Accessible Streaming Algorithms for the Chi-Square Test

Emily Farrow, Junbo Li, Farhan Zaki, Ashwin Lall
Denison University

ABSTRACT

We present space-efficient algorithms for performing Pearson's chi-square goodness-of-fit test in a streaming setting. Since the chi-square test is one of the most well known and commonly used tests in statistics, it is surprising that there has been no prior work on designing streaming algorithms for it. The test is not based on a specific distribution assumption and has one-sample and two-sample variants. Given a stream of data, the one-sample variant tests if the stream is drawn from a fixed distribution. The two-sample variant tests if two data streams are drawn from the same or similar distributions. One major advantage of using statistical tests over other quantities commonly measured by streaming algorithms is that these tests do not require parameter tuning and have results that can be easily interpreted by data analysts. The problem that we solve in this paper is how to compute the chi-square test on streams with minimal parameter configuration and assumptions. We give rigorous proofs showing that it is possible to compute the chi-square statistic with high fidelity and an almost quadratic reduction in memory in the continuous case, but the categorical case only admits heuristic solutions. We validate the performance and accuracy of our algorithms through extensive testing on both real and synthetic data sets.

CCS CONCEPTS

•**Information systems → Data stream mining;**

KEYWORDS

data streams, chi-square, one sample, two sample, categorical

ACM Reference format:
Emily Farrow, Junbo Li, Farhan Zaki, Ashwin Lall. 2016. Accessible Streaming Algorithms for the Chi-Square Test. In *Proceedings of ACM Conference, Washington, DC, USA, July 2017 (Conference'17),* 12 pages.
DOI: 10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Over the last few decades, modern computer and networking systems have created the ability to continuously generate enormous volumes of data at a very high speed. This data is being generated by sensors, mobile devices, routers, scientific devices, and every large system at a global level from the functioning of almost every major business, government operation, scientific enterprise, and social media exchange. Big Data has become so large that it is unfeasible to translate, store, and process. This exponential growth of data is unavoidable in the modern age. However, efficient technologies to store the generated information without massive down-sampling have not been developed. The inability of fast memory capacity to keep up with the size of this data has become one of the most pressing challenges of Big Data.

The streaming model of computation was introduced to solve the abovementioned problem. In the streaming model, the input is presented as a stream of updates and the challenge is to answer some question about the stream, as it goes by in a single pass, without storing all of it. There has been considerable research done in this model (see [1, 23] for surveys on the topic). There has been work done to design sublinear memory data structures, known as *sketches*, for fundamental operators such as frequency moments [2] and quantiles [8], as well as more complex queries such as entropy [4], information divergences [11], and subpopulation distributions [14].

A significant barrier for practitioners to adopt many of these techniques is that they do not have the expertise or time to learn how to apply them or how to configure their parameters. On the other hand, there are plenty of statistical hypothesis tests that can be applied with minimal technical knowledge (e.g., interpreting a p-value), that have few parameter knobs to configure and that are already extensively being used by engineers and data analysts. It is for this reason that we propose accessible streaming. We define *accessible streaming* to be streaming algorithms that are easy for a practitioner to use with easy-to-understand guarantees and few to no parameter knobs to be tuned so that the algorithm can be used "out-of-the-box." The goal of this paper is to provide accessible streaming algorithms for Pearson's chi-square goodness-of-fit test.

There are many statistical tests for determining the validity of a hypothesis, but few are as well-known or widely used as Pearson's chi-square goodness-of-fit test (henceforth referred to as the chi-square test in this paper). This test has the advantages of being non-parametric, i.e., it is not tied to any specific distribution, such as the Gaussian, and can be used for continuous as well as categorical data. It is therefore surprising that there have been no sub-linear memory algorithms for this test proposed in the literature. In this paper, we will show how to compute the chi-square test to concisely yet accurately check if a particular stream of continuous data comes from a fixed known distribution, if two streams of continuous data come from the same source, and if two categorical data streams have a similar underlying distribution.

To perform the continuous version of the chi-square test, we need to partition the data into bins, where the expected frequency of each bin is compared to the observed frequency to calculate the test statistic. There is a commonly used rule of thumb that each bin has an expected value of at least five (see, e.g., Knuth [13, p. 45]). In our setting, the distribution of the stream is unknown beforehand, therefore picking bins without any foreknowledge of the distribution can lead to one or more bins having fewer than five

Conference'17, Washington, DC, USA

DOI: 10.1145/nnnnnnn.nnnnnnn

samples—violating the above requirement. An alternate approach is to sample an initial section of the stream and choose bins based upon this sample. The downside to this is that we need to make an identical and independently distributed (i.i.d.) assumption about the stream, something that is not always the case. For example, network traffic data is notorious for being very bursty, exhibiting non-stationary distributions. The main challenge overcome by the algorithms in our paper is that we are able to compute the chi-square statistic in each of these cases without any prior knowledge about the distribution being measured and making few assumptions about the stream.

1.1 Contributions

The contributions of this paper are as follows:

- We propose a streaming algorithm for the one-sample chi-square test to check if a stream of continuous data comes from a fixed known distribution. The space complexity is $O(K^2 \log(N)\sqrt{N})$ for a stream of length at most N and number of bins at most K. While we need to know an upper bound on the size of the stream and the number of buckets, there is no need to know the distribution of the stream, the distribution being tested, or the number of bins in advance.
- We propose a streaming algorithm for the two-sample chi-square test to check whether two streams of continuous data come from the same source. The space complexity is $O(K^2 \log(N)\sqrt{N})$. Similar to the one-sample variant, we make few assumptions about the stream.
- We show that it is impossible to similarly summarize the categorical chi-square test by establishing lower bounds for this problem. We give a heuristic algorithm for computing the categorical chi-square test on two streams to show that reasonable space savings are still possible in practice.
- We conduct extensive experiments on both synthetic and empirical data to verify the accuracy of the proposed algorithms and to show that they perform well in practice.

Organization: In Section 2, we discuss the most relevant related work. Section 3 contains the formal definition of this problem. Streaming algorithms for the one-sample, two-sample, and categorical chi-square tests are given in Sections 4, 5, and 6 respectively, along with their analysis for accuracy. In Section 7, we evaluate the algorithms on both synthetic and empirical data. Section 8 contains the conclusion of this paper and possible future work.

2 RELATED WORK

Our work is the first to propose algorithms for the chi-square test in the streaming model of computation. In the streaming model, the input data is presented as a sequence of updates [23]. The advantage of this model is that it is possible to summarize the data in sketches that use memory sub-linear in the input size. Streaming algorithms have been designed for operators such as frequency moments [2], quantiles [8], and counting distinct elements [6], more complex queries such as entropy [16], information divergences [11], and data mining applications such as clustering [12], outlier detection [3], and wavelet decomposition [7].

Our work builds on work done on quantile sketches. A quantile sketch is a summary that can be queried for the quantiles of a large stream of data. The idea of the quantile sketch was initially formalized in a paper by Munro and Paterson [22] in which they proposed an algorithm that uses $\Omega(n^{1/p})$ memory, where n is the length of the stream and p is the number of sequential scans of the input. Manku et al. [19, 20] proposed a single pass deterministic algorithm that uses $O(\frac{1}{\epsilon}\log^2(\epsilon n))$ memory, where n is the length of the stream and the quantile returned has at most ϵ error, and randomized algorithms with slightly smaller space bounds. Subsequently, their results were improved by Greenwald and Khanna [8] who gave the GK algorithm that only requires $O(\frac{1}{\epsilon}\log(\epsilon n))$ memory. Another quantile sketch, Q-digest, uses $O(\frac{1}{\epsilon}\log U)$ memory, where U is the size of the input domain [25]. Wang et al. [26] and Luo et al. [18] performed extensive experimental comparisons of the performance of these quantile sketch algorithms. An equivalent alternative to using quantile sketches is to use equi-depth histograms [9, 21].

In [17], stable random projections are used to compute the chi-square similarity (as opposed to the chi-square test) on streaming data. This work doesn't give guarantees for computing the chi-square statistic in a stream. Another related work [15] proposes space-efficient streaming algorithms for Kolmogorov-Smirnov test. The goal of this test is to detect differences between large datasets, similar to the work presented here. A major advantage of the chi-square test is that it is used much more extensively in practice. It also has a categorical variant, for which we also provide an algorithm in this paper.

We show that it is impossible to design a sublinear solution for the general case of the categorical chi-square test, even when we allow approximation and randomization, using techniques from [10]. As a result, we focus on a sampling-based solution for this test. Specifically, we make use of a technique known as coordinated sampling [5] to maximize the overlap between the pair of streams.

