 4. It iterates through all dimensions and processes in each iteration the two slices that contain R_{min} and R_{max} in that dimension (A and B in line 5). If the SliceHist summary has $k > 1$, the algorithm proceeds recursively to compute the estimates of $A \cap R$ and $B \cap R$ using the $k-1$-SliceHist (line 7). When the last level of the summary is reached, the queries are eventually answered by querying the grid histogram (lines 9 and 10). The counts of the grid histograms are stored as cumulative sums (or prefix sums/cubes) to enable constant-time querying. In line 11, the processed parts of the query are subtracted from the query range R. After terminating this loop, R contains the grid-aligned part of the query range, which is processed on the grid histogram of the 1-level SliceHist (line 12).

Algorithm 2: SLICEHIST QUERYING

Input: k-level $[\varepsilon_1, \ldots, \varepsilon_k]$-SliceHist SH and query range $(s \times t)$
Output: approximate count of points C in box spanned by s and t

```
Let G be the 1-level grid histogram of SH at the top-level;
R ← (φ(s) × φ(t));
C ← 0;
foreach dimension i do
    Let A and B be the slices of G containing, resp., φ(s) and φ(t);
    if k > 1 then
        s ← s + μ_A(A ∩ R) + μ_B(B ∩ R), where μ_A and μ_B are
          estimates of the (k−1)-SliceHist of A and B, respectively;
    else
        Let C and D be the grid-aligned bounding box of,
          respectively, A ∩ R and B ∩ R in G;
        C ← C + (vol(A∩R)/vol(C)) · μ(C) + (vol(B∩R)/vol(D)) · μ(D);
    R ← R \ (A ∪ B);
C ← C + μ(R);
return s
```

3.4. Complexity Analysis

We treat k in the following as a small constant, because the construction algorithm always picks $k \in \{1, 2, 3, 4\}$.

Lemma 5. *The query complexity of a k-level SliceHist with precision ε is $O\left(dk \log \frac{1}{\varepsilon} 2^d (1 + (2d)^{k-1}\right)$ and $O_d(\log \frac{1}{\varepsilon})$ when d and k are treated as constants.*

Proof. The query is rank transformed dk times in $O(\log \frac{1}{\varepsilon})$ and runs $(1 + (2d)^{k-1})$ grid-aligned queries, which are answered in $O(2^d)$ as the grid histograms are stored as prefix cubes/sums. □

We start by demonstrating that a multi-level SliceHist indeed provides ε-approximation guarantees that depend on the granularity parameters of the per-level grid histograms.

Lemma 6. *A k-level $[\varepsilon_1, \ldots, \varepsilon_k]$-SliceHist in d dimensions is an ε-histogram with $\varepsilon = (2d)^k \cdot \prod_{i=1}^k \varepsilon_i$.*

Proof. The only error in querying a k-level SliceHist results from partially intersected grid slices at the lowest level k of SliceHist; all other buckets are fully contained in the query and, thus, cannot introduce an error. By the construction of SliceHist, each such level k slice contains at most a fraction $\prod_{i=1}^k \varepsilon_i$ of the data points, while the recursive querying process ensures that the number of partially intersected slices is at most $(2d)^k$, which proves the result. □

Given a desired approximation accuracy ε, the following Lemma 7 shows how to determine the per-level parameters $\varepsilon_1, \ldots, \varepsilon_k$ that minimize the overall SliceHist storage costs.

Lemma 7. *Given a desired accuracy guarantee ε and a number of levels k, the minimal storage costs are achieved by picking the vector $\varepsilon_1, \ldots, \varepsilon_k$ such that*

$$\varepsilon_i = \begin{cases} \left(\frac{\varepsilon}{(2d)(2d-1)^{k-1}}\right)^{\frac{1}{A_k}} \left(\frac{1}{\prod_{j=1}^{k-1}(d-1)^{A_j}}\right)^{\frac{1}{A_k}} & \text{for } i = 1, \\ \varepsilon_{i-1}^{(1-\frac{1}{d})}(d-1)^{1/d} & \text{for } i > 1, \end{cases}$$

where $A_j = 1 - (1 - \frac{1}{d})^j$.

Proof. The storage costs of a k-level SliceHist are determined by the total space required by the grid histograms for all slices across all levels of the summary. It is not difficult to see that, at a given level $i \in \{1, \ldots, k\}$, the total number of slice histograms (for slices created at the previous level) is exactly $\frac{d^{i-1}}{\varepsilon_1 \cdots \varepsilon_{i-1}}$, which implies that the total storage cost (i.e., number of grid histogram buckets) at level i is $\frac{d^{i-1}}{\varepsilon_i^d \prod_{l=1}^{i-1} \varepsilon_l}$. Thus, minimizing storage costs for our synopsis can be formalised as the following (non-linear) optimization problem:

Minimize
$S(\varepsilon_1, \ldots, \varepsilon_k) = \sum_{i=1}^k \frac{d^{i-1}}{\varepsilon_i^d \prod_{l=1}^{i-1} \varepsilon_l}$

under the constraint $\varepsilon_1 \cdots \varepsilon_k = C$ ($= \varepsilon/(2d)^k$, constant). The objective function can easily be rewritten as

$$S(\varepsilon_1, \ldots, \varepsilon_k) = \frac{\sum_{i=1}^k d^{i-1}(\varepsilon_1 \cdots \varepsilon_{i-1})^{d-1}(\varepsilon_{i+1} \cdots \varepsilon_k)^d}{(\varepsilon_1 \cdots \varepsilon_k)^d},$$

and as the denominator is constant, we seek to minimize

$$S'(\varepsilon_1, \ldots, \varepsilon_k) = \sum_{i=1}^k d^{i-1} C_{-\{i\}}^{d-1} (\varepsilon_{i+1} \cdots \varepsilon_k)$$

under the constraint $\prod_i \varepsilon_i = C$, where for notational convenience we define $C_{-\{i,j\}} = \prod_{l=1, l \notin \{i,j\}}^k \varepsilon_l$. We can now write out the Lagrangian function:

$$L(\varepsilon_1, \ldots, \varepsilon_k, \lambda) = S'(\varepsilon_1, \ldots, \varepsilon_k) + \lambda(C - \prod_{i=1}^k \varepsilon_i)$$

and its partial derivative with respect to ε_i:

$$\frac{\partial L}{\partial \varepsilon_i} = -\lambda C_{-\{i\}} + d\varepsilon_i^{d-1} \sum_{j=1}^{i-1} d^{j-1} C_{-\{i,j\}}^{d-1} \prod_{l=j+1, l \neq i}^{k} \varepsilon_l + (d-1)\varepsilon_i^{d-2} \sum_{j=i+1}^{k} d^{j-1} C_{-\{i,j\}}^{d-1} \prod_{l=j+1}^{k} \varepsilon_l$$

At the optimal solution point, we have $\frac{\partial L}{\partial \varepsilon_i} = \frac{\partial L}{\partial \varepsilon_{i+1}} = 0$ for all i. Solving this equation (with some symbol manipulation) gives $\varepsilon_{i+1} = \varepsilon_i^{(1-1/d)}(d-1)^{1/d}$.

