Deterministic Sampling and Range Counting in Geometric Data Streams

Amitabha Bagchi* Amitabh Chaudhary† David Eppstein‡ Michael T. Goodrich‡

December 6, 2004

Abstract

We present memory-efficient deterministic algorithms for constructing ϵ-nets and ϵ-approximations of streams of geometric data. Unlike probabilistic approaches, these deterministic samples provide guaranteed bounds on their approximation factors. We show how our deterministic samples can be used to answer approximate online iceberg geometric queries on data streams. We use these techniques to approximate several robust statistics of geometric data streams, including Tukey depth, simplicial depth, regression depth, the Thiel-Sen estimator, and the least median of squares. Our algorithms use only a polylogarithmic amount of memory, provided the desired approximation factors are at least inverse-polylogarithmic. We also include a lower bound for non-iceberg geometric queries.

1 Introduction

With the proliferation of streams of packets on the Internet, as well as data streaming from embedded systems, digital monitors, sensor networks, and scientific instruments, there is a need for new algorithms that can compute approximations or answer approximate queries on data streams. The main challenge in these contexts is that the data volumes are often much larger than the memory size of a typical computer. Thus, there is a considerable amount of interest in methods that can process data streams using limited memory (e.g., see recent surveys by Muthukrishnan [34] and Babcock [2]). The model we choose to work in is the so called *Time Series* model in which each time instant reveals a new element of the data stream "signal."

A typical approach in data streaming algorithms is to maintain a random sample of the input data and perform computations on the sample with the hope that information about the sample can be used to infer properties of the entire set. Naturally, such inferences come with an associated probability that they are inaccurate. In this paper, we are interested in deterministically constructing samples of a data stream that have guaranteed approximation properties for the original set. Moreover, because of the limited memory restriction of data streaming applications, we are interested in deterministic samples that can be constructed using space that is polylogarithmic in the data stream's length.

In addition, because much of the streaming data is coming from sensors and scientific instruments, we are interested in this paper in studying streaming algorithms for geometric data. Such data could include multi-dimensional points in the color space of astrophysical data or two-dimensional lines defined by a point-line duality of a stream of points in the plane. Of particular interest, then, is data streaming algorithms for constructing ϵ-nets and ϵ-approximations, which are general structures developed in the computational geometry literature for deterministically sampling geometric data. Indeed, ϵ-nets and ϵ-approximations are developed in a very general context of bounded-dimensional range spaces, where we are given a ground set and a polynomial-sized family of ranges on that set (which constitute the queries or sampling statistics we are interested in). Hence, results for constructing such deterministic samples should have a considerable number of applications.

*Department of Computer and Information Science, Polytechnic University, 6 Metrotech Center, Brooklyn, NY 11215, USA. bagchi@cis.poly.edu

†Department of Computer Science and Engineering, Notre Dame University, Notre Dame, IN 46556, USA.Amitabh.Chaudhary.1@nd.edu

‡School of Information and Computer Science. University of California. Irvine, CA 92697-3425, USA. {eppstein,goodrich}@ics.uci.edu

1.1 Related work on Streaming Algorithms

Data streaming problems have engendered a large amount of interest among the algorithms community over the last few years. For a comprehensive survey of the work done so far and some interesting directions for the future, the reader is referred to Muthukrishnan's work [34]. An earlier survey by Babcock et. al. [2] explores the issues arising in building data stream systems.

There has been very little work so far on constructing ϵ-nets or ϵ-approximations in streaming models, although these structures have been extensively studied in full-memory contexts (e.g., see the chapter by Matoušek [33]). The two lines of research we are aware of are by Suri, Tóth and Zhou [40] and by Brönimann et. al [7, 6]. We will discuss these two in detail at the end of this section once we lay out the larger context of research in this area.

Apart from the papers mentioned in the previous paragraph the closest previous work is done in the *iceberg query* [15] frameworkq. Manku and Motwani [29] provide $1+\epsilon$ approximations for the frequency counts of items in a data stream that occur more than ϵN times (which are the so-called "icebergs"). An alternative approach which requires two passes but uses less space can be found in [25]. Another set of improved results for determining the top k frequency counts is given in [13]. Algorithms for computing the quantiles of a data stream have been given by Greenwald and Khanna [18] guaranteeing a precision of ϵN, which is similar to the guarantees that are provided by ϵ-approximations, while using $O(\frac{1}{\epsilon}\log \epsilon N)$ space. This limitation of an additive ϵN error in every quantile is overcome by Gupta and Zane [19]. The latter's method provides relative error for all quantiles but uses $O(\log^2 N/\epsilon^3)$ space and requires knowledge of an upper bound on the stream size.

Although the area of geometric property testing has generated interest in algorithms with sublinear time complexity (see e.g. [8]), the first geometric problem to be studied in the streaming model as we understand it was that of finding the diameter of a set of points. Feigenbaum, Kannan and Zhang [16] gave an $O(1/\epsilon)$ space algorithm for computing the diameter of points in two dimensions in the streaming model and a $O(\frac{1}{\epsilon^{3/2}} \cdot \log^3 N(\log R + \log\log N + \log(\frac{1}{\epsilon})))$ space algorithm for computing it in the sliding window model where R is the maximum, over all windows, of the ratio of the diameter to the distance between the closest two points in the window. Indyk [22] gave a streaming algorithm which maintains a c-approximate diameter of points in d dimensions using $O(dn^{1/(c^2-1)})$ space taking $O(dn^{1/(c^2-1)})$ time per new point, for $c > \sqrt{2}$.

Cormode and Muthukrishnan generalized the exponential histograms used on single dimensional data sets in earlier works on streaming algorithms [12, 26] and defined *radial histograms* [11], which allowed them to give a $O(1+\epsilon)$ approximation to the diameter using $O(1/\epsilon)$ space. They were also able to use these structures to approximate convex hulls in the sense that no point in the input stream is more than ϵD outside the approximate hull, where D is the diamter of the point set. Constructing an approximate hull takes them $O(q/\epsilon)$ space. Hershberger and Suri [21] improve this to give a sampling-based algorithm for approximating the convex hull of a streaming point set, showing how to maintain an adaptive sample of at most $2r$ points such that the distance between the hull of their sample and the true convex hull is $O(D/r^2)$, where D is the current diameter of the sample. Some of the other geometric problems that have been studied in a streaming model include minimum spanning tree and minimum weight matching [23] and certain facility location and nearest neighbour kind of queries [11].

As mentioned earlier, the work by Suri et. al. [40], which appeared in the same place as the first version of our current work [3], is very closely related to our paper. Suri et. al. present an algorithm for answering axis-parallel box range counting queries using cross products of Greenwald and Khanna's quantile summaries [18]. They also use Chazelle and Welzl's results on matchings with low crossing number [9] to give an algorithm for directly merging and reducing ϵ-approximations for axis-aligned box ranges. Their techniques are very interesting, synthesizing many different areas of research. Their results parallel the ones we provide here although they are presented in a less general setting (and hence their space and time bounds are slightly better.) The other major advantage of our work over their's is that our constructions are deterministic.