3 PROBLEM DEFINITION

This section reviews the fundamental hypothesis setup and the procedure of conducting the chi-square test. We work under the model that a stream is processed over a finite period (e.g., one hour or one day at a location), as is done for example with anomaly detection [27], into a sketch with the constraints that storing the whole stream is infeasible and that processing time per item is small. Our algorithms apply this test to detect differences between a sample (sketch) and a fixed known distribution in the one-sample continuous case, and determine if two samples come from the same source in the two-sample continuous and categorical cases.

3.1 One-Sample Continuous

The one-sample case checks if a set of data follows a specified distribution. Hence, the null and alternative one-sample hypotheses are:

H_0: The sample is from the fixed known distribution.
H_a: The sample is not from the fixed known distribution.

For this case, the data must be binned into non-overlapping ranges. The frequency of the data observed in each bin is then compared to the expected frequency of each bin. We will use N to denote the size of the stream and K to denote the number of bins. For each $1 \le i \le K$, if we let the observed frequency of bin i be O_i and the expected frequency be E_i, then the chi-square statistic is

defined as

$$\chi^2 = \sum_{i=1}^{K} \frac{(O_i - E_i)^2}{E_i}.$$

An important characteristic of this test is that it is sensitive to the choice of bins, i.e, changing K or the bin ranges gives a different statistic. Though there is no standard method to choose the bins or even the number of bins, each bin must contain at least five data points. One common convention that we follow in this paper is to have equi-probable bins, i.e., each bin has approximately the same expected frequency, N/K.

At the decision making stage, in order to accept or reject the null hypothesis, the statistic is compared to critical values from the chi-square distribution. The critical values are read from the table based on the significance level α, where $0 < \alpha < 1$, and the degrees of freedom. For the chi-square test, the degrees of freedom is $K - c$, where c is the number of estimated parameters (such as the location, scale and shape parameters) plus one. For example, for a two-parameter normal distribution (mean and standard deviation), $c = 3$. Therefore, the null hypothesis is rejected if $\chi^2 > \chi^2_{1-\alpha,K-c}$, where $\chi^2_{1-\alpha,K-c}$ is the appropriate value from the chi-square distribution.

3.2 Two-Sample Continuous

We define the size of the first stream as N, the second as M, and K as the number of bins. The two streams must use the same bins, and the frequency in the i^{th} bin is defined as S_i for the first stream and R_i for the second stream. Since there are two streams, the expected frequency for the first stream in the i^{th} bin is $\frac{(S_i+R_i)}{N+M}N$, and the expected frequency for the second stream in the i^{th} bin is $\frac{(S_i+R_i)}{N+M}M$. Therefore, based on the contingency table definition of two-sample (this generalizes to the k-sample case, but we limit ourselves to two for the purpose of this paper), the chi-square statistic χ^2 is calculated and simplified [24] to:

$$\chi^2 = \sum_{i=1}^{K} \frac{(S_i\sqrt{\frac{M}{N}} - R_i\sqrt{\frac{N}{M}})^2}{S_i + R_i}.$$

At the decision making stage, the rules are fairly similar as the rules for the one-sample version, which is that the null hypothesis is rejected if $\chi^2 > \chi^2_{1-\alpha,K-c}$. The only difference is that when computing the degrees of freedom $K - c$, we use $c = 1$ if the two samples have the same size ($N = M$) and $c = 0$ otherwise.

The two sample variant compares two streams to determine if they come from the same distribution. The null and alternative one-sample hypotheses are:

H_0: Two samples have the same underlying distribution.
H_a: Two samples have different underlying distributions.

3.3 Categorical

The categorical case checks if two streams of discrete data come from the same distribution. We do not study the one-sample categorical case as it would be unfeasible to specify expected counts for a large number of categories. The hypotheses are the same as for the continuous two-sample case.

We once again define the length of the two streams as N and M. We now define the number of categories to be K as each category corresponds to its own bin. Each stream consists of a sequence of category labels, where the frequency of a category is the number of times its label appears in the stream. If we let the frequencies of the i^{th} category be R_i and S_i for the first and second streams, respectively, then the chi-square statistic is identical to that for the two-sample continuous case:

$$\chi^2 = \sum_{i=1}^{K} \frac{(S_i\sqrt{\frac{M}{N}} - R_i\sqrt{\frac{N}{M}})^2}{S_i + R_i}.$$

The rejection conditions are identical as well: $\chi^2 > \chi^2_{1-\alpha,K-c}$, with c once again being the indicator of whether the streams have the same length.

symbol	description
N	length of stream (one- and two-sample)
M	length of second stream (two-sample)
K	number of bins
O_i	observed frequency of bin i (one- and two-sample)
E_i	expected frequency of bin i (one- and two-sample)
S_i	count of stream1 bin i (two-sample, categorical)
R_i	count of stream2 bin i (two-sample, categorical)
F	CDF of comparison distribution (one-sample)
Q	quantile sketch for the stream (one-sample)
Q_1	quantile sketch for stream1 (two-sample)
Q_2	quantile sketch for stream2 (two-sample)
$\chi^2_{1-\alpha,d}$	critical value for $p = \alpha$ with d degrees of freedom
H_0	the null hypothesis
H_a	the alternative hypothesis

Table 1: List of notation

3.4 Accessible Streaming

Notice that in all the definitions listed above, the user of the streaming algorithm is not expected to know much about the stream prior to deploying the algorithm and does not have to configure any parameters to use the algorithm. To obtain the guarantees shown in this paper, the user should know an upper bound on the length of the stream and the number of buckets to compute the size of the sketch needed, but absent this she can simply provision the largest sketch feasible and still be able to perform the test. This is explained in more detail later. Also note that the desired significance level (α) and number of bins (K) can be specified after the stream has been processed. There are fairly standard choices available to the user, such as $\alpha = 0.05$ and a value of K between 10 and 100.

For ease of reading, the notation for this paper is summarized in Table 1.

4 ONE-SAMPLE TEST

This section describes the algorithm for the one-sample continuous variant of the chi-square test, in which the observed input can be compared against an arbitrary distribution function specified after the stream has been collected. The algorithm achieves this by creating a compact summary of the data, also known as a sketch, which can be used to compare with an arbitrary distribution afterwards.

Algorithm 1 OneSample(Q, N, K, F, α,c)

Input: Quantile ϵ-sketch Q of stream size N; K number of bins; cumulative distribution function F; significance level α; the number of estimated parameters + 1, c

Output: Whether the test rejects the null hypothesis

1: $\hat{\chi^2} = 0$
2: **for** $i = 1$ to K **do**
3: $E_i = \frac{N}{K}$
4: Let $l = F^{-1}(\frac{i-1}{K})$.
5: Let $u = F^{-1}(\frac{i}{K})$.
6: Let $\hat{i_l}$ be the approximate fraction of the stream less than l, as computed in Theorem 4.1.
7: Let $\hat{i_u}$ be the approximate fraction of the stream less than u, as computed in Theorem 4.1.
8: $\hat{O_i} = N(\hat{i_u} - \hat{i_l})$
9: $\hat{\lambda_i} = |\hat{O_i} - E_i|$
10: **if** $\hat{\lambda_i} > 2\sqrt{N}$ **then**
11: return *true*
12: $\hat{\chi^2} = \hat{\chi^2} + \frac{(\hat{\lambda_i})^2}{E_i}$
13: Let $\chi^2_{1-\alpha,K-c}$ be the critical value at significance level α and degree of freedom $K - c$.
14: return $\hat{\chi^2} > \chi^2_{1-\alpha,K-c}$

If we know the comparison distribution or the bin ranges beforehand, then computing the chi-square statistic is fairly trivial to accomplish by keeping a count for each bin and incrementing the count for the appropriate bin for each insertion into the stream. Our algorithm does not need to know the comparison distribution, the bin ranges, or even the number of bins beforehand. It makes a sketch of the stream and computes the bin ranges after the stream has been processed. This is important because choosing bin ranges beforehand can be catastrophic for unknown distributions since it is impossible to know where the input data may fall and it may be that several of the bins will have expected count less than five. In contrast, our algorithm guarantees that all bins have roughly the same counts (the ideal case for the chi-square test) so that the expected count for each bin is very large.

As for the sketch, it follows that it must guarantee some error bounds for any distribution. We use an ϵ-quantile sketch to store the quantiles of the data in a streaming fashion. An ϵ-quantile sketch is defined as a data structure with input stream $X_1, \ldots, X_N (X_1 \leq X_2 \leq \ldots \leq X_N)$ in arbitrary order that can be queried to return for a fixed $\epsilon > 0$ and any rank $1 \leq r \leq N$, a value X_i such that i is in the range $[r - \epsilon N, r + \epsilon N]$ [15].