The result follows by combining the above expression with the expression for the overall error ε in Lemma 6. □

We can now define a *k-level ε-SliceHist* as an (ε-approximate) $[\varepsilon_1, \ldots, \varepsilon_k]$-SliceHist with $\varepsilon = (2d)^k \cdot \prod_{i=1}^k \varepsilon_i$, where the ε_i's are defined based on Lemma 7.

4. Analysis of ε-error Summaries

In this section, we analytically investigate existing data summarization approaches with respect to the offered ε-error guarantees for a limited space budget. Formally, a summary approach has ε-error guarantees if a summary can be created for any multiset of n points that approximates the multiset's counts along all axis-aligned boxes with at at most additive error εn. As the error guarantees depend only on the data size, summary size and data dimensionality,

we can predict the practical performance through formulas and do not need to run experiments for this purpose. In our analysis, space budget numbers are assumed to be in bytes, and the summary values/counts are assumed to be stored in $64bit = 8byte$ words.

4.1. Histogram-based ε-Approximations

Theorem 1 (SliceHist properties)**.** *A k-level SliceHist summary with parameters $\varepsilon_1, \ldots, \varepsilon_k$ has size in bytes:*

$$\min_{\alpha_1,\ldots,\alpha_k \in (1,2]} \sum_{i=1}^{k} \left(\prod_{j=1}^{i-1} d \left\lceil \frac{\alpha_j}{\varepsilon_j} \right\rceil \right) \left(\left\lceil \frac{\alpha_i}{\varepsilon_i} \right\rceil^d + d^2 \left\lceil \frac{\frac{\alpha_i}{\alpha_i - 1} 3}{\varepsilon_i} \right\rceil \right) 8$$

It can be constructed in $(k+1)$ passes over an ordered stream of multipoints by performing $\sum_{i=1}^{k} d^i$ quantile summary insertions and $\sum_{i=1}^{k} d^i$ binary searches per point. It answers any box count query by performing $\sum_{i=1}^{k} (2d)^{i-1} d \leq k(2d)^{k-1}$ binary searches and $\sum_{i=1}^{k} (2d)^{i-1} 2^d \leq k d^{k-1} 2^{d+k-1}$ lookups. It achieves ε-error guarantees by choosing $\varepsilon_1, \ldots, \varepsilon_k$ according to Lemma 7 s.t. $\prod_{i=1}^{k} \varepsilon_i = \frac{\varepsilon}{2d(2d-1)^{k-1}}$, which results in a summary size of $O_d(\frac{1}{\varepsilon}^{d/(d-d(\frac{d-1}{d})^k)})$ for a fixed dimensionality and a summary size of $O_\varepsilon(O(d)^{O(d)})$ for a fixed ε.

Proof. A 1-level ε_1-SliceHist summary with α_1 of n points is constructed in two data passes by performing d quantile summary insertions and d binary searches per data point. In the first data pass, $\frac{\alpha_1 - 1}{\alpha_1} \frac{\varepsilon_1}{6}$-approximate quantile summaries $q_1, \ldots, q_d$ are created for each dimension. Additionally, a grid histogram G_1 with $\left\lceil \frac{\alpha_1}{\varepsilon_1} \right\rceil$ grid slices per dimension is created, whose counts will remain zero until the second iteration. Then each data point is inserted in each of the d quantile summaries, which performs d quantile operations per point (insertions). At the end of the first iteration, each of the quantile summaries $q_1, \ldots, q_d$ are queried for $\lceil \frac{\alpha_1 - 1}{\alpha_1} \varepsilon_1 \rceil$ quantiles, which totals to $d\lceil \frac{\alpha_1 - 1}{\alpha_1} \varepsilon_1 \rceil$ quantile operations (in this case quantile queries), which are negligibly few compared to the quantile operations performed per data point. The obtained quantile points then comprise a δ-rank-transformation ϕ_1 with $\delta = \frac{\alpha_1 - 1}{\alpha_1} \varepsilon_1$. In a second data pass, each point is transformed using ϕ_1 by performing d quantile operations (in this case quantile queries). Then the slice containing the point can be determined for each dimension in $O(1)$ arithmetic operations and the count at the grid cell of G_1 is incremented.

A k-level SliceHist summary with $\varepsilon_1, \ldots, \varepsilon_k$ and $\alpha_1, \ldots, \alpha_k$ is created by first creating a 1-level ε_1-SliceHist summary (ϕ_1, G_1) and then creating a $(k-1)$ SliceHist summary of the ϕ_1-transformed points for each grid slice of G_1. Naively, this would result in $2k+1$ data passes, but since nothing depends on the grid counts obtained in the second iteration of the 1-level construction, the second iterations of the 1-level SliceHist can be all moved to a last pass over the data. Thus, only $k+1$ data passes have to be performed. In the first data pass, d quantile summary insertions are performed per point, ..., in the ℓ-th data pass $d^{\ell-1}$ binary searches, and d^ℓ quantile summary insertions are performed per point. In the $(k+1)$-th data pass, d^k binary searches are performed per data point. Thus, the total number of quantile summary insertions and binary searches is $(d + d^2 + \ldots + d^k)$.

Let $A_x = 1 - (1 - 1/d)^x$. From $d \geq 1$ follows $\frac{1}{d} \leq A_x < 1$ for any x. It is also easy to see that $A_x / A_y \leq 1$ for $x < y$.