Also in the streaming context, Brönnimann and others [7, 6] use ϵ-approximations for association rule mining in large databases. They give an algorithm for deterministically computing an ϵ-approximation on streams of transactions (each of which is a set of items). This algorithm operates by repeatedly halving the current sample till the approximation bound required in violated.

1.2 Our Results

In this paper, we present memory-efficient deterministic algorithms for constructing ϵ-nets and ϵ-approximations of streams of geometric data. Our algorithms use a polylogarithmic amount of memory, provided ϵ is at least inverse-polylogarithmic. As mentioned above, ϵ-nets and ϵ-approximations are of interest in their own right and have many applications in computational geometry. We show how our deterministic samples can be used to answer online iceberg geometric queries on data streams, such as in multi-dimensional iceberg range searching. Because the information typically of interest from data streams is statistical, we focus in this paper primarily on the use of ϵ-nets and ϵ-approximations to compute approximations to several robust statistics of geometric data streams, including Tukey depth, simplicial depth, regression depth, the Thiel-Sen estimator, and the least median of squares. Thus, we additionally give polylogarithmic-space data streaming algorithms for computing approximations to these statistics. We also include a lower bound for non-iceberg range queries in data streams.

2 Preliminaries on ϵ-Nets and ϵ-Approximations

We recap certain aspects of ϵ-Nets and ϵ-approximations [42, 33] which are part of a general framework for modelling a number of interesting problems in computational geometry and derandomizing divide-and-conquer type algorithms.

A *range space* is a set system, i.e., a pair $\Sigma = (X, \mathcal{R})$, where X is a set and $\mathcal{R}$ is a set of subsets of X. We call the elements of $\mathcal{R}$ the *ranges* of Σ, as $\mathcal{R}$ is typically defined in terms of some well structured geometry. If Y is a subset of X, we denote by $\mathcal{R}|_Y$ the set system *induced by $\mathcal{R}$ on Y*, i.e., $\{R \cap Y | R \in \mathcal{R}\}$[1].

We say a subset $Y \subseteq X$ is *shattered* if every possible subset of Y is induced by $\mathcal{R}$, i.e., if $\mathcal{R}|_Y = 2^Y$. The *VC-dimension* of Σ is the maximum size of a shattered subset of X. If there are shattered subsets of any size, then the VC-dimension is infinite. A related and simpler notion is the *scaffold dimension* [17] of Σ. It is based on the notion of the *shatter function* $\pi_{\mathcal{R}}(m)$, which we define as the maximum possible number of sets in a subsystem of Σ induced by an m-sized subset of X. In other words, it is the $\sup\{|R|_Y| : Y \subseteq X, |Y| = m\}$. We now define the scaffold dimension of $(X, \mathcal{R})$ as the infimum of all numbers d such that $\pi_{\mathcal{R}}(m)$ is $O(m^d)$. It turns out that the shatter function of a set system of VC-dimension d' is bounded by $\binom{m}{0} + \binom{m}{1} + \cdots + \binom{m}{d'} = \Theta(m^{d'})$ [38, 42]. Thus the scaffold dimension is always at most the VC-dimension. Conversely, if the scaffold dimension is bounded by a constant, the VC-dimension too is bounded by a constant. There are, however, many natural geometric set systems of scaffold dimension strictly smaller than the VC-dimension; for instance, the scaffold dimension of a set system defined by halfplanes in the plane is 2, while the VC-dimension is 3. In the rest of the paper, we will always refer to the scaffold dimension of a set system. In addition, we consider only those set systems whose scaffold dimensions are bounded by a constant.

We are now ready to define ϵ-nets and ϵ-approximations. A subset $S \subseteq X$ is an *ϵ-net for* $(X, \mathcal{R})$ provided that $S \cap R \neq \emptyset$ for every $R \in \mathcal{R}$ with $|R|/|X| > \epsilon$. A subset $A \subseteq X$ is an *ϵ-approximation for* $(X, \mathcal{R})$ provided that

$$\left| \frac{|A \cap R|}{|A|} - \frac{|X \cap R|}{|X|} \right| \leq \epsilon \tag{1}$$

for every set $R \in \mathcal{R}$. Note that every ϵ-approximation is automatically an ϵ-net, but the converse need not be true. A remarkable property about set systems of scaffold dimension d is that, for any $\epsilon \in [0, 1)$, they admit an ϵ-approximation whose size depends only on d and ϵ, *not* on the size of X. The first basic result in this vein is the following lemma.

Lemma 2.1 *For any set system $(X, \mathcal{R})$, with a finite X, and a scaffold dimension at most d, where $d \geq 1$, there exists, for any $\epsilon \in [0, 1]$, an ϵ-net of size at most $C_1\epsilon^{-1} \lg(\epsilon^{-1})$, and an ϵ-approximation of size at most $C_2\epsilon^{-2} \lg(\epsilon^{-1})$. Here C_1, C_2 depend on only d.*

Note that, in general, the $\lg(\epsilon^{-1})$ factor cannot be removed from the bound.

Matoušek [32] gave a deterministic algorithm for efficiently computing small sized ϵ-approximations (and thereby, ϵ-nets) for set systems with constant-bounded scaffold dimensions. Such an algorithm needs that the set system to be given in a form more "compact" than simply the listing of the elements in each set. For this we assume the existence of a *subsystem oracle*, i.e. an algorithm (depending on the specific geometric application) that, given any subset $Y \subseteq X$, lists all sets of $\mathcal{R}|_Y$. We say that the subsystem oracle is *of dimension at most d* if it lists all sets in time $O(|Y|^{d+1})$. This corresponds to the scaffold dimension; the maximum number of sets in $\mathcal{R}|_Y$ is $\pi_{\mathcal{R}}(|Y|)$, and the "+1" in the exponent accounts for the fact that each output set is given by a list of size up to $|Y|$. Matoušek's result is summarized by the following lemma.

Lemma 2.2 *Let $(X, \mathcal{R})$ be a set system with a subsystem oracle of dimension d, where d is a constant. Given any $\epsilon \in [0, 1)$, we can compute an ϵ-approximation of size $O(\epsilon^{-2} \lg(\epsilon^{-1}))$ and an ϵ-net of size $O(\epsilon^{-1} \lg(\epsilon^{-1}))$ in time $O(|X|\epsilon^{-2d} \lg^d(\epsilon^{-1}))$.*

[1] Note that although many sets of $\mathcal{R}$ may intersect Y in the same subset, this intersection appears only once in $\mathcal{R}|_Y$.