Our algorithm will make use of the ability to compute inverse quantiles from these quantile sketches. Prior work shows how to extract from an ϵ-quantile sketch the inverse quantile of some value, i.e., the approximate fraction of the stream that a given value is bigger than [8, 15]. We use the following result from [15]:

Theorem 4.1 ([15]). *For any value $x \in \mathbb{R}$, it is possible to compute the fraction of the stream that has value less than x to within 3ϵ error using an ϵ-quantile sketch.*

4.1 One-Sample Algorithm

We use this result to compute the chi-square statistic in Algorithm 1 with the following guarantee:

Theorem 4.2. *Algorithm 1 computes an estimate of the chi-square statistic with at most ± 0.0812 additive error.*

Proof. We want to approximate $\chi^2 = \sum_{i=1}^{K} \frac{(O_i - E_i)^2}{E_i}$. Fix some $1 \leq i \leq K$. For the known cumulative distribution function that we are comparing against, F, let $l = F^{-1}(\frac{i-1}{K})$ and $u = F^{-1}(\frac{i}{K})$, so that $[l, u)$ corresponds to the ith equi-probable interval from the distribution. Let i_l and i_u be the fractions of the stream less than u and l, respectively, computed from the ϵ-sketch using the result from Theorem 4.1. We use these values to get an approximation to O_i, denoted $\hat{O_i} = N(i_u - i_l)$. By Theorem 4.1, we know that estimating i_l and i_u gives at most $\pm 3\epsilon$ error each, so the total error of $\hat{O_i}$ is $\pm 6\epsilon N$.

In order to find the total error of χ^2, we must calculate the error at every i. It follows that estimating O_i with $\hat{O_i}$ leads to an estimate $\hat{\chi}^2 = \sum_{i=1}^{K} \frac{((O_i - E_i) \pm 6\epsilon N)^2}{E_i} = \sum_{i=1}^{K} (\frac{(O_i - E_i)^2}{E_i} \pm \frac{12(O_i - E_i)\epsilon N}{E_i} + \frac{36\epsilon^2 N^2}{E_i})$. Each bin in the expected distribution F has the same frequency, $E_i = \frac{N}{K}$, as defined in line 2 of Algorithm 1. For ease of presentation and completeness, assume that $\epsilon = \frac{1}{300\sqrt{N}K^2}$. Using these definitions,

$$\hat{\chi}^2 = \sum_{i=1}^{K} \left(\frac{(O_i - E_i)^2}{E_i} \pm \frac{0.04(O_i - E_i)}{\sqrt{N}K} + \frac{0.0004}{K^3} \right).$$

Now consider $\lambda_i = |O_i - E_i|$, where $\hat{\lambda_i} = |\hat{O_i} - E_i|$. As we noted earlier the total error of estimating O_i by $\hat{O_i}$ is $\pm 6\epsilon N$. Therefore it follows that the error of estimating λ_i by $\hat{\lambda_i}$ is also $\pm 6\epsilon N$, or $\lambda_i - 6\epsilon N \leq \hat{\lambda_i} \leq \lambda_i + 6\epsilon N$. Since $\epsilon = \frac{1}{300\sqrt{N}K^2}$, it follows that $\lambda_i - \frac{0.02\sqrt{N}}{K^2} \leq \hat{\lambda_i} \leq \lambda_i + \frac{0.02\sqrt{N}}{K^2}$. We can further say that since $\frac{0.02\sqrt{N}}{K^2} < 0.02\sqrt{N}$, $\lambda_i - 0.02\sqrt{N} \leq \hat{\lambda_i} \leq \lambda_i + 0.02\sqrt{N}$. From here there are two cases.

In case 1, suppose there exists at least one i where $1 \leq i \leq K$, such that $\hat{\lambda_i} > 2\sqrt{N}$. By our inequality found above, it follows then that $\lambda_i > 1.98\sqrt{N}$. Then consider $\frac{(O_i - E_i)^2}{E_i} \geq \frac{(1.98\sqrt{N})^2}{N/K} = K(1.98)^2$. This is only one out of K bins, however $K(1.98)^2$ is already significantly greater than the critical value at any significance level of 0.001 or greater, so we should reject the null hypothesis. This is the reason for the test in lines 10-11 of Algorithm 1.

In case 2, $\hat{\lambda_i} \leq 2\sqrt{N}$ for all i such that $1 \leq i \leq K$. From the inequality above, it follows that $\lambda_i \leq 2.02\sqrt{N}$, and therefore

$$\begin{aligned} \hat{\chi}^2 &= \sum_{i=1}^{K} \left(\frac{(O_i - E_i)^2}{E_i} \pm \frac{0.04(2.02\sqrt{N})}{\sqrt{N}K} + \frac{0.0004}{K^3} \right) \\ &= \sum_{i=1}^{K} \frac{(O_i - E_i)^2}{E_i} \pm (0.0808) + \frac{0.0004}{K^2}. \end{aligned}$$

As K must at least be 1, at maximum the error will be $\pm(0.0808) + 0.0004$, and hence we will have at most ± 0.0812 error for the chi-square statistic. □

As this amount of error is so small, it is almost negligible when comparing to the critical value at some given significance level. For example, the critical values for $\alpha = 0.1, 0.05$, and 0.025 with $K = 20$ are 28.412, 31.410, 34.170, respectively, so an error of 0.0812 will have very little bearing on the efficacy of the test. Moreover, we can decrease this error bound further by changing the threshold on Line 10 of Algorithm 1 while slightly increasing the space requirement, but this would unnecessarily complicate the analysis.

4.2 Computational Analysis

The analysis of the streaming part of the algorithm depends on the specific quantile used in the implementation. For example, the Greenwald-Khanna sketch uses at most $O(\frac{\log(\epsilon N)}{\epsilon})$ space and time-per-insertion. Earlier we assumed $\epsilon = \frac{1}{300\sqrt{N}K^2}$, giving us

$$O\left(\log\left(\frac{\sqrt{N}}{300K^2}\right)K^2\sqrt{N}\right) = O(K^2\log(N)\sqrt{N})$$

memory required. For example, we can summarize a stream of size in the terabytes ($N = 10^{12}$) using hundreds of megabytes of memory. Moreover, the fraction of memory needed decreases as the stream size increases—for example, a stream 100 times larger would only need about 10 times more memory for the sketch.

In terms of the running time of Algorithm 1, the majority of time is spent finding $\hat{i_l}$ and $\hat{i_u}$. As described in [15], this involves a binary search which takes $O(\log N)$ time. Algorithm 1 iterates K times and performs $O(\log N)$ computations for the binary searches, giving a running time of $O(K \log N)$, which is very small, even for large streams.

4.3 Accessible Streaming

The above algorithm is very convenient for a data analyst to use since it does not require any parameter configurations while collecting the stream. The distribution being compared against, the desired significance level (α), and number of bins (K) can all be specified after the stream has been processed.

At first blush, it might appear that the memory requirement of $O(K^2\log(N)\sqrt{N})$ indicates that we need to know both the length of the stream (N) and the number of bins (K) to provision memory for the sketch before the stream is processed. However, note that if we have some (perhaps loose) upper bound on N and K, this will suffice, since all the guarantees listed above still apply for all stream lengths $N' < N$ and number of bins $K' < K$. Moreover, the user could also provision the largest sketch that was feasible in her infrastructure and still use this algorithm with no prior knowledge of N and K, with the caveat that there would be no guarantee on the fidelity of estimation of the test statistic beforehand.

5 TWO-SAMPLE TEST

In this section we describe the algorithm for the two-sample continuous variant of the chi-square test, where data from two different datasets are compared to determine if they come from the same distribution. As this requires no prior knowledge of the underlying distribution of either dataset, the two sample case has a significant advantage over the one sample, and is therefore more often used in practice.

Algorithm 2 TwoSample($Q_1, Q_2, N, M, K, \alpha$)

Input: Quantile ϵ-sketches Q_1 and Q_2 of streams of length N and M, respectively ($N \geq M$); K number of bins; significance level α
Output: Whether the test rejects the null hypothesis

1: $\hat{\chi}^2 = 0$
2: **for** $i = 1$ to K **do**
3: $\quad \hat{S_i} = \frac{N}{K}$
4: $\quad$ Let $l = Q_1(\frac{i-1}{K})$.
5: $\quad$ Let $u = Q_1(\frac{i}{K})$.
6: $\quad$ Let $\hat{j_l}$ be the approximate fraction of the second stream less than l, as computed from Q_2.
7: $\quad$ Let $\hat{j_u}$ be the approximate fraction of the second stream less than u, as computed from Q_2.
8: $\quad \hat{R_i} = M(\hat{j_u} - \hat{j_l})$
9: $\quad$ **if** $|\hat{R_i} - M/K| > 20\sqrt{\frac{M}{K}}$ **then**
10: $\quad\quad$ return *true*
11: $\quad \hat{\chi}^2 = \hat{\chi}^2 + \frac{(\hat{S_i}\sqrt{\frac{M}{N}} - \hat{R_i}\sqrt{\frac{N}{M}})^2}{\hat{S_i} + \hat{R_i}}$
12: Let c be 1 if N and M are equal, 0 otherwise.
13: Let $\chi^2_{1-\alpha,K-c}$ be the critical value at significance level α and degrees of freedom $K - c$.
14: return $\hat{\chi}^2 > \chi^2_{1-\alpha,K-c}$

As with the one-sample algorithm, the two-sample algorithm does not require knowledge of the bin ranges or number of bins before running. It creates sketches of the two streams, and without loss of generality, computes the bin ranges from one of the two sketches after both have been processed. Just as the one-sample case, this helps avoid situations where at least one of the bins has a count less than five. We use the ϵ-sketch for inverse quantiles, similarly to the one-sample algorithm, but we also make use of the ability to compute the quantiles themselves.

