Discrete Comput Geom 36:633–655 (2006)
DOI: 10.1007/s00454-006-1269-4

Discrete & Computational
Geometry

Range Counting over Multidimensional Data Streams*

Subhash Suri,[1] Csaba D. Tóth,[2] and Yunhong Zhou[3]

[1]Department of Computer Science, University of California,
Santa Barbara, CA 93106, USA
suri@cs.ucsb.edu

[2]Department of Mathematics, Massachusetts Institute of Technology,
Cambridge, MA 02139, USA
toth@math.mit.edu

[3]Hewlett-Packard Laboratories, 1501 Page Mill Road,
Palo Alto, CA 94304, USA
yunhong.zhou@hp.com

Abstract. We consider the problem of approximate range counting over a stream of d-dimensional points. In the *data stream model* the algorithm makes a single scan of the data, which is presented in an arbitrary order, and computes a compact summary data structure. The summary, whose size depends on the approximation parameter ε, can be used to count the number of points inside a query range within additive error εn, where n is the size of the stream seen so far. We present several results, deterministic and randomized, for both rectangle and halfspace ranges.

1. Introduction

Data streams have emerged as an important paradigm for processing data that arrives and needs to be processed continuously. For instance, telecom service providers routinely monitor packet flows through their networks to infer usage patterns and signs of attack, or to optimize their routing tables. Financial markets, banks, web servers, and news organizations also generate rapid and continuous data streams.

Many data streams are "geometric" in the sense that each datum is best thought of as a point in some coordinate space. For example, IP packets with four important header fields (source address, destination address, protocol, and time stamp) can be mapped uniquely to points in a four-dimensional space. The growing adoption of GPS-based devices is making it possible to localize and track users of laptops or PDAs, drivers of cars, trucks, or emergency vehicles. In other applications, such as environmental monitoring,

* The research by the first two authors was partially supported by NSF Grants CCR-9901958 and ANI-9813723.

astrophysical or geological measurements, location coordinates are an intrinsic part of the data.

In the data stream model the goal is to summarize the stream of points seen so far (or recent past) into a compact synopsis, which can answer useful queries approximately [27]. The model is motivated by the reality that in many of the applications mentioned above, data is too large to store in its entirety or even to scan for real-time query processing. For instance, a telecom network manager may wish to understand the *distribution* of network traffic in the source–destination space. He may want to query the packet log collected at routers: How much TCP data went across a backbone router R that was sent from a subnet S to a subnet D between 1 and 4 p.m. yesterday? A quick size estimate of the packet log shows that a single Internet Service Provider with one hundred 2.4 gigabytes per second routers will need more than 50 terabytes per day to store just the header information of all the packets. Similarly, after a nightly scan of the sky, an astrophysicist may want to obtain, *in retrospect*, spatial range query estimates of the number of events that occurred in certain regions of interest. Motivated by these applications, we consider in this paper the fundamental problem of approximate range counting over streams of d-dimensional point sets.

1.1. *Problem Definition*

We assume that the input stream is a sequence of d-dimensional points, $p_1, p_2, \ldots, p_n, \ldots$, where p_1 is the oldest point and p_n the most recent point. The total number of points is not known in advance, and the stream can be potentially infinite. We want to maintain a data structure that is able to answer range counting queries in the *real RAM model* on the set of points seen so far. The size of our data structure should ideally be independent of n, the number of points seen so far, or at worst it might be polylogarithmic in n.

Many non-trivial problems, including range counting, are impossible to solve *exactly* in the data stream model. In particular, a classical result by Munro and Paterson [26] shows that any algorithm that computes quantiles of a set of n *numbers* in at most p passes over the data requires at least $\Omega(n/p)$ space. Therefore, the data stream algorithms aim for approximate answers using space that is polynomial in ε^{-1} and $\log n$. Indeed, one can compute ε-approximate quantiles of n numbers in one pass using $O(\varepsilon^{-1}\log(\varepsilon n))$ space [13].

A standard tool for compressing a point set for range counting queries is the ε-approximation introduced by Vapnik and Chervonenkis [28]. For any range space with finite VC dimension, a random sample of size $O(\varepsilon^{-2}\log(\varepsilon\delta)^{-1})$ is an ε-approximation with probability at least $1-\delta$. An on-line sampling method, such as the reservoir sampling of Vitter [30], can be used to compute a random sample of the data stream almost linearly in time and space with respect to the sample size. In our view, two key challenges for range counting in data streams are: to find deterministic data stream algorithms with small space complexity, and to find randomized data stream algorithms that improve the naïve space complexity $O(\varepsilon^{-2}\log\varepsilon^{-1})$. To simplify notations and complexity analysis, throughout the paper, we assume that d, the dimension of the space, is a finite constant, and the big-O notation hides a constant which only depends on d.

1.2. *Related Work*

Exact Range Counting [1], [21]. Range searching and counting are fundamental problems in computational geometry, with a long history. The main focus has been on achieving fast query time with as little space as possible. However, since the entire input is stored, the space is at least linear, and the data structure construction requires multiple passes over the data. Among the specific results, Chazelle [9] gives an $O(n)$ size data structure that can answer axis-parallel rectangle range counting queries in the plane in $O(\log n)$ time, and Matoušek [20] gives an $O(n)$ size data structure that can answer simplex queries in the plane in $O(\sqrt{n})$ time. An interested reader should consult one of the surveys [1] or [21] for many related results and extensions to d dimensions.

ε-Approximations and Geometric Discrepancy [10], [23]. An *ε-approximation* of a point set P for a range space $\mathcal{Q}$ is a set $S \subset P$ such that, for any range $Q \in \mathcal{Q}$,

$$\left| \frac{|P|}{|S|} \cdot |S \cap Q| - |P \cap Q| \right| \leq \varepsilon |P|.$$

A range query is answered by counting how many points of the ε-approximation fall in the query range, and scaling up the answer proportionately.

Matoušek [22], [24] has shown (slightly improving an earlier result of Matoušek et al. [25]) that there exists an ε-approximation of size $O(\varepsilon^{-2d/(d+1)})$ for halfspace ranges in $\mathbb{R}^d$, which is the best possible bound apart from constant factors due to discrepancy results of Beck [5], [7], [10]. No polynomial-time algorithm is known for computing an ε-approximation of this size. For any constant $\delta > 0$, an ε-approximation of size $O(\varepsilon^{-2d/(d+1)+\delta} \operatorname{polylog} \varepsilon^{-1})$ can be obtained in $O(n+\varepsilon^{-2}\log^2\varepsilon^{-1})$ time through random sampling [28], and iterative halving steps based on randomized partial coloring by Beck [6], [7], [23] and constructing matchings of low crossing numbers by Matoušek [19].

A random sampling algorithm that picks every point with probability $\Theta(\varepsilon^{-2}\log\varepsilon^{-1}/n)$ provides an ε-approximation with probability close to one [28]. Chazelle and Matoušek [11], [10] have given a deterministic algorithm for computing an $O(\varepsilon^{-2}\log\varepsilon^{-1})$ size ε-approximation in $O(\varepsilon^{-2d}\log^d\varepsilon^{-1}\cdot n)$ time. Recently, Bagchi et al. [3] have adapted this algorithm to the data stream model. They have noticed that it can be implemented so that it reads the input in a single pass, and maintains an ε-approximation of size $O(\varepsilon^{-2}\log\varepsilon^{-1})$. As opposed to the easy-to-implement random sampling of Vitter [30], this *deterministic* algorithm uses $O(\log n \cdot s + s^{d+1/2})$ working space and $O(\log n \cdot s^{d+1})$ per-item processing time, where $s = O(\varepsilon^{-2}\log^{2c} n(\log\log n + \log\varepsilon^{-1}))$, and $c > 1$ is a constant.

Range Counting over Data Streams. In one dimension Greenwald and Khanna [13] construct an ε-approximate quatile summary of size $O(\varepsilon^{-1}\log(\varepsilon n))$ that can also be used for answering range counting queries with an absolute error of εn. We describe their algorithm in Section 2.1 because we use it as a building block in some of our algorithms for axis-aligned box ranges. For a parameter $q \in [0, 1]$ given in advance, Manku et al. [18] can find an ε-approximate q-quantile (an element whose rank is in

the interval $[(q-\varepsilon)n, (q+\varepsilon)n]$) with a randomized algorithm in $O(\varepsilon^{-1}\log^2\varepsilon^{-1} + \varepsilon^{-1}\log^2\log\delta^{-1})$ space with probability $1-\delta$.

Finally, Hershberger et al. [16] present a deterministic sketch of size $O(\varepsilon^{-1}\log^{2d-1}R)$ for the *discrete* version of the axis-aligned box range queries: they assume that input points are from the universe $\{1, 2, \ldots, R\}^d \subset \mathbb{N}^d$. Two recent survey papers [27], [2] are available on a wide range of data stream problems and models.

1.3. *Contribution*

We propose several algorithms for computing small size synopses of d-dimensional streams of points that can give approximate answer to range counting queries.

In Section 2 we consider a *deterministic* sketching algorithm for *axis-aligned box ranges* only. We first present a data stream algorithm that maintains a sketch of size $O(\varepsilon^{-d}\log^d(\varepsilon n))$ in $\mathbb{R}^d$. If we are allowed to make d passes over the data, then we show a deterministic sketch of size $O(\varepsilon^{-1}\log(\varepsilon n)\log^d(\varepsilon^{-1}\log(\varepsilon n)))$. Thus, additional passes can significantly improve the space complexity.

In Section 3 we present a *randomized* algorithm that maintains a weighted ε-approximation of size $O(\varepsilon^{-2d/(d+1)}\log^d(\varepsilon^{2/(d+1)}n)\log\varepsilon^{-1})$ with probability $1-o(1)$ for axis-aligned box ranges. It is an easy combination of our deterministic algorithm (Section 2.2) and a new Chernoff bound (Section 3.1).

In Sections 4 and 5 we present randomized one-pass algorithms for computing ε-approximations that have *optimal* size, up to logarithmic factors. For halfspace ranges in $\mathbb{R}^d$, we can maintain an $O(\varepsilon^{-2d/(d+1)}\log^{d+2}\varepsilon^{-1})$ size weighted ε-approximation with constant probability. This result generalizes to any range space whose dual shatter function is $\pi^*(m) = O(m^d)$. For axis-aligned box ranges in $\mathbb{R}^d$, we can maintain an $O(\varepsilon^{-1}\log^{2d+2}\varepsilon^{-1})$ size ε-approximation with probability greater than $\frac{3}{4}$. In both cases the probability can be increased to $1-\delta$, for any $\delta \in (0,1)$, with an additional $O(\log\delta^{-1})$ factor increase in space. The latter two sketching algorithms maintain almost optimal size ε-approximations at all times. Both use, however, $O(\varepsilon^{-1}\log\varepsilon^{-1}\log(\varepsilon n))$ *merge steps* that are based on sophisticated techniques and may be computationally intensive.