We shall use the algorithm above as a sub-routine for our streaming algorithm for ϵ-approximations (see Section 4). It is based on two observations that we state below. They correspond to two basic operations of our algorithm, the *merge step* and the *reduce step*. Many algorithms for computing ϵ-approximations (certainly the one Matoušek gave, and the one we shall give) start by partitioning X into small pieces, and then alternate between the two steps until they get the desired approximation.

Observation 2.3 (Merge Step) *Let $X_1, \ldots, X_m \subseteq X$ be disjoint subsets of equal cardinality and let A_i be an ϵ-approximation of cardinality b for $(X_i, \mathcal{R}|_{X_i})$, $i = 1, \ldots, m$. Then $A_1 \cup \ldots \cup A_m$ is an ϵ-approximation for the subsystem induced by $\mathcal{R}$ on $X_i \cup \ldots \cup X_m$.*

Observation 2.4 (Reduce Step) *Let A be an ϵ-approximation for $(X, \mathcal{R})$ and let A' be a δ-approximation for $(A, \mathcal{R}|_A)$. Then A' is an $(\epsilon + \delta)$-approximation for $(X, \mathcal{R})$.*

Lemma 2.2 can be extended to a weighted case, as in the following result by Matoušek [32].

Lemma 2.5 *Let X be a finite set equipped by a probabilistic measure μ (given by a table) and let $\Sigma = (X, \mathcal{R})$ be a range space satisfying the assumptions of Lemma 2.2. Then an ϵ-approximation for Σ with respect to the measure μ can be computed with the same asymptotic efficiency in the running time and size of the ϵ-approximation in the case of uniform measure in Lemma 2.2.*

When X is associated with a probabilistic measure μ, an ϵ-approximation of $(X, \mathcal{R})$ is a multi-set A such that

$$\left| \frac{|A \cap R|}{|A|} - \frac{\mu(X \cap R)}{\mu(X)} \right| \leq \epsilon$$

for every $R \in \mathcal{R}$. Though we call μ a probabilistic measure, the result extends to measure in which $\mu(X)$ to take values other than 1; specifically $\mu(X)$ can be $|X|$.

3 Additional Extensions for Weighted Sets

While the extension described above is useful in our context, we nevertheless need some further generalizations, which will be useful in the data streaming model. In particular, we need to be able to merge ϵ-approximations of sets of different cardinalities. To this end we generalize Observations 2.3 and 2.4. To the best of our knowledge, this is the first time such observations are being made.

Our observations generalize the fact that in the un-weighted case, for an ϵ-approximation A for $(X, \mathcal{R})$, each element in A "represents" $|X|/|A|$ elements in X. This is easy to see if we write Requirement (1) in the following form

$$\left| |A \cap R| \frac{|X|}{|A|} - |X \cap R| \right| \leq \epsilon |X|$$

In general, an element p in the ϵ-approximation$A$ need not represent the same number of elements of X as every other element in A, especially if A has been created by merging two ϵ-approximationsof sets with different cardinalities. So, instead of having an element $p \in A$ represent $|X|/|A|$ elements, we can assign it a weight $\gamma(p)$ equal to the number of elements in X that it represents. In this generalized scenario, a subset $A \subseteq X$, is a *weighted ϵ-approximation for* $(X, \mathcal{R})$ if $\sum_{p \in A} \gamma(p) = |X|$, and for every $R \in \mathcal{R}$,

$$\left| \sum_{p \in A \cap R} \gamma(p) - |X \cap R| \right| \leq \epsilon |X|$$

We are now ready to state observations related to weighted merging and weighted in a form that will be of use in the streaming algorithm.

Observation 3.1 (Weighted Merge Step) *Let $X_1, \ldots, X_m \subseteq X$ be disjoint subsets (of possibly differing cardinalities) and let A_i be a weighted ϵ-approximation of $(X_i, \mathcal{R}|_{X_i})$, $i = 1, \ldots, m$. Then $A_1 \cup \ldots \cup A_m$ is a weighted ϵ-approximation for the subsystem induced by $\mathcal{R}$ on $X_i \cup \ldots \cup X_m$, where the weight $\gamma(p), p \in A_1 \cup \ldots \cup A_m$ remains unchanged.*

For the next observation, we define a measure $\mu(\cdot)$ as follows: $\mu(Y) = \sum_{p \in Y} \gamma(p)$, where $\gamma(p)$ is the weight of point p. Recall Lemma 2.5 which talks about computing an ϵ-approximation for a set equipped with such a measure.

Observation 3.2 (Weighted Reduce Step) *Let A be a weighted ϵ-approximation for $(X, \mathcal{R})$ and let A' be a δ-approximation for $(A, \mathcal{R}|_A)$ with respect to measure $\mu(\cdot)$. Then A' is an $(\epsilon + \delta)$-approximation for $(X, \mathcal{R})$.*

4 Computing ϵ-Approximations in Geometric Streams

In this section we describe an algorithm that takes polylogarithmic space and processing time per object to compute a compact ϵ-approximation of a data stream of geometric objects of bounded scaffold dimension, provided ϵ is at least inverse polylogarithmic,

Let $x_1, \ldots, x_n, \ldots$ be a stream of geometric objects in the time series model with insert only. Let X be the set of all the objects in the stream that have arrived till now. Let $\mathcal{R}$ be a set of ranges defined on X, and $\Sigma = (X, \mathcal{R})$ be the current range space. In addition, let d, where d is a constant, be the scaffold dimension of Σ.

We begin by imposing a natural tree hierarchy over the data stream: define a *canonical set* $S_{j,k}$ as $\{x_i | j2^k \leq i < (j+1)2^k\}$ for $j, k \geq 0$. Canonical sets are inter-related through a natural tree hierarchy. The *children* of set $S_{j,k}$, $k \geq 1$, are the canonical sets $S_{2j,k-1}$ and $S_{2j+1,k-1}$. We say that a canonical set $S_{j,k}$ becomes *available* when the last element in it, i.e., $x_{(j+1)2^k-1}$, arrives. A *maximal canonical set* is one that is available but whose parent is not yet available.

Observe that when x_n arrives, there are at most $\lg n$ maximal canonical sets. Also, the union of all the maximal canonical sets is the set X of all elements that have arrived till now.