5.1 Two-Sample Algorithm

Algorithm 2 computes the chi-square statistic with the following guarantee. Note that even though we require truly massive streams for this theorem, this is simply for the worst case bounds. We show experimentally that the algorithm works well even for modestly large streams.

Theorem 5.1. *Algorithm 2 returns an estimate of the chi-square two-sample test with about* $0.048K$ *additive error, assuming that* $N, M \geq 10^{14}, 10 \leq K \leq 1000$.

Proof. We want to compute

$$\chi^2 = \sum_{i=1}^{K} \frac{(S_iK_1 - R_iK_2)^2}{S_i + R_i}$$

where $K_1 = \sqrt{\frac{M}{N}}$ and $K_2 = \sqrt{\frac{N}{M}}$. For any i, consider S_i and R_i. By the definition of the ϵ-quantile sketch, querying for a value at an index i returns the value X_j such that $j \in [i - \epsilon N, i + \epsilon N]$. It follows that querying for equi-probable intervals on the quantile Q_1 will give this error for both the lower and upper estimate, giving $S_i = \frac{N}{K} \pm 2\epsilon N$. We compare this with $\hat{S_i}$ as approximated in

line 2 of Algorithm 2, where $\hat{S_i} = \frac{N}{K}$. Therefore it follows that approximating S_i with $\hat{S_i}$ gives $\pm 2\epsilon N$ error.

Now consider R_i. Let $l = Q_1(\frac{i-1}{K})$ and $u = Q_1(\frac{i}{K})$, the bounds for the interval of S_i. Let j_l and j_u be the fraction of the second stream less than u and l, respectively. It follows that $R_i = M(j_u - j_l)$. We compare this with $\hat{R_i}$ as computed in Algorithm 2. By Theorem 4.1, we know that estimating j_l with $\hat{j_l}$ gives at most $\pm 3\epsilon$ error, and similarly estimating j_u with $\hat{j_u}$ gives at most $\pm 3\epsilon$ error. Therefore it follows that approximating $R_i = M(j_u - j_l)$ with $\hat{R_i} = M(\hat{j_u} - \hat{j_l})$ gives $\pm 6\epsilon M$ error.

In order to find the total error of χ^2, we must calculate the error at every i. We separate the error into two cases, where we look separately at the largest overestimate the error can give, and the largest underestimate the error can give. In order to do these calculations, we must have an estimate for how close R_i is to $\frac{M}{K}$, i.e., bound $E \equiv |R_i - \frac{M}{K}|$. Let $\epsilon = \frac{1}{320\sqrt{N}K^2}$, and recall that $|S_i - \frac{N}{K}| \leq 2\epsilon N$. We can bound the chi-square statistic term for the ith bin,

$$\lambda_i \equiv \frac{(S_i\sqrt{M/N} - R_i\sqrt{N/M})^2}{S_i + R_i},$$

by

$$\lambda_i \geq \frac{(\frac{N}{K}\sqrt{M/N} + 2\epsilon N\sqrt{M/N} - \frac{M}{K}\sqrt{N/M} - E\sqrt{N/M})^2}{N/K + 2\epsilon N + M/K + E}.$$

Simplifying this, we get

$$\lambda_i \geq \frac{(2\epsilon\sqrt{M} - E/\sqrt{M})^2}{\frac{1}{K}(1 + 2\epsilon K + \frac{M}{N} + \frac{EK}{N})}.$$

We assume that $\epsilon = \frac{1}{320\sqrt{N}K^2}$, $M \leq N$, and $EK \leq N$, so it follows that

$$\lambda \geq \frac{(2\epsilon\sqrt{M} - E/\sqrt{M})^2}{\frac{4}{K}} = \epsilon^2 KM - \epsilon EK + \frac{E^2K}{4M}.$$

We assume that $\epsilon EK < 1$, so that $\lambda \geq \frac{E^2K}{4M} - 1$. Suppose that $E \geq 20\sqrt{M/K}$. Then $\lambda \geq 99$, and therefore if $E \geq 20\sqrt{M/K}$, we know that we should reject the null hypothesis when there are a small number of bins since the critical value will always be much less than 99. As the leading constant 20 is arbitrary, it can be changed to account for a larger number of bins. Lines 9-10 in Algorithm 2 check for the case that E is above this threshold, and so for the remainder of this analysis we will assume that $|R_i - \frac{M}{K}| \leq 20\sqrt{M/K}$.

We assume that $\epsilon = \frac{1}{320\sqrt{N}K^2}$, and in order for the algorithm to provably bounded error, we will also assume that $N \geq 10^{14}, M \geq 10^{14}$, (i.e., the case for truly big data that cannot be stored in main memory) and $10 \leq K \leq 1000$, which are typical values for the number of bins. We will later show in the evaluation section that our algorithm also works for considerably smaller streams as well.

As opposed to the analysis of the one-sample algorithm, here we look at the under and overestimates of the actual statistic that can be given by the algorithm. First we will look at the overestimate, β, where we will calculate the difference between the actual statistic and the estimated statistic given by Algorithm 2. We are looking for the error that gives the largest possible actual chi-square statistic and smallest estimated chi-square statistic. Therefore it follows that for the first fraction the error will maximize the numerator and minimize the denominator. Inversely for the second fraction, the error will minimize the numerator and maximize the denominator.

We bound

$$\begin{aligned}
\beta &= \sum_{i=1}^{K}\left(\frac{(S_iK_1 - R_iK_2)^2}{S_i + R_i} - \frac{(\hat{S_i}K_1 - \hat{R_i}K_2)^2}{\hat{S_i} + \hat{R_i}}\right) \\
&\leq \sum_{i=1}^{K}\left(\frac{((\frac{N}{K} + 2\epsilon N)\sqrt{\frac{M}{N}} - (\frac{M}{K} - E)\sqrt{\frac{N}{M}})^2}{\frac{N}{K} + \frac{M}{K} - 2\epsilon N - E} - \right. \\
&\left. \frac{(\frac{N}{K}\sqrt{\frac{M}{N}} - (\frac{M}{K} + E + 6\epsilon M)\sqrt{\frac{N}{M}})^2}{\frac{N}{K} + \frac{M}{K} + E + 6\epsilon M}\right) \\
&= \sum_{i=1}^{K}\left(\frac{(2\epsilon\sqrt{NM} + E\sqrt{\frac{N}{M}})^2}{\frac{N}{K} + \frac{M}{K} - 2\epsilon N - E} - \frac{(-6\epsilon\sqrt{NM} - E\sqrt{\frac{N}{M}})^2}{\frac{N}{K} + \frac{M}{K} + E + 6\epsilon M}\right) \\
&= \sum_{i=1}^{K}\left(\frac{(\frac{2\sqrt{M}}{320K^2} + 20\sqrt{\frac{N}{K}})^2}{\frac{N}{K} + \frac{M}{K} - \frac{2\sqrt{N}}{320K^2} - 20\sqrt{\frac{M}{K}}} - \right. \\
&\left. \frac{(\frac{-6\sqrt{M}}{320K^2} - 20\sqrt{\frac{N}{K}})^2}{\frac{N}{K} + \frac{M}{K} + \frac{6M}{320\sqrt{N}K^2} + 20\sqrt{\frac{M}{K}}}\right).
\end{aligned}$$