1.4. *Proof Technique*

We use a slight generalization of ε-approximations: a *weighted ε-approximation* of a set P for a range space $\mathcal{Q}$ is a set S where every $s \in S$ also has a weight $w_s \in \mathbb{R}$ such that for every range $Q \in \mathcal{Q}$,

$$\left|\sum_{s\in S\cap Q} w_s - |P\cap Q|\right| \le \varepsilon|P|.$$

In our constructions the sum of weights $\sum_{s\in S} w_s$ always equals the total number of points $|P|$. Specifically, every weighted ε-approximation $S = \{s_i\colon i \in I\}$ corresponds to a *partition* $\{F_i\colon i \in I\}$ of the input set P, such that $s_i \in F_i$ and the weight of s_i is $w_i = |F_i|$, $i \in I$.

Next, we introduce two key concepts, that of representative systems and deficiency, that allow us to achieve almost optimal summary sizes in Sections 3–5. For a point set $P \subset \mathbb{R}^d$, we call a system $\mathcal{F} = (\{F_i : i \in I\}, \{r_i : i \in I\})$ a *representative system* (RS, for short) if $\{F_i : i \in I\}$ is a partition of P and $r_i \in \mathbb{R}^d$, $i \in I$. Instead of storing the entire RS in memory, we only store the pairs $(|F_i|, r_i)$, $i \in I$, in $O(|I|)$ space.

In our deterministic schemes the weighted ε-approximation S is simply the set of representatives with the cardinality of the corresponding sets as weights. The error in our estimate is $||P \cap Q| - \sum_{i:\, r_i \in Q} |F_i||$, which equals the number of *misrepresented* points. We call a data point p *misrepresented* for a query range Q, if $p \in Q$ but the representative of p is not in Q, or vice versa. Therefore, the representatives give a weighted ε-approximation for $\mathcal{Q}$ if the system $\mathcal{F}$ satisfies the following ε-deficiency property for the range space $\mathcal{Q}$:

> A representative system $\mathcal{F}$ over a point set P is **ε-deficient** for $\mathcal{Q}$, if for any range $Q \in \mathcal{Q}$, the total number of points separated from their representative by Q is at most $\varepsilon|P|$. Formally,
>
> $$\sum_{i \in I} |\{p \in F_i : (p \in Q) \neq (r_i \in Q)\}| \leq \varepsilon |P|.$$

In our randomized schemes we actually use the representatives only to maintain the partition. Our weighted ε-approximation is a set of samples $S = \{s_i : i \in I\}$, such that each s_i is a point chosen uniformly at random from F_i. Thus, we maintain two RSs on the same partition $\{F_i : i \in I\}$: the system $\mathcal{F}$ and a system of samples $(\{F_i : i \in I\}, \{s_i : i \in I\})$. We show (in Lemma 3.2 below) that the representatives form a weighted ε-approximation of $|P \cap Q|$ with probability close to one if $\mathcal{F}$ is $O(\varepsilon^\alpha)$-deficient for $\mathcal{Q}$ for some $\alpha \in (0, 2)$, and $|F_i| = O(\varepsilon^{2-\alpha-o(1)}n)$ for every $i \in I$.

The ε-deficiency property for halfspaces in $\mathbb{R}^d$ is also related to *simplicial partitions of low crossing numbers*, a powerful technique developed by Matoušek [19], originally for exact range searching (and, in turn, related to the so-called ε-cuttings). Such a simplicial partition of a point set P in $\mathbb{R}^d$ is a system $(\{F_i : i = 1, 2, \ldots, k\}, \{R_i : i = 1, 2, \ldots, k\})$ of size k, $k \in \mathbb{N}$, where

1. $\{F_i : i = 1, 2, \ldots, k\}$ is a partition of P such that $|F_i| = \Theta(n/k)$ for every $i = 1, 2, \ldots k$,
2. $F_i \subset R_i$ such that R_i is a simplex, and
3. any hyperplane intersects only $O(k^{(d-1)/d})$ simplices of $\{R_i : i = 1, 2, \ldots, k\}$.

If we choose an arbitrary representative point from each set F_i of a Matoušek partition in $\mathbb{R}^d$, then we obtain a representative system satisfying the $O(k^{-1/d})$-deficiency property for halfspaces. This forms a weighted $O(k^{-1/d})$-approximation [19], since the approximation error is bounded by the number of points lying in simplices along the query hyperplane, which totals to $O(k^{(d-1)/d} \cdot n/k) = O(k^{-1/d} \cdot n)$ (see Fig. 1). If we choose a random sample point from each F_i, $i = 1, 2, \ldots, k$, of a Matoušek partition, then we obtain an $O(k^{-1/d})$-deficient RS, again, but it now forms a weighted $O(k^{-(d+1)/2d})$-approximation with large probability (see jittered sampling [7], [10]).

Off-line algorithms for constructing a Matoušek partition use numerous passes over the input and do not seem to adapt to the data stream model. We maintain, instead, random samples $s_i \in F_i$ from an $O(k^{-1/d})$-deficient representative system $\mathcal{F} = (\{F_i : i \in I\},$

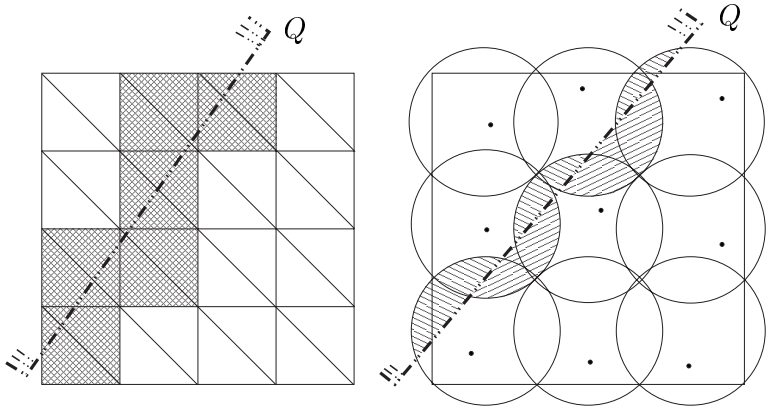

Fig. 1. In a simplicial partition the approximation error comes from the simplices pierced by the boundary of a query Q (left). The right figure shows parts of the sets in a representative system misrepresented for a halfspace Q.

$\{r_i\colon\ i \in I\}$) with weights $|F_i| = \Theta(n/k)$. A hyperplane is allowed to cross the convex hull of any number of sets F_i. Our key result (Lemma 3.2) guarantees that our samples achieve the same approximation quality as samples from an equal size simplicial partition with low crossing number.

2. Deterministic Rectangle Range Counting

2.1. *One Dimension*: *Quantile Summaries*

The ε-approximate range counting over one-dimensional data streams is equivalent to the ε-approximate *quantile summaries*. Greenwald and Khanna [13] present a deterministic algorithm for building a data structure of size $O(\varepsilon^{-1}\log(\varepsilon n))$ that can report the rank of any query point $q \in \mathbb{R}$ among the n points seen so far with an error within the interval $[-\varepsilon n, 0]$. We refer to this algorithm as GK(ε) in what follows.

The algorithm GK(ε) maintains a list of triples $((r_i, g_i, \Delta_i),\ i = 1, 2, \ldots, k)$, $r_1 \le r_2 \le \cdots \le r_k$, corresponding to a representative system $\mathcal{F} = (\{F_i\colon\ i = 1, 2, \ldots, k\}, \{r_i\colon\ i = 1, 2, \ldots, k\})$ of size $k = O(\varepsilon^{-1}\log(\varepsilon n))$, where $g_i = |F_i|$ and Δ_i is an error term. Their representative system $\mathcal{F}$ satisfies two properties: (1) r_i is the rightmost point of F_i and (2) $\mathcal{F}$ is ε-deficient for halfline ranges $\mathcal{Q}^- = \{(-\infty, q]\colon\ q \in \mathbb{R}\}$. This immediately implies that the GK(ε) data structure can answer grounded range counting queries $Q \in \mathcal{Q}^-$ with an error within the interval $[-\varepsilon n, 0]$ (there is no over-counting), therefore it can also answer interval range counting queries with an absolute error of at most εn.

We point out two more characteristics of the GK(ε) algorithm, as we use it as a building block below. First, the key operations in the construction of the summary GK(ε) are (i) inserting a new one-element set F_i, and (ii) forming the union $F_i \cup F_j$ of two sets F_i and F_j such that $r(F_i \cup F_j) = \max\{r_i, r_j\}$. Second, the GK($\varepsilon$) algorithm combines

multiple merge operations into one single *compress* operation which is called periodically after every $1/2\varepsilon$ input points. The amortized per-item processing time of the algorithm is $O(\log(\varepsilon^{-1}\log(\varepsilon n)) + \log(\varepsilon n))$.

2.2. *Cross Product of Quantile Summaries*

In this section we present a *deterministic* data stream algorithm for axis-parallel box range counting queries. In the plane it is essentially the cross product of two GK(ε) summaries. In Section 3 we present more efficient *randomized* summaries based on this algorithm.

Theorem 2.1. *For a stream of points in the plane, we can deterministically maintain a weighted ε-approximation of size $O(\varepsilon^{-2}\log^2(\varepsilon n))$ for axis-aligned rectangle ranges.*

Proof. We simultaneously run two Greenwald–Khanna algorithms, $\mathrm{GK}_x(\varepsilon/2)$ and $\mathrm{GK}_y(\varepsilon/2)$, on the x- and the y-coordinates of the data point. Each generates a partition of the ground set into $O((1/\varepsilon)\log(\varepsilon n))$ subsets, $\{G_1, G_2, \ldots, G_{k_x}\}$ and $\{H_1, H_2, \ldots, H_{k_y}\}$. We store in memory the cardinalities $|G_i|$ and $|H_j|$, and $r(G_i) \in \mathbb{R}$ and $r(H_j) \in \mathbb{R}$, which are the maximal values of each subset according to the x- and y-coordinates, respectively.