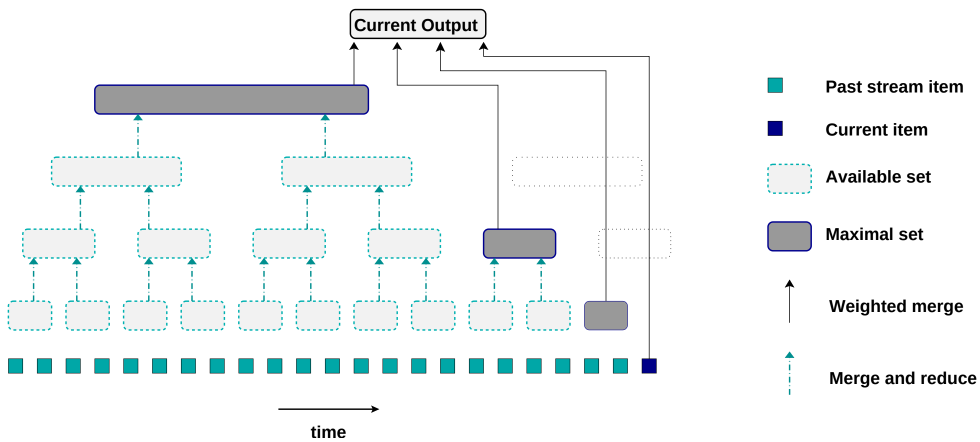

Figure 1: Schematic: Computing an ϵ-approximation of a data stream

We now outline our streaming algorithm. It simulates the divide-and-conquer approach of the static algorithm in a bottom up fashion. Interestingly, we do not need to know the value of n in advance.

The algorithm proceeds as follows (see Figure 1): At every stage, the algorithm stores a δ-approximation for all available maximal canonical sets, where δ varies with the set, but is always at most $\epsilon/2$. The δ-approximation $A_{j,k}$ for $S_{j,k}$ is constructed through merging the approximations $A_{2j,k-1}$ and $A_{2j+1,k-1}$ which were earlier computed for $S_{j,k}$'s two children. Lemma 2.2 states that ϵ-approximations of the same size can be constructed on two different input sets of the same cardinality, provided the same $\epsilon$is used for both. Thus, by induction, we can see that $A_{2j,k-1}$ and $A_{2j+1,k-1}$ have the same cardinality, and by Observation 2.3 can be merged. After this merge the ϵ-approximation is reduced to the required size.

The ϵ-approximation of the set X at any point, the *stream output*, is determined by merging the ϵ-approximations for the maximal canonical sets in a weighted manner as given by Observation 3.1. Each element $p \in A_{j,k}$ is assigned a weight $\gamma(p) = |S_{j,k}|/|A_{j,k}|$ for this purpose.

Formally, we need the following building blocks:

- ϵ-approx(): An algorithm for deterministically computing ϵ-approximation of small size (see Lemma 2.2),

- weighted_ϵ-approx(): An algorithm for computing deterministically ϵ-approximations of *weighted* items of small size (see Lemma 2.5).

Note that we will never use ϵ-approx() on an input that is larger than logarithmic, as otherwise we will not remain the space and time bounds that we are aiming for.

Our algorithm, we call it ϵ-stream_approx(), follows the basic merge and reduce technique [33] for constructing ϵ-approximations. To follow this technique we need to use a sequence $w_1, \ldots, w_u, \ldots$ with the property that $W \triangleq \sum_{u=1}^{\infty} w_u = O(1)$. Here we shall use $w_i = i^{-c}$, for some $c > 1$.

In Figure 2 we give the fomrmal specification of **ϵ-stream_approx()**. In reading the specification, assume that $A_{j,0}$ is the element itself in the singleton set $S_{j,0}$.

```
ε-stream_approx()
  When the next element x_n in the stream arrives
    For each canonical set S_{j,k} that becomes available,
    taken in the order of increasing k, where k ≥ 1
        /* Combine approximations of its children
           for the parent */
        B ← A_{2j,k−1} ∪ A_{2j+1,k−1}.
        /* Reduce the size of the approximation */
        A_{j,k} ← (ε/2 · w_k/W)-approximation of B
           using ε-approx().
        /* Assign weights to elements */
        For all p ∈ A_{j,k}: γ(p) ← |S_{j,k}|/|A_{j,k}|.
    /* Combine approximations of maximal canonical sets
       for the stream */
    A' ← ⋃_{S_{j,k} is available} A_{j,k}.
       Each element in A' retains its weight from its orginal A_{j,k}.
    /* Reduce the size of the approximation */
    A ← (ε/2)-approximation of A' using weighted_ε-approx().
    Output A.
```

Figure 2: Algorithm for computing an ϵ-approximation of a geometric stream.

We are now ready to state our main theorem.

Theorem 4.1 *Given an algorithm for computing an ϵ-approximation of an n-point range space (equipped with a probabilistic measure) of size $\sigma(\epsilon)$ in time $T(n, \epsilon)$ and taking space $S(n, \epsilon)$, the algorithm **ϵ-stream_approx()** can compute an ϵ-approximation for a stream of objects, of which n have been seen, such that:*

1. *The size of the approximation is $O(\sigma(\epsilon))$;*
2. *The processing time per input object is $O(\lg n \cdot T(s, O(\epsilon/\lg^c n)) + T(\lg n \cdot s, \epsilon/2))$;*
3. *The space taken is $O(\lg n \cdot s + S(s, O(\epsilon/\lg^c n)) + S(\lg n \cdot s, \epsilon/2))$;*

where $s = \sigma(O(\epsilon/\lg^c n))$, and $c > 1$ is a constant.

Proof. Observations 2.3 and 2.4 imply that $A_{i,j}$ is a δ-approximation for $S_{j,k}$, where

$$\delta \leq \sum_{u=1}^{k} \frac{\epsilon}{2} \cdot \frac{w_u}{W} < \frac{\epsilon}{2}.$$

Together with Observation 3.1, this implies that A' is a weighted $(\epsilon/2)$-approximation for the set X of elements in the stream. By Observation 3.2, it now follows that A is an ϵ-approximation of $(X, \mathcal{R})$. The size of A is $O(\sigma(\epsilon/2))$.

The data structure needs to store just the ϵ-approximations of the maximal canonical sets i.e. the '$A_{j,k}$'s; all other sets are intermediate results that can be discarded. Denote the size of largest such set, i.e., $A_{j,\lg n}$, by s, which is $O(\sigma(\epsilon/\lg^c n))$; recall that the size is determined by just the last reduction step. The size of the data structure is, therefore, $O(\lg n \cdot s)$.

Consider the space and time requirements for calls to ϵ-approx() (there can be at most $\lg n$ such calls per input object) and weighted_ϵ-approx() (1 per input object). Note that s is an upper bound for the size of the input to ϵ-approx(), and $s \lg n$ an upper bound for the size of the input to weighted_ϵ-approx(). Thus, for the calls to ϵ-approx(), the time required is $O(\lg n \cdot T(s, O(\epsilon/\lg^c n))$, and the space required is $S(s, O(\epsilon/\lg^c n))$. For the call to weighted_ϵ-approx(), the time required is $T(\lg n \cdot s, \epsilon/2))$, and the space required is $S(\lg n \cdot s, \epsilon/2))$.