From here we would like to have common denominators in the fractions, which can be achieved by considering the error terms as fractions of $\frac{N}{K} + \frac{M}{K}$. With our earlier assumptions about N, M, and K, it follows that $\frac{2\sqrt{N}}{320K^2} / \frac{N}{K} \leq \frac{2}{3}(10^{10})$, $\frac{6M}{320\sqrt{N}K^2} / \frac{N}{K} \leq 2(10^{10})$, and $20\sqrt{\frac{M}{K}} / \frac{M}{K} \leq 2(10^{-7/2})$. Using these fractions, it follows that:

$$\begin{aligned}
\beta &\leq \sum_{i=0}^{K}\left(\frac{(\frac{2\sqrt{M}}{320K^2} + 20\sqrt{\frac{N}{K}})^2}{\frac{N+M}{K}(1 - 0.00006)} - \frac{(\frac{-6\sqrt{M}}{320K^2} - 20\sqrt{\frac{N}{K}})^2}{\frac{N+M}{K}(1 + 0.00006)}\right) \\
&\leq \sum_{i=0}^{K}\left(\frac{(\frac{2\sqrt{M}}{320K^2} + 20\sqrt{\frac{N}{K}})^2(1.00006)}{\frac{N+M}{K}} - \right. \\
&\left. \frac{(\frac{-6\sqrt{M}}{320K^2} - 20\sqrt{\frac{N}{K}})^2(0.9994)}{\frac{N+M}{K}}\right) \\
&\leq \frac{K^2}{N+M}\left(\frac{-0.0003M}{K^4} + \frac{0.048N}{K} - \frac{0.4999\sqrt{NM}}{K^2\sqrt{K}}\right) \\
&= \frac{N}{N+M}\left(0.048K - \frac{0.0003M}{NK^2} - \frac{0.4999\sqrt{M}}{\sqrt{NK}}\right) \\
&\leq 0.048K - \frac{0.0003}{K^2} - \frac{0.4999}{\sqrt{K}}.
\end{aligned}$$

As for the underestimate, γ, we are again calculating the difference between the actual chi-square statistic and the estimate generated from Algorithm 2, however here we are looking for the error that minimizes the actual statistic and maximizes the estimated statistic. Just as with the overestimate, we consider the error in terms of $\frac{N}{K} + \frac{M}{K}$ to further simplify the equations.

We have

$$\begin{aligned}
\gamma &= \sum_{i=1}^{K}\left(\frac{(S_iK_1 - R_iK_2)^2}{S_i+R_i} - \frac{(\hat{S}_iK_1 - \hat{R}_iK_2)^2}{\hat{S}_i+\hat{R}_i}\right)\\
&= \sum_{i=1}^{K}\left(\frac{((\frac{N}{K}-2\epsilon N)\sqrt{\frac{M}{N}} - (\frac{M}{K}+E)\sqrt{\frac{N}{M}})^2}{\frac{N}{K}+\frac{M}{K}+2\epsilon N+E} - \right.\\
&\left.\frac{(\frac{N}{K}\sqrt{\frac{M}{N}} - (\frac{M}{K}-E-6\epsilon M)\sqrt{\frac{N}{M}})^2}{\frac{N}{K}+\frac{M}{K}-E-6\epsilon M}\right)\\
&= \sum_{i=1}^{K}\left(\frac{(\frac{-2\sqrt{M}}{320K^2} - 20\sqrt{\frac{N}{K}})^2}{\frac{N}{K}+\frac{M}{K}+\frac{2\sqrt{N}}{320K^2}+20\sqrt{\frac{M}{K}}} - \right.\\
&\left.\frac{(\frac{6\sqrt{M}}{320K^2} + 20\sqrt{\frac{N}{K}})^2}{\frac{N}{K}+\frac{M}{K}-\frac{6M}{320\sqrt{N}K^2}-20\sqrt{\frac{M}{K}}}\right)\\
&\le \sum_{i=1}^{K}\left(\frac{(\frac{-2\sqrt{M}}{320K^2} - 20\sqrt{\frac{N}{K}})^2}{\frac{N+M}{K}(1+0.00006)} - \frac{(\frac{6\sqrt{M}}{320K^2} + 20\sqrt{\frac{N}{K}})^2}{\frac{N+M}{K}(1-0.00006)}\right)\\
&\le \frac{K^2}{N+M}\left(\frac{-0.0003M}{K^4} - \frac{0.048N}{K} - \frac{0.5001\sqrt{NM}}{K^2\sqrt{K}}\right)\\
&\le -0.048K - \frac{0.0003}{K^2} - \frac{0.50001}{\sqrt{K}}.
\end{aligned}$$

Therefore the total error, X, is bounded by $-0.048K - \frac{0.0003}{K^2} - \frac{0.50001}{\sqrt{K}} \le X \le 0.048K - \frac{0.0003}{K^2} - \frac{0.4999}{\sqrt{K}}$, or approximately $0.048K$. □

The error is dependent on K, however this is not a problem as the critical value centers around K or $K-1$, so an error of $0.048K$ will have a negligible effect on the result of the test. Also note that, while the above theorem requires a truly massive stream ($N \ge 10^{14}$), this is simply for the worst guarantee to work. We will show that there are considerable savings even for much smaller streams in the experimental evaluation section.

5.2 Computational Analysis

The analysis of the space required by the two-sample test is almost identical to that of the one-sample, with the only difference being storing two streams of lengths N and M instead of storing a single stream of length N. However since we have assumed that $N \ge M$, the analysis remains the same, such that the Greenwald-Khanna sketch uses at most $O(K^2 \log(N)\sqrt{N})$ space. As before, we can summarize terabytes of data using hundreds of megabytes of main memory. The running time of Algorithm 2 is $O(K \log N)$, similar to Algorithm 1.

5.3 Accessible Streaming

As with the one-sample algorithm, this algorithm does not need much prior information about either stream other than an upper bound on the stream sizes and number of bins. Moreover, the sketch for both streams can be computed independently of one another (e.g., at two different locations) with no information or communication necessary until it is time to compute the test. The tight space bounds allow these sketches to be efficiently stored and transferred between locations.

6 CATEGORICAL DATA

In this section we show that computing the chi-square test on categorical data will entail using a large (linear) amount of memory. We then give an algorithm that reduces the memory requirement by a (significant) constant amount and show in Section 7 that it performs well in practice.

6.1 Lower Bound

To show that the categorical chi-square test requires linear memory, we use a result from [10] about sketching decomposable distances. For a pair of data streams, p and q in which the fraction of the stream that comprise item i is given by p_i and q_i, respectively, the decomposable distance d_ϕ is defined as $d_\phi(p,q) = \sum_i \phi(p_i, q_i)$. This relates to the chi-square categorical test which, for the case of equal length streams (say, length N), can be written in this form as $\sum_i \phi(p_i, q_i)$ with $\phi(x,y) = \frac{N(x-y)^2}{x+y}$.

The following Shift Invariant Theorem from [10] shows that any decomposable distance of the above form with certain properties cannot be sketched using sublinear space:

Theorem 6.1 ([10]). *For ϕ such that $\phi(x,x) = 0$ for all $x \in [0,1]$, if for sufficiently large n there exists $a, b, c > 0$ such that*

$$\max\left(\phi\left(\frac{a+c}{t}, \frac{a}{t}\right), \phi\left(\frac{a}{t}, \frac{a+c}{t}\right)\right) > \frac{\alpha^2 w}{4}\left(\phi\left(\frac{b+c}{t}, \frac{b}{t}\right) + \phi\left(\frac{b}{t}, \frac{b+c}{t}\right)\right)$$

where $t = \alpha n/4 + bn + cn/2$, then any streaming algorithm over values in $[5n/4]$ for estimating $\sum_i \phi(p_i, q_i)$ within factor α with probability at least 3/4 for a stream of length $O((a+b+c)n)$ must use $\Omega(w)$ space.

We now use the above theorem to show the desired lower bound for the chi-square test.

Theorem 6.2. *The categorical chi-square test statistic cannot be approximated to within a factor of 2 with probability greater than 3/4 using a sketch unless it uses $\Omega(\min(N/K, K))$ space.*

Proof. Assume that the two streams are of the same length ($N = M$). (Note that if the equal length stream case is hard to approximate, it follows that the general case will no less hard.) For each category i, let $s_i = S_i/N$ and $r_i = R_i/N$ be the fractions of the two streams that make up this category. The chi-square statistic when the two streams are of the same length is

$$\begin{aligned}
\chi^2 &= \sum_{i=1}^{K}\frac{(S_i - R_i)^2}{S_i+R_i}\\
&= \sum_{i=1}^{K}\frac{N^2(s_i-r_i)^2}{N(s_i+r_i)}\\
&= \sum_{i=1}^{K}\frac{N(s_i-r_i)^2}{s_i+r_i}.
\end{aligned}$$

Applying Theorem 6.1 with $\phi(x,y) = \frac{N(x-y)^2}{x+y}$, $n = 4K/5$, $a = 1$, $b = 5N/4K - 2$, $c = 1$, $\alpha = 2$, $w = 5N/12K - 1$, it is easy to

verify that all the conditions for ϕ are satisfied. Moreover, the number of categories is $5n/4 = K$ and the length of the stream is $N = (a + b + c)n$. Finally, it is always possible to estimate the statistic by storing the counts for each category using K counters. Hence, there is no hope for sketching the categorical chi-square test using less than $\Omega(\min(w, K)) = \Omega(\min(N/K, K))$ space even when we allow a 2-factor approximation error and a 1/4 probability of failure. □

6.2 Algorithm

Algorithm 3 Insert(S_1, S_2, p, R)

Input: Two streams of categories S_1 and S_2, sampling rate p, range of hash table R.