We also maintain the partition formed by the cross product of $\mathrm{GK}_x(\varepsilon/2)$ and $\mathrm{GK}_y(\varepsilon/2)$: $\{G_i \cap H_j\colon\ i = 1, 2, \ldots, k_x,\ j = 1, 2, \ldots, k_y\}$. Let the representative of each set $G_i \cap H_j$ be the point $r(G_i \cap H_j) := (r(G_i), r(H_j)) \in \mathbb{R}^2$. This means that all points of $G_i \cap H_j$ lie in a box whose upper right corner is $r(G_i \cap H_j)$ (since $r(G_i)$ is the x-coordinate of the rightmost point of G_i and $r(H_j)$ is the y-coordinate of the highest point of H_j). The two operations of the GK($\varepsilon/2$) algorithm can be easily represented to update the cross product: (1) when a new point arrives, it is placed in a one-element set G_{k_x} in $\mathrm{GK}_x(\varepsilon/2)$ and H_{k_y} in $\mathrm{GK}_y(\varepsilon/2)$, so we insert into the cross product the singleton set $G_{k_x} \cap H_{k_y}$ as well as empty sets $G_{k_x} \cap H_j$, $G_i \cap H_{k_y}$ for all $j = 1, 2, \ldots, k_y - 1$ and $i = 1, 2, \ldots, k_x - 1$; (2) whenever the summary $\mathrm{GK}_x(\varepsilon/2)$ forms the union of two sets G_i and $G_{i'}$, we form the union of $G_i \cap H_j$ and $G_{i'} \cap H_j$ for every $j = 1, 2, \ldots, k_y$. Similarly, whenever $\mathrm{GK}_y(\varepsilon/2)$ forms the union of H_j and $H_{j'}$, we form the union of $G_i \cap H_j$ and $G_i \cap H_{j'}$ for every $i = 1, 2, \ldots, k_x$.

For a query rectangle $Q = [q_1, q_2] \times [q_3, q_4]$, we report the sum of the cardinalities of sets $G_i \cap H_j$ whose representative $r(G_i \cap H_j)$ lies in Q.

Due to the $(\varepsilon/2)$-deficiency property of GK($\varepsilon/2$), there are at most $(\varepsilon/2)n$ misrepresented points for any axis-parallel slab query. The number of misrepresented points is at most $2(\varepsilon/2)n = \varepsilon n$ for any axis-parallel rectangle query Q because if p is misrepresented for Q, then it is also misrepresented for at least one of the axis-parallel slabs spanned by Q. Therefore, the set of representatives $r(G_i \cap H_j)$ with weights $|G_i \cap H_j|$, $\forall i, j$, forms a weighted ε-approximation for axis-parallel rectangles. □

Since every operation of $\mathrm{GK}_x(\varepsilon/2)$ and $\mathrm{GK}_y(\varepsilon/2)$ is followed by k_y and k_x operations in the cross product, the amortized per-item processing time of the algorithm is $O(\varepsilon^{-1}\log n \log(\varepsilon n))$. The above theorem generalizes to arbitrary Euclidean space $\mathbb{R}^d$, $d \in \mathbb{N}$. The proof is straightforward and is omitted.

Theorem 2.2. *For a stream of points in $\mathbb{R}^d$, we can deterministically maintain a weighted ε-approximation of size $O(\varepsilon^{-d}\log^d(\varepsilon n))$ for axis-parallel box ranges. The algorithm requires the same working space and $O(\varepsilon^{1-d}\log n\log^d(\varepsilon n))$ amortized per-item processing time.*

2.3. *Range Counting in d Passes*

In many data stream applications, $\Omega(\varepsilon^{-d})$ storage can be prohibitive. For instance, with $\varepsilon = 10^{-3}$ or 10^{-5}, the space requirement may be too large even in two dimensions. In this section we show that by allowing *d passes* over the data, we can significantly improve the space requirement of a deterministic data stream algorithm. Multipass algorithms are meaningful when the data is stored on some external tape drives, and sequential access is the only practical approach to the data [15], [27]. Specifically, we describe a data structure that can answer rectangular range counting queries with εn absolute error using $O(\varepsilon^{-1}\,\mathrm{polylog}(\varepsilon^{-1}, \varepsilon n))$ space. Because our algorithm will make multiple passes, we assume that the size of the data stream is known in advance.

Theorem 2.3. *For n points in the plane, we can deterministically construct in* two passes *over the data a summary of size*

$$O\left(\frac{1}{\varepsilon}\log(\varepsilon n)\cdot\log^2\left(\frac{1}{\varepsilon}\log(\varepsilon n)\right)\right)$$

that can answer axis-parallel rectangle range counting queries with an absolute error of at most εn.

Proof. Let $\varepsilon_1 \in (0, \varepsilon)$ be a constant to be specified later. In the first pass we build a sketch according to the x-coordinates of the points with the algorithm GK(ε_1), and obtain a partition into $\ell = O(\varepsilon_1^{-1}\log(\varepsilon_1 n))$ subsets. By adding empty subsets, if necessary, we may assume that ℓ is a power of 2 and $\ell \geq 2$. Let $r_1, r_2, \ldots, r_\ell$ denote the representative elements in increasing order, and let $r_0 := -\infty$. By the ε_1-deficiency property, every interval $(r_{i-1}, r_i]$ corresponds to a vertical slab in the plane with at most $\varepsilon_1 n$ data points, $i = 1, 2, \ldots, \ell$.

After the first pass over the data is completed, we build a binary tree T of height $\log \ell$ on the intervals $(r_{i-1}, r_i]$, $i = 1, 2, \ldots, \ell$. Every node of T corresponds to a vertical slab: Every leaf corresponds to the slabs spanned by one interval $(r_{i-1}, r_i]$, every non-leaf node v corresponds to the union of the slabs spanned by the (consecutive) intervals at the descendants of v. Thus, at level i of the tree, we have $\ell/2^i$ slabs, which partition the plane. If we denote by $n_{i,j}$ the number of points in slab j at level i, where $j = 1, 2, \ldots, \ell/2^i$, then we have $\sum_j n_{i,j} = n$, $\forall i$. Note that after the first pass, we do not know the exact value of $n_{i,j}$, but GK(ε_1) gives us an estimate $n^*_{i,j}$ such that $|n^*_{i,j} - n_{i,j}| \leq \varepsilon_1 n$.

In the second pass we build a sketch using the y-coordinates in each vertical slab corresponding to a node of T whenever $n^*_{i,j} > \varepsilon_1 n$, and do nothing otherwise. For a slab containing $n_{i,j}$ points, we apply the algorithm GK($\varepsilon_1 n/n^*_{i,j}$). This ensures that in each slab, we can answer range counting queries with respect to *horizontal query* slabs with

an absolute error of $(n_{i,j}/n^*_{i,j})\varepsilon_1 n \leq 2\varepsilon_1 n$. The size of the secondary data structure for a vertical slab of size $n_{i,j}$ is $O((n^*_{i,j}/\varepsilon_1 n) \cdot \log(\varepsilon_1 n))$. At level i of the tree T, the total size is

$$\sum_{j=1}^{\ell/2^i} O\left(\frac{n^*_{i,j}}{\varepsilon_1 n} \cdot \log(\varepsilon_1 n)\right) = O\left(\frac{1}{\varepsilon_1} \log(\varepsilon_1 n)\right).$$

Thus the total size of all secondary data structures is $O(\log \ell \cdot (1/\varepsilon_1) \cdot \log(\varepsilon_1 n))$.

It remains to show how the primary and secondary data structures are used for answering a rectangular range counting query $Q = [q_1, q_2] \times [q_3, q_4]$. We locate the intervals $(r_{a-1}, r_a]$ and $(r_b, r_{b+1}]$ containing q_1 and q_2, respectively. We can cover the vertical slab spanned by the interval $(r_a, r_b]$ with $2 \log \ell$ disjoint vertical slabs corresponding to nodes of T. In each vertical slab we query the interval $[q_3, q_4]$, and report the sum of approximate answers.

There are two sources of error in our estimate for the range query: (1) we have ignored the points of Q that lie in the vertical slabs of $(r_{a-1}, r_a]$ and $(r_b, r_{b+1}]$ and (2) we have an absolute error of $2\varepsilon_1 n$ in each of the $2 \log \ell$ vertical slabs. The total absolute error, therefore, is bounded by $4 \log \ell \cdot \varepsilon_1 n$. Let us choose ε_1 such that $4 \log \ell \cdot \varepsilon_1 = \varepsilon$, then the absolute error is at most εn. Given the value of ε_1 and ℓ, the total sketch size is thus bounded by

$$\begin{aligned} O\left(\log \ell \cdot \frac{1}{\varepsilon_1} \cdot \log(\varepsilon_1 n)\right) &= O\left(\log^2 \ell \cdot \frac{1}{\varepsilon} \cdot \log(\varepsilon n)\right) \\ &= O\left(\frac{1}{\varepsilon} \cdot \log(\varepsilon n) \cdot \log^2\left(\frac{1}{\varepsilon} \log(\varepsilon n)\right)\right). \end{aligned}$$ □

The total processing time in the first pass is $O(n \log n)$. The total processing time for the second pass is

$$\sum_{i=0}^{\log \ell} \sum_{j=0}^{\ell/2^i} O(n_{ij} \log n_{ij}) = \sum_{i=0}^{\log \ell} \sum_{j=0}^{\ell/2^i} O(n_{ij} \log n) = \sum_{i=0}^{\log \ell} O(n \log n) = O(n \log n \log \ell).$$

Thus the amortized per-item processing time of the algorithm is $O(\log n \log(\varepsilon^{-1} \log(\varepsilon n)))$. Our construction generalizes to $\mathbb{R}^d$ in a straightforward manner, and we omit the details of the proof.

Theorem 2.4. *For n points in $\mathbb{R}^d$, we can deterministically construct in d* passes *over the data a sketch of size*

$$O\left(\frac{1}{\varepsilon} \log(\varepsilon n) \cdot \log^{2d-2}\left(\frac{1}{\varepsilon} \log(\varepsilon n)\right)\right)$$

that can answer axis-parallel rectangle range counting queries with an absolute error of at most εn. The algorithm requires the same working space and $O(\log n \log^{d-1}(\varepsilon^{-1} \log(\varepsilon n)))$ amortized per-item processing time.

3. Randomized Range Counting

In our randomized sketching algorithms, we generate a representative system ($\{F_i\}$, $\{r_i\}$: $i \in I$) over the input and for every $i \in I$ we maintain a random sample point $s_i \in F_i$. The points r_i are used for updating the representative system as new elements of the data stream arrive, while the sample points s_i form an ε-approximation to answer range counting queries. We can bound the approximation quality with the following Chernoff bound.

