Data Streaming Algorithms for the Kolmogorov-Smirnov Test

Ashwin Lall
Department of Mathematics and Computer Science
Denison University
Granville, OH, USA
lalla @ denison.edu

***Abstract*—We propose space-efficient algorithms for performing the Kolmogorov-Smirnov test on streaming data. The Kolmogorov-Smirnov test is a non-parametric test for measuring the strength of a hypothesis that some data is drawn from a fixed distribution (one-sample test), or that two sets of data are drawn from the same distribution (two-sample test). Unlike some other tests, Kolmogorov-Smirnov does not assume that the distribution has a known form (e.g., it is normal), and in the two-sample case it need not know anything about the distribution, other than that it is continuous. Motivated by the challenges of big data, we present algorithms for both the one-sample and the two-sample tests for data processed in a stream. We demonstrate the accuracy of our algorithms via extensive experimentation on both real and synthetic datasets. We show that our algorithms are superior to sampling and that they accurately perform the test with several orders of magnitude reduction in data.**

I. Introduction

One of the characteristic challenges of Big Data is the lack of capacity to store all of it. It is currently infeasible to collect data at the core of the Internet or on many scientific measurement devices without massive downsampling. Even simple tasks such as detecting significant changes in the data have become challenging because sampling tends to discard much of the information. This paper addresses the need to identify the source distribution of a stream of data, or to compare the distributions of different streams. For instance, we may study packet inter-arrival in a high-speed network and want to know whether the distribution of these times is identical on different days or on different routers, or sensor motes may need to maintain months of data in their limited storage. Unfortunately, in many of these applications, to maintain a complete record would very quickly overwhelm local storage capacity—in the abovementioned examples, the data rate is orders of magnitude greater than available memory. To overcome this problem, it is necessary to perform comparative distributional tests on summaries of the data. In this paper, we show how one such statistic for comparing distributions, the Kolmogorov-Smirnov statistic, can be estimated succinctly yet accurately in a stream.

The Kolmogorov-Smirnov test (henceforth referred to as the *KS test*) is a means for measuring whether given data are drawn from a specific distribution. The power of this test comes from the fact that it is non-parametric, i.e., it does not assume some fixed type of distribution (such as the normal distribution in the case of the Student t-test) and can be applied to any distribution with a continuous distribution function. This lack of restriction makes it invaluable for inferring whether data fits a given distribution when it is preferable to not assume a fixed parametric distribution, or when the distribution does not have well-established tests of its own. It is also superior to χ^2 tests in that there is no need to determine how to bin the data or ensure that there is sufficient density in each bin. The test computes a statistic of the distribution function of the data which is used to reject the hypothesis that the distributions are identical up to a given significance level (e.g., $\alpha = 0.05$).

The KS test has both a one-sample and a two-sample variant. In the one-sample variant, empirical data (e.g., packet inter-arrival times or luminosity) can be tested against a fixed, known distribution to see whether the data are drawn from this distribution. This test is useful for verifying whether the data is from a known distribution that does not have its own parametric test. The two-sample version of the test allows for the comparison of two (not necessarily equal-size) datasets without any foreknowledge of the underlying distributions. This test can be used to check whether the two sources are different. We provide the first streaming algorithms for both these tests in this paper.

Much of the literature on statistical testing of streams has focused on statistics of the frequencies of the items in the stream (see, e.g., [16], [17]). This is due, in part, to the fact that frequency-based problems are challenging in the streaming model. In contrast, problems that have to do with measuring statistics on the stream values themselves (e.g., mean, standard deviation, etc.) are usually easy to compute in a stream. The KS-test is a notable exception to this rule and hence makes for an interesting problem to study.

A significant advantage of using the KS test for tasks such as change and anomaly detection is that it can be applied to wide range of very heterogeneous types of data. For instance, the same algorithm can be used for detecting changes in quantities as diverse as round-trip-time (RTT), latency, throughput, jitter, and loss in the networking domain. Moreover, there is barely any parametrization required to deploy the test on a real system. Simply by using the two-sample test to compare against data pre-collected during some periods in which "normal" behavior is observed, one can detect significant deviations in the distribution. This makes the KS test very easy for system engineers to deploy. We believe that it has not found more widespread use because of the obstacle of prohibitive cost, which can be removed by the techniques in this paper.

A. *Applications*

We outline below a few of the applications for which streaming algorithms for the KS test would be useful.

Astronomy: The KS test is commonly used in the field of astronomy to measure the distance between distributions of astronomical measurements [25]. The recent increase in the amount of data available to astronomers will soon make storing these measurements very challenging. For instance, the Chandra space telescope [22] is capable of recording data at the rate of 1.8 gbps, but it has a downlink capacity of only 1 mbps to Earth. Another telescope under development, the Square Kilometre Array [23], will generate data at the rate of several times the traffic of the entire Internet!

Wireless Sensor Networks: One of the most common uses of wireless sensor networks is to perform scientific measurements at remote or wide-spread locations. These sensor networks consist of sensor motes that have limited resources such as battery-life, memory, processing power, etc. [18]. Being able to perform statistical tests to detect changes in or to measure properties of the distributions of the measurements necessitates the ability to retain the sensed data in the mote's limited storage. The techniques in this paper could be used to perform light-weight tests on the data to detect significant changes in measurement.

Internet Measurement: The inter-arrival time between packets is a common metric for network measurements [4], [13]. An algorithm for measuring the KS-statistic would give network operators the ability to detect when the packet arrival rate changes significantly or to match an arrival pattern with known distributions of previously-identified behavior. Since this data is generated at the rate of many gigabytes per second across a large ISP, it is infeasible to keep a long-term record of this data. The algorithms proposed in this paper would allow for succinct storage of these measurements. Other quantities that could be compared in this way are round-trip-time (RTT), packet size, delay, loss, and latency.

B. *Contributions*

The contributions of this paper are as follows:

- We propose an algorithm for the one-sample KS test to test whether a source of data is drawn from a fixed (known) distribution. The algorithm can compress n items of information into $\Theta(\sqrt{n}\log n)$ space (e.g., terabytes into megabytes). It does not need to know the distribution being tested against *a priori*.
- We design an algorithm for the two-sample KS test to test whether two sources of data are from the same (unknown) distribution, once again using $\Theta(\sqrt{n}\log n)$ memory. In this case, nothing needs to be known about either underlying source, other than the fact that they have continuous distributions.
- We performed extensive experiments on both real and synthetic data to demonstrate that the proposed algorithms do perform well in practice, and give considerable benefit over simple strategies such as sampling the data.