Initialize empty hash tables t_1 and t_2.
Initialize function h that uniformly maps categories into the range $[0, R-1]$.
for each category c in S_1 **do**
    $hash = h(c)$
    if $hash < pR$ **then**
        if c is in t_1 **then**
            Increment value of c in t_1
        else
            Insert c into t_1 with value 1
Similarly, insert stream S_2 into t_2 with same h

Algorithm 4 CalculateStatistic($N, M, K, t_1, t_2, \alpha$)

Input: Stream lengths N and M; number of categories K; hash tables t_1, t_2; significance level α.
Output: Whether the test rejects the null hypothesis

$\hat{\chi}^2 = 0$
for each category i in $t_1 \cup t_2$ **do**
    R_i = value for key i in t_1 (0 if not present)
    S_i = value for key i in t_2 (0 if not present)
    $\hat{\chi}^2 = \hat{\chi}^2 + \frac{(R_i\sqrt{\frac{M}{N}} - S_i\sqrt{\frac{N}{M}})^2}{R_i + S_i}$
Let c be 1 if N and M are equal, 0 otherwise.
Let $\chi^2_{1-\alpha, K-c}$ be the critical value at significance level α and degrees of freedom $K - c$.
return $\hat{\chi}^2 > \chi^2_{1-\alpha, K-c}$

Here we present our heuristic algorithm for the two-sample categorical variant of the chi-square test. Similar to the continuous variant, data from two different streams are compared to determine if they come from the same distribution. Like the two sample continuous variant, no prior knowledge of the underlying distribution is required to calculate the test statistic.

Our algorithm makes use of the concept of coordinated sampling [5] which is superior to uniform sampling. If we were to perform uniform sampling on the two streams, say with a rate of 10%, then the overlap of the two streams would be only 1% of all the possible categories. We get around this by coordinating the sampling so that both streams measure the same random categories, making the final sample size 10% as well.

In the insertion algorithm (see Algorithm 3), every category is passed into the same hash function which returns a hash value uniformly within a particular range. (In our experiments, we used a hash range of $[0, 2^{32})$). By the method shown in the algorithm, each category has precisely p probability of being sampled (where p is the sampling rate). Since both streams use the same hash function, the sampled categories will be the same in both streams. This achieves coordinated sampling and gives our technique a strong advantage over uniform sampling.

We use a hash table to keep track of the frequency of each sampled category. Once the frequency of each category is calculated, the algorithm computes the statistic using the two-sample formula, as shown in Algorithm 4. Depending on the significance level we get the critical value to determine whether to reject the null hypotheses. We will show how this algorithm performs experimentally in the next section.

The insertion algorithm (Algorithm 3) simply samples the stream, taking $O(1)$ time per insertion. The memory requirement of the algorithm is linear, but has a $1/p$ factor savings over storing the frequencies of all the categories. Algorithm 4 has a running time of $O(K)$.

6.3 Accessible Streaming

The categorical chi-square algorithm is also easy to deploy with minimal configuration necessary. The only parameter to be tuned is the sampling rate p, which can be selected based on the memory constraints of the user. There is no need to know the number of categories (K) beforehand. The coordinated hash function can be any hash function implementation that is known to give a uniformly random distribution across its range.

7 EXPERIMENTAL EVALUATION

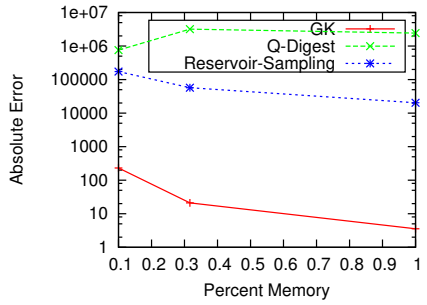
In this section we show the results from our experimental evaluations on synthetic and empirical data sets. We ran all of our experiments on 3.2 GHz Intel quad-core i5 processors with 8 GB memory. All of our code is in C++.

We generated the synthetic data from several different distributions, namely the normal, Pareto, and uniform distributions, as these are commonly seen in practice. In order to get an accurate measure, the results were averaged over ten independently generated streams of data. For the one-sample and two-sample continuous variants of the chi-square test we ran experiments using different quantile sketches and varying different parameters, such as stream size, number of bins, and percent of the stream size that the sketches used. Unless otherwise specified, all experiments used stream sizes of ten million ($N = 10^7$), 20 bins ($K = 20$), and 1% of the memory needed to store the entire stream.

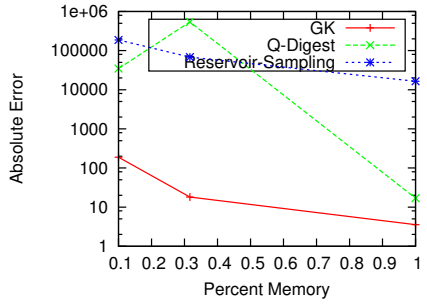
For the real data sets we looked at the following streams:

- **Light Data**: We used radiant light energy measurements, specifically the irradiance level (micro W/cm^2) collected by Columbia University's EnHANTs project [1], which contains two data streams coming from Trace A (1133636 values) and Trace B (1081793 values) that were gathered in 2009 and 2010, respectively.

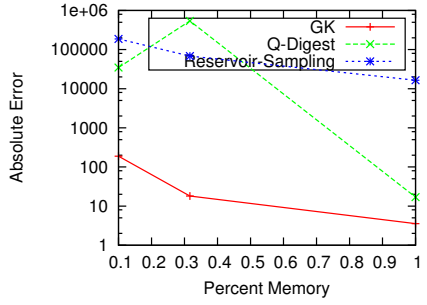
[1] http://www.crawdad.org/columbia/enhants/20110407/



(a) N(0, 1) vs. N(0, 1)



(b) U(0, 1) vs. U(0, 1)



(c) P(1, 2) vs. P(1, 2)

Figure 1: Varying memory ($N = 10^7, K = 20$) for one-sample data drawn from various distributions

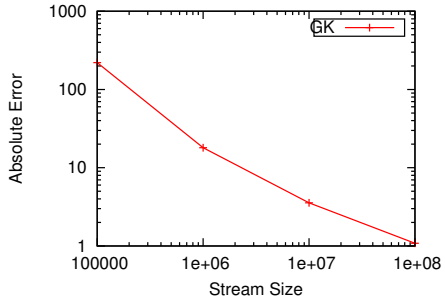


Figure 2: Varying one-sample n, $\mathbf{N}(0, 1)$ vs. $\mathbf{N}(0, 1)$ (1% memory, $K = 20$)

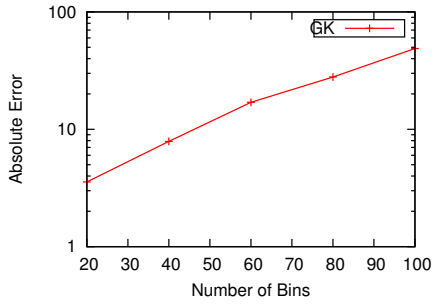


Figure 3: Varying one-sample k, $\mathbf{N}(0, 1)$ vs. $\mathbf{N}(0, 1)$ (1% memory, $n = 10^7$)

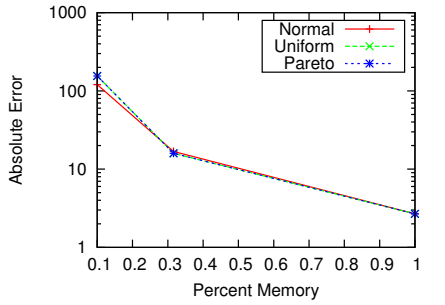


Figure 4: Varying two-sample memory ($N = 10^7, K = 20$)

- **Power Consumption Data**: We compared household electric power consumption in 2006-2007 (543661 values) against consumption in 2008-2009 (1505619 values), collected from EDF R&D, Clamart, France [2].
- **U.S. Census Data (1990)**: We gathered the data in our categorical variant experiments from a one percent sample (2458285 values) of the Public Use Microdata Samples (PUMS) person records drawn from the 1990 census sample. [3] The categories were combinations of demographic information such as age, gender, and marital status.

[2]http://archive.ics.uci.edu/ml/datasets/Individual+\household+electric+power+consumption
[3]https://archive.ics.uci.edu/ml/datasets/US+Census+\Data+(1990)

As for the specific quantile sketches implemented and used, first is the Greenwald-Khanna (GK) sketch [8], which uses $O(\frac{1}{\epsilon}\log(\epsilon n))$ memory. We also run tests on the Q-Digest [25], which uses $O(\frac{1}{\epsilon}\log U)$ memory where U is the size of the input. Additionally, we compared these with a uniform sampling technique, namely reservoir sampling.