3.1. *A Chernoff Bound*

Our Chernoff bound is independent of the dimension of the space. It is a weighted variant of a classical Chernoff bound in [14].

Theorem 3.1. *For a natural number m, assume that we are given probabilities $p_1, p_2, \ldots, p_m$, and real-valued weights $w_1, w_2, \ldots, w_m$, where $|w_i| < W$ for a fixed constant $W > 0$. Let $X_1, X_2, \ldots, X_m$ be mutually independent random variables such that $\Pr(X_i = -p_i) = 1 - p_i$ and $\Pr(X_i = 1 - p_i) = p_i$. Set $X = \sum_{i=1}^{m} w_i X_i$ and $M = \sum_{i=1}^{m} |w_i| p_i$. For any $\delta \in (0, 1)$, we have*

$$\Pr(|X| \geq \delta M) \leq 2\, e^{-\delta^2 M/3W}.$$

Proof. For any $t > 0$, we have the following chain of inequalities:

$$\begin{aligned}
\Pr(X \geq \delta M) &= \Pr(e^{tX} \geq e^{t\delta M}) \leq e^{-t\delta M} E[e^{tX}] \\
&= e^{-t\delta M} \prod_{i=1}^{m} E[e^{tw_i X_i}] = e^{-t\delta M} \prod_{i=1}^{m} e^{-tw_i p_i} (1 + p_i(e^{tw_i} - 1)) \\
&\leq e^{-t\delta M} \prod_{i=1}^{m} e^{-tw_i p_i + p_i(e^{tw_i} - 1)} = e^{-t\delta M + \sum_{i=1}^{m} p_i (e^{tw_i} - 1 - tw_i)}.
\end{aligned}$$

For $|x| \leq 1$, it is easy to verify the following inequality: $e^x - 1 - x \leq \frac{3}{4}x^2$. Let $t = 2\delta/3W$, then $|tw_i| \leq tW \leq 1, \forall i$. We bound the exponent of the last expression in the above inequality:

$$\begin{aligned}
-t\delta M + \sum_{i=1}^{m} p_i (e^{tw_i} - 1 - tw_i) &\leq -t\delta M + \sum_{i=1}^{m} p_i (\tfrac{3}{4} t^2 w_i^2) \\
&= -t\delta M + \tfrac{3}{4} t^2 \sum_{i=1}^{m} p_i w_i^2 \leq -t\delta M + \tfrac{3}{4} t^2 MW \\
&= -\frac{\delta^2 M}{3W}.
\end{aligned}$$

Thus $\Pr(X \geq \delta M) \leq e^{-\delta^2 M/3W}$. Through an almost identical calculation with $-X$ rather than X, we have $\Pr(-X \geq \delta M) \leq e^{-\delta^2 M/3W}$. The combination of the above two

inequalities gives the desired result:

$$\Pr(|X| \geq \delta M) = \Pr(X \geq \delta M) + \Pr(-X \geq \delta M) \leq 2e^{-\delta^2 M/3W}. \qquad \square$$

We use this Chernoff bound in the main lemma below, which is the key tool in our analysis.

Lemma 3.2. *Assume we are given a set P of n points in a range space $(X, \mathcal{Q})$ and an α-deficient representative system $(\{F_i\colon i \in I\}, \{r_i\colon i \in I\})$ over P such that $|F_i| \leq \beta n$ for every $i \in I$, where $\alpha, \beta > 0$. For every $i \in I$, independently, let s_i be a sample point chosen uniformly at random from F_i. Then for any range $Q \in \mathcal{Q}$, the sum $\sum_{i \in I, s_i \in Q} |F_i|$ approximates $|P \cap Q|$ with an absolute error of at most εn with probability at least $1 - 2e^{-\varepsilon^2/3\alpha\beta}$.*

Proof. Fix a range $Q \in \mathcal{Q}$ from the range space. For every $i \in I$, we split F_i into two disjoint subsets A_i and B_i as follows:

$$A_i = \{p \in F_i\colon (p \in Q \text{ and } r_i \in Q) \text{ or } (p \notin Q \text{ and } r_i \notin Q)\} \quad \text{and} \quad B_i = F_i \backslash A_i.$$

That is, r_i represents the points in A_i correctly but it misrepresents the points in B_i. Let $a_i = |A_i|$, $b_i = |B_i|$, and $w_i = a_i + b_i$. By the α-deficiency property, $\sum_{i=1}^{k} b_i \leq \alpha n$.

For every $i \in I$, we set $p_i = b_i/w_i$ and define a random variable X_i such that $X_i = 1 - p_i$ if $s_i \in B_i$ and $X_i = -p_i$ if $s_i \in A_i$. Observe that $\Pr(X_i = 1 - p_i) = p_i$ and $\Pr(X_i = -p_i) = 1 - p_i$. Note that the random variables X_i, $i \in I$, are mutually independent. The total error of our answer for Q is

$$\begin{aligned} |P \cap Q| - \sum_{i \in I, s_i \in Q} |F_i| &= \sum_{i\colon r_i \in Q} \left(\sum_{s_i \notin Q} a_i - \sum_{s_i \in Q} b_i \right) + \sum_{i\colon r_i \notin Q} \left(\sum_{s_i \notin Q} b_i - \sum_{s_i \in Q} a_i \right) \\ &= \sum_{i\colon r_i \in Q} w_i X_i - \sum_{i\colon r_i \notin Q} w_i X_i. \end{aligned}$$

Let $X = \sum_{i=1}^{k} w_i^* X_i$ where $w_i^* = w_i$ if $r_i \in Q$ and $w_i^* = -w_i$ otherwise. Thus $X = |P \cap Q| - \sum_{i \in I, s_i \in Q} |F_i|$ and it denotes the total error of the query answer. Let $M = \sum_{i=1}^{k} |w_i^*| p_i$, then $M = \sum_{i=1}^{k} w_i p_i = \sum_{i=1}^{k} b_i \leq \alpha n$. We set $\delta = \varepsilon n/M$ and $W = \beta n$, and apply Theorem 3.1 (Chernoff bound):

$$\Pr(|X| \geq \varepsilon n) \leq 2\, e^{-\delta^2 M/3W} \leq 2\, e^{-\varepsilon^2 n^2/3MW} \leq 2\, e^{-\varepsilon^2/3\alpha\beta}. \qquad \square$$

3.2. *Randomized Rectangular Range Counting*

We apply Lemma 3.2 to guarantee that a random sample drawn from a cross-product partition of one-dimensional quantile summaries can approximate the number of points in a query rectangle.

Theorem 3.3. *For a stream of points in $\mathbb{R}^2$, we can maintain a summary of size*

$$O(\varepsilon^{-4/3}\log^2(\varepsilon^{2/3}n))$$

that answers axis-parallel rectangle range counting queries with at most εn absolute error with probability $1-o(1)$.

Proof. Let $\varphi = \varepsilon^{2/3}$ and $w = \varphi^2 n/\log^2(\varphi n)$. Similarly to the proof of Theorem 2.1, we maintain the cross-product partition of two data structures $\mathrm{GK}_x(\varphi)$ and $\mathrm{GK}_y(\varphi)$. By the φ-deficiency property, the size of each set of the partitions G_i and H_j is below φn. We modify the cross-product partition so that if a set $G_i \cap H_j$ with more than w elements will be generated by a union operation, then we postpone the union operation and keep multiple sets to represent $G_i \cap H_j$. Therefore $G_i \cap H_j$ may contain multiple sets with the cardinality of each set $\leq w$. Since n (and also w) is monotone increasing as new points are streaming in, we merge two small sets later when their union becomes $\leq w$. We obtain a partition $\{F_i\colon i \in I\}$, whose size is still $|I| = O(\varphi^{-2}\log^2(\varphi n))$, asymptotically the same as the size of the cross product partition.

Every F_i is part of a subset $G_j \cap H_h$, and so we let $r_i = r(G_j \cap H_h)$. Since at most φn points are misrepresented by the r_i's with respect to any vertical or horizontal slab, at most $2\varphi n$ points are misrepresented by the r_i's with respect to axis-aligned boxes. That is, $(\{F_i\colon i = 1, 2, \ldots k\}, \{r_i : i = 1, 2, \ldots k\})$ is (2φ)-deficient for axis-aligned rectangles.

We choose a sample point $s_i \in F_i$ uniformly at random from every F_i. A sample point can easily be maintained under the insertion and union operation. From every new one-element set F_i, we choose a unique sample $s_i \in F_i$, and for a union $F_h = F_i \cup F_j$, we choose $s_h \in \{s_i, s_j\}$ randomly with probabilities $|F_i|/|F_h|$ and $|F_j|/|F_h|$, respectively. For a query rectangle Q, we report $\sum_{i:s_i\in Q}|F_i|$. Observe that the expected value of our answer is exactly the number of points in Q.

It remains to show that the absolute error is at most εn with high probability. We apply Lemma 3.2 to our partition $\{F_i\colon i \in I\}$ with $\alpha = 2\varphi$ and $\beta = \varphi^2 \cdot \log^{-2}(\varphi n)$. We conclude, that for every query rectangle Q, the probability that the absolute error $|\sum_{i\colon s_i\in Q}|F_i| - |P \cap Q||$ is above εn is bounded by

$$2e^{-\varepsilon^2/6\varphi\beta} = 2e^{-(\varepsilon^2\log^2(\varphi n))/6\varphi^3} = 2e^{-(\log^2(\varepsilon^{2/3}n))/6} = o(1).$$

Thus each axis-aligned rectangle range counting query has absolute error bounded by εn with probability $1-o(1)$. □

The techniques used for $\mathbb{R}^2$ readily generalize to any finite-dimensional Euclidean space. The next theorem is an extension of Theorem 3.3.

Theorem 3.4. *For a stream of points in $\mathbb{R}^d$, we can maintain a summary of size*

$$O(\varepsilon^{-2d/(d+1)}\log^d(\varepsilon^{2/(d+1)}n))$$

that answers axis-aligned box range counting queries with εn absolute error with probability $1-o(1)$.