□

The algorithm given by Matoušek [32] (see Lemmas 2.2 and 2.5) has $\sigma(\epsilon) = O(\epsilon^{-2}\lg(\epsilon^{-1}))$, $T(n, \epsilon) = O(n(\epsilon^{-2}\lg(\epsilon^{-1}))^d)$, and, we claim, $S(n, \epsilon) = O(n + \sqrt{n} \cdot (\epsilon^{-2}\lg(\epsilon^{-1}))^d)$. Thus, in effect, we have the following result.

Corollary 4.2 *There is an algorithm for computing ϵ-approximations for a stream of objects, of which n have been seen, with an associated set of ranges, such that the size of the approximation is $O(\epsilon^{-2}\lg(\epsilon^{-1}))$, the processing time per object is $O(\lg n \cdot s^{d+1})$, and the space taken is $O(\lg n \cdot s + s^{d+1/2})$, where $s = O(\epsilon^{-2} \cdot \lg^{2c} n(\lg\lg n + \lg(\epsilon^{-1})))$, and $c > 1$ is a constant.*

5 Applications: Robust Statistics

ϵ-Nets and ϵ-approximations have a number of applications in computational geometry, and even other areas like learning theory — see, e.g., [31]. Many of the problems in these have streaming versions. One basic application is *range counting*. In this, we are given a set S of n points in $\mathbb{R}^d$, and a family $\mathcal{R}$ (the ranges) of subsets of $\mathbb{R}^d$. Each query consists of a range $R \in \mathcal{R}$ and asks for the number of points in it. Typical range families are axes-orthogonal ranges, spherical ranges (proximity queries), and simplical ranges. The corresponding range spaces for these all have a bounded scaffold dimension. In the streaming version, the point set S comes as a continuous stream, interspersed with queries. It is easy to see how our algorithm would work here: use ϵ-stream_approx() to maintain an ϵ-approximation A of the current $(S, \mathcal{R})$. When queried with range $R \in \mathcal{R}$, output $|A \cap R| \cdot n/|A|$. This is within an additive ϵn of the true value; this is akin to the iceberg queries mentioned earlier.

The above technique has implications in a lot of specific applications. To get a flavor of this, we delve deeper into the specific area of robust statistic in the next few paragraphs.

Robust statistics concerns the study of statistical estimators that can tolerate high numbers of *outliers*, while maintaining an accuracy of estimation that depends only on the remaining uncorrupted data points. In contrast, ordinary least squares estimators, while trivial to compute even in the streaming model, can be forced to produce estimates that are arbitrarily far from the correct model even in the presence of a single outlier. The number of outliers that an estimator can tolerate while preserving its accuracy is called its *breakdown point*; in general, methods with high breakdown points are preferred but other criteria are also important including statistical efficiency (number of samples needed to achieve a given accuracy) and computational efficiency (amount of time it takes to compute a given estimate from a set of samples). Many robust statistical methods also have the advantage of being *non-parametric*, not requiring the statistician to produce a prior probability distribution or other arbitrary parameters before producing a fit. The paradigmatic example of a robust statistic is the median of one-dimensional data, which, unlike the mean, is robust with a breakdown point of $\frac{1}{2}$. Much research on streaming algorithms has gone into methods for maintaining approximate medians or more general quantiles [18], and we would like to find similar methods for higher dimensional statistics.

Two of the critical problems studied in robust statistics are *location* (finding a central point in a cloud of data points) and *regression* (fitting the data to a model in which a dependent variable or variables is a linear function of the independent variables). Many methods in this area are based on various concepts of *depth*, which measures the quality of fit of an estimate. It is natural to seek the estimate maximizing the depth, but it is also of importance to be able to compute depths of non-optimal estimates, in order to form *depth contours* that produce a center-outward ordering of the data.

For many of these robust statistical methods, a computationally efficient streaming approximation to the depth measure can be obtained from an ϵ-approximation of the sample data. The deepest fit can be approximated by a deepest fit to the ϵ-approximation, and this approximate fit often has similar breakdown point properties to the non-approximate fit on which it is based. We describe below several of the methods to which this technique applies:

5.1 Tukey Depth

This quantity [14] measures the quality of fit of a center, as the minimum proportion of sample points among all halfspaces that contain the center. The Tukey depth of a point can be computed in time $O(n^d \log n)$, where n denotes the number of sample points [37]. The *Tukey median* is the point of maximum depth. It is known that any Tukey median has depth at least $1/(d+1)$, and the breakdown point of the Tukey median as an estimate of location is also $1/(d+1)$. There are known static algorithms for finding Tukey medians, or other points of high depth, in two or three dimensions [24, 27, 30], but in higher dimensions only inefficient linear-programming based exact solutions are known and it is necessary to resort to more efficient approximation algorithms [10].

The Tukey depth is based on counting points in halfspaces. Hence it can be approximated effectively using ϵ-approximations for halfspace ranges [10]: the depth of a point within an ϵ-approximation of a sample is within an additive error of ϵ of its depth in the original sample data. In particular, the Tukey median of an ϵ-approximation has depth within ϵ of that of the true Tukey median. The breakdown point of this approximate Tukey median is $1/(d+1)-\epsilon$. Thus, by using our streaming ϵ-approximation algorithm, we can efficiently maintain not only an approximate Tukey median of the data set, but also a space-efficient data structure from which we can compute accurate approximations of the Tukey depth of any point.

5.2 Simplicial Depth

This is another measure of quality of fit for location, introduced by Liu [28]. The simplicial depth of a fit point is defined to be the proportion of simplices, among all the $\binom{n}{d+1}$ simplices formed by convex hulls of $(d+1)$-tuples of sample points, that contain the fit point. Equivalently, it is the probability that a randomly chosen $(d+1)$-tuple contains the fit point in its convex hull. As we now argue, for points in the plane, the simplicial depth in a sample set is accurately approximated by the simplicial depth of an ϵ-approximation for wedge ranges (that is, ranges formed by intersecting two halfplanes). Therefore, as for Tukey depth, we can answer approximate depth queries and maintain an approximate deepest point in a space-efficient manner for streaming data.

