Variance Reduction in Frequency Estimators via Control Variates Method

Rameshwar Pratap[1] **Raghav Kulkarni**[2]

[1]Indian Institute of Technology (IIT), Mandi H.P., India.
[2]Chennai Mathematical Institute (CMI) Chennai, India.

Abstract

Generating succinct summaries (also known as *sketches*) of massive data streams is becoming increasingly important. Such a task typically requires fast, accurate, and small space algorithms in order to support the downstream applications, mainly in areas such as data analysis, machine learning and data mining. A fundamental and well-studied problem in this context is that of estimating the frequencies of the items appearing in a data stream. The Count-Min-Sketch Cormode and Muthukrishnan [2005] and Count-Sketch Charikar et al. [2004] are two known classical algorithms for this purpose. However, a limitation of these techniques is that the variance of their estimate tends to be large. In this work, we address this problem and suggest a technique that reduces the variance in their respective estimates, at the cost of little computational overhead. Our technique relies on the classical Control-Variate trick Lavenberg and Welch [1981] used for reducing variance in Monte-Carlo simulation. We present a theoretical analysis of our proposal by carefully choosing the control variates and complement them with experiments on synthetic as well as real-world datasets.

1 INTRODUCTION

Many real-life applications such as managing web clicks and crawls, sensor/IoT readings Madden and Franklin [2002], real-time IP traffic analysis Estan and Varghese [2002], email/tweets/SMS, time-series data Zhu and Shasha [2002], metagenomics Elworth et al. [2020] and other text sources are instances of massive data streams. In these applications, new data arrives at a rapid rate, and often there is not enough space available to store all the data. Therefore, managing such a large stream of data requires algorithmic techniques that maintain a small footprint of the data stream with the facility of incorporating fast updates. Such a small footprint is also known as the *sketch* of the entire data stream which is used for getting important statistics of the data stream for post-hoc queries Muthukrishnan [2005].

In this work, we focus on the problem of estimating the frequency of items appearing in the data stream. Formally, suppose we have a stream $\sigma = \{a_1, a_2, \ldots, a_m\}$ of length m, where each element of the stream is drawn from the universe $[n] := \{1, 2, \ldots, n\}$, that is, $\forall i \in [m]$, we have $a_i \in [n]$. Given a query item (say a) the problem is to give an estimate of the number of times the item a has appeared in the stream. Of course, the problem can be solved exactly by creating a histogram over each element of the universe. However, given the space constraint this is not a feasible approach. Therefore, the aim is to develop a space-efficient sketch (data structure) that outputs an accurate estimate of the frequency of the query point. This problem has been studied extensively in the literature and there are two classical techniques available for this problem – Count-Min-Sketch (CMS) Cormode and Muthukrishnan [2005] and Count-Sketch (CS) Charikar et al. [2004]. These techniques have been applied successfully in computing sketches of many large scale streaming datasets that enable post-hoc frequency estimation queries Cormode and Muthukrishnan [2005], Charikar et al. [2004]. However, a major limitation of these techniques is that the variance of their estimate is large due to which the predicted frequency of a query item can be far from its ground truth frequency. In this work, we address this challenge and suggest a technique that can reduce the variance in their respective estimates and as a consequence leads to a more accurate estimation.

1.1 OUR APPROACH – VARIANCE REDUCTION USING CONTROL VARIATE TRICK

Our technique relies on utilizing the control variate (CV) trick for reducing the variance that occurs while computing the estimate of the frequency of a query item from the

Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence (UAI 2021), PMLR 161:183–193.

sketches obtained by the sketching algorithm mentioned above. The control variate trick is a classical technique used for reducing variance in Monte-Carlo simulations, by looking at correlated errors from the same random numbers Lavenberg and Welch [1981]. Suppose there is a random number generator that generates a random variable X, and we are interested in estimating the $\mathbb{E}[X]$. Suppose that the random number generator is used to generate another random variable Z, and we know its true mean $\mathbb{E}[Z]$. Then, for a constant c, the term $X + c(Z - \mathbb{E}[Z])$ is an unbiased estimator of X,

$$\mathbb{E}[X + c(Z - \mathbb{E}[Z])] = \mathbb{E}[X] + c\mathbb{E}[Z - \mathbb{E}[Z]] = \mathbb{E}[X]. \quad (1)$$