Proof. Let $\varphi = \varepsilon^{2/(d+1)}$ and $w = \varphi^d n/\log^d(\varphi n)$. As in the proof of Theorem 3.3, we maintain the cross-product of d data structures $\mathrm{GK}_{x_i}(\varphi)$, one for each coordinate axis. We keep the cardinality of each partition set bounded by w by postponing the union operation if a set size would exceed w. It is easy to verify that the size of the partition is bounded by $O(\varphi^{-d} \cdot \log^d(\varphi n))$. It is also easy to check that the cross-product representative system is $(d\varphi)$-deficient for axis-aligned rectangle ranges. We apply Lemma 3.2 with $\alpha = d\varphi$ and $\beta = \varphi^d \cdot \log^{-d}(\varphi n)$ to the RS and the probability for the absolute error to be over εn is bounded by

$$2e^{-\varepsilon^2/(3(d\varphi)\beta)} = 2e^{-(\varepsilon^2 \log^d(\varphi n))/3d\cdot\varphi^{d+1}} = 2e^{-(\log^d(\varepsilon^{2/(d+1)}n))/3d} = o(1). \qquad \square$$

Similarly to the scheme in Theorem 2.2, it is easy to verify that the above algorithm takes $O(\varepsilon^{-2d/(d+1)} \log^d(\varepsilon^{2/(d+1)}n))$ working space and $O(\varepsilon^{-2(d-1)/(d+1)} \log n \log^d(\varepsilon^{2/(d+1)}n))$ amortized per-item processing time.

3.3. *Extension to ε-Approximations*

In Theorem 3.3 we proved that one can maintain a weighted RS that can approximate *any* rectangular range with εn error with probability $1 - o(n)$. This is not necessarily an ε-approximation, which has to approximate *all* rectangles *simultaneously* with at most εn absolute error. Since axis-parallel rectangle ranges have bounded VC-dimension, we can extend Theorem 3.3 to obtain an ε-approximation with probability close to 1 at the expense of an $O(\log \varepsilon^{-1})$ factor increase in space.

Theorem 3.5. *For a stream of points in $\mathbb{R}^d$, we can maintain a weighted ε-approximation of size*

$$O\left(\varepsilon^{-2d/(d+1)} \log^d(\varepsilon^{2/(d+1)}n) \log \frac{1}{\varepsilon}\right)$$

for axis-aligned box ranges with probability $1 - o(1)$.

Proof. We run the data stream algorithm in the proof of Theorem 3.4 with parameter $\varepsilon/2$ instead of ε, and $\varphi = (\varepsilon/2)^{2/(d+1)} \log^{-1/(d+1)}(18d^2/\varepsilon)$. We obtain an RS of size $O(\varphi^{-d} \cdot \log^d(\varphi n))$ such that the probability that it approximates an axis-aligned box with less than $(\varepsilon/2)n$ absolute error is above $1 - \varepsilon^{3d} o(1)$.

Consider the set P_n of the first n points of the stream. Since the rectangular ranges have bounded VC-dimensions, there exists an $(\varepsilon/2)$-approximation R of size $\ell = O(\varepsilon^{-2} \log \varepsilon^{-1})$ by a celebrated result of Vapnik and Chervonenkis [28]. Let $\mathcal{Q}_R$ be a maximal family of axis-aligned boxes such that the intersection sets $Q \cap R$, $Q \in \mathcal{Q}_R$, are pairwise disjoint. The cardinality $|\mathcal{Q}_R|$ is bounded by $O(\ell^d) = O(\varepsilon^{-2d} \log^d \varepsilon^{-1})$. (In general, it is bounded by the (*primal*) *shatter function* $\pi(\ell)$ of the range space.)

The probability that our RS approximates all boxes of $\mathcal{Q}_R$ simultaneously with an absolute error of at most $\varepsilon n/2$ is at least $1 - O(\ell^d)\varepsilon^{3d} o(1) = 1 - o(1)$. Since an $(\varepsilon/2)$-approximation of an $(\varepsilon/2)$-approximation is an ε-approximation, our RS is an ε-approximation with probability at least $1 - o(1)$ at any time $n \in \mathbb{N}$. $\square$

4. Randomized Halfspace Range Counting

Our main result in this section is a weighted ε-approximation of size $O(\varepsilon^{-2d/(d+1)} \log^{d+1} \varepsilon^{-1})$ for halfspace ranges that can be maintained over d-dimensional point streams. Our algorithm crucially depends on merging two summaries of the same size into one, *without significantly worsening the approximation error.*

4.1. *The Merge Step*

To merge two summaries we can apply a result of Chazelle and Welzl [12] (see Theorem 5.17 in [23]) about matchings with low crossing numbers, which is tight apart from a constant factor.

Lemma 4.1. *Given $2k$ points in $\mathbb{R}^d$, there is a matching M of size k such that any hyperplane crosses at most $c_1 \cdot k^{(d-1)/d}$ edges of M, where $c_1 > 0$ is a constant depending on d only.*

There are deterministic algorithms for constructing a matching described in Lemma 4.1, based on a de-randomization of a randomized polynomial-time algorithm [23]. The deterministic merge step, however, requires $O(k^d)$ memory in d-dimensions. A matching satisfying a slightly weaker bound, however, can be computed efficiently by a result due to Matoušek [19].

Lemma 4.2. *We are given $2k$ points in $\mathbb{R}^d$. For any constant $\delta > 0$, one can compute in $O(k \log k)$ time a matching M of size k such that any hyperplane crosses at most $c_1 \cdot k^{(d-1)/d+\delta}$ edges of M, where $c_1 > 0$ is a constant depending on d only.*

For simplicity, we use the bound of Lemma 4.1 in our computations. Any implementation using this merge step must apply an efficient subroutine with the slightly weaker bound of Lemma 4.2. We next describe how we merge two samples of size k into one sample of size k.

Lemma 4.3. *Assume that we are given a representative system $\mathcal{F} = (\{F_i\colon i = 1, 2, \ldots, 2k\}, \{r_i\colon i = 1, 2, \ldots, 2k\})$, $k \in \mathbb{N}$, over a ground set of size n in $\mathbb{R}^d$. Furthermore, $\mathcal{F}$ is α-deficient for halfspaces and $|F_i| \leq \beta n$ for every $i = 1, 2, \ldots, 2k$, where $\alpha, \beta > 0$. Then one can construct an $(\alpha + c_1 \cdot k^{(d-1)/d}\beta)$-deficient representative system of size k for halfspaces.*

Proof. We invoke Lemma 4.1 on the set of $2k$ representatives $\{r_i\colon i = 1, 2, \ldots, 2k\}$. We define the representative system $\mathcal{G} = (\{G_h\colon h = 1, 2, \ldots, k\}, \{q_h\colon h = 1, 2, \ldots, k\})$ as follows. Every matching edge $r_i r_j \in M$ determines a set $G_h = F_i \cup F_j$, and the corresponding representative q_h is chosen uniformly at random from $\{r_i, r_j\}$.

Fix a halfspace $Q \in \mathcal{Q}$, and let Q_0 denote the boundary hyperplane of Q. At most αn points are misrepresented by the representative r_i of $\mathcal{F}$. If a line segment $r_i r_j \in M$ does not cross Q_0, then $q_h \in \{r_i, r_j\}$ misrepresents the same points in $\mathcal{G}$ as r_i and r_j jointly

misrepresent in $\mathcal{F}$. If $r_i r_j$ crosses Q_0, however, then $q_h \in \{r_i, r_j\}$ misrepresents at most βn more points in $\mathcal{G}$ than r_i and r_j jointly misrepresent in $\mathcal{F}$. Since at most $c_1 \cdot k^{(d-1)/d}$ edges of M cross Q_0, the system $\mathcal{G}$ misrepresents at most $\alpha n + c_1 \cdot k^{(d-1)/d}\beta n$ points for any Q, as required. □

4.2. *Summarizing Fixed Size Streams*

By a well-known result of Vapnik and Chervonenkis [28], a random sample of size $\Theta(\varepsilon^{-2}\log\varepsilon^{-1})$ is an ε-approximation for halfspaces in $\mathbb{R}^d$ with probability close to 1. Since an ε_2-approximation of an ε_1-approximation is an $(\varepsilon_1+\varepsilon_2)$-approximation, it is enough to find an ε_2-approximate data stream algorithm whose input is a sequence of $\Theta(d\varepsilon_1^{-2}\log\varepsilon_1^{-1})$ random samples.

First we show a weak bound where we assume that the total number of points in the stream is known in advance. We use this algorithm to compress a stream *of* $\Theta(d\varepsilon_1^{-2}\log\varepsilon_1^{-1})$ *sample points* in the next subsection.

Theorem 4.4. *For a stream of N points in $\mathbb{R}^d$ (where $N \geq 2\varepsilon^{-2d/(d+1)}$ is known in advance), we can construct a sample of size $O(\varepsilon^{-2d/(d+1)}\log^d(\varepsilon^{2d/(d+1)}N))$ that can answer halfspace range counting queries such that the absolute error is at most εN with probability at least $1 - 2\exp(-2\log(\varepsilon^{2d/(d+1)}N)) \geq \frac{1}{2}$.*

Proof. Let $\varphi = \varepsilon^{2/(d+1)}$ and $k = c_2 \cdot \varphi^{-d}\log^d(\varphi^d N)$, where $c_2 > 1$ is a sufficiently large constant. For convenience, we assume that $N = 2^m k$, with $m \in \mathbb{N}$ (otherwise we append dummy points to the stream to match this constraint). We divide the input stream of N points into blocks of size k, and build a balanced binary tree T on the blocks (see Fig. 2). T has $m = \log(N/k)$ levels, where level 0 corresponds to the leaves and the root is at level m.

For a node $v \in T$ at level i, let P_v be the set of $2^i \cdot k$ points in descendant blocks of v. For every $v \in T$, we compute inductively a representative system $\mathcal{F}_v$ and a corresponding random sample S_v. For every leaf $v \in T$, we let $\mathcal{F}_v$ be the trivial partition of P_v into one-element sets and let $S_v = P_v$. At every non-leaf node $v \in T$, we construct $\mathcal{F}_v$ by merging the representative systems of its two children by Lemma 4.3. For every $F_j \in \mathcal{F}_v$, which is generated as $F_j = G_{i_1} \cup G_{i_2}$ from partitions of v's children, we choose the sample point $s(F_j)$ uniformly at random from $\{s(G_{i_1}), s(G_{i_2})\}$.