The goal of these algorithms is to compute the chi-square statistic with enough accuracy that the hypothesis is or is not rejected at some given significance level. This is done by comparing the chi-square statistic with the critical value from a table. We will show in this section that the error added by using our algorithm is very small compared with the critical value, thereby giving confidence that the algorithm does not erroneously reject (or fail to reject) the hypothesis.

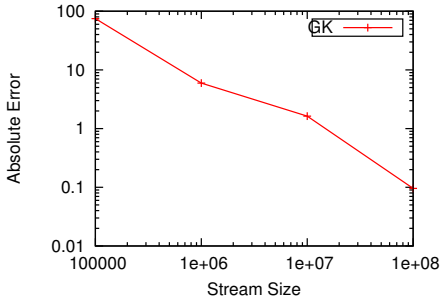


Figure 5: Varying N, M, $\mathbf{N}(0,1)$ vs. $\mathbf{N}(0,1)$ (1% memory, $K = 20$)

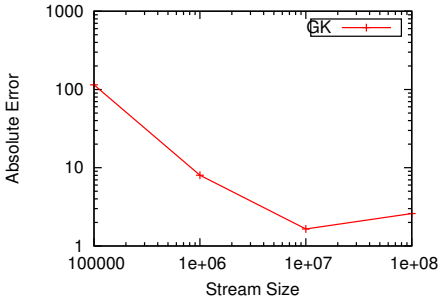


Figure 6: Varying N only, $\mathbf{N}(0,1)$ vs. $\mathbf{N}(0,1)$ (1% memory, $K = 20$)

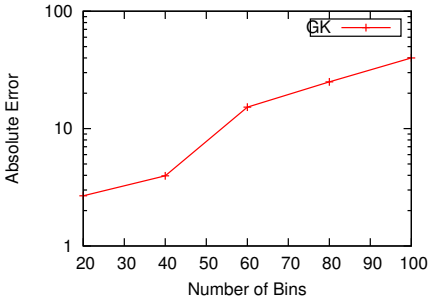
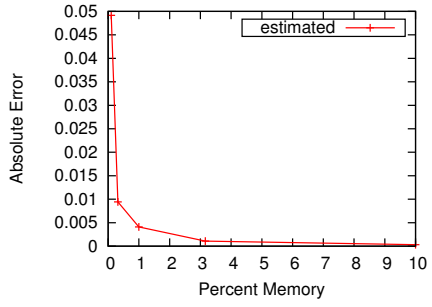
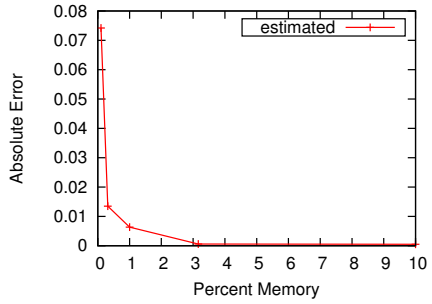


Figure 7: Varying K, $\mathbf{N}(0,1)$ vs. $\mathbf{N}(0,1)$ (1% memory)

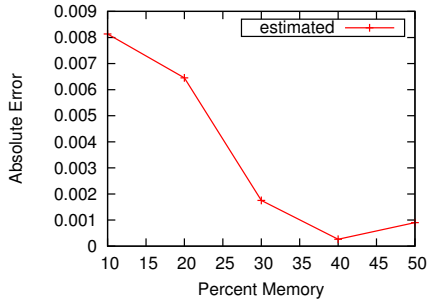


(a) radiant light energy

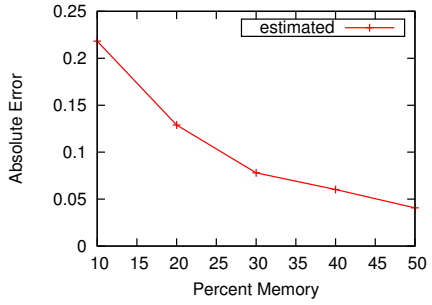


(b) household power consumption

Figure 8: Real data for two sample continuous



(a) S1 and S2 data



(b) stream1 and stream2

Figure 9: Real data for the categorical algorithm

7.1 One-sample Continuous Data

In these experiments, we compared the synthetic data to the same distribution it was generated from. We first looked at the three different distributions while varying memory, illustrated in Figure 1. The Reservoir Sampling and Q-Digest sketches give very high error when using 0.1% of memory and the error from sampling remains high even at 1% of memory. In contrast, the GK sketch gives relatively low error and reduces to an error of about 1 while using just 1% of the cost of storing the entire stream. Comparing this error with the critical value for $\alpha = .05$ of 27.587 [4], we can see that the error is considerably smaller and hence will not have a significant effect on the result of the test. This error can be diminished further by increasing the size of the sketch by a few percent.

For the remainder of this subsection, we focus on the normal distribution as the results are similar to the other two distributions. In Figure 2 we hold memory at 1% while varying the size of the stream. While the error is not negligible, the plot shows a downward trend as the stream size increases. At $N = 10^8$ the absolute error is around 1, which is fairly small error. It follows that the one-sample algorithm requires a very large sample size for best results. This resonates with the theory, for our assumption of $\epsilon = \frac{1}{300\sqrt{N}K^2}$ requires N to be very large in order to produce significant savings.

[4]When $K = 20$, $\alpha = 0.05$, and we have a distribution with two parameters ($c = 2+1$), the critical value to compare against is $\chi^2_{1-\alpha,K-c} = \chi^2_{0.95,17} = 27.587$.

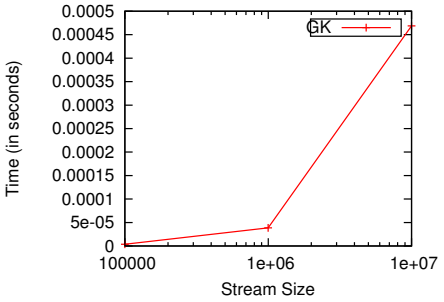


Figure 10: Varying one sample n, $\mathbf{N}(0,1)$ data (1% memory, $K = 20$)

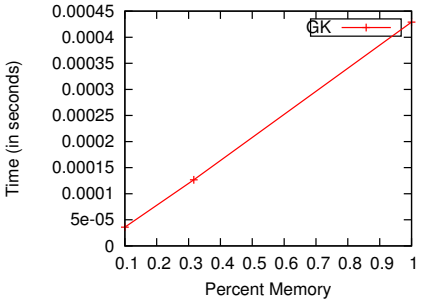
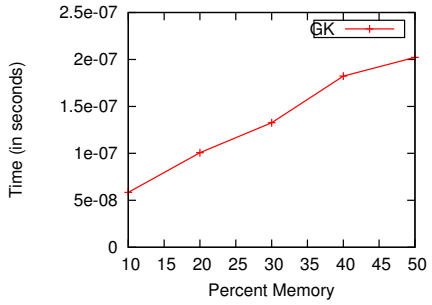
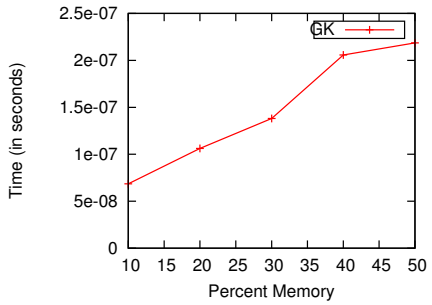


Figure 11: Varying memory ($N = 10^7, K = 20$) for one-sample $\mathbf{N}(0,1)$ data



(a) S1 and S2 data



(b) stream1 and stream2

Figure 12: Time per insertion using the categorical algorithm with real data

The one deficiency in our algorithm is that for larger number of bins there is more error, as seen in Figure 3 which varies the number of bins while keeping stream size and memory constant. This is because the memory dependence on K is quadratic, hence the accuracy of our algorithm declines as we increase K and keep the memory constant.

7.2 Two-sample Continuous Data

As we already saw that the Q-Digest and sampling techniques perform very poorly, we omit testing them with the two-sample case. Instead we just compare the three different distributions using only the GK sketch while varying memory, as seen in Figure 4. As with the one-sample case, at very small amounts of memory the test performs poorly in all distributions, however around 1% of memory the test performs well. There is slightly more error here than in the one-sample case, however this is expected as the theory states that the two-sample case requires huge stream sizes. For the following experiments we will again focus on the normal distribution.

In Figure 5 we vary the size of both streams, such that they always are of the same length, and we are increasing the sizes together. Each point of this plot is an average over five (rather than ten) repetitions because of the longer running time. There is a very sharp drop from $N = M = 10^6$ to $N = M = 10^7$, just as in the one-sample case, once again because 1% memory gives too small of a sketch for the former case but starts to become effective by the latter. The performance will continue to improve as the stream size increases.