The variance of $X + c(Z - \mathbb{E}[Z])$ is given by

$$\begin{aligned} &\mathrm{Var}[X + c(Z - \mathbb{E}[Z])] \\ &= \mathrm{Var}[X] + c^2\mathrm{Var}[Z] + 2c\mathrm{Cov}[X,Z]. \end{aligned} \quad (2)$$

By elementary calculus we can find the appropriate value of c which minimise the above expression. Suppose we denote that value by $\hat{c}$, then

$$\hat{c} = -\frac{\mathrm{Cov}[X,Z]}{\mathrm{Var}[Z]}. \quad (3)$$

Equations (2), (3) give us the following

$$\mathrm{Var}[X + c(Z - \mathbb{E}[Z])] = \mathrm{Var}[X] - \frac{\mathrm{Cov}[X,Z]^2}{\mathrm{Var}[Z]}. \quad (4)$$

Thus, for a random variable X, we are able to generate another random variable $X + c(Z - \mathbb{E}[Z])$ such that both of these have the same expected value – latter is an unbiased estimator of the former. Further, due to Equation (4) the random variable $X + c(Z - \mathbb{E}[Z])$ has lower variance than that of X, as the term $\mathrm{Cov}[X,Z]^2/\mathrm{Var}[Z]$ is always non-negative, with the equality if there is no correlation between X and Z. The random variable Z is called the control variate, and the term $\hat{c}$ is called the control variate correction. Of course, there are some practical considerations that need to be addressed carefully such as for a given application defining an appropriate random variable Z, and computing the term c, etc.

1.2 OUR RESULTS

Exploiting the control variate trick, we are able to show significant variance reductions that occur in the frequency estimation of Count-Min-Sketch and Count-Sketch algorithms. Suppose we have a stream $\sigma = \{a_1, a_2, \ldots, a_m\}$ of length m, where each element of the stream is drawn from the universe $[n] := \{1, 2, \ldots, n\}$, that is $\forall i \in [m]$ we have, $a_i \in [n]$. This also implicitly define the frequency vector over the elements of the stream $\mathbf{f} = \langle f_1, f_2, \ldots, f_n \rangle$, where f_i denotes the occurrence of the element a_i in the stream σ. Further, let $||\mathbf{f}||_1$ denotes the ℓ_1 norm of the frequency vector $\mathbf{f}$, we have $||\mathbf{f}||_1 = m$. We present our theoretical results as follows:

Theorem 1. *For a fixed a, let X be the random variable denoting the estimate of frequency f_a obtained using Count-Min-Sketch Cormode and Muthukrishnan [2005] having (sketch) size k. Then there exists a control variate random variable Z, and the corresponding control variate coefficient $\hat{c}$ such that*

$$\mathbb{E}[X + \hat{c}(Z - \mathbb{E}[Z])] = \mathbb{E}[X] = f_a + \frac{||\mathbf{f}||_1 - f_a}{k}, \quad (5)$$

$$\mathrm{Var}(X + \hat{c}(Z - \mathbb{E}[Z])) = \mathrm{Var}(X) - \frac{(||\mathbf{f}||_1 - f_a)^2}{(n-1)k}\left(1 - \frac{1}{k}\right), \quad (6)$$

where $\mathrm{Var}(X) = \frac{||\mathbf{f}||_2^2 - f_a^2}{k}\left(1 - \frac{1}{k}\right)$ follows from Cormode and Muthukrishnan [2005] (Theorem 5).

Theorem 2. *For a fixed a, let X be the random variable denoting the estimate of frequency f_a obtained using Count-Sketch Charikar et al. [2004] having (sketch) size k. Then there exists a control variate random variable Z, and the corresponding control variate coefficient $\hat{c}$ such that*

$$\mathbb{E}[X + \hat{c}(Z - \mathbb{E}[Z])] = \mathbb{E}[X] = f_a,$$

$$\mathrm{Var}(X + \hat{c}(Z - \mathbb{E}[Z])) = \mathrm{Var}(X) - \frac{(||\mathbf{f}||_1 - f_a)^2}{(n-1)k},$$

where $\mathrm{Var}(X) = \frac{||\mathbf{f}||_2^2 - f_a^2}{k}$ follows from Charikar et al. [2004] (Theorem 6).