Note that the 2^{m-i} RSs on level i jointly form an RS of the total ground set of size N. The joint RS on the leaf level is 0-deficient. The deficiency increases by $c_1 \cdot k^{-1/d}$ on each level by Lemma 4.3. By induction, the joint RS at level i is $(i \cdot c_1 \cdot k^{-1/d})$-deficient for halfspaces. The root $r \in T$ is at level $m = \log(N/k)$. So the number of points misrepresented for halfspaces by $\mathcal{F}_r$ is at most

$$m \cdot c_1 \cdot k^{-1/d} \cdot N = c_1 \cdot \log\left(\frac{N}{k}\right) \cdot \frac{N}{k^{1/d}} = \frac{c_1}{c_2^{1/d}} \cdot \left(1 - \frac{\log c_2 + d\log\log(\varphi^d N)}{\log(\varphi^d N)}\right) \cdot \varphi N.$$

This is less than φN, if $N \geq 2\varphi^{-d}$ and c_2 is large enough, and so $\mathcal{F}_r$ is φ-deficient for halfspaces.

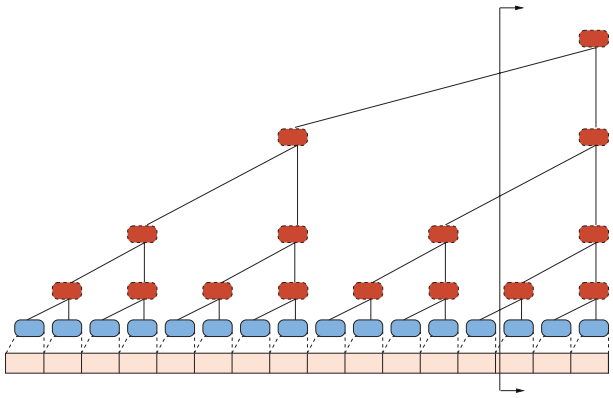

Fig. 2. We generate the summary of a stream of size $N = 2^4 \cdot k$ inductively in four levels.

The size of every set in $\mathcal{F}_r$ is $2^m = N/k = \varphi^d N/(c_2 \log^d(\varphi^d N))$. We apply Lemma 3.2 to $\mathcal{F}_r$ with $\alpha = \varphi$ and $\beta = \varphi^d/(c_2 \log^d(\varphi^d N))$. By choosing $c_2 \geq 6$, we conclude that the probability that the approximation error $|2^m \cdot |S_r \cap Q| - |P \cap Q||$ is above εN is less than

$$2e^{-\varepsilon^2/3\alpha\beta} = 2e^{-(c_2 \log^d(\varphi^d N))/3} \leq 2e^{-2\log(\varphi^d N)} \leq 2(\varphi^d N)^{-2} \leq 2 \cdot 2^{-2} = \tfrac{1}{2}. \qquad \square$$

Observe that $O(k) = O(\varepsilon^{-2d/(d+1)} \log^d(\varepsilon^{2d/(d+1)} N))$ is only the size of the final summary. If we construct the summary while reading the input, then we need to keep in memory at most one summary for each of the $\log(N/k)$ different levels of the hierarchy (see Fig. 2).

Corollary 4.5. *For a stream of N points in $\mathbb{R}^d$ (where $N \geq 2\varepsilon^{-2d/(d+1)}$ is known in advance), we can* maintain *at any time n, $1 \leq n \leq N$, a summary of size*

$$O(\varepsilon^{-2d/(d+1)} \log^d(\varepsilon^{2d/(d+1)} N) \max(\log(\varepsilon^{2d/(d+1)} n), 1))$$

that can answer halfspace range counting queries such that the absolute error is at most εn with probability at least $1 - 2\exp(-2\log(\varepsilon^{2d/(d+1)} N)) \geq \frac{1}{2}$.

Proof. At any time n, $1 \leq n \leq N$, the tree T has at most $\log(n/k) \leq \log(\varphi^d n)$ different levels.

During the inductive construction, we keep in memory at most one representative system and corresponding sample of size k from each level: At the leaf level, we feed points of the data stream into $\mathcal{F}_v$ and S_v. The leaf v is *complete* when all k elements of P_v have arrived. When two children v and w of a non-leaf node u are complete, we construct $\mathcal{F}_u$ and S_u, declare u *complete*, and delete all information regarding v and w. This establishes the claimed space bound.

Let S_i denote the level i sample. The representative systems in memory jointly give an RS of the first n points P_n. The joint RS is φ-deficient for halfspaces and the maximal set

size is $\max(1, 2^{\lfloor \log(n/k) \rfloor}) \leq \max(1, n/k)$. We aggregate the samples of different levels into a *weighted sample* such that sample points on level i have weight 2^i. For $n < 2k$, the sample is a 0-approximation. For $n \geq 2k$, we can apply Lemma 3.2 with $\alpha = \varphi$ and $\beta = \varphi^d/(c_2 \log^d(\varphi^d N))$. We conclude that the weighted sum $\sum_{i=0}^{m} 2^i |S_i \cap Q|$ approximates $|P_n \cap Q|$ with an absolute error of at most εn with probability at least $1 - 2\exp(-2\log(\varphi^d N)) \geq \frac{1}{2}$. □

4.3. *Summarizing General Data Streams*

In general, the stream size, n, is not known in advance. An algorithm of Vitter [30] can maintain a uniformly random sample of a fixed size $\Theta(\varepsilon^{-2} \log \varepsilon^{-1})$. This algorithm, however, occasionally deletes sample points (it replaces a randomly chosen old sample point by a new one), and so it cannot be fed into another insert-only data stream algorithm.

To overcome this difficulty, we deploy a simple idea of Manku et al. [18]: after having seen n_0 points of the stream, we know that the total stream size is at least n_0 and we can choose sample points at a rate $\Omega(d\varepsilon^{-2} \log \varepsilon^{-1})/n$. This results in a non-uniform sample, but we can assign weights to the sample points so that the output data structure has a uniform error bound.

Theorem 4.6. *For a stream of points in $\mathbb{R}^d$, we can maintain a summary of size $O(\varepsilon^{-2d/(d+1)} \log^{d+1} \varepsilon^{-1})$ that can answer halfspace range counting queries with an absolute error εn with probability $\frac{1}{2}$.*

We first describe a data stream algorithm, and then we prove that this algorithm meets the required theoretical bounds.

Algorithm `halfspace-summary`. Let $K = c_3 d\varepsilon^{-2} \log \varepsilon^{-1}$, where c_3 is a sufficiently large constant. Assume, for convenience, that K is a power of 2. We partition the input stream into buckets $B_0, B_1, B_2, \ldots$, etc. The size of B_0 is K; later buckets have exponentially increasing sizes: $|B_i| = 2^{i-1}K$ for $i = 1, 2, \ldots$. From each bucket B_i, we draw a sample D_i of size K. The sampling algorithm of Vitter [29] can choose, if the bucket size $|B_i|$ is known in advance, the sample D_i online in one pass over B_i using only $O(1)$ working space and an expected $O(K)$ time.

Since we cannot afford to keep the entire D_i in memory, we compress it on-line, while reading the bucket B_i. We feed D_i into the algorithm of Theorem 4.4 and prepare a summary S_i of size $k = c_2 \cdot \varepsilon^{-2d/(d+1)} \log^d \varepsilon^{-1} = c_2 \cdot \varphi^{-d} \log^d \varepsilon^{-1}$, where $\varphi = \varepsilon^{2/(d+1)}$ and c_2 is a large enough constant. The summary set S_i corresponds to a representative system $\mathcal{F}_i$ over D_i. Note that if only an initial portion B_i' of B_i has arrived, then only an initial portion D_i' of D_i is selected. We maintain a partial summary S_i' of D_i' of size $k \log(K/k)$ according to Corollary 4.5. S_i' corresponds to a representative system $\mathcal{F}_i'$ over D_i'.

A bucket B_i and the sample D_i are *complete* when all points of B_i have arrived and we have chosen all K elements of D_i. At that time, the summary S_i of size k is also complete. Then we prepare a summary (representative system and sample) of size k for the set $\bigcup_{j=0}^{i} B_j$ for every i inductively: The base case is $\mathcal{F}_0^* = \mathcal{F}_0$ and $S_0^* = S_0$. For

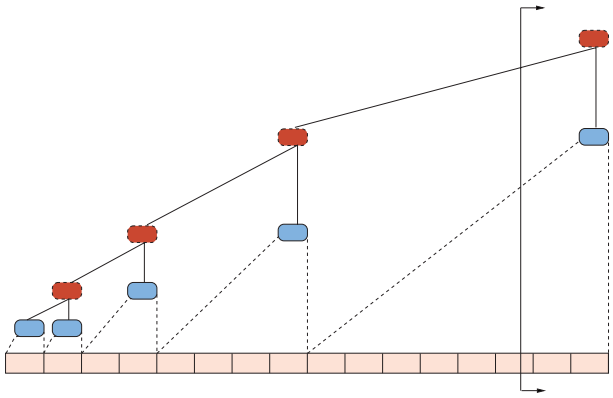

Fig. 3. We merge inductively S_i and S^*_{i-1}.

$i = 1, 2, \ldots$, we construct $\mathcal{F}^*_i$ by merging $\mathcal{F}^*_{i-1}$ and $\mathcal{F}_i$ using Lemma 4.3, and generate the corresponding summary S^*_i from S^*_{i-1} and S_i (Fig. 3).

The nth point lies in bucket B_m, $m = \lceil \log(n/K) \rceil$. When n points have arrived, our current sample set is composed of two parts that jointly form a weighted sample: (1) S^*_{m-1}, each element with weight $2^{m-1} \cdot K/k$. (2) S'_m, which consists of at most $\log(K/k)$ samples of size k coming from different levels of a tree T_m; the weight of every sample point at level j is 2^j. The size of our summary is therefore $(1 + \log(K/k)) \cdot k = O(\varphi^{-d} \log^{d+1} \varepsilon^{-1}) = O(\varepsilon^{-2d/(d+1)} \log^{d+1} \varepsilon^{-1})$.

Proof of Theorem 4.6. We show now that Algorithm `halfspace-summary` maintains a required summary. If $n \leq K$, then Corollary 4.5 with $N = K$ completes the proof. Otherwise $n > K$, and the nth point arrives at a bucket B_m with $m > 0$. At that time, the first $m - 1$ buckets containing $2^{m-1}K$ points are complete, we have seen $n - 2^{m-1}K$ points $B'_m \subset B_m$, and we have selected samples $D'_m = D_m \cap B'_m$. Let $B = \bigcup_{i=0}^{m-1} B_i \cup B'_m$ be the set of n points seen so far and let $D = \bigcup_{i=0}^{m-1} D_i \cup D'_m$ be a weighted sample set where elements of D_0 have unit weight while the weight of every $a \in D_i$, $i = 1, 2, \ldots, m$, is $w_a = 2^{i-1}$. Our current summary is $S = S^*_{m-1} \cup S'_m$ corresponding to a representative system $\mathcal{F} = \mathcal{F}^*_{m-1} \cup \mathcal{F}'_m$ over D.