Organization: In Section II the work most directly related to this problem is discussed. We define the problem and introduce quantile sketches in Section III. The algorithms for the one-sample and two-sample KS test are given in Sections IV and V, respectively. In Section VI we show how to pick the error parameters in our algorithms so as to guarantee a reliable answer to the KS test. The algorithms are evaluated on both real and synthetic data in Section VII. Lastly, we discussion our conclusions and future work in Section VIII.

II. Related Work

The Kolmogorov-Smirnov test is a commonly used means to distinguish distributions. Its strength lies in that it does not assume that the data are from a fixed distribution (e.g., Gaussian) and can be applied for arbitrary continuous distributions. For this reason it has commonly been suggested as a way to distinguish change in data. In one such work [14], perhaps most closely related with this paper, the authors use the KS test (among several others) to detect significant changes in a stream. However, their sliding window algorithm assumes that the entire window is stored and instead focuses on how to re-compute the KS-statistic quickly. The purpose of our work is to estimate the same quantity using considerably less space than holding the entire stream in memory. Another significant difference between our methodologies is that, while [14] focuses on detecting change in a single stream, the goal of this paper is to compute a succinct summary, called a sketch, that can be used to differentiate streams processed at different times or locations. Another recent work [9] defines a different version of the KS test for high-dimensional data. The variations they introduce for more dimensions make their results incomparable with ours.

Computing distances in streams has been done for a long time. For instance, [12] examines how to compute a general class of information distances in a stream. This model is dissimilar from ours, though, in that they study the item frequency distribution, rather than the raw values the appear in the stream. Batu et al. [1] studied the estimation of the L_1-distance in a stream and proposed a near-tight algorithm to match the $\Omega(n^{2/3})$ lower bound that they show for the L_1 case. One major difference between much previous work and ours is that we study a distance that has easy-to-understand, statistical significance—it is possible to reject a hypothesis upto arbitrary levels of significance using the KS test. The algorithms in this paper can hence be deployed with minimal fine-tuning or parametrization.

This paper uses previous work on computation of quantiles in a stream, one of the longest-studied problems in streaming algorithms. (See [2] for a survey.) This line of inquiry was initiated by Munro and Paterson [21], and their results were subsequently improved by Manku et al. [19] and Greenwald and Khanna [10]. The Greenwald-Khanna algorithm needs only $O(\frac{1}{\epsilon}\log(\epsilon n))$ space, where n is the length of the stream and ϵn is a bound on the rank error, and is considered the state of the art algorithm for computing quantiles in streams. By way of contrast, there is a folklore result that random sampling needs $\theta(1/\epsilon^2)$ samples [2], which is much worse for small ϵ. There have also been several other developments in quantile computation, such as the q-digest sketch [24] that uses $O(\frac{1}{\epsilon}\log U)$ memory, where U is the size of the input domain. The q-digest sketch also has the property that it can be efficiently aggregated in distributed environments, e.g., sensor networks. Also, in [6] Cormode and Muthukrishnan discuss how their count-min sketch can be used to extend the work in [8] to get a more space-efficient streaming algorithm for quantiles. Zhang and Wei [27] gave a fast-update algorithm for quantiles, but this comes at the cost of a $O(\log(\epsilon n))$ increase in the storage requirement over the Greenwald-Khanna algorithm. There has been some recent work that focuses on randomly ordered streams [3], [11] that takes advantage of this randomness to reduce the space complexity for measurement. Very recently, there has been an extensive experimental study to determine which quantile algorithms perform best in practice [26].

There has been a revived interest in the benefits of sampling of late [5], [7], [20]. One of the major contributions of this paper is to demonstrate that our streaming algorithm can outperform sampling-based techniques, indicating that sampling is not a good solution for this technique. We demonstrate this both theoretically as well as experimentally.

III. Preliminaries

A. Problem Definition

In this paper, we will denote the length of the stream by n; since it is possible to maintain the length of a stream using only $\lceil \log n \rceil$ bits, we assume that this value is available after processing the stream. The streams are of real-valued numbers that can be stored with unit cost (with some fixed precision), but cannot have any computation other than the standard comparison operations performed on them. We denote the items by the values $\{X_1, X_2, X_3, \ldots, X_n\}$. In order to simplify notation later, we assume that $X_1 \leq X_2 \leq X_3 \leq \ldots \leq X_n$ and that the stream is presented in some order $[X_{\pi(1)}, \ldots, X_{\pi(n)}]$, where π is an arbitrary permutation of the values $\{1, \ldots, n\}$, i.e., the stream is not sorted in any particular order.

Now, in order to define the problem addressed in this paper, we remind the reader about the definition of several terms from probability theory. The *distribution function* (sometimes referred to as the *cumulative distribution function* or c.d.f.) of a distribution is defined as the function $F(x) = \Pr(X \leq x)$, where X is a random variable drawn from the distribution. The *empirical distribution function* of a series of observations of some random quantity $X_1, \ldots, X_n$ is defined as $F_n(x) = \frac{|\{i \mid X_i \leq x\}|}{n}$. Both the distribution and the empirical distribution functions are defined over all of $\mathbb{R}$ and take values within the range $[0, 1]$.

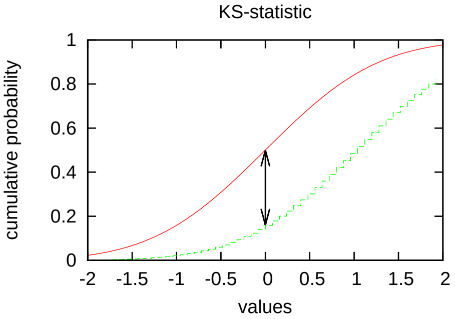


Figure 1. KS-statistic: greatest difference between distribution functions

The one-sample KS test compares a set of discrete data with a fixed, continuous distribution function to see if the data are drawn from this distribution. Specifically, if the empirical distribution function of a stream of length n is given by F_n, then the KS-statistic indicating the distance between this empirical distribution and some fixed distribution F is given by

$$D_n = \sup_x |F_n(x) - F(x)|.$$

The KS test simply computes the largest absolute difference between the empirical distribution and the distribution it is being tested against. This is illustrated in Figure 1. The value of this statistic can be shown to be independent of the distribution in question (i.e., it is called distribution free) and there are tables of values available for the critical region of the test. That is, for $0 < \alpha < 1$, there is some fixed value K_α such that the null hypothesis (the data $\{X_i\}$ is drawn from the distribution F) is rejected at level α if $\sqrt{n}D_n > K_\alpha$.