Next, in Figure 6, we fix the second stream at size $M = 10^7$ and vary the size of the first stream. We once again see an increase in accuracy as the stream size increases. The results are slightly better than for Figure 5 since, except for the last point, the streams and hence the sketches are larger.

As with the one sample case, the one deficiency to our algorithm is the direct relationship between the increasing number of bins and an increasing amount of error, as seen in Figure 7. This relates directly to the theory in Algorithm 2, for the error is related to K, and therefore increases as K increases.

Finally we tested the two sample continuous algorithm on the empirical data sets, the first of which compares household power consumption, and the second of which looks at radiant light energy measurements. As can be seen in Figure 8, the absolute error is negligible (under 0.1) even when using just 1% of the memory needed to store the entire stream. We thus see that even for streams of length considerably smaller than $N = 10^{14}$ (see Section 5) we can obtain up to two orders of magnitude reduction in the amount of memory needed.

7.3 Categorical Data

We performed tests on our categorical algorithm with empirical data. In Figure 9 we show the absolute error of our algorithm when performing tests on the data sets described above. The results show that we have low absolute error even when using an order of magnitude less memory than storing the entire stream. These plots are less smooth than the ones before because each experiment is

run on the same data set (as opposed to an average of ten in the synthetic case).

While the results for the categorical case are not as impressive as for the continuous tests, this is expected from the theory. We can only hope for constant savings in this scenario unless we allow for huge error and probability of failure. Still, the results show how one can trade off the memory footprint and error when performing this version of the test.

7.4 Running Time

Finally, we performed some of the same tests as earlier, but instead evaluated the average time per insertion. In Figure 10, we looked at the time per insertion of increasing stream sizes with the same 1% of memory and 20 bins. As the plot shows, there is a sharp increase, almost 10-fold, between the stream sizes of $n = 10^6$ and $n = 10^7$. This is due to the fact that with a common memory percent, the 10^7 stream size is storing about 10 times more values in the quantile than the 10^6 sized stream. As our implementation of the quantile uses a list instead of a tree structure, each subsequent insertion takes more time, resulting in a higher average time per insertion for a longer stream. This effect would be ameliorated by a tree implementation.

A similar effect is also seen when we look at the time per insertion when varying memory in Figure 11. While more linear, the time per instruction clearly goes up when more values are added to the quantile. Still, it is clear that our implementation is able to handle thousands of items per second. A more efficient and optimized tree implementation would potentially allow for millions of insertions per second.

We also looked at the average time per insertion for categorical data, using our aforementioned streams of real data, see Figure 12. These average times per insertion are much smaller than that of the previous graphs, since the insertion is an $O(1)$ time operation. Here, the algorithm is able to perform about 5 million insertions per second.

8 CONCLUSIONS

In conclusion, motivated by a wide range of needs from real world applications, we provided streaming algorithms for the one-sample continuous and the two-sample continuous and categorical versions of Pearson's chi-square goodness-of-fit test. The algorithms have rigorous proofs demonstrating their accuracy, further confirmed in our experimental evaluation. We found that using the Greenwald-Khanna quantile sketch gives the best result, and our algorithm greatly outperforms the strategy of uniform sampling.

There is still much future work that is possible. One avenue of future work is to determine if the

$$O(K^2 \log(N)\sqrt{N})$$

space utilization of our algorithms for continuous data can be improved upon or if there is a corresponding lower bound. In the case of categorical data, we showed that coordinated sampling can give good results with reduced memory, but there may still be further improvements possible. All the tests proposed in this paper are for single-dimensional data; similar tests for higher-dimensional or structured data are also possible. Finally, there are many other statistical tests that do not have obvious streaming algorithms that allow them to be computed in sub-linear space. Determining which ones can and cannot be categorized as such would be of interest to scientists and practitioners in industry that need to perform these tests on big data sets.

REFERENCES

[1] Charu C Aggarwal. 2007. *Data streams: models and algorithms*. Vol. 31. Springer Science & Business Media.

[2] Noga Alon, Yossi Matias, and Mario Szegedy. 1996. The space complexity of approximating the frequency moments. In *STOC*.

[3] Fabrizio Angiulli and Fabio Fassetti. 2007. Detecting Distance-based Outliers in Streams of Data. In *CIKM*.

[4] Amit Chakrabarti, Khanh Do Ba, and S. Muthukrishnan. 2006. Estimating Entropy and Entropy Norm on Data Streams. In *STACS*.

[5] Edith Cohen and Haim Kaplan. 2013. What You Can Do with Coordinated Samples. In *RANDOM/APPROX*.

[6] Philippe Flajolet and G. Nigel Martin. 1985. Probabilistic counting algorithms for data base applications. *J. Comput. Syst. Sci.* 31, 2 (1985).

[7] Anna C. Gilbert, Yannis Kotidis, S. Muthukrishnan, and Martin J. Strauss. 2003. One-Pass Wavelet Decompositions of Data Streams. *IEEE Trans. on Knowl. and Data Eng.* 15, 3 (March 2003).

[8] Michael Greenwald and Sanjeev Khanna. 2001. Space-Efficient Online Computation of Quantile Summaries. In *SIGMOD*.

[9] Michael B Greenwald and Sanjeev Khanna. 2016. Quantiles and equidepth histograms over streams. *In Data Stream Management: Processing High-Speed Data Streams. Springer* (2016).

[10] Sudipto Guha, Piotr Indyk, and Andrew McGregor. 2008. Sketching information divergences. *Machine Learning* 72, 1 (2008), 5–19.

[11] Sudipto Guha, Andrew Mcgregor, and Suresh Venkatasubramanian. 2006. Streaming and sublinear approximation of entropy and information distances. In *SODA*.

[12] Sudipto Guha, Adam Meyerson, Nina Mishra, Rajeev Motwani, and Liadan O'Callaghan. 2003. Clustering Data Streams: Theory and Practice. *IEEE TKDE* 15, 3 (March 2003).

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

[14] Abhishek Kumar, Minho Sung, Jun Xu, and Ellen W. Zegura. 2005. A Data Streaming Algorithm for Estimating Subpopulation Flow Size Distribution. In *Proc. of ACM SIGMETRICS*.

[15] A. Lall. 2015. Data streaming algorithms for the Kolmogorov-Smirnov test. In *IEEE Big Data*.

[16] Ashwin Lall, Vyas Sekar, Mitsunori Ogihara, Jun Xu, and Hui Zhang. 2006. Data Streaming Algorithms for Estimating Entropy of Network Traffic. In *SIGMETRICS*.

[17] Ping Li, Gennady Samorodnitsk, and John Hopcroft. 2013. Sign Cauchy Projections and Chi-Square Kernel. In *Advances in Neural Information Processing Systems 26*.

[18] Ge Luo, Lu Wang, Ke Yi, and Graham Cormode. 2016. Quantiles over Data Streams: Experimental Comparisons, New Analyses, and Further Improvements. *The VLDB Journal* 25, 4 (Aug. 2016), 449–472. DOI:http://dx.doi.org/10.1007/s00778-016-0424-7

[19] Gurmeet Singh Manku, Sridhar Rajagopalan, and Bruce G. Lindsay. 1998. Approximate Medians and Other Quantiles in One Pass and with Limited Memory. In *SIGMOD*.

[20] Gurmeet Singh Manku, Sridhar Rajagopalan, and Bruce G. Lindsay. 1999. Random Sampling Techniques for Space Efficient Online Computation of Order Statistics of Large Datasets. *SIGMOD Rec.* 28, 2 (June 1999).

[21] Hamid Mousavi and Carlo Zaniolo. 2011. Fast and Accurate Computation of Equi-depth Histograms over Data Streams. In *EDBT/ICDT*.

[22] J. I. Munro and M. S. Paterson. 1978. Selection and sorting with limited storage. In *SFCS*.

[23] Shanmugavelayutham Muthukrishnan. 2005. *Data streams: Algorithms and applications.* Now Publishers Inc.

[24] NIST. 2015. Chi-Square Two Sample Test. http://www.itl.nist.gov/div898/software/dataplot/refman1/auxillar/chi2samp.htm. (Oct 2015).

[25] Nisheeth Shrivastava, Chiranjeeb Buragohain, Divyakant Agrawal, and Subhash Suri. 2004. Medians and beyond: new aggregation techniques for sensor networks. In *SenSys*.

[26] Lu Wang, Ge Luo, Ke Yi, and Graham Cormode. 2013. Quantiles over Data Streams: An Experimental Study. In *SIGMOD*.

[27] Nong Ye and Qiang Chen. 2001. An anomaly detection technique based on a chifiifftsquare statistic for detecting intrusions into information systems. *Quality and Reliability Engineering International* 17 (03 2001), 105 – 112.