We focus on the last $\mu = \lceil \log(6/\varepsilon) \rceil$ buckets. The remaining $m - \mu$ buckets together contain no more than $2n/2^{\lceil \log(6/\varepsilon) \rceil} \leq \varepsilon n/3$ points. Our *approximation error* εn will come from three sources: (1) for $i = 0, 1, \ldots, m - \mu$, we assume that sample D_i is a 1-approximation of B_i; (2) in the last μ buckets, we assume that D_i (and B'_m) is an $\varepsilon/3$-approximation of B_i (resp., B'_m); (3) the sample S is an $(\varepsilon/3)$-approximation of the weighted set D. The *probability* that we err more than εn has, therefore, two sources: (i) the probability that not all of the last μ sets D_i approximate the last μ buckets B_i with at most $\varepsilon |B_i|/3$ absolute error and (ii) the weighted sample S does not approximate D with at most $\varepsilon n/3$ absolute error.

Analysis. Every sample D_i is an $\varepsilon/6$-approximation of B_i for halfspaces with probability at least $1 - \varepsilon/4$ for $i = 0, 1, \ldots, m - 1$ if c_3 is large enough (see Theorem 4.9

of [10]). D'_m is not necessarily an ε-approximation of B'_m, but by analogous argument, $|Q \cap D'_m| \cdot 2^{m-1}$ approximates $|Q \cap B'_m|$ with an absolute error of at most $(\varepsilon/6)|B_m|$ with probability at least $1 - \varepsilon/4$. Therefore, assuming this approximation quality for the last μ buckets with probability at least $1 - \mu(\varepsilon/4) = 1 - \lceil \log(6/\varepsilon) \rceil (\varepsilon/4) \geq \frac{3}{4}$, the weighted set D is a $2\varepsilon/3$-approximation of B. That is, for any halfspace query Q,

$$\left| \sum_{a \in Q \cap D} w_a - |Q \cap B| \right| \leq \sum_{i \leq m-\mu} |B_i| + \sum_{i=1}^{\mu-1} \frac{\varepsilon}{6} |B_{m-i}| + \frac{\varepsilon}{6} |B_m| \leq (\tfrac{1}{3} + \tfrac{1}{6} + \tfrac{1}{6})\varepsilon n = \frac{2\varepsilon n}{3}$$

with probability greater than $\frac{3}{4}$.

S_i^* is a result of inductive merges of $i + 1$ sample sets $S_0, S_1, \ldots, S_i$. Following the proof of Theorem 4.4, each $\mathcal{F}_i$, $i = 0, 1, \ldots, m - 1$, is φ-deficient for halfspaces. Similarly, $\mathcal{F}'_m$ is also φ-deficient for halfspaces over the weighted set D'_m. This implies that the union of these RSs is φ-deficient over the weighted set D.

The RSs $\mathcal{F}^*_{i-1}$ and $\mathcal{F}_i$ together cover the weighted set $\bigcup_{j=0}^{i} D_i$ of total weight $2^i K$, for $i = 1, 2, \ldots, m - 1$. By Lemma 4.3, merging these two RSs increases the weight of misrepresented points by at most $c_1 \cdot k^{(d-1)/d}(2^i K/k) = c_1 \cdot k^{-1/d} \cdot 2^i K \leq \varphi 2^i K$. The total weights of misrepresented points at $\mathcal{F}^*_{m-1}$ is therefore $\varphi \cdot (|\bigcup_{j=0}^{m-1} D_j| + 2 \cdot 2^{m-1} K) \leq 3\varphi \cdot |\bigcup_{j=0}^{m-1} D_j|$.

$\mathcal{F} = \mathcal{F}^*_{m-1} \cup \mathcal{F}'_m$ is (3φ)-deficient RS for halfspaces over the weighted set D. The total weight of each set in $\mathcal{F}$ is at most $2^{m-1} \cdot K/k \leq n/k = (\varphi^d \log^{-d} \varepsilon^{-1}/c_2)n$. We denote the partition in $\mathcal{F}$ by $\{F_s \colon s \in S\}$, associating a set $F_s \subset D$ to every sample point $s \in S$ such that $s \in F_s$. We apply Lemma 3.2 to the representative system $\mathcal{F} = (\{F_s \colon s \in S\}, \{s \in S\})$ with $\alpha = 3\varphi$ and $\beta = (\varphi^d \log^{-d} \varepsilon^{-1})/c_2$. For every query halfspace Q, the probability that

$$\left| \sum_{s \in S \cap Q} \sum_{b \in F_s} w_b - \sum_{a \in Q \cap D} w_a \right| > \frac{\varepsilon n}{3}$$

is bounded by

$$2e^{-(\varepsilon/3)^2/(3 \cdot 3\varphi \cdot \varphi^d \log^{-d}(1/\varepsilon)/c_2)} = 2e^{-(c_2 \log^d \varepsilon^{-1})/81} \leq \tfrac{1}{4} \tag{1}$$

if c_2 is sufficiently large. □

We next estimate the amortized per-item processing time of Algorithm `halfspace-summary`. Since we use the bound of Lemma 4.2 instead of that of Lemma 4.1, we choose $\varphi = \varepsilon^{2/(d+1)-\delta/d}$ and $k = c_2 \cdot \varepsilon^{-2d/(d+1)+\delta} \log^d \varepsilon^{-1}$, for a small $\delta > 0$. The total processing time for Algorithm `halfspace-summary` is composed of the linear running time of the sampling algorithm of Vitter [30] and the iterative merge steps. By Lemma 4.2, one can merge two RSs of size k in $O(k \log k)$ time. Compressing a sample D_i of size K into sample S_i of size k by the algorithm of Theorem 4.4 requires $(K/k)\log(K/k) = O(\varepsilon^{-2/(d+1)} \log^{2-d} \varepsilon^{-1})$ merge steps. Merging the samples S^*_{i-1} and S_i, for $i = 1, 2, \ldots, m - 1$, requires an additional $m = O(\log(n/K))$ merge steps. For $n \geq K$, the total processing time amounts to $O(n) + m \cdot (K/k)\log(K/k) \cdot O(k \log k) = O(n + K \log(n/K) \log(K/k) \log k) = O(n + \varepsilon^{-2} \log(\varepsilon^2 n) \log^3 \varepsilon^{-1})$ time.

The amortized per-item processing time is, therefore, $O(1+(1/\varepsilon^2 n)\log(\varepsilon^2 n)\log^3 \varepsilon^{-1})$, which is $O(1)$ if $n \to \infty$.

Similarly to Theorem 3.5, we can extend Theorem 4.6 and maintain an *ε-approximation* of size $O(\varepsilon^{-2d/(d+1)}\log^{d+1}\varepsilon^{-1}\log(\varepsilon\delta)^{-1})$ with at least $1-\delta$ probability for halfspace ranges in $\mathbb{R}^d$. Theorem 4.6 and its extension to ε-approximations also hold for any range space where the order of magnitude of the *dual shatter function* is $\pi^*(n) = O(n^d)$: Our argument made assumptions on the underlying range space only in Lemmas 4.1 and 4.2, which hold for any range space whose dual shatter function is $O(n^d)$ [23]. The argument of Theorem 3.5 also goes through because the primal shatter function is a bounded degree polynomial, too. We can summarize our result as follows.

Theorem 4.7. *We are given a data stream of points in a range space whose dual shatter function is of order $\pi^*(m) = O(m^d)$. For any constant $\delta > 0$, we can maintain in $O(1)$ amortized per-item processing time an $O(\varepsilon^{-2d/(d+1)+\delta}\log^{d+2}\varepsilon^{-1})$ size weighted sample set which is an ε-approximation with probability $\frac{1}{2}$.*

5. Rectangle Range Counting Revisited

The core of the algorithm in Section 4 was a merging step that compressed two representative systems for halfspace ranges into one. In the case of axis-aligned rectangular ranges, we can use a similar merging step that compresses ε-approximations directly.

Lemma 5.1. *Given a set A of $2k$ points in $\mathbb{R}^d$, we can compute a subset $A' \subset A$ of size at most k which is a $(c_4\log^{2d}(2k)/k)$-approximation of A,[1] where $c_4 > 0$ is a constant depending on d only.*

The proof is based on the combinatorial discrepancy of $2k$ points for axis-aligned boxes and on the Beck–Fiala theorem [8] (see Theorem 4.14 and remarks in [23]). The application of the Beck–Fiala theorem involves solving $O(k)$ linear programs on $2k$ variables. Using the interior-point method [17], for example, each linear program can be solved in $O(k^3)$ time and $O(k^2)$ space.

Equipped with a merging algorithm for two ε-approximation of equal size, we can follow the data stream algorithms of the previous section. Instead of a representative system and sample set, we maintain only one sample set A, which is an ε_1-approximation for some $\varepsilon_1 < \varepsilon$.

Theorem 5.2. *For a stream of N points in $\mathbb{R}^d$ (where $N \geq 2/\varepsilon$ is known in advance), we can deterministically construct an ε-approximation for axis-aligned rectangles of*

[1] The term $\log^{2d}(2k)/k$ is not the optimal bound: the best known upper bound (for which only existence proofs are known) is $O(\log^{d+1/2}(2k)\sqrt{\log\log 2k}/k)$ [23]. The best known lower bound, due to Baker [4], is $\Omega(\log^{(d-1)/2}(2k)\cdot(\log\log(2k)/\log\log\log(2k))^{1/(2d-1)}/k)$, $d \geq 3$.

size

$$O\left(\frac{1}{\varepsilon}\log(\varepsilon N)\log^{2d}\left(\frac{1}{\varepsilon}\log(\varepsilon N)\right)\right).$$

Proof. We follow the main steps in the proof of Theorem 4.4. Let

$$k = c_5 \cdot \frac{1}{\varepsilon}\log(\varepsilon N)\cdot\log^{2d}\left(\frac{1}{\varepsilon}\log(\varepsilon N)\right),$$

where $c_5 > 1$ is a sufficiently large constant depending on the dimension d only. We divide the input stream into blocks of size k, by adding dummy points if necessary. Also, we assume that N/k is a power of 2. We build a balanced binary tree T on the blocks.