Note that it is not necessary to compute the above supremum over infinitely many values of x since the empirical distribution function only changes at n discrete values. Specifically, if the set of points are $X_1 \leq \ldots \leq X_n$ and the distribution is defined by the c.d.f. F, then the KS-statistic can alternatively be defined as (see, e.g., Knuth [15])

$$D_n = \max_{1 \leq i \leq n} \max\left(\frac{i}{n} - F(X_i), F(X_i) - \frac{i-1}{n}\right).$$

The two-sample KS test allows one to compare two sets of samples and test the likelihood that they came from the same underlying distribution. More formally, assume that we have two collections of points with cardinality n and m, and empirical distribution functions F_n and G_m, respectively. Then, the two-sample KS statistic for these points is

$$D_{n,m} = \max_x |F_n(x) - G_m(x)|.$$

In practice, the two sample test is even more useful because it can be used to compare two streams with no knowledge about their underlying distributions. For the two-sample test, the null hypothesis is rejected at level α when $\sqrt{\frac{nm}{n+m}} D_{n,m} > K_\alpha$.

B. Quantile Sketches

The algorithms in this paper will make use of streaming data structures, known as sketches, for computing the quantiles of the values in a stream. We define such a sketch as follows:

Definition 1: A *quantile ϵ-sketch* is a data structure that, given an input stream $X_1, \ldots, X_n$ ($X_1 \le X_2 \le X_3 \le \ldots \le X_n$) in arbitrary order, can then be queried to return, for any $1 \le i \le n$, a value X_j such that $j \in [i - \epsilon n, i + \epsilon n]$.
These sketches are assumed to need less memory than storing all the values exactly (i.e., $o(n)$ bits), and can typically be updated very quickly.

The quantile values are in some sense the "inverse" of what we would like to have to compute the KS statistic. The main challenge in this paper is to extract this inverse and to bound the error of the KS statistic.

IV. One-sample Test

In this section we describe a streaming algorithm for the one-sample variant of the test. The algorithm must create a succinct summary, or sketch, of the stream so that its distribution can be compared with any distribution afterwards. This version of the test is useful when we do not know the type of distribution (e.g., normal, Pareto) we are going to compare against or its parameters (e.g., mean, variance) *a priori*. In other words, the sketch must be able to guarantee bounded error for any possible distribution function.

Our algorithm uses a quantile sketch (e.g., [10]) to maintain the quantiles in a stream (in a single pass) with the following guarantee (for any fixed $\epsilon > 0$): For any given rank r, the quantile sketch will return an element whose rank is within the range $[r - \epsilon n, r + \epsilon n]$. Note that computation of the KS-statistic requires not the quantiles themselves but the "inverse" of the quantiles, and so the main technical challenge of the algorithm and analysis is to carefully computes these values at all the necessary points. In order to do this, we prove the following observations (needed in the analysis of our algorithms):

Observation 1: It is possible to extract from a quantile ϵ-sketch a subset $\{X_{i_1}, \ldots X_{i_k}\} \subseteq \{X_1, \ldots, X_n\}$ (where $X_1 \le X_2 \le X_3 \le \ldots \le X_n$) such that $i_1 < i_2 < i_3 < \ldots < i_k$ and, for all $1 \le j < k$, $i_{j+1} - i_j < 2\epsilon n$.

Algorithm 1 OneSample(Q, n, F)

Input: Quantile ϵ-sketch Q of a stream of size n, and a distribution function F
Output: $\hat{D}$, an estimate of the KS-statistic D

```
1: Let $X_{i_1} \le \ldots \le \ldots \le X_{i_k}$ be the values in $Q$, as
   described in Observation 1.
2: $\hat{D} = 0$
3: for each $x \in \{X_{i_1}, \ldots, X_{i_k}\}$ do
4:     Let $j = \max\{p \mid X_{i_p} \le x\}$.
5:     Let $\hat{i}_j$ be the approximate index of $X_{i_j}$, computed as
       described in Observation 2.
6:     $\hat{E}_x = |\hat{i}_j/n - F(x)|$
7:     $\hat{D} = \max(\hat{D}, \hat{E}_x)$
8: return $\hat{D}$
```

Proof: Since a quantile sketch is guaranteed to return only values from the original data stream, any values that the sketch contains can be extracted via querying for the ith largest element (for $1 \le i \le n$) from the sketch. Let $X_{i_1} \le \ldots \le X_{i_k}$ be the values that result from these queries.

Now, fix any $j \in \{1, \ldots, k-1\}$. If it is the case that $i_{j+1} - i_j > 2\epsilon n$, then there must be some i', $i_j < i' < i_{j+1}$, such that querying the sketch for the i'th largest element will give a value with rank error at least ϵn, a contradiction. Hence, it must be that $i_{j+1} - i_j \le 2\epsilon n$. ■

Observation 2: Given some value X_i returned by a quantile ϵ-sketch (where X_i is the ith largest element in the input), it is possible to estimate i to within ϵn additive error.

Proof: Performing a binary search among the indices $1, \ldots, n$ for the value X_i will give the desired approximation to the index. ■

Recall that our goal is to compute the KS-statistic of the empirical distribution of a set of points $X_1, \ldots, X_n$ (where $X_1 \le X_2 \le X_3 \le \ldots \le X_n$), denoted by F_n, from some arbitrary distribution F via the formula $D_n = \sup_x |F(x) - F_n(x)|$. We achieve an approximation $\hat{D}$ to this value D_n by using a quantile ϵ-sketch to summarize the data being streamed and then comparing the result to the fixed distribution F. The pseudocode for the comparison is given in Algorithm 1. Note that the streaming part of the algorithm (the quantile sketch) is independent of the distribution F.

Theorem 1: Algorithm 1 returns an estimate of the KS-statistic with at most 3ϵ additive error.

Proof: For any x, let $E_x = |F(x) - F_n(x)|$. Recall that our goal is to compute $D_n = \max_x E_x$. Now, let $X_1 \le \ldots \le X_n$ be the data in the stream (in ascending order) and let $X_{i_1} \le \ldots \le X_{i_k}$ be the values in the sketch, as computed in line 1 of Algorithm 1. We first show how to estimate $F_n(x)$ approximately using the sketch Q.

For any x, let i be such that $X_i \le x < X_{i+1}$, i.e., the largest index of the data that is at most x. Then, by definition, $F_n(x) = i/n$. Let j be the largest index such that $X_{i_j} \le x <$

$X_{i_{j+1}}$. Note that this corresponds with the value j computed on line 4 of Algorithm 1. We now use the fact that i was chosen to be the largest index such that $X_i \leq x < X_{i+1}$, and the fact that $\{X_{i_1}, \ldots, X_{i_k}\} \subseteq \{X_1, \ldots, X_n\}$ to get that $X_{i_j} \leq X_i \leq x < X_{i+1} \leq X_{i_{j+1}}$. This follows since i was chosen to be the maximal such value and the sketch has a subset of the $\{X_i\}$'s.