From picking $\alpha_\ell = 2$ for any $1 \leq \ell \leq k$ we obtain summary size $O(\sum_{i=1}^{k} \frac{2^{d+i-1} d^{i-1} (2d)(2d-1)^{k-1}}{{\varepsilon_i}^d \prod_{j=1}^{i-1} \varepsilon_j} + \frac{d^{i-1} d^2}{\prod_{j=1}^{i-1} \varepsilon_j})$. ε_i can be written as

$$\varepsilon_i = \frac{\varepsilon}{(2d)^k \prod_{j=1}^{k-1} (d-1)^{A_j}}^{\frac{(1-1/d)^{i-1}}{d A_k}} \prod_{j=0}^{i-2} \sqrt[d]{d-1}^{1-(1/d)^j},$$

and by inserting the formulas for ε_i into the initial formula, we obtain

$$O(\textstyle\sum_{i=1}^{k} 2^{d+i-1} \left(\frac{d}{d-1}\right)^{i-1} (d-1)^{\frac{\sum_{j=1}^{k-1} A_j}{A_k} \frac{1}{A_k}} \frac{2d(2d-1)^{k-1}}{\varepsilon} + d^{i+2} (d-1)^{(\sum_{j=0}^{i-1} (1-1/d)^j - 1) + (\sum_{j=1}^{k-1} A_j) \frac{A_i}{A_k}} \left(\frac{2d(2d-1)^{k-1}}{\varepsilon}\right)^{\frac{A_i}{A_k}})$$

By upper bounding any d-related terms by $2d$ and by exploiting that for any $1 \leq i \leq k$ it holds that $\frac{A_i}{A_k} \leq 1$, $1 \leq \frac{1}{A_i} \leq d$ and $(1 - 1/d) < 1$, we can reduce it to $O_\varepsilon(2^{(d+2dk+4k)} d^{(2kd+3k+2)})$ which is in $O_\varepsilon(O(d)^{O(d)})$. □

Theorem 2 (Equi-depth summary complexity)**.** *An equi-depth summary achieves ε-error guarantees with parameter B chosen as the smallest integer s.t. $\sum_{i=0}^{d-1} (\frac{2(B-2)}{B})^i \frac{4}{B} \leq \varepsilon$, which leads to $B = O(\frac{d}{\varepsilon})$ and a summary size in $O(\frac{O(d)^{d+2}}{\varepsilon^d})$. It answers queries with time complexity $O(\frac{1}{\varepsilon^{d-1}} \log(\frac{1}{\varepsilon}))$ and can be constructed in $(d+1)$ passes over an ordered stream of multipoints by performing in total $O(d)$ quantile operations per point.*

Proof. A $(1, B)$ summary can be constructed in one pass over the stream. In the first pass, a $\frac{1}{2B}$-approximate quantile summary is created and each data point is inserted into it. After the first pass, B approximate quantile points are obtained, such that there are at most $\frac{2}{B}$ points between two approximate quantiles. A (d, B) summary can be constructed in d passes over the stream. It first constructs a $(1, B)$ equi-depth summary and then constructs for each bucket a $(d-1, B)$ summary.

A d-dimensional equi-depth summary creates first a (d, B)-summary and then does another data pass to count the number of points in each bucket, so that no error is accumulated over buckets completely contained in the query region. Thus, the summary consists of $d \sum_{i=1}^{d} B^{i-1}(B-1) \leq d^2 B^d$ quantile point coordinates and $\sum_{i=1}^{d} B^i \leq d B^d$ counts.

A 1-dimensional equi-depth summary has a maximum error of $f_1(B) = 2\frac{2}{B}$ and a d-dimensional summary a maximum error of $f_d(B) = f_1(B) + (B-2)\frac{2}{B} f_{d_1}(B)$. Unraveling the recursion yields $f_d(B) = \sum_{i=0}^{d-1} (\frac{2(B-2)}{B})^i \frac{4}{B} = O(\frac{d}{B})$. Thus, equi-depth achieves ε-error guarantees for a summary size $O(\frac{O(d)^{d+2}}{B^d})$. □

Dyadic histograms [7] can be described as a recursive data structure as follows. A (p, d)-dyadic histogram operates in d dimensions and divides the space along the first dimension into 2^p space regions $r_1, \ldots, r_{2^p}$, each containing at most $\frac{1}{2^p}$ points. An interesting observation here is that a query box might partially intersect all of these regions, but then completely contains all but two regions along the first dimension. This can be exploited by constructing a binary tree over the regions and maintaining for each node a $(q, d-1)$-dyadic histogram that operates only in the last $d-1$ dimensions. In the worst case the query box can be split into $4 + 2(p-2)$ parts of which 2 can be handed off to a node with 2 regions, 2 to a node with 4 regions, ..., 2 to a node with 2^{p-1} regions and 4 are handled directly.

Theorem 3 (Dyadic histogram complexity)**.** *Let $err_d(p,n) = 2\,err_1(p,n) + 2\sum_{q=1}^{p-2} err_{d-1}(q, 2^q \frac{n}{2^p}) \leq n$ be the recursively defined maximal error of (p, d)-dyadic histograms with base case $err_1(p,n) = \frac{2}{2^p}n$. A dyadic histogram achieves ε-error guarantees with parameter $p = O(\log\left(\frac{1}{\varepsilon}\log^{d-1}\frac{2^d}{\varepsilon}\right))$ chosen as the smallest integer s.t. $err_d(p,n) \leq \varepsilon n$, for which it has a summary size of $O(\log^{d-1}\left(\frac{1}{\varepsilon}\log^{d-1}\frac{2^d}{\varepsilon}\right)\frac{1}{\varepsilon}\log^{d-1}\frac{2^d}{\varepsilon})$.*

Proof. The complexity of the error is $err_d(p,n) = O(\frac{p^{d-1}}{2^p}n)$, because each expanded error formula for $d \geq 2$ contains the sum $\sum_{i=1}^{p-1} P(i)$ with a polynomial $P(i) = O(p^{d-2})$ of degree $d-2$. By applying Faulhaber's formula $\sum_{i=1}^{p-1} P(i)$ can be rewritten as one polynomial $\mathcal{P}(p) = O(p^{d-1})$. For instance, $err_2(p,n) = max(n, (4p-4)\frac{n}{2^p})$ and $err_3(p,n) = max(n, (4p^2-20p+32)\frac{n}{2^p})$. In order to achieve $err_d(p,n) < \varepsilon n$, we need to satisfy $\mathcal{P}(p)\frac{n}{2^p} \leq \varepsilon n$. The minimal value for p to provide ε-error guarantees can be obtained by first upper bounding p using a crude analysis and then refining it through the inequality. A crude analysis exploits that a (p, d)-dyadic histogram is comprised of an equi-depth histogram with 2^p buckets, which guarantees that $\varepsilon \leq O(\frac{1}{2^{p/d}})$. Thus, $p \geq O(d \log \frac{1}{\varepsilon})$ ensures ε-error guarantees. Substituting p with the crude upper bound in the second part of the inequality $p^{d-1}\frac{1}{2^p} \leq \varepsilon$ results in $d^{d-1}\log^{d-1}\frac{1}{\varepsilon}\frac{1}{2^p} \leq \varepsilon$, which solved for p yields $p \geq O(\log\left(\frac{d^{d-1}}{\varepsilon}\log^{d-1}\frac{1}{\varepsilon}\right)) = O(\log\left(\frac{1}{\varepsilon}\log^{d-1}\frac{2^d}{\varepsilon}\right))$.