For every $v \in T$, we compute inductively a sample A_v of size at most k. For every leaf $v \in T$, we set $A_v = P_v$. At every non-leaf node $v \in T$, we construct A_v by merging the samples of its two children by Lemma 5.1.

The union of all samples on level i of T jointly forms an $(ic_4\log^{2d}(2k)/k)$-approximation of the total input. We clearly have a 0-approximation at the leaf level, $i = 0$. Then at each level the approximation quality deteriorates by $c_4\log^{2d}(2k)/k$ according to Lemma 5.1. The root is at level $\log(N/k) \le \log(\varepsilon N)$, so the approximation error of the sample A_r is

$$\log(\varepsilon N)\cdot\frac{c_4\log^{2d}(2k)}{k}\,N = \frac{c_4}{c_5}\cdot\frac{\log^{2d}(2k)}{\log^{2d}(\varepsilon^{-1}\log(\varepsilon N))}\cdot\varepsilon N \le \varepsilon N$$

if c_5 is large enough. □

Note that k is the size of the final summary only. If we construct the summary while reading the input, then we need to keep in memory one summary for each of the $\log(n/k)$ different levels of the hierarchy.

Corollary 5.3. *For a data stream of N points in $\mathbb{R}^d$ (where $N \ge 2/\varepsilon$ is known in advance), we can* maintain *a weighted ε-approximation of size*

$$O(\varepsilon^{-1}\log(\varepsilon N)\log^{2d}(\varepsilon^{-1}\log(\varepsilon N))\max(\log(\varepsilon n), 1))$$

for axis-aligned rectangles at any time n, $1 \le n \le N$.

Proof. We keep in memory at most one sample from each level: At the leaf level, we feed points of the data stream directly into our sample A_v. The leaf v is *complete* when all k elements of P_v have arrived. When two children v and w of a non-leaf node u are complete, then we construct A_u, declare v *complete*, and delete the samples of u's children.

At any time n, $1 \le n \le N$, we keep at most $\log(n/k) \le \log(\varepsilon n)$ samples in memory. The size of each sample set is k, which gives the claimed space bound.

These samples are from disjoint subsets of the set P_n of the first n points of the stream. We aggregate the samples into a *weighted sample* such that sample points on level i have weight 2^i. Each sample set is an ε-approximation of the corresponding base set, so their weighted union is also an ε-approximation of the first n points. □

If the stream size n is not known in advance, then we follow a two-phase procedure similar to the structure of Algorithm `halfspace-summary`: In a first phase we construct a weighted $(2\varepsilon/3)$-approximation D by a sampling algorithm of Vitter [29], then in the second phase we compact these samples by applying iteratively the merge step in Lemma 5.1.

Theorem 5.4. *For a stream of points in $\mathbb{R}^d$, we can maintain a weighted ε-approximation of size $O((1/\varepsilon)\log^{2d+2}(1/\varepsilon))$ for axis-aligned rectangles with probability at least $\frac{3}{4}$.*

Algorithm `box-summary`. Let $K = c_3 d\varepsilon^{-2}\log\varepsilon^{-1}$, where c_3 is a sufficiently large constant. Assume for convenience that K is a power of 2. We partition the input stream into buckets $B_0, B_1, B_2, \ldots$. The size of B_0 is K, and then $|B_i| = 2^{i-1}K$ for $i = 1, 2, \ldots$. From each bucket B_i, we draw a sample D_i of size K, exactly as in the first phase of Algorithm `halfspace-summary`.

We compress D_i by the algorithm of Theorem 5.2 and prepare a sample A_i of size $k = c_5\varepsilon^{-1}\log^{2d}\varepsilon^{-1}$, where c_5 is a sufficiently large constant depending on d only. On an initial portion D_i' of D_i, we maintain a partial sample A_i' of D_i' of size $k\log(K/k) = O(\varepsilon^{-1}\log^{2d+2}\varepsilon^{-1})$ according to Corollary 5.3.

When bucket B_i is complete, we prepare the sample A_i^* of size k for the set $\bigcup_{j=0}^{i} D_j$ for every i inductively: The base case is $A_0^* = A_0$. Then for $i = 1, 2, \ldots$, we construct A_i^* by merging A_{i-1}^* and A_i according to Lemma 5.1.

Proof of Theorem 5.4. We show that Algorithm `box-summary` maintains a required summary. Let $B = \bigcup_{i=0}^{m-1} B_i \cup B_m'$ and $D = \bigcup_{i=0}^{m-1} D_i \cup D_m'$ similarly to the proof of Theorem 4.6. We have seen that D is a weighted $(2\varepsilon/3)$-approximation of B with probability at least $\frac{3}{4}$ if c_3 is large enough.

Following the steps in the proof of Theorem 5.2, A_i, $i = 0, 1, \ldots, m-1$, is an $(\varepsilon/6)$-approximation of D_i with probability at least $1 - \varepsilon/4$ if c_3 is sufficiently large. A_i^* is a result of inductive merges of $i+1$ sample sets $A_0, A_1, \ldots, A_i$.

By Lemma 5.1, merging A_{i-1}^* and A_i increases the approximation error by $(\log^{2d}(2k)/2k)\cdot 2^i K$. The total approximation error of A_{m-1}^* on the weighted set $\bigcup_{i=0}^{m-1} D_i$ is therefore $\varepsilon n/6 + (\log^{2d}(2k)/2k)\cdot 2\cdot 2^{m-1}K \le \varepsilon n/6 + (n/k)\log^{2d}(2k) \le \varepsilon n/3$. Together with A_m', which is a weighted $(\varepsilon/6)$-approximation of D_m', we have a weighted $(\varepsilon/3)$-approximation of D. This is also a weighted ε-approximation of B with probability at least $\frac{3}{4}$. □

6. Closing Remarks

We have studied the problem of approximate range counting among multidimensional points in the data stream model, and presented deterministic and randomized summaries. We have shown that one can maintain almost optimal size summaries for range spaces whose dual shatter function is a bounded degree polynomial. Our deterministic schemes in the last section offers almost optimal summary size for axis-aligned box ranges. However, it relies on a halving step (Lemma 5.1), for which the space complexity of the

currently known deterministic or randomized algorithms exceeds the summary size. It is an open problem to find almost linear-space exact or approximation algorithms for this subroutine.

References

1. P.K. Agarwal. Range searching. In *Handbook of Discrete and Computational Geometry* (J. E. Goodman and J. O'Rourke, eds.), pp. 575–598. CRC Press, Boca Raton, FL, 1997.
2. B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom. Models and issues in data stream systems. In *Proc. Conf. on Principles of Database Systems*, pp. 1–16. ACM Press, New York, 2002.
3. A. Bagchi, A. Chaudhary, D. Eppstein, and M.T. Goodrich. Deterministic sampling and range counting in geometric data streams. In *Proc. 20th ACM Sympos. on Computational Geometry*, 2004, pp. 144–151.
4. R. C. Baker. On irregularities of distribution, II. *J. London Math. Soc.* (2) **59** (1999), 50–64.
5. J. Beck. Irregularities of distribution, I. *Acta Math.* **159**(1–2) (1987), 1–49.
6. J. Beck. Irregularities of distribution, II. *Proc. London Math. Soc.* (3) **56** (1988), 1–50.
7. J. Beck and W. Chen. *Irregularities of Distributions*. Cambridge University Press, Cambridge, 1987.
8. J. Beck and T. Fiala. Integer-making theorems. *Discrete Appl. Math.* **3** (1981), 1–8.
9. B. Chazelle. A functional approach to data structures and its use in multidimensional searching. *SIAM J. Comput.* **17**(3) (1988), 427–462.
10. B. Chazelle. *The Discrepancy Method*. Cambridge University Press, Cambridge, 2000.
11. B. Chazelle and J. Matoušek. On linear-time deterministic algorithms for optimization problems in fixed dimension. *J. Algorithms* **21** (1996), 579–597.
12. B. Chazelle and E. Welzl. Quasi-optimal range searching in spaces of finite VC-dimension. *Discrete Comput. Geom.* **4**(5) (1989), 467–489.
13. M. Greenwald and S. Khanna. Space-efficient online computation of quantile summaries. *ACM SIGMOD Record* **30**(2) (2001), 58–66.
14. T. Hagerup and C. Rüb. A guided tour of Chernoff bounds. *Inform. Process. Lett.* **33** (1990), 305–308.
15. M. Henzinger, P. Raghavan, and S. Rajagopalan. Computing on data streams. Technical Note 1998-011, Digital Systems Research Center, Palo Alto, CA, May 1998.
16. J. Hershberger, N. Shrivastava, S. Suri, and Cs. D. Tóth. Adaptive spatial partitioning for multidimensional data streams. *Algorithmica* **46**(1) (2006), 97–117.
17. N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica* **4** (1984), 373–395.
18. G.S. Manku, S. Rajagopalan, and B.G. Lindsay. Random sampling techniques for space efficient online computation of order statistics of large datasets. *ACM SIGMOD Record* **28**(2) (1999), 251–262.
19. J. Matoušek. Efficient partition trees. *Discrete Comput. Geom.* **8** (1992), 315–334.
20. J. Matoušek. Range searching with efficient hierarchical cuttings. *Discrete Comput. Geom.* **10** (1993), 157–182.
21. J. Matoušek. Geometric range searching. *ACM Comput. Surveys* **26**(4) (1994), 422–461.
22. J. Matoušek. Tight upper bounds for the discrepancy of half-spaces. *Discrete Comput. Geom.* **13** (1995), 593–601.
23. J. Matoušek. *Geometric Discrepancy*: *An Illustrated Guide*. Springer-Verlag, Berlin, 1999.
24. J. Matoušek. *Lectures on Discrete Geometry*. Springer-Verlag, Berlin, 2002.
25. J. Matoušek, E. Welzl, and L. Wernisch. Discrepancy and approximations for bounded VC-dimension. *Combinatorica* **13** (1993), 455–466.
26. J.I. Munro and M.S. Paterson. Selection and sorting with limited storage. *Theoret. Comput. Sci.* **12** (1980), 315–323.
27. S. Muthukrishnan. Data streams: algorithms and applications. Preprint, 2003.
28. V.N. Vapnik and A.Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory Probab. Appl.* **16** (1971), 264–280.
29. J.S. Vitter. Faster methods for random sampling. *Comm. ACM* **27** (1984), 703–718.
30. J.S. Vitter. Random sampling with a reservoir. *ACM Trans. Math. Software* **11** (1985), 37–57.

Received July 19, 2004, *and in revised form July* 20, 2005. *Online publication September* 12, 2006.