We use the fact that Observation 1 tells us that $i_{j+1} - i_j \leq 2\epsilon n$. Combining this with the above inequalities, and the fact that the sequences $\{X_i\}$ and $\{X_{i_j}\}$ are monotonic, we get that $i - i_j \leq 2\epsilon n$.

Now, line 4 of Algorithm 1 gives the value of X_{i_j}, but not the value of the index i_j. To compute this, we make use of Observation 2 to compute an estimate $\hat{i_j}$ in line 5. This estimate is guaranteed to have at most ϵn additive error. Putting this together with $i - i_j \leq 2\epsilon n$ and using the triangle inequality, we get that $|i - \hat{i_j}| \leq 3\epsilon n$.

We now have, for any given x, an estimate of $F_n(x)$ computed as $\hat{i_j}/n$ with at most 3ϵ additive error from the actual value i/n.

Lastly, instead of computing $\hat{E_x}$ (the estimate of E_x) for every x, we restrict the computation to just the values extracted from the sketch since these are the critical values at which the empirical distribution function changes. ■

A. *Computational Analysis*

The streaming part of the algorithm is identical to that of whichever quantile ϵ-sketch is employed by the algorithm. For instance, in the case of the Greenwald-Khanna sketch, summarizing n data points with up to ϵ error can be done using at most $O(\frac{1}{\epsilon}\log(\epsilon n))$ space and time per update. Since the bound given by the algorithm with $\epsilon' = 3\epsilon$ is off by a constant from this guarantee, it is easy to see that the same asymptotic guarantee is possible (replacing ϵ with ϵ').

The computational complexity of measuring the KS-distance is less important since this can be done offline, well after the stream is processed, but we analyze it here anyway. The running time of Algorithm 1 is dominated by the time needed to extract the the values X_{i_j} from the quantile sketch. This takes $O(n)$ query operations to the sketch. In contrast, the rest of the algorithm is relatively fast since, if there are $s = o(n)$ unique values stored in the sketch (e.g., $s = O(\frac{1}{\epsilon}\log(\epsilon n))$ for the Greenwald-Khanna sketch), then the algorithm iterates s times and performs $O(\log s)$ computations for the binary search on line 5 of Algorithm 1, giving a running time of $O(s \log s)$, which is much less than the initial query operations.

V. Two-Sample Test

The two-sample KS test is used in situations in which two datasets need to be compared to see if they come from the same distribution. A significant advantage it has over the one-sample test is that there is no need to assume anything about the distribution that both samples are drawn from. As a result, it is more commonly used in practice.

```
Algorithm 2 TwoSample($Q_1$, $n$, $Q_2$, $m$)
Input: Quantile $\epsilon$-sketches $Q_1$ and $Q_2$ of streams with sizes $n$ and $m$, respectively
Output: $\hat{D}$, an estimate of the KS-statistic $D$
 1: Let $X_{i_1} \leq \ldots \leq X_{i_k}$ be the values in $Q_1$, as described in Observation 1.
 2: Let $Y_{j_1} \leq \ldots \leq Y_{j_l}$ be the values in $Q_2$, as described in Observation 1.
 3: $\hat{D} = 0$
 4: for each $x \in \{X_{i_1}, \ldots, X_{i_k}\} \cup \{Y_{j_1}, \ldots, Y_{j_l}\}$ do
 5:     Let $a = \max\{j \mid X_{i_j} \leq x\}$.
 6:     Let $\hat{i}_a$ be the approximate index of $X_{i_a}$, computed as described in Observation 2.
 7:     Let $b = \max\{i \mid Y_{j_i} \leq x\}$.
 8:     Let $\hat{j}_b$ be the approximate index of $Y_{j_b}$, computed as described in Observation 2.
 9:     $\hat{E}_x = |\hat{i}_a/n - \hat{j}_b/m|$
10:     $\hat{D} = \max(\hat{D}, \hat{E}_x)$
11: return $\hat{D}$
```

Just as for the one-sample test algorithm, we use quantile ϵ-sketches to solve this problem. The major difference here is that we assume that the sketches from the two streams (samples) are shipped to a common location for the computation to be performed. Note that this algorithm allows for pairwise comparison of any number of streams, as long as the sketches are all in the same location. Moreover, transmitting these sketches is much more bandwidth-efficient than sending the entire stream in distributed settings.

A. *Two-sample algorithm*

To compute the KS-statistic, we need to be able to find the maximum of $|F_n(v) - G_m(v)|$ over all values v. Fortunately, rather than having to check all (possibly infinite) such values, we can take advantage of the fact that the empirical distribution is discrete and only check at the values v such that $F_n(v)$ or $G_m(v)$ is in the set $\{i/n \mid 0 \leq i \leq n\} \cup \{i/m \mid 0 \leq i \leq m\}$, where n and m are the lengths of the two streams.

Theorem 2: Algorithm 2 returns an estimate of the KS-statistic with at most 6ϵ additive error.

Proof: Our goal is to compute $D_{n,m} = \sup_x |F_n(x) - G_m(x)|$. For any x, let $E_x = |F_n(x) - G_m(x)|$. Let $i = \max\{i \mid X_i \leq x\}$. Similarly, define $j = \max\{j \mid Y_j \leq x\}$. We know that $F_n(x)$ must be i/n and $G_m(x)$ must be j/m, by definition, so we have that $E_x = |i/n - j/m|$. We compare this value with that of $\hat{E}_x$ computed in line 9 of Algorithm 2 below.

Let $X_{i_1} \leq \ldots \leq \ldots X_{i_k}$ and $Y_{j_1} \leq \ldots \leq \ldots Y_{j_l}$ be the values stored in the sketches, defined as in lines 1-2 of Algorithm 2. Let $a = \max\{j \mid X_{i_j} \leq x\}$ and

$b = \max\{i \mid Y_{j_i} \leq x\}$. Since i is defined such that $X_i \leq x < X_{i+1}$, we have that $X_{i_a} \leq X_i \leq x \leq X_{i+1} \leq X_{i_{a+1}}$, where the first inequality follows from the fact that X_i was chosen as the largest value among $X_1, \ldots, X_n$ that is at most x and because $\{X_{i_1}, \ldots, X_{i_k}\} \subseteq \{X_1, \ldots, X_n\}$. Similarly, the last inequality follows from the fact that X_{i+1} is the smallest value among $X_1, \ldots, X_n$ that is greater than x.

We know from Observation 1 that the indexes i_a and i_{a+1} are such that $i_{a+1} - i_a \leq 2\epsilon n$. Combining with the result above, this implies that $i - i_a \leq 2\epsilon n$.