A (p, d)-dyadic histogram of n points stores $s(1,p) = 2^p$ bucket boundaries for $d = 1$ and $s(d,p) = s(1,p) + \sum_{q=1}^{p-1} 2^{p-q}\, s(d-1, q)$ bucket boundaries for $d \geq 2$. The total number of bucket boundaries is in $O(p^{d-1}2^p)$, because each expanded size formula for $d \geq 2$ contains the sum $\sum_{i=1}^{p-1} P(i)$ with a polynomial P of degree $d-2$, and using Faulhaber's formula the sum can be reformulated as a polynomial of degree $d-1$. For instance, $s(2,p) = p(2^p)$ for $d = 2$ and $s(3,p) = (p^2 - 3p + 2)2^p$ for $d = 3$. Inserting the complexity of $p = O(\log\left(\frac{1}{\varepsilon}\log^{d-1}\frac{2^d}{\varepsilon}\right))$ as a function of ε into the size formula, we obtain the size bound $O(p^{d-1}2^p) = O(\log^{d-1}\left(\frac{1}{\varepsilon}\log^{d-1}\frac{2^d}{\varepsilon}\right)\frac{1}{\varepsilon}\log^{d-1}\frac{2^d}{\varepsilon}) = O(\log^{2d-2}\frac{2^d}{\varepsilon}\log^{d-1}\left(\log\frac{1}{\varepsilon}\right))$, which is $O_\varepsilon(O(d)^{O(d)})$. □

4.2. Deterministic ε-approximation Sampling

Existing ε-approximation approaches [7], [6], [8], [11], [12], [5] to create samples with ε-error guarantees for multidimensional range queries are either astronomically expensive to construct or do not offer non-trivial guarantees. As this is not directly evident from the asymptotic formulas, we take a deeper look in the next Theorem.

Theorem 4. *Existing deterministic ε-approximation sampling approaches [7], [6], [8], [11], [12], [5] require at least 10^{24} arithmetic operations to construct an ε-approximation sample with error guarantee $\varepsilon \leq 1\%$.*

Proof. All noted approaches are based on applying a halving procedure that splits the data into two halves such that one half is an ε-approximation of the data. Clearly, an approach solely based on cleverly applying the halving procedure cannot be more precise than the halving procedure itself. The approaches can be split into two groups. They are either based on (a) a halving procedure related to Spencer's discrepancy bounds [26] or (b) a halving procedure using the Beck-Fiala [27] theorem and dyadic range decomposition [7].

(a) Approaches related to Spencer's discrepancy bounds such as [6], [8], [11], [12] require in two dimensions at least 10^{24} arithmetic operations and 10^{32} in three dimensions (which is a very optimistic lower bound) to construct an ε-sample of size $O(\frac{1}{\varepsilon^2})$ with $\varepsilon \leq 1\%$ for axis-aligned range queries. The halving procedure outlined in Proposition 3.1 of [11] performs more than $n^{2d+1}/4^d$ arithmetic operations to achieve $\varepsilon \geq \sqrt{\frac{16d\log(4n)+4}{n}}$ for arbitrary range queries. For $d = 2$, in order to achieve $\varepsilon \leq 1\%$ the data size n has to be larger than a million, which leads to construction costs beyond 10^{28} arithmetic operations. Exploiting the fact that each axis-aligned query can be substituted with 2^d anchored queries (i.e., min-corner has to be the origin) in combination with the inclusion-exclusion-principle, the number of arithmetic operations can be improved to n^{d+1} and $\varepsilon \geq 2^d\sqrt{\frac{16d\log(4n)+4}{n}}$, but then from $\varepsilon \leq 1\%$ follows that $n > 10^8$ which still results in construction costs beyond 10^{24} arithmetic operations for two dimensions and more than 10^{32} in three dimensions.

(b) Approaches using the Beck-Fiala theorem such as [7], [5] require at least 10^{36} arithmetic operations to construct an ε-sample of size $O(\frac{1}{\varepsilon}\log^{2d}\frac{1}{\varepsilon}\,\text{polylog}(\log\frac{1}{\varepsilon}))$ with $\varepsilon \leq 1\%$ for axis-aligned range queries. The halving procedure of these methods constructs for a data set of size n a sample of size $\frac{n}{2}$ by solving n linear equation systems of size $n \times n$, which requires at least n^4 arithmetic operations. The resulting sample is an ε-approximation with $\varepsilon \leq \frac{2t-1}{n}$, where t is the maximal number of distinct queries in which a data point appears. This requires the value t and the number of queries to be smaller than n. In general $t = O(n^d)$, and the total number of queries is $O(n^{2d})$. To reduce these two numbers, the query set can be limited to dyadic (canonical) ranges. Dyadic ranges are simply cross products of dyadic intervals. We can think of dyadic intervals as buckets of

	Dyad. Histograms			SliceHist ($k = O(1)$)			Equi-depth			Dyad. Sketches		
size for $d = O(1)$	$O_d(\frac{1}{\varepsilon}\log^{2d-2}\frac{\log\frac{1}{\varepsilon}}{\varepsilon})$			$O_d(\frac{1}{\varepsilon}^{d/(d-d(\frac{d-1}{d})^k)})$			$O_d(\frac{1}{\varepsilon d})$			$O_d(\frac{1}{\varepsilon}(\log^{2d} n)\log\frac{1}{p})$		
size for $\varepsilon = O(1)$	$O_\varepsilon(O(d)^{O(d)})$			$O_\varepsilon(O(d)^{O(d)})$			$O_\varepsilon(O(d)^{d+2})$			$O_\varepsilon(O(d)^{O(d)}\log\frac{1}{p})$		
	$d=2$	$d=3$	$d=4$	$d=2$	$d=3$	$d=4$	$d=2$	$d=3$	$d=4$	$d=2$	$d=3$	$d=4$
size for $\varepsilon = 5\%$	152.0KB	36.4MB	3.4GB	79.6KB	736.7KB	8.1MB	96.9KB	37.4MB	18.0GB	89.5GB	437TB	2.1EB
size for $\varepsilon = 1\%$	1.6MB	430.5MB	102.2GB	463.8KB	7.4MB	193.5MB	2.4MB	4.8GB	11.7TB	447GB	2.2 PB	10.4EB
size for $\varepsilon = 0.1\%$	15.5MB	10.3GB	2.7TB	5.6MB	271.1MB	27.7GB	244.0MB	4.7TB	116.2 PB	4.5TB	21.9 PB	103.9EB
size for $\varepsilon = 0.01\%$	312.0MB	233.6GB	66.4TB	83.8MB	11.5GB	1.2TB	23.8GB	4.6 PB	$\geq$ 1EB	44.7TB	219 PB	1ZB
size for $\varepsilon = \frac{1}{1000}\%$	5.9GB	2.3TB	1.5 PB	1.3GB	225.0GB	56.0TB	2.3TB	$\geq$ 1EB	$\geq$ 1EB	447TB	2.19EB	10.4ZB
size for $\varepsilon = \frac{1}{10000}\%$	53.0GB	47.9TB	34.2 PB	23.2GB	4.8TB	1.7 PB	232.8TB	$\geq$ 1EB	$\geq$ 1EB	4.47PB	21.9EB	103.9ZB
construction time	$O_d(n\log^d\frac{1}{\varepsilon})$			$O_d(n\log\frac{1}{\varepsilon})$			$O_d(n\log\frac{1}{\varepsilon})$			$O_d(n\log^d n\ \log\frac{1}{p})$		
passes over data stream	d			$k+1$			$d+1$			1		
query time	$O_d(\log^{d-1}\frac{1}{\varepsilon})$			$O_d(\log\frac{1}{\varepsilon})$			$O_d(\frac{1}{\varepsilon d})$			$O_d(\log^d n\log\frac{1}{p})$		