Let δ be a value to be determined later and imagine the following process for measuring approximately the simplicial depth of a fit point: first, let L be a set of $1/\delta$ lines through the fit point, partitioning the plane into $2/\delta$ wedges having the fit point as a common apex, with at most a δ fraction of the sample points in any wedge. Let e_1 be the proportion of triangles, determined by three input points, that are not all on one side of one of a line in L. Then e_1 is an overestimate of the simplicial depth, but the amount by which it overestimates the depth is $O(\delta)$: the only triangles incorrectly included in the estimate are ones that have two points in opposite wedges, there are $O(\delta^2 n^3)$ such triangles per pair of opposite wedges, and $O(1/\delta)$ such pairs. Next, let e_2 be the proportion of triangles, determined by three points in an ϵ-approximation of the sample, that are not all on one side of a line in L. For the same reasons as before, e_2 is within $O(\delta)$ of the simplicial depth for the ϵ-approximation. Further, e_1 and e_2 are within $O(\epsilon/\delta)$ of each other:

$$e_1 = 1 - \sum_i \frac{\binom{w_i}{3} + \binom{w_i}{2}(h_i - w_i) + w_i\binom{h_i - w_i}{2}}{\binom{n}{3}},$$

where w_i is the number of sample points in the ith wedge and h_i is the number of sample points in the halfplane containing the ith wedge on its counterclockwise boundary. Each term in the sum is approximated within $O(\epsilon)$ by the corresponding term where w_i and h_i are replaced by numbers of points in the ϵ-approximation, and there are $O(1/\delta)$ terms, so the total difference between e_1 and e_2 is $O(\epsilon/\delta)$. Putting together the errors in going from the original simplicial depth to e_1 to e_2 to the simplicial depth of the approximation, and setting $\delta = \sqrt{\epsilon}$, we see that the ϵ-approximation approximates the simplicial depth to within $O(\sqrt{\epsilon})$.

As far as we are aware, this deterministic ϵ-approximation based method for approximating simplicial depth is novel even for static, non-streaming data, although it is trivial to approximate simplicial depth randomly in the static case by sampling triangles. It seems likely that similar deterministic and streaming approximation guarantees, with worse dependence on ϵ, can be shown to hold also in higher dimensions.

5.3 Regression Depth

This statistic was introduced by Rousseeuw and Hubert [35] as a measure of the quality of fit of a regression hyperplane. It is defined as being the minimum proportion of sample points that can be removed to turn the fit plane into a *nonfit*, that is, a hyperplane combinatorially equivalent to a vertical hyperplane. Amenta et al. [1] showed that, like Tukey depth, for regression depth a fit always exists with depth at least $1/(d+1)$, and the breakdown point of the maximum-depth fit is $1/(d+1)$. Their proof technique shows that the regression depth of a query hyperplane can be measured by performing a certain projective transformation of the space containing the sample points, and measuring the Tukey depth of a certain point in the transformed space. Due to the transformation, a halfspace in the transformed space may correspond to a *double wedge* (symmetric difference of two halfspaces) in the original space. Therefore, the same ϵ-approximation technique used for Tukey depth, but with double wedge ranges, also applies to regression depth, and lets us compute depths and maintain an approximate deepest fit with high breakdown point for streaming data. Bern and Eppstein [4] generalized regression depth to the context of multivariate regression, in which the sample data have more than one dependent variable; in their definition, the depth of a fit is the minimum proportion of sample data contained in any double wedge, one boundary of which contains the fit and the other of which is parallel to the dependent coordinate axes; this is again well approximated by ϵ-approximations for double wedge ranges.

5.4 The Thiel-Sen Estimator

This estimator [39, 41] is a method for two-dimensional linear regression, in which one first finds the median among all $\binom{n}{2}$ slopes determined by the lines through pairs of sample points, and then selects a regression line having that median slope and bisecting the sample set. It has a breakdown point of $1 - \sqrt{1/2} \approx 0.293$. This has long been a testbed for geometric optimization algorithms, and several $O(n \log n)$ time static algorithms for it are known, among them one based on using ϵ-cuttings in a prune-and-search technique [5]. However these algorithms seem to require repeatedly scanning the data in a way that is unavailable to a streaming algorithm. Instead, we apply an approximation technique very similar to that for simplicial depth, above.

To begin with, suppose that we are given a query slope s, and must determine the approximate position of s within the sorted sequence of slopes, normalized by dividing the position by $\binom{n}{2}$. This can be solved exactly by a reduction to computing the number of inversions in a permutation, but we are interested in approximations that can be computed by a streaming algorithm that does not know s in advance. To do this, let δ be a parameter to be determined later, and imagine subdividing the sample points into a grid by $O(1/\delta)$ lines that are vertical and parallel to s, in such a way that at most a δ proportion of the points lie in the slab between any two adjacent parallel grid lines. Let e_1 be an estimate of the position of s, formed by summing up the normalized number of pairs of points that form a line with lower slope than s and that are in a pair of grid cells that are separated both by a vertical line of the grid and by a line parallel to s from the grid. Then e_1 is within $O(\delta)$ of the true position of s since the only lines through a given point that are omitted from the count are the ones where the other point determining the line is in one of the two slabs containing s, and e_1 can be expressed as a sum with $O(\delta^{-2})$ terms, each term being a product of the number of points in two parallelograms. Let e_2 be a similar normalized sum, with the number of sample points in each parallelogram replaced by the number of points of an ϵ-approximation for parallelogram ranges, and let e_3 be the normalized position of s within the set of lines determined by pairs of points from the ϵ-approximation. Then e_1 differs from e_2 by $O(\epsilon\delta^{-2})$ and e_2 differs from e_3 by $O(\delta + \epsilon\delta^{-1})$. Therefore, the overall error caused by using e_3 as our approximation to the position of s is $O(\delta + \epsilon\delta^{-2})$. Setting $\delta = \epsilon^{1/3}$ makes this total error equal $O(\epsilon^{1/3})$.

To compute an approximate Thiel-Sen estimator, we use the same ϵ-approximation for parallelograms. We compute the median slope among pairs of points from the approximation, and then find a line with that median slope bisecting the approximation. The resulting line has slope with a normalized position within $O(\epsilon^{1/3})$ of the median slope, partitions the sample points within ϵ of exact bisection, and has a breakdown point of $1 - \sqrt{1/2} - O(\epsilon^{1/3})$.

5.5 Least Median of Squares (LMS)

These methods [36] in robust statistics seek a fit that minimizes the median residual value separating the fit from the sample points. This is not a depth-based criterion, but it leads to fits which are highly robust against outliers. For location problems, the least median of squares fit is the center of the minimum radius sphere that contains at least half of the sample data [20]. It has a breakdown point of $\frac{1}{2}$: if fewer than half the sample data points are outliers, then the sphere defining the LMS fit has smaller radius than the circumsphere of the non-outliers, and it contains at least one non-outlier, so its center must be an accurate fit. Clearly, this is the best breakdown point possible for any location method. The natural type of ϵ-approximation to use for this problem is one with balls as its ranges. If we form the LMS fit of such an ϵ-approximation, the result may not be robust. Instead, we approximate the LMS fit by finding the center of the minimum radius sphere that contains at least a $\frac{1}{2} + \epsilon$ proportion of the points in the ϵ-approximation. Such a sphere must therefore contain at least half of the sample data, and has a radius at least as small as the smallest sphere containing at least a $\frac{1}{2} + 2\epsilon$ fraction of the sample data. It is robust with a breakdown point of $\frac{1}{2} - 2\epsilon$.