Now, keep in mind that even though the sketch returns the value X_{i_a}, it does not have the exact value of i_a available to it. However, we can approximate this value as $\hat{i}_a$ (line 6 of Algorithm 2) by performing a binary search of the quantile sketch, as described in Observation 2. We have that this approximation $\hat{i}_a$ of i_a is such that $|\hat{i}_a - i_a| \leq \epsilon n$.

Putting together the above two inequalities and using the triangle inequality we get that $|i - \hat{i}_a| \leq |i - i_a| + |\hat{i}_a - i_a| \leq 2\epsilon n + \epsilon n = 3\epsilon n$.

In exactly the same way we can show that the estimate $\hat{j}_b$ computed in line 8 of Algorithm 2 is such that $|j - \hat{j}_b| \leq 3\epsilon m$.

Putting all this together, we get that since the error of estimating i/n by $\hat{i}_a/n$ is at most 3ϵ and the error of estimating j/m by $\hat{j}_b/m$ is at most 3ϵ, the error of estimating $E_x = |i/n - j/m|$ by $\hat{E}_x = |\hat{i}_a/n - \hat{j}_b/m|$ is at most 6ϵ.

Finally, note that we do not have to repeat the process above for *every* value of x, just the ones that give a different answer. Since the approximation only changes for values of x among $\{X_{i_1}, \ldots, X_{i_k}\} \cup \{Y_{j_1}, \ldots, Y_{j_l}\}$, it suffices to approximate E_x for these values. ■

B. Computational Analysis

The analysis of the online computation for the two-sample case is identical to that of the one-sample case. Hence, we focus on just the offline computation cost here. Once again, this cost is dominated by the $n + m$ queries performed in lines 1-2 of Algorithm 2. The running time of the following iterations is $o(n + m)$, once again depending on the number of samples that the quantile ϵ-sketches end up storing.

VI. Picking ϵ

So far, we have discussed how to bound the error on the KS-statistic in comparison with the error (ϵ) of the quantile ϵ-sketch. The obvious question that arises (and that we address in this section) is:

> What types of error bounds are necessary for computing the KS-statistic in practice?

Recall that the form of the one-sample KS test is to compute the KS-statistic D and then find the significance level α at which $\sqrt{n}D > K_\alpha$. For instance, if the desired significance level is $\alpha = 0.05$, then we reject the hypothesis that the data is drawn from the distribution being tested against exactly when $\sqrt{n}D > K_{0.05} \approx 1.358$. On the other hand, if this value were to exceed $K_{0.01} \approx 1.628$, then we could reject the hypothesis at the 0.01 significance level. Clearly, our goal should be to select the absolute error of the quantile ϵ-sketch to be sufficiently low so as not to adversely affect this comparison—to (not) reject the null hypothesis erroneously.

Suppose that the KS test is being applied to reject the null hypothesis at the $\alpha = 0.05$ significance level. Then, the test simply needs to check if $\sqrt{n}D > K_{0.05} \approx 1.358$. The error introduced by using our sketch, rather than all of the data in the stream, should be small enough to not degrade the quality of the test. The form of approximation this would take is that we should always reject the hypothesis at the $\alpha = 0.04$ (or lower) significance level, but never reject it at the $\alpha = 0.06$ (or higher) significance level. Since $K_{0.04} = 1.399$ and $K_{0.06} = 1.324$, it is clear that in this example an error in $\sqrt{n}D$ of up to 0.03 can be tolerated.

Let us say that our goal is to compute the quantity $\sqrt{n}D_n$ to within δ precision. To achieve this level of accuracy, we have to determine how many samples are necessary in the quantile ϵ-sketch. For the purposes of this analysis we will use the Greenwald-Khanna sketch as an example since it is considered the state-of-the-art for computing quantiles on a stream. The Greenwald-Khanna sketch needs $O(\frac{1}{\epsilon}\log(\epsilon n))$ samples to guarantee an error of ϵ. Now, recall from Section IV that an error of ϵ for a sketch translates to an error of 3ϵ for our estimate of D_n. Hence, we need to pick ϵ such that $3\epsilon\sqrt{n} \leq \delta$. Choosing $\epsilon = \delta/(3\sqrt{n})$ suffices, so we substitute this into the memory requirement of the sketch to get that $O(\frac{\sqrt{n}}{\delta}\log(\delta\sqrt{n}))$ space is required, which turns out to be $O(\sqrt{n}\log n)$ for constant δ (e.g., 0.03 in the above example). Thus, the overall space needed by the one-sample test is $O(\sqrt{n}\log n)$. For example, a terabyte of data would get compressed down to the order of megabytes.

The result for the two-sample test is similar. Recall that the two-sample test allows one to reject the hypothesis at the α significance level when $\sqrt{\frac{nm}{n+m}}D > K_\alpha$, where n and m are the lengths of the two streams. Let us say, without loss of generality, that $n \geq m$. We then have that $\sqrt{\frac{nm}{n+m}} \leq \sqrt{\frac{n^2}{n}} = \sqrt{n}$ and an analysis similar to that above shows that $O(\sqrt{n}\log n)$ space is needed by the quantile sketches to reliably perform this test. Unfortunately, this space requirement is needed for both the n and the $m \leq n$ length stream, which means that in the case that $m \ll n$ the space requirement of the smaller stream may become unreasonable. Hence, the two-sample test is only feasible when n and m are not too far apart in magnitude. This is demonstrated in the experimental section.

We next compare our $O(\sqrt{n}\log n)$ result with that for random sampling. There is a folklore result that random sampling needs $\theta(1/\epsilon^2)$ samples to ϵ-approximate the quantiles [2]. Substituting the $\epsilon \leq \delta/(3\sqrt{n})$ requirement from above into this formula gives us that this amounts to $\Omega(n)$ samples. Hence, we expect that sampling should need many more samples than using the Greenwald-Khanna sketch to attain the same level of accuracy.

VII. Experimental Evaluation

We experimentally evaluated our algorithms on both real and synthetic datasets to test their accuracy. We measured the absolute error in the KS-statistic for both the one-sample and the two-sample KS tests using different quantile ϵ-sketches to evaluate which ones are most effective in practice. All our code was written in Java. The experiments were all run on a 3.0 GHz Intel Core i3 Mac with 4GB memory running Mac OS X 10.6.8.

We used synthetic data drawn from uniform, Gaussian, and power-law (Pareto) distributions, as these commonly appear in real data. We averaged the results over 10 independently generated datasets in each case. Unless otherwise stated, our experiments used ten thousand points ($n = 10000$) and used 1% of the space needed to store the entire stream.