TABLE 1: Comparison of approaches with error guarantees using asymptotic complexities and exact summary sizes (cf. Theorem 1, Theorem 2 and Theorem 3), and dyadic sketches that fail to provide ε-guarantees with probability p, where we use $p = 5\%$ and $\log n = 32$ for summary sizes; best approaches are highlighted. Existing ε-approximation sampling approaches [7], [6], [8], [11], [12], [5] are not featured due to astronomical construction costs (cf. Theorem 4).

one-dimensional equi-depth histograms with 2^x buckets for $x \in \mathbb{N}$. In computational geometry literature it is commonly known that any axis-aligned range can be built from the union of $m = (2\log_2(n) - 2)^d$ dyadic ranges and each point is contained in at most $t = \log_2(n+1)^d$ dyadic ranges. Thus, an $\frac{\varepsilon}{m}$-approximation over the dyadic ranges is an ε-approximation over all ranges. There are in total $O(n^d)$ dyadic ranges, but since each point is contained in at most t ranges, it follows that only $O(n)$ of the dyadic ranges can contain more than $\frac{\varepsilon}{m}$ points. Taken all together, the Beck-Fiala theorem applied to canonical ranges results in a halving procedure with $\varepsilon n \leq m(2t-1) = (2\log_2(n) - 2)^d(2(\log_2(n+1))^d - 1)$. □

4.3. Summary of Results

As Theorems 1–3 provide exact summary sizes to achieve ε-error guarantees, we can directly compare how compact the summaries are for a given summary size. The results are shown in Table 1. As an additional comparison, we also report the results for dyadic sketches [28], [1] that provide probabilistic ε-guarantees as discussed in the related work section. While Table 1 is just a snapshot, the formulas stipulate that SliceHist offers the tightest ε-error guarantees for summary sizes up to exabytes and is vastly smaller for a fixed precision, especially in more dimensions.

While dyadic histograms asymptotically scale better with ε, SliceHist offers the best query time complexity. Both Equi-depth and SliceHist offer very fast construction, which is comparable to sorting the data, while dyadic histograms are much more expensive to construct. From the formulas it is not clear which approach scales best with dimensionality, but from the exact sizes only SliceHist appears to be functional up to four dimensions.

SliceHist, equi-depth and dyadic histograms can be constructed by performing for each point a number of quantile operations (insertions into quantile summaries, or binary searches). Dyadic histograms perform $O_d(\log^{d-1}(\frac{1}{\varepsilon}))$ operations whereas SliceHist and equi-depth only perform $O(1)$. We therefore expect SliceHist's and equi-depth's construction time to be much smaller than of dyadic histograms, especially in more dimensions.

5. Experimental Evaluation

In this section, we present experimental results on the performance of SliceHist to support the theoretical analyses of Section 4 (cf. Table 1). We use the term "selectivity of a query box" as the fraction of all points contained in it.

Setup and Datasets. We use a machine with an Intel(R) Xeon(R) W-2145 CPU @ 3.70GHz with 16 cores and 512GB RAM. The proposed approach only has access to 4GB of RAM and all approaches use only one core. All algorithms are implemented by the same author in C++ and compiled with GCC 7.4.0 using -O3.

We use four different real-world datasets. *OSM2D*: a 46GB spatial dataset from the OpenStreetMap project that records 2.9 billion two-dimensional GPS-coordinates [29]. *CLOUD3D*: a point cloud of 15GB based on laser scans of a public street comprised of 429 million points. *GAIA3D* and *GAIA4D*: a scientific dataset with 1.7 billion multi-dimensional entries from the European Space Agency Mission Gaia to detect celestial objects and map the universe [30]. *GAIA3D* has dimensions "right ascension","declination" and "G flux", and a size of 40GB. *GAIA4D* additionally has "G magnitude" and a size of 54GB. The attributes relate to the location, energy, and brightness of celestial bodies.

We create for each dataset 5000 queries, with 1000 queries for each of the targeted selectivities 0.01%, 0.1%, 1%, 5% and 10%. Each query box is created as described in the following. The shape of the box is determined by drawing two uniform points as opposing corners of a box and only using its proportions (ignoring size and position). The center of the box is determined by drawing a random data point. After fixing the proportions and position, the size of the box is chosen such that it contains roughly as many points as the targeted selectivity (tolerating a 0.5x