The same LMS approach can also be applied to regression problems. The least median of squares regression hyperplane can be defined as the central hyperplane in a slab bounded by two parallel hyperplanes, with minimum vertical separation between them, that contains at least half of the sample data; again this is robust with a breakdown point of $\frac{1}{2}$. As above, we can use an ϵ-approximation, with slab ranges, and find the slab with minimum vertical separation containing a $\frac{1}{2}+\epsilon$ fraction of the ϵ-approximation points, to produce an approximate LMS fit with breakdown point $\frac{1}{2}-2\epsilon$.

6 A Lower Bound on Range Counting

We provide a simple lower bound on the space required to count approximately the number of items in a range that is not necessarily an iceberg. When we say that an algorithm f-approximates the range counting problem we mean that if a given range contains l points, the algorithm gives us an answer which lies between l/f and $l \cdot f$.

The bound is stated in terms of *two-sided ranges*: a point (x, y) is said to belong to the two sided range located at (p, q) if $x \geq p$ and $y \geq q$.

Theorem 6.1 *Any f-approximate algorithm to the two-sided range counting problem must use space $\Omega(n/f^2)$.*

We begin by assuming there is an algorithm A which gives an f approximation to the two-sided range counting problem for a stream of points in two dimensions. Further we assume that this algorithm uses space $o(n/f^2)$.

Now consider a set of n points which are grouped in n/f^2 equally sized groups, we call them G_i, where $1 \leq i \leq n/f^2$, in the following way (Figure 3):

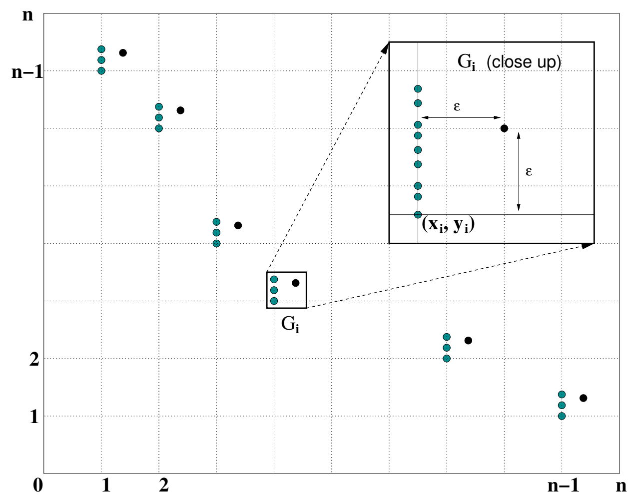

Figure 3: Input sequence for the lower bound on approximate range counting

- Each point in G_i has the same x coordinate, we call it x_i. For simplicity of presentation assume that all x_i values are integers. Additionally, we require $x_i > x_{i-1}$.
- All the points in G_i have y coordinates closely clustered at a given value, we call it y_i. Here too we assume that all y_i values are integers. Formally, for every $p_j \in G_i$, we say that $0 \leq y(p_j) - y_i < 1/2$.
- Every point $p_j \in G_i$ has y-coordinate strictly smaller than the y-coordinates of all the points in G_{i-1}.
- Each group i has an additional point $q_i = (x_i + \epsilon, y_i + \epsilon)$, for some $\epsilon < 1/2$, associated with it.

Note that this family of input sequences has the property that a two-sided query made at (x_i, y_i) should return a count of $f^2 + 1$ and one made at $(x_i + \frac{\epsilon}{2}, y_i + \frac{\epsilon}{2})$ should return a count of 1. This radical change in the counts will not occur between two such queries at any point which is not actually (x_i, y_i) for some value of i. As an extension to this simple observation, we note that since all the x_is and y_is are chosen out of the integers $1, 2, \ldots n$, it is possible to extract the exact values of all the x_i with $O(\frac{n \log n}{f^2})$ queries by using binary search.

Let us see if the algorithm A can be the query mechanism which we can deploy to this end. Since A is an f approximation, it should return a value of at most f at $(x_i + \frac{\epsilon}{2}, y_i + \frac{\epsilon}{2})$ and a value between $f + 1/f$ and $f^3 + f$ at (x_i, y_i). This means that A can indeed act as the oracle which identifies the locations of the groups in our set.

Hence, using A as a subroutine we can extract $\theta(n/f^2)$ information about the input set. This contradicts the assumption that A uses space $o(n/f^2)$. $\square$

Seen in the context of streaming algorithms, Theorem 6.1 implies that is not possible to approximate the range counting problem in polylogarithmic space. One of the implications of this, among others, is that it is not possible to count inversions in lists [19] in the sliding window model.

Acknowledgments. We would like to thank David Mount for helpful discussions of robust statistics in the context of the topics of this paper, and S. Muthukrishnan for helpful discussions on geometric streaming algorithms in general.

References

[1] N. Amenta, M. W. Bern, D. Eppstein, and S.-H. Teng. Regression depth and center points. *Discrete & Computational Geometry*, 23(3):305–323, 2000. arxiv:cs.CG/9809037.

[2] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom. Models and Issues in Data Stream Systems. In *Proc. of the 22nd Annual ACM Symp. on Principles of Databases Systems*, 2002. `dbpubs.stanford.edu:8090/pub/2002-19`.

[3] A. Bagchi, A. Chaudhary, D. Eppstein, and M. T. Goodrich. Deterministic sampling and range counting in geometric data streams. In *Proc. of the 20th ACM Symposium on Computational Geometry*, pages 144–151, Brooklyn, NY, 2004.

[4] M. W. Bern and D. Eppstein. Multivariate regression depth. *Discrete & Computational Geometry*, 28(1):1–17, July 2002. arxiv:cs.CG/9912013.

[5] H. Brönnimann and B. Chazelle. Optimal slope selection via cuttings. *Computational Geometry: Theory and Applications*, 10:23–39, 1998.

[6] H. Brönnimann, B. Chen, M. Dash, P. Haas, Y. Qiao, and P. Scheuerman. Efficient data-reduction methods for on-line association rule discovery. In *Data Mining: Next Generation Challenges and Future Directions, Selected papers from the NSF Workshop on Next-Generation Data Mining (NGDM '02)*, pages 190–208. MIT Press, 2003.

[7] H. Brönnimann, B. Chen, M. Dash, P. Haas, and P. Scheuermann. Efficient data reduction with ease. In *Proc. of the 9th Annual ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data mining (KDD '03)*, pages 59–68, 2003.

[8] B. Chazelle, D. Liu, and A. Magen. Sublinear geometric algorithms. In *Proc. of the 35th Annual ACM Symp. on Theory of Computing*, pages 531–540, 2003.