For our experiments on real data we used the following three traces:

- Astronomy data: We collected magnitude data for stars and galaxies from the Sloan Digital Sky Survey[1]. We queried for data on all objects within 60 arcminutes of the location (180, 0) and got 35697 stars and 62091 galaxies. The KS-statistic was computed for the magnitudinal distribution of the stars versus the galaxies in the green part of the spectrum.
- Light data: We also used irradiance measurements (in units W/cm^2) taken from photometric sensors as part of Columbia University's EnHANTs (Energy Harvesting Active Networked Tags) project[2]. We compared the irradiance levels between Traces A and B of this dataset.
- Inter-arrival time data: For our third dataset we used inter-arrival times collected from wireless networks in the Portland area[3]. We compared the inter-arrival times (measured in nanoseconds) of five minutes of data collected at the Portland State University CS department (260325 values) against those collected at Pioneer Square (517631 values).

We tested our algorithms using the following quantile ϵ-sketches:

- The Greenwald-Khanna [10] (GK) algorithm has one of the best space usage guarantees of $O(\frac{1}{\epsilon}\log(\epsilon n))$ and has been demonstrated to be very competitive in practice [26].

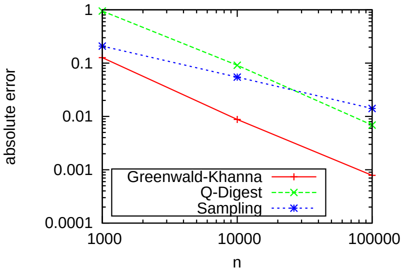


Figure 3. Varying one-sample data size (n), for data drawn from $\mathrm{N}(0,1)$ compared with $\mathrm{N}(0.1,1)$, using 1% memory

- The q-digest [24] (QD) sketch uses space $O(\frac{1}{\epsilon}\log U)$ memory, where U is the size of the universe. In the case of real-valued data, we quantized the data into bins of size 1e-5 and executed the algorithm on this quantized stream. Since the KS-statistic is a function of the relative size of data, rather than the absolute values, we did not expect this to affect the result.

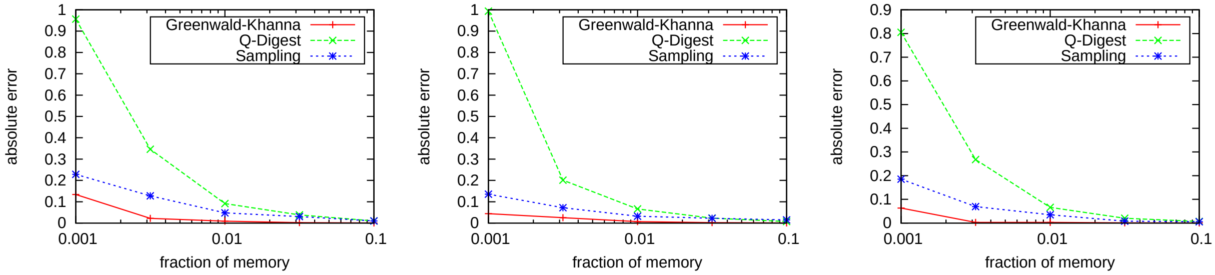
We also compared the above algorithms with the naive methodology of sampling the data. This was included to give a comparison with the obvious solution for this problem. Note that there are no other existing algorithms for this problem. In every experiment, all the algorithms were allocated identical amounts of memory.

A. One Sample

In our experiments, we focused on computing the KS-statistic between distributions that are so close that we require accurate estimates to be able to distinguish them with high confidence. The case in which distributions are far apart is relatively easy to handle because more coarse-grained estimates suffice to distinguish them. The summarization of the data causes an absolute error that increases with the degree to which the data are compressed. This is illustrated using normal, uniform, and Pareto-distributed data in Figure 2. We found that comparing data from $\mathrm{N}(0.1,1)$ with the distribution $\mathrm{N}(0,1)$ gave a KS-statistic close to the threshold for distinguishing distributions using $n = 10000$ points, where $\mathrm{N}(\mu,\sigma^2)$ represents the Gaussian distribution with mean μ and variance σ^2. The uniform distributions $\mathrm{U}(0,1)$ and $\mathrm{U}(0.1,1)$, where $\mathrm{U}(a,b)$ is the uniform distribution on the range $[a,b]$, and the Pareto distributions $\mathrm{P}(1,1)$ and $\mathrm{P}(1.1,1)$, where $\mathrm{P}(x_m,\alpha)$ is the Pareto distribution with scale x_m and shape α, were picked for similar reasons. In all these cases, the Greenwald-Khanna sketch out-performed the sampling algorithm which in turn out-performed the q-digest sketch. The Greenwald-Khanna sketch also gave low enough error at 1% memory to be able to distinguish the distributions. For the rest of our experiments we focus on the normal distribution as it appears to have the highest error.

In Figure 3 we fixed the algorithms to all use 1% of the memory it would take to store all the data and compared

[1] http://cas.sdss.org/astro/en/tools/search/radial.asp
[2] http://crawdad.org/columbia/enhants/
[3] http://crawdad.org/pdx/vwave/



(a) $\mathrm{N}(0, 1)$ vs. $\mathrm{N}(0.1, 1)$

(b) $\mathrm{U}(0, 1)$ vs. $\mathrm{U}(0.1, 1)$

(c) $\mathrm{P}(1, 1)$ vs. $\mathrm{P}(1.1, 1)$

Figure 2. Varying memory ($n = 10000$) for one-sample data drawn from various distributions

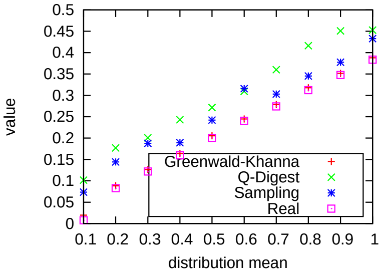


Figure 4. Varying mean of one-sample distribution ($\mathrm{N}(x, 1)$) compared with $\mathrm{N}(0, 1)$, using $n = 10000$ and 1% memory

how the algorithms performed when the data size increased. Note that both the axes have logarithmic scales. We see that as the data size grows, the absolute error drops rapidly in all cases. For smaller data sizes the sampling algorithm out-performs the q-digest sketch, but the Greenwald-Khanna sketch is clearly the best at all sizes. This drop in error is to be expected since the increase in data size corresponds with an increase in the number of samples stored since the memory size is fixed to 1%.

Next, we studied how the accuracy of the estimate changed as the actual KS-statistic between the data and comparative distribution varied. In Figure 4 we varied the mean of the normally-distributed data and compared with the distribution $\mathrm{N}(0, 1)$, comparing the estimate of each sketch to the exact value. Once again, the Greenwald-Khanna sketch is the clear winner, almost indistinguishable from the real value in the figure. In contrast, for this distribution and these parameter values, the q-digest sketch and the sampling solution were equally bad, tending to over-estimate the actual distance.

We omit experiments for the real data in the one-sample case as there are no known analytical distributions for these datasets.