measure	75th percentile q-error					75th percentile q-bounds				
selectivity [%]	0.01	0.1	1	5	10	0.01	0.1	1	5	10
summary size	OSM2D (46 GB)									
1 MB	1.64 RS	1.16 RS	1.035 DH	1.011 DH	1.0086 DH	no winner	0.81 DY	0.96 DY	0.99 DY	0.99 DY
10 MB	1.16 DH	1.045 DH	1.0083 DH	1.0025 DH	1.0016 DH	67977 DH	1.062 DH	0.98 DH	0.98 DH	0.99 DH
100 MB	1.03 DH	1.0051 DH	1.00081 DH	1.00027 DH	1.00017 DH	1.067 DH	0.99 DY	0.99 DY	0.99 DY	0.99 DY
1000 MB	1.0073 DH	1.00094 DH	1.00015 DH	1.000056 DH	1.000033 DH	1.029 DH	0.99 DH	0.99 DH	0.99 DH	0.99 DH
summary size	CLOUD3D (15 GB)									
1 MB	2 RS	2 RS	1.38 RS	1.16 RS	1.1 RS	no winner	no winner	1.11 DH	1.14 DH	1.11 DH
10 MB	1.86 RS	1.49 RS	1.11 RS	1.039 DH	1.026 DH	no winner	5.9 DH	1.21 DH	1.058 DH	1.03 DH
100 MB	1.96 RS	1.12 RS	1.022 RS	1.0078 DH	1.0055 DH	no winner	1.37 DH	1.015 DH	0.99 DH	0.99 DH
1000 MB	1.26 RS	1.035 RS	1.005 RS	1.0016 RS	1.001 RS	4 DH	1.17 DH	1.011 DH	1.0014 DH	1.00028 DH
summary size	GAIA3D (40 GB)									
1 MB	1.81 RS	1.42 RS	1.3 RS	1.14 RS	1.1 RS	no winner	no winner	<0.1 ALL	0.38 ED	0.58 ED
10 MB	1.13 RS	1.21 RS	1.11 RS	1.038 RS	1.026 RS	no winner	<0.0001 ALL	0.11 ED	0.52 ED	0.69 ED
100 MB	1.068 RS	1.08 RS	1.015 RS	1.0057 RS	1.0042 RS	<0.1 ALL	0.31 DY	0.87 DY	0.97 DY	0.98 DY
1000 MB	1.049 RS	1.05 RS	1.0048 RS	1.0016 RS	1.0011 RS	0.041 DH	0.24 DY	0.82 DY	0.95 DY	0.97 DY
summary size	GAIA4D (54 GB)									
1 MB	5.7 RS	2.2 RS	2.6 RS	1.85 RS	1.55 RS	no winner	no winner	no winner	<0.0001 ALL	<0.0001 ALL
10 MB	1.61 RS	3 RS	1.5 RS	1.21 RS	1.12 RS	no winner	no winner	<0.0001 ALL	0.052 ED	0.22 ED
100 MB	3.8 RS	1.38 RS	1.17 RS	1.05 RS	1.033 RS	no winner	<0.0001 ALL	0.053 ED	0.38 ED	0.56 ED
1000 MB	2.3 RS	1.2 RS	1.076 RS	1.02 RS	1.014 RS	no winner	<0.0001 ALL	0.11 ED	0.48 ED	0.66 ED

TABLE 2: Comparison of precision for SliceHist (SH), Dyadic Histograms (DY), Random Sampling (RS), Equi-Depth (ED), Dyadic Sketches, and DigitHist (DH). The numbers indicate the error of SH as a factor of the indicated best competitor's error. Cells where SH is the best approach are highlighted in yellow. If all competitors have q-bounds > 100 we reference the best competitor by the wildcard "ALL". If all approaches have q-bounds > 100 we declare "no winner".

smaller and 1.5x larger selectivity). Each box is stored with the correct count to evaluate how precisely the summaries approximate the counts.

We compare the following summary techniques. *Sampling* takes a simple random sample (which is what AQP systems commonly do). *SliceHist* is a SliceHist summary with ε large enough such that the predicted summary size is within the given space budget, while parameter k is chosen s.t. the summary size is minimized. *Dyad. Hist.* [7] is the state-of-the-art technique for histogram-based summaries with ε-error guarantees and implemented using the same quantile summaries as SliceHist. *Equi-depth* is an equi-depth histogram [20] constructed using $(d+1)$ passes and the same quantile summaries as SliceHist. DigitHist [24] is a state-of-the-art multidimensional histogram that offers per query bounds, but no ε-error guarantees. *Sketch* reduces range queries to point queries through dyadic ranges as proposed in [28] and supports point queries using CM-sketches [31].

Results. For each of the datasets with sizes ranging from 15GB to 54GB we create a summary of different sizes (1MB, 10MB, 100MB, 1000MB) using the five different summary techniques. We evaluate the count estimates and count bounds provided by the techniques for 5000 box queries (1000 for each selectivity 0.01%, 0.1%, 1.0%, 5% and 10%), and compute how much these estimates deviate from the correct count (q-error) and how tight the bounds are (q-bounds). We aggregate the 75th percentiles of the q-errors/q-bounds and then compare the summaries over the same datasets and summary size with each other. The q-error is the multiplicative error between estimates and correct counts $max(T/E, E/T)$, and q-bounds are the quotient between upper and lower bounds U/L, where T is the correct count, E is the summary's estimate and $L \leq T \leq U$ are the count bounds. Table 2 shows the results. We can see that SliceHist is the best approach in 32% of the settings with a winner. Interestingly, it fills the niche of being precise for difficult problems, e.g., tight bounds for the GAIA4D dataset. This is predicted by its theoretical properties, as it is guaranteed to do well for any data or query distribution.

For the next experiment we fix the summary size to approximately 100MB and use a query selectivity of 1% and 0.1%. Table 3 reports the results in terms of achieved ε-error guarantee, construction time, average query time, 75th percentile q-error, and 75th percentile q-bounds. As it is difficult to construct competitor summaries for exact sizes, we make sure that SliceHist uses less space than the competitors to achieve a fair comparison. We can see that dyadic histograms and dyadic sketches are very expensive to construct, where as a timeout we use two days. SliceHist is more expensive to construct compared to heuristic approaches that do not provide ε-error guarantees, and also more expensive than Equi-depth, but in turn provides by far the best ε-error guarantees in all cases. This is reflected by its tight q-bounds, as a ε-error guarantee translates to q-bounds and q-error < 2 for any queries with at most 2ε selectivity. In terms of query times, SliceHist is always among the best approaches, while Sampling is by far the slowest. When comparing the ε-error guarantees of the techniques for the dataset with different dimensionalites, we see that DyadicHist and Equi-depth become significantly worse with increasing dimensionality for a limited space budget. SliceHist is the only approach that offers ε-error guarantees less than 1% for a 4-dimensional dataset using a summary with at most 128MB. It is also the only approach