Comment on the overhead of our estimates: For both Count-Min-Sketch and Count-Sketch algorithms, our control variate random variable Z is independent of the actual values of the stream. Therefore, our new estimate $X + \hat{c}(Z - \mathbb{E}[Z])$ can be estimated with a small computational overhead, that is, $O(\log n)$ space and $O(n)$ time, to Count-Min-Sketch/Count-Sketch algorithm.

The Count-Min-Sketch gives a biased estimator of the frequency, and it is known to overestimate f_a, whereas Count-Sketch gives an unbiased estimate of the same. For both Count-Min-Sketch and Count-Sketch algorithms, we can exactly compute the true mean of their respective control variate random variable Z, which gives $\mathbb{E}[X + c(Z - \mathbb{E}[Z])] = \mathbb{E}[X]$. Therefore, our new estimate does not introduce any additional bias to the respective estimate of Count-Min-Sketch and Count-Sketch.

The optimum value of c, given by $\hat{c}$ in Equation (3) turns out to be

$$\hat{c} = -\frac{\mathrm{Cov}[X,Z]}{\mathrm{Var}[Z]} = -\frac{||\mathbf{f}||_1 - f_a}{n-1},$$

for both Count-Min-Sketch and Count-Sketch (see Equations (21) and 30). In practice, we choose an approximation for $\hat{c}$, where the f_a is approximated using an estimate from Count-Min-Sketch/Count-Sketch, respectively as a proxy.

The value of $||\mathbf{f}||_1$ is the length of the stream σ and its value is m are already known to us. Therefore, the overall complexity remains the same as that of Count-Min-Sketch/Count-Sketch.

Comment on the variance reduction: Note that when the original variance in Count-Min-Sketch/Count-Sketch is large, i.e., when $(||\mathbf{f}||_2^2 - f_a^2)$ is large (see Theorems 1, 2), then $(||\mathbf{f}||_1 - f_a)$ is also expected to be large. Hence we would expect bigger variance reduction in absolute terms given by $\frac{(||\mathbf{f}||_1 - f_a)^2}{(n-1)k}\left(1 - \frac{1}{k}\right)$ for Count-Min-Sketch, and $\frac{(||\mathbf{f}||_1 - f_a)^2}{(n-1)k}$ for Count-Sketch, respectively. If the value of the term $(||\mathbf{f}||_2^2 - f_a^2)$ is small, then we would expect lesser variance reduction in absolute terms. Thus when the original variance is large, our method will mitigate the problem by bigger amount and when the original variance is small, our method will only mitigate the problem by a small amount.

We further give a pictorial comparison of the theoretical variance reduction obtained by our results on the Count-Min-Sketch and Count-Sketch. Note that for a query item a both (i) variance occurred in its estimation using Count-Min Sketch/Count Sketch, (ii) and the variance reduction obtained via our methods depends only on frequency of the remaining items (see Theorems 1, 2). We, therefore, wish to understand the behaviour of variance reduction with the frequency of the remaining items. To do so, we synthetically generate a vector such that each entry is randomly sampled between $1 - 10$. We consider this as our frequency vector. We steadily increase the frequency of all but one feature (which we consider as the frequency of our query item – f_a). We note the original variance obtained via Count-Sketch and Count-Min-Sketch, and the corresponding variance reduction obtained via our methods using the expression mentioned in Theorems 1, 2. We plot it with respect to the ℓ_1 norm of the frequency vector. We summarise our results in Figure 1. The results indicate that the variance of our proposal is significantly smaller *w.r.t.* the variance of Count-Sketch and Count-Min-Sketch. Moreover, the variance of Count-Sketch and Count-Min-Sketch increases rapidly with the increase of frequency of other items, whereas our variance of our estimate somewhat remains constant.