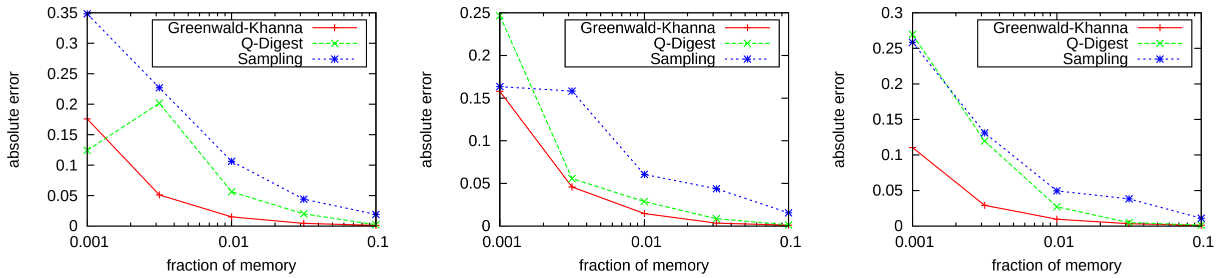
B. Two Sample

Similarly to the one-sample case, we compared our two-sample algorithm using both sketches against the sampling technique on normal, uniform, and Pareto-distributed data. This can be seen in Figure 5. In all these cases, the q-digest sketch does slightly better than sampling, but the Greenwald-Khanna sketch gives the best performance at almost all levels of summarization. Once again, it can be seen that using a sketch with as little as 1% of the original data can shrink the error in the KS-statistic small enough to reliably apply this test.

Figure 6 shows a sharp drop in the error as the data size increases, just as in the one-sample case. The reason for this drop is the same as in the one-sample case. We were also curious about what the effect of changing one of the data sizes while keeping the other one fixed would be. Since the KS test can be applied to two samples of differing sizes, we were interested to see what varying the relative sample sizes would do to the error. Figure 7 shows this result when one dataset was fixed to $n = 10000$ points while the other's size (m) varied. We can see in this figure that there is a drop in error, but that it seems to level out after m becomes larger than n. This seems to indicate that the accuracy of the test is dependent on the size of the smaller of the two samples, as predicted in the previous section.

Next, we studied the accuracy of the algorithms as the actual distance between the datasets was varied. Figure 8 shows how each of the algorithms performs by plotting the estimated value against the real value. For reference, the $y = x$ line that indicates the ideal answer is given as well. It is again clear from this figure that the Greenwald-Khanna sketch gives the best performance.

Finally, we tested the two-sample algorithm on our real datasets. The results are shown in Figure 9. In the case of the astronomy dataset, the Greenwald-Khanna sketch peformed excellently, needing less than 0.1% memory to give a very accurate estimate of the KS-statistic. In contrast, for the light dataset, the Greenwald-Khanna sketch performed poorest at very small fractions of memory, but by the 1% mark had started to best the other algorithms. For the inter-arrival time data, the Q-Digest sketch performs very poorly at high compression, but then soon gets better than sampling; as always, the Greenwald-Khanna version has the best performance.

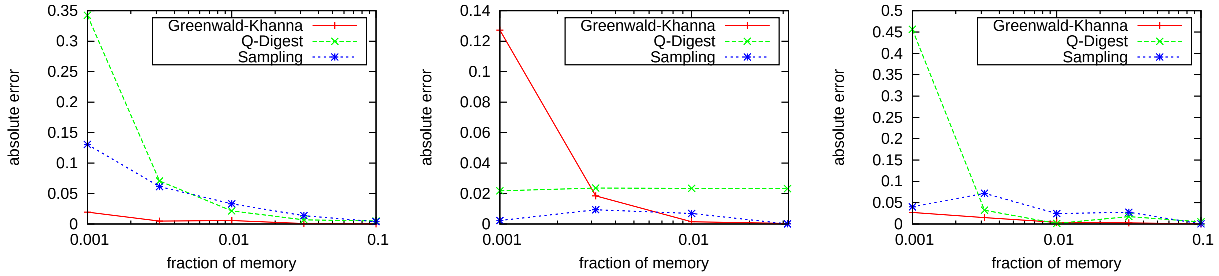


(a) $\mathrm{N}(0, 1)$ vs. $\mathrm{N}(0.1, 1)$

(b) $\mathrm{U}(0, 1)$ vs. $\mathrm{U}(0.1, 1)$

(c) $\mathrm{P}(1, 1)$ vs. $\mathrm{P}(1.1, 1)$

Figure 5. Varying memory ($n = m = 10000$) for two-sample data drawn from various distributions



(a) Astronomy data

(b) Light data

(c) Inter-arrival data

Figure 9. Varying memory for real two-sample data

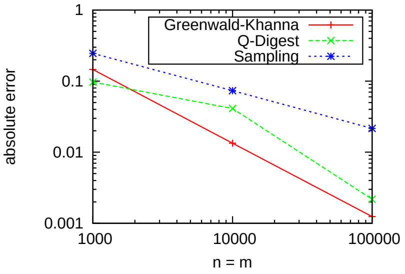


Figure 6. Varying two-sample data size ($n = m$), for data drawn from $\mathrm{N}(0, 1)$ and $\mathrm{N}(0.1, 1)$, using 1% memory

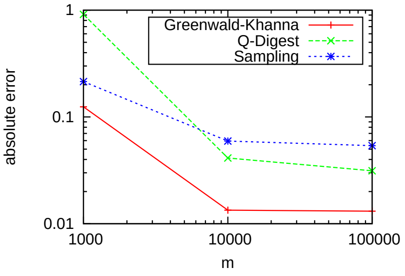


Figure 7. Varying data size of one sample (m) keeping other sample fixed ($n = 10000$) for data drawn from $\mathrm{N}(0, 1)$ and $\mathrm{N}(0.1, 1)$, using 1% memory

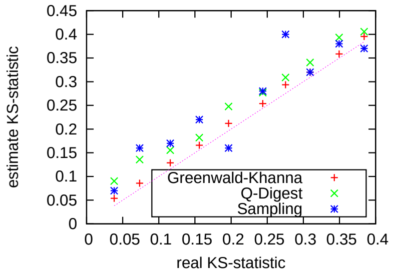


Figure 8. Scatter plot of estimated vs. real values of KS-statistic ($n = m = 10000$) between two-sample data drawn from $\mathrm{N}(0, 1)$ and various distributions of the form $\mathrm{N}(x, 1)$, using 1% memory. The $y = x$ line is also shown for reference.