	SliceHist	Dy.Hist.	Sampling	Eq.Depth	DigitHist	Sketch
OSM2D (46 GB)						
summary size	85MB	126MB	143MB	237MB	101MB	123MB
ε-err. guarantee	0.005%	0.025%	100%	0.22%	100%	100%
constr. time	2.1h	9h	1.33min	32min	2h	18h
avg query time	<1ms	<1ms	665ms	1.8ms	349ms	14ms
75-th percentile of q-error						
selectivity 0.1 %	1.007	1.0059	1.011	1.04	1.0018	80
selectivity 1 %	1.001	1.00068	1.0036	1.0082	1.00023	9.2
75-th percentile of q-bounds						
selectivity 0.1 %	1.046	1.053	>100	1.66	1.075	>100
selectivity 1 %	1.0047	1.0059	>100	1.063	1.014	>100
CLOUD3D (15 GB)						
summary size	70MB	157MB	143MB	105MB	124MB	131MB
ε-err. guarantee	0.13%	1.92%	100%	7.8%	100%	100%
constr. time	1.11h	16h	14sec	5.3min	47min	1.2day
avg query time	<1ms	<1ms	492ms	14ms	282ms	6ms
75-th percentile of q-error						
selectivity 0.1 %	1.14	1.26	1.014	1.53	1.036	>100
selectivity 1 %	1.027	1.04	1.0046	1.11	1.0055	63
75-th percentile of q-bounds						
selectivity 0.1 %	2.1	2.8	>100	26	1.53	>100
selectivity 1 %	1.12	1.22	>100	3.1	1.11	>100
GAIA3D (40 GB)						
summary size	107MB	142MB	143MB	247MB	135MB	-
ε-err. guarantee	0.1%	1.92%	100%	5.8%	100%	-
constr. time	2.4h	1.75day	46sec	13min	1.38h	timeout
avg query time	0.2ms	0.2ms	505ms	27ms	426ms	-
75-th percentile of q-error						
selectivity 0.1 %	1.096	1.32	1.015	1.89	1.81	-
selectivity 1 %	1.02	1.058	1.0045	1.15	2.4	-
75-th percentile of q-bounds						
selectivity 0.1 %	1.58	5	>100	>100	>100	-
selectivity 1 %	1.1	1.25	>100	3.2	>100	-
GAIA4D (54 GB)						
summary size	128MB	-	143MB	-	179MB	-
ε-err. guarantee	0.85%	-	100%	-	100%	-
constr. time	2.3h	timeout	46sec	timeout	1.84h	timeout
avg query time	<1ms	-	398ms	-	421ms	-
75-th percentile of q-error						
selectivity 0.1 %	1.41	-	1.017	-	>100	-
selectivity 1 %	1.18	-	1.0054	-	31	-
75-th percentile of q-bounds						
selectivity 0.1 %	7.2	-	>100	-	>100	-
selectivity 1 %	1.72	-	>100	-	>100	-

TABLE 3: Comparison of $\approx$ 100MB summaries.

to obtain q-bounds < 10 for both query selectivities and q-bounds < 2 for 1% selectivity.

6. Related Work

ε-approximations are samples that approximate the selectivity in any range query with at most εn absolute error, where ε is a precision parameter and n is the data size. The concept is widely used in the discrepancy and computational geometry literature [9], [10]. Similarly, relative ε-approximations guarantee a limited multiplicative error for sufficiently selective queries [32], [33]. We extend the notion of bounding the maximal absolute error to εn to non-sampling structures and refer to it as ε-error guarantees. Existing deterministic sampling approaches [7], [6], [8], [11], [12], [5] are based either (a) on a halving procedure related to Spencer's discrepancy bounds [26] or (b) a halving procedure using the Beck-Fiala [27] theorem and dyadic range decomposition [7]. The techniques do not offer non-trivial guarantees for smaller data sizes and are too slow for larger data sizes, i.e., in order to obtain non-trivial guarantees such as $\varepsilon = 1\%$ it requires at least 10^{24} arithmetic operations. The attempted workaround to split the data into smaller chunks only leads to improvements in the asymptotic formulas at the cost of increased constants.

One-dimensional ε-approximations, such as equi-depth histograms or the GK summary [25], only require $O(\frac{1}{\varepsilon}\log(\varepsilon n))$ storage and can be efficiently constructed either by sorting the data or by incrementally constructing the summary while reading the data (see also Q-digest [34]). They are used as a building block for multi-dimensional summaries, such as dyadic histograms [7] or our approach.

Multi-dimensional data summaries in the database literature [1], [13], [14], [15], [16], [17], [18], [19] generally do not have error guarantees and, while working well on many datasets, cannot guarantee good precision for all datasets and queries. An exception are equi-depth histograms [20], where a very large histogram with $O(\frac{1}{\varepsilon^d})$ buckets has an ε-error guarantee as each query partially intersects at most $\frac{1}{\varepsilon^{(d-1)}}$ buckets, each containing at most ε^d of all points.

In more recent works, histogram-based summaries with ε-error have been proposed [7], [35] in the streaming literature, which can provide deterministic guarantees in practice. We will refer to them as dyadic histograms, because they share the basic idea of decomposing ranges along dyadic intervals, which is also the key idea behind range trees [36] and how to utilize sketches for range queries [1]. Dyadic histograms can be thought of as approximate variants of range trees. They are composed of $O(\log^{d-1}\frac{1}{\varepsilon})$ equi-depth histograms with varying numbers of space divisions (that are powers of two) per dimension. The knowledge of the equi-depth histograms is combined during querying to improve error bounds. Examples of dyadic histograms are GK-Multipass [7] that can obtain in d data scans a summary with $O(\frac{1}{\varepsilon}\log(\varepsilon n)\log^{2d-2}(\frac{1}{\varepsilon}\log(\varepsilon n)))$ storage, $O(\log^d\frac{1}{\varepsilon})$ querying complexity and $O(n\log^d\frac{1}{\varepsilon})$ construction time. Making some trivial changes it can be improved to $O(\frac{1}{\varepsilon}\log^{2d-2}\frac{\log(\varepsilon)}{\varepsilon})$ storage and $O(\log^{d-1}\frac{1}{\varepsilon})$ querying time using ideas from range trees and [35]. For two dimensions, there exists a specialized approach that offers $O(\frac{1}{\varepsilon}(\log^2\frac{1}{\varepsilon}+\log n))$ storage and $O(\log\frac{1}{\varepsilon})$ querying time [35].

Sketch-based summaries [1] that offer probabilistic error guarantees have been applied to one-dimensional ranges through dyadic ranges, and some works [28], [1] hint at possible multidimensional generalisations. In principle, a sketch-based summary is very desirable as it can be constructed in one data pass, supports updates and can provide deterministic upper bounds by using CM-sketches [31] and deterministic lower bounds by issuing 2^d range queries that form the complement. Since a continuous domain is at least as large as the data size n, a sketch-based summary that offers ε-error guarantees with probability $1-p$ has size $O(\frac{2^d}{\varepsilon}(\log^{2d}n)\log\frac{1}{p})$, query time $O(\log^d n\log\frac{1}{p})$ and

construction time $O(n(\log^d n)\log\frac{1}{p})$. Due to the $\log^{2d} n$ factor, the technique is not competitive for more than one dimension which is confirmed by our numerical evaluation (cf. Table 1) and experiments (cf. Table 2 and Table 3). The factor is rooted in the decomposition into dyadic ranges that adds for each dimension a $\log n$ factor not just to the number of sketches but also to their size (to counteract error accumulation). As evidenced by empirical results in [28], even spatial joins pose a much simpler problem to sketches, because they do not require such a troublesome range decomposition.

7. Conclusion

In this work, we have proposed a novel histogram-based summary for multisets of multidimensional points, termed SliceHist, that offers the best ε-error guarantees for a limited space budget (up to exabytes). It offers meaningful error guarantees up to four dimensions, whereas previous approaches already struggled with three dimensions. While it offers more reliable empirical performance than other approaches, it comes at a cost of a larger construction time, which could be improved by faster quantile summaries.