[9] B. Chazelle and E. Welzl. Quasi-optimal range searching in spaces of finite VC-dimension. *Discrete Comput. Geom.*, 4:467–489, 1989.

[10] K. Clarkson, D. Eppstein, G. L. Miller, C. Sturtivant, and S.-H. Teng. Approximating center points with iterated Radon points. *Intl. Journal of Computational Geometry & Applications*, 6(3):357–377, 1996.

[11] G. Cormode and S. Muthukrishnan. Radial Histograms for Spatial Streams. Technical Report 2003-11, DIMACS, 2003.

[12] M. Datar, A. Gionis, P. Indyk, and R. Motwani. Maintaining Stream Statistics over Sliding Windows. In *Proc. of the 13th Annual ACM-SIAM Symp. on Discrete Algorithms*, pages 635–644, 2002.

[13] E. D. Demaine, A. López-Ortiz, and J. I. Munro. Frequency estimation of internet packet streams with limited space. In *Proc. of the 10th Annual European Symp. on Algorithms (ESA 2002)*, pages 348–360, 2002.

[14] D. L. Donoho. *Breakdown properties of multivariate location estimators*. PhD thesis, Harvard University, 1982.

[15] M. Fang, N. Shivakumar, H. Garcia-Molina, R. Motwani, and J. D. Ullman. Computing iceberg queries efficiently. In *Proc. of the 1998 Intl. Conf. on Very Large Data Bases*, pages 299–310, 1998.

[16] J. Feigenbaum, S. Kannan, and J. Zhang. Computing Diameter in the Streaming and Sliding-Window Models. Technical Report YALEU/DCS/TR-1245, Yale University, 2002.

[17] M. T. Goodrich and E. A. Ramos. Bounded-independence derandomization of geometric partitioning with applications to parallel fixed-dimensional linear programming. *Discrete Comput. Geom.*, 18:397–420, 1997.

[18] M. Greenwald and S. Khanna. Space-efficient online computation of quantile summaries. In *Proc. of the 2001 ACM SIGMOD Intl. Conf. on Management of Data*, pages 58–66, 2001.

[19] A. Gupta and F. X. Zane. Counting inversions in lists. In *Proc. of the 14th Annual ACM-SIAM Symp. on Discrete Algorithms*, pages 253–254, 2003.

[20] S. Har-Peled and S. Mazumdar. Fast algorithms for computing the smallest k-enclosing disc. In *Proc. of the 11th Annual European Symp. on Algorithms*, Lecture Notes in Computer Science. Springer-Verlag, 2003.

[21] J. Hershberger and S. Suri. Convex hulls and related problems in data streams. In *Proc. of the ACM/DIMACS Workshop on Management and Processing of Data Streams*, 2003. Available online at http://www.research.att.com/conf/mpds2003/.

[22] P. Indyk. Better algorithms for high-dimensional proximity problems via asymmetric embeddings. In *Proc. of the 14th Annual ACM-SIAM Symp. on Discrete Algorithms*, pages 539–545, 2003.

[23] P. Indyk. Stream-based geometric algorithms. In *Proc. of the ACM/DIMACS Workshop on Management and Processing of Data Streams*, 2003. Available online at http://www.research.att.com/conf/mpds2003/.

[24] S. Jadhav and A. Mukhopadhyay. Computing a centerpoint of a finite planar set of points in linear time. In *Proc. of the 9th Annual ACM Symp. on Computational Geometry*, pages 83–90, 1993.

[25] R. M. Karp, S. Shenker, and C. H. Papadimitriou. A simple algorithm for finding frequent elements in streams and bags. *ACM Transactions on Database Systems*, 28(1):51–55, March 2003.

[26] F. Korn, S. Muthukrishnan, and D. Srivastava. Reverse nearest neighbour aggregates over data streams. In *Proc. of the 2002 Intl. Conf. on Very Large Data Bases*, pages 814–825, 2002.

[27] S. Langerman and W. Steiger. Optimization in arrangements. In H. Alt and M. Habib, editors, *Proc. 20th Intl. Symp. Theoretical Aspects of Computer Science*, number 2607 in Lecture Notes in Computer Science, pages 50–61. Springer-Verlag, 2003.

[28] R. Y. Liu. On a notion of data depth based on random simplices. *Annals of Statistics*, 18:405–414, 1990.

[29] G. S. Manku and R. Motwani. Approximate Frequency Counts over Data Streams. In *Proc. of the 2002 Intl. Conf. on Very Large Data Bases*, pages 346–357, 2002.

[30] J. Matoušek. Computing the center of planar point sets. In J. E. Goodman, R. Pollack, and W. Steiger, editors, *Discrete and Computational Geometry: Papers from the DIMACS Special Year*, number 6 in DIMACS Series in Discrete Mathematics and Theoretical Computer Science, pages 221–230. AMS, 1991.

[31] J. Matoušek. Epsilon-nets and computational geometry. In J. Pach, editor, *New Trends in Discrete and Computational Geometry*, volume 10 of *Algorithms and Combinatorics*, pages 69–89. Springer-Verlag, Heidelberg, 1993.

[32] J. Matoušek. Approximations and optimal geometric divide-and-conquer. *J. Comput. Syst. Sci.*, 50:203–208, 1995.

[33] J. Matoušek. Derandomization in computational geometry. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 559–595. Elsevier Science Publishers B.V. North-Holland, Amsterdam, 2000.

[34] S. Muthukrishnan. Data Streams: Algorithms and Applications. Invited talk at SODA 2003. Available on request by email to `muthu@research.att.com`.

[35] P. J. Rousseeuw and M. Hubert. Regression depth. *Journal of the American Statistical Association*, 94(446):388–402, 1999.

[36] P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection*. John Wiley & Sons, 1987.

[37] P. J. Rousseeuw and A. Struyf. Computing location depth and regression depth in higher dimensions. *Statistics and Computing*, 8:193–203, 1998.

[38] N. Sauer. On the density of families of sets. *J. Combin. Theory Ser. A*, 13:145–147, 1972.

[39] P. K. Sen. Estimates of the regression coefficient based on Kendall's tau. *Journal of the American Statistical Association*, 63:1379–1389, 1968.

[40] S. Suri, C. D. Tóth, and Y. Zhou. Range counting over multidimensional data streams. In *Proc. of the 20th ACM Symposium on Computational Geometry*, pages 160–169, Brooklyn, NY, 2004.

[41] H. Thiel. A rank-invariant method of linear and polynomial regression analysis, part 3. *Proc. of Koninalijke Nederlandse Akademie van Weinenschatpen A*, 53:1397–1412, 1950.

[42] V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory Probab. Appl.*, 16:264–280, 1971.