VIII. Conclusions

In this paper, we considered the problem of performing the Kolmogorov-Smirnov test on streaming data. We gave algorithms for both the one-sample and the two-sample versions of the test, along with guarantees of their performance. Our algorithms make use of the techniques for computing quantiles on a stream and hence may have improved results when further progress is made on this problem. Moreover, we show via experiments on real and synthetic data that the proposed algorithms are capable of performing the test with a two order magnitude reduction in the size of the data. Our experiments also showed that the Greenwald-Khanna sketch is best suited for our algorithm, and that it is considerably superior to other simple techniques such as sampling.

There are several open problems that still remain. The algorithms proposed in this paper need $O(\sqrt{n}\log n)$ samples, and it remains open whether this can be further reduced. While the algorithm proposed in this paper makes use of quantile sketches, and hence has the same space usage as them, it is unclear whether a testing algorithm exists that uses asymptotically less space than any quantile sketch. It would be interesting to find such an algorithm, or alternatively to prove that any quantile sketch must use as much space as any testing algorithm. From the experimental side, it would be interesting to design other quantile algorithms that are targeted towards improving the accuracy of the result of the KS test algorithm in this paper.

There are also many other statistical tests, both parametric and non-parametric, that do not have known streaming algorithms. Identifying which ones have sublinear space algorithms and developing these algorithms is another avenue for future work.

Acknowledgement: We thank the anonymous reviewers for their helpful comments on the paper. We would also like to thank Erin Lall for her help in proofreading an earlier version of this paper and Ryan Johnson for his help with the astronomy data set. This project was funded in part by NSF award CNS 1217758.

REFERENCES

[1] T. Batu, L. Fortnow, R. Rubinfeld, W. D. Smith, and P. White. Testing that distributions are close. In *FOCS*, pages 259–269, 2000.

[2] C. Buragohain and S. Suri. Quantiles on streams. In *Encyclopedia of Database Systems*, pages 2235–2240. 2009.

[3] A. Chakrabarti, T. S. Jayram, and M. Pătraşcu. Tight lower bounds for selection in randomly ordered streams. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '08, pages 720–729, Philadelphia, PA, USA, 2008.

[4] K. C. Claffy, G. C. Polyzos, and H.-W. Braun. Application of sampling methodologies to network traffic characterization. *SIGCOMM Comput. Commun. Rev.*, 23(4):194–203, Oct. 1993.

[5] E. Cohen, G. Cormode, and N. G. Duffield. Don't let the negatives bring you down: sampling from streams of signed updates. In *SIGMETRICS*, pages 343–354, 2012.

[6] G. Cormode and S. Muthukrishnan. An improved data stream summary: The count-min sketch and its applications. *J. Algorithms*, 55:29–38, 2004.

[7] N. Duffield. Fair sampling across network flow measurements. *SIGMETRICS Perform. Eval. Rev.*, 40(1):367–378, June 2012.

[8] A. C. Gilbert, Y. Kotidis, S. Muthukrishnan, and M. J. Strauss. How to summarize the universe: dynamic maintenance of quantiles. In *Proceedings of the 28th international conference on Very Large Data Bases*, VLDB '02, pages 454–465. VLDB Endowment, 2002.

[9] A. Glazer, M. Lindenbaum, and S. Markovitch. Learning high-density regions for a generalized kolmogorov-smirnov test in high-dimensional data. In *NIPS*, pages 737–745, 2012.

[10] M. Greenwald and S. Khanna. Space-efficient online computation of quantile summaries. In *SIGMOD*, pages 58–66, 2001.

[11] S. Guha and A. McGregor. Stream order and order statistics: Quantile estimation in random-order streams. *SIAM J. Comput.*, 38(5):2044–2059, 2009.

[12] S. Guha, A. Mcgregor, and S. Venkatasubramanian. Streaming and sublinear approximation of entropy and information distances. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 733–742, 2006.

[13] T. Karagiannis, M. Molle, M. Faloutsos, and A. Broido. A nonstationary poisson view of internet traffic. In *Proceedings of IEEE INFOCOM*, 2004.

[14] D. Kifer, S. Ben-David, and J. Gehrke. Detecting change in data streams. In *Proceedings of the Thirtieth international conference on Very large data bases - Volume 30*, VLDB '04, pages 180–191. VLDB Endowment, 2004.

[15] D. E. Knuth. *The Art of Computer Programming, Volume II: Seminumerical Algorithms, 2nd Edition*. Addison-Wesley, 1981.

[16] A. Kumar, M. Sung, J. Xu, and E. W. Zegura. A data streaming algorithm for estimating subpopulation flow size distribution. In *Proc. of ACM SIGMETRICS*, June 2005.

[17] A. Lall, V. Sekar, M. Ogihara, J. Xu, and H. Zhang. Data streaming algorithms for estimating entropy of network traffic. *SIGMETRICS Perform. Eval. Rev.*, 34(1):145–156, June 2006.

[18] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. Tinydb: an acquisitional query processing system for sensor networks. *ACM Trans. Database Syst.*, 30(1):122–173, Mar. 2005.

[19] G. S. Manku, S. Rajagopalan, and B. G. Lindsay. Random sampling techniques for space efficient online computation of order statistics of large datasets. In *SIGMOD*, pages 251–262, 1999.

[20] A. McGregor, A. Pavan, S. Tirthapura, and D. Woodruff. Space-efficient estimation of statistics over sub-sampled streams. In *Proceedings of the 31st Symposium on Principles of Database Systems*, PODS '12, pages 273–282, New York, NY, USA, 2012. ACM.

[21] J. I. Munro and M. S. Paterson. Selection and sorting with limited storage. In *Proceedings of the 19th Annual Symposium on Foundations of Computer Science*, SFCS '78, pages 253–258, Washington, DC, USA, 1978. IEEE Computer Society.

[22] NASA. Chandra X-ray Observatory Quick Facts. http://www.nasa.gov/centers/marshall/news/background/facts/cxoquick.html.

[23] S. K. A. Organization. The Square Kilometre Array. http://www.skatelescope.org/the-technology/signal-processing/.

[24] N. Shrivastava, C. Buragohain, D. Agrawal, and S. Suri. Medians and beyond: new aggregation techniques for sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, SenSys '04, pages 239–249, New York, NY, USA, 2004. ACM.

[25] J. V. Wall and C. R. Jenkins. *Practical Statistics for Astronomers*. Cambridge University Press, 2003.

[26] L. Wang, G. Luo, K. Yi, and G. Cormode. Quantiles over data streams: An experimental study. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, SIGMOD '13, pages 737–748, New York, NY, USA, 2013. ACM.

[27] Q. Zhang and W. Wang. A fast algorithm for approximate quantiles in high speed data streams. In *Proceedings of the 19th International Conference on Scientific and Statistical Database Management*, SSDBM '07, pages 29–, Washington, DC, USA, 2007. IEEE Computer Society.