Acknowledgement

The work is supported by European Research Council grant ERC-2014-CoG 647557. We also thank Graham Cormode for sharing his expertise on sketching-related matters.

References

[1] G. Cormode, M. N. Garofalakis, P. J. Haas, and C. Jermaine, "Synopses for massive data: Samples, histograms, wavelets, sketches," *Foundations and Trends in Databases*, vol. 4, no. 1-3, pp. 1–294, 2012.

[2] G. Moerkotte, T. Neumann, and G. Steidl, "Preventing bad plans by bounding the impact of cardinality estimation errors," *PVLDB*, vol. 2, no. 1, pp. 982–993, 2009.

[3] J. M. Phillips, "Chernoff-hoeffding inequality and applications," *CoRR*, vol. abs/1209.6396, 2012.

[4] S. Har-Peled, *Geometric approximation algorithms*. American Mathematical Soc., 2011, no. 173.

[5] J. M. Phillips, "Algorithms for epsilon-approximations of terrains," in *ICALP (1)*, ser. LNCS, vol. 5125. Springer, 2008, pp. 447–458.

[6] P. K. Agarwal, G. Cormode, Z. Huang, J. M. Phillips, Z. Wei, and K. Yi, "Mergeable summaries," *ACM Trans. Database Syst.*, vol. 38, no. 4, pp. 26:1–26:28, 2013.

[7] S. Suri, C. D. Tóth, and Y. Zhou, "Range counting over multidimensional data streams," *Discrete & Computational Geometry*, vol. 36, no. 4, pp. 633–655, 2006.

[8] A. Bagchi, A. Chaudhary, D. Eppstein, and M. T. Goodrich, "Deterministic sampling and range counting in geometric data streams," *ACM Trans. Algorithms*, vol. 3, no. 2, p. 16, 2007.

[9] B. Chazelle, "The discrepancy method in computational geometry," in *Handbook of Discrete and Computational Geometry, Second Edition.*, 2004, pp. 983–996.

[10] ——, *The discrepancy method: randomness and complexity*. Cambridge University Press, 2001.

[11] J. Matoušek, "Approximations and optimal geometric divide-and-conquer," *J. Comput. Syst. Sci.*, vol. 50, no. 2, pp. 203–208, 1995.

[12] B. Chazelle and J. Matoušek, "On linear-time deterministic algorithms for optimization problems in fixed dimension," *J. Algorithms*, vol. 21, no. 3, pp. 579–597, 1996.

[13] D. Gunopulos, G. Kollios, V. J. Tsotras, and C. Domeniconi, "Approximating multi-dimensional aggregate range queries over real attributes," in *SIGMOD*, 2000, pp. 463–474.

[14] V. Poosala and Y. E. Ioannidis, "Selectivity estimation without the attribute value independence assumption," in *VLDB*, 1997, pp. 486–495.

[15] S. Acharya, V. Poosala, and S. Ramaswamy, "Selectivity estimation in spatial databases," in *SIGMOD*, 1999, pp. 13–24.

[16] J. Lee, D. Kim, and C. Chung, "Multi-dimensional selectivity estimation using compressed histogram information," in *SIGMOD*, 1999, pp. 205–214.

[17] Y. Matias, J. S. Vitter, and M. Wang, "Wavelet-based histograms for selectivity estimation," in *SIGMOD*, 1998, pp. 448–459.

[18] N. Bruno, S. Chaudhuri, and L. Gravano, "Stholes: A multidimensional workload-aware histogram," in *SIGMOD*, 2001, pp. 211–222.

[19] L. Getoor, B. Taskar, and D. Koller, "Selectivity estimation using probabilistic models," in *SIGMOD*, 2001, pp. 461–472.

[20] M. Muralikrishna and D. J. DeWitt, "Equi-depth histograms for estimating selectivity factors for multi-dimensional queries," in *SIGMOD*, 1988, pp. 28–36.

[21] J. L. Bentley and J. H. Friedman, "Data structures for range searching," *ACM Comput. Surv.*, vol. 11, no. 4, pp. 397–409, 1979.

[22] P. Jaworski, F. Durante, W. K. Hardle, and T. Rychlik, *Copula theory and its applications*. Springer, 2010, vol. 198.

[23] J. Ma and Z. Sun, "Mutual information is copula entropy," *Tsinghua Science & Technology*, vol. 16, no. 1, pp. 51–54, 2011.

[24] M. Shekelyan, A. Dignös, and J. Gamper, "Digithist: a histogram-based data summary with tight error bounds," *PVLDB*, vol. 10, no. 11, pp. 1514–1525, 2017.

[25] M. Greenwald and S. Khanna, "Space-efficient online computation of quantile summaries," in *SIGMOD*, 2001, pp. 58–66.

[26] J. Spencer, "Six standard deviations suffice," *Transactions of the American mathematical society*, vol. 289, no. 2, pp. 679–706, 1985.

[27] J. Beck and T. Fiala, ""Integer-making" theorems," *Discrete Applied Mathematics*, vol. 3, no. 1, pp. 1–8, 1981.

[28] A. Das, J. Gehrke, and M. Riedewald, "Approximation techniques for spatial data," in *SIGMOD*, 2004, pp. 695–706.

[29] OpenStreetMap contributors, "Planet dump retrieved from https://planet.osm.org ," https://www.openstreetmap.org , 2017.

[30] A. G. Brown *et al.*, "Gaia data release 1-summary of the astrometric, photometric, and survey properties," *Astronomy & Astrophysics*, vol. 595, p. A2, 2016.

[31] G. Cormode and S. Muthukrishnan, "An improved data stream summary: the count-min sketch and its applications," *J. Algorithms*, vol. 55, no. 1, pp. 58–75, 2005.

[32] S. Har-Peled and M. Sharir, "Relative (p, ϵ)-approximations in geometry," *Discrete & Computational Geometry*, vol. 45, no. 3, pp. 462–496, 2011.

[33] Y. Li, P. M. Long, and A. Srinivasan, "Improved bounds on the sample complexity of learning," *J. Comput. Syst. Sci.*, vol. 62, no. 3, pp. 516–527, 2001.

[34] N. Shrivastava, C. Buragohain, D. Agrawal, and S. Suri, "Medians and beyond: new aggregation techniques for sensor networks," in *SenSys*, 2004, pp. 239–249.

[35] Z. Wei and K. Yi, "Tight space bounds for two-dimensional approximate range counting," *ACM Trans. Algorithms*, vol. 14, no. 2, pp. 23:1–23:17, 2018.

[36] J. L. Bentley, "Decomposable searching problems," *Inf. Process. Lett.*, vol. 8, no. 5, pp. 244–251, 1979.