Importance of variance reduction and its implication on the space saving: We note that the standard way to achieve the variance reduction, in the pairwise similarity estimation, is to generate several *i.i.d.* copies of the sketches (or hash values) of given pairs of data points. Needless to say that this is an expensive task. A major advantage of our approach is that it significantly reduces the variance occurred in the similarity estimation by proposing a new estimator and exploiting the available sketches. Therefore, we require generating a smaller number of *i.i.d.* copies of the sketches in order to achieve the same accuracy during the similarity estimation. We state that the sketch sizes required by our control variate estimators for Count-Min Sketch and Count-Sketch are

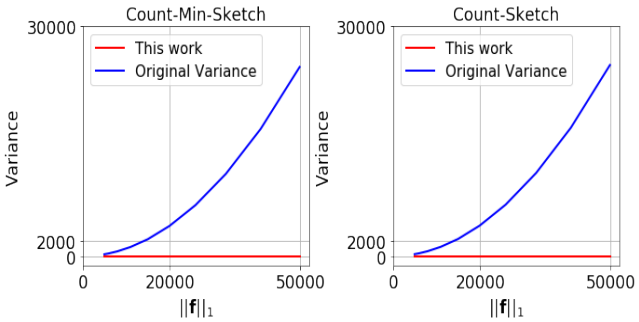

Figure 1: Comparison of the variance reduction achieved via our method *w.r.t.* Count-Min-Sketch/Count-Sketch in the estimation of the frequency of item a. X-axis corresponds to the ℓ_1 norm of frequency vectors with increase in frequency of all but item a.

much lesser than their respective vanilla estimates while offering the same concentration guarantee. We quantify it in Corollaries 7, 8 of Theorems 1, 2, respectively. For both Count-Min Sketch and Count-Sketch the ratio of the sketch size for CV estimate with the corresponding vanilla estimate is given as follows:

$$\begin{aligned}&\text{Sketch ratio}\\ &= \frac{\text{sketch size of CV estimate for CMS (resp. CS)}}{\text{sketch size of vanilla CMS (resp. CS) estimate}}.\\ &= 1 - \frac{(||\mathbf{f}||_1 - f_a)^2}{(n-1)\left(||\mathbf{f}||_2^2 - f_a^2\right)}. \qquad (7)\end{aligned}$$

We further pictorially illustrate the space saving (Figure 2) by computing it on the synthetic datasets described above. It is evident that with increase of $||\mathbf{f}||_1$, we achieve significant savings on the space required to store the sketch.

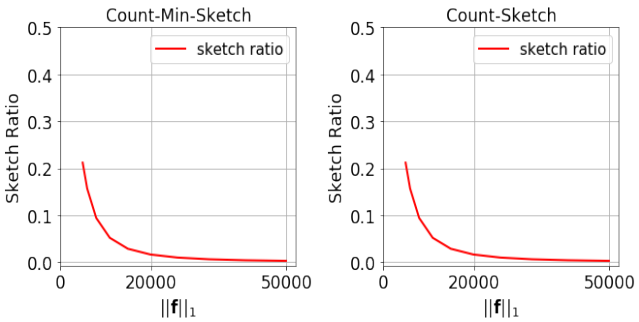

Figure 2: Comparison of the space reduction achieved via our method *w.r.t.* Count-Min-Sketch/Count-Sketch in the estimation of the frequency of item a. X-axis corresponds to the ℓ_1 norm of frequency vectors with increase in frequency of all but item a, and Y-axis corresponds to the ratio of sketch size illustrated in Equation (7). A lower value is an indication of better performance.

Known applications of control variates: Control variate

technique has been used recently for reducing the variances of the estimates obtained in several Monte-Carlo simulations. Kang *et. al.* Kang and Hooker [2017], Kang [2017] used it for improving the estimates for inner product and Euclidean distance obtained from random projection. However, to the best of our knowledge this approach has not been tried for reducing the variance of the sketching algorithm. In this work, we initiate this study.

Organisation of the paper: The rest of the paper is organized as follows: in Section 2, we state some definitions which are used in the paper, also for the sake of completeness we briefly revisit the Count-Min-Sketch Cormode and Muthukrishnan [2005], and Count-Sketch Charikar et al. [2004] algorithms and their analysis; in Section 3, we give proofs of our results stated in Theorems 1, 2; in Section 4, we complement our theoretical results with experiments on synthetic and real-world datasets; finally in Section 5, we conclude our discussion and state some potential open questions of the work.

2 BACKGROUND

Notations: We denote a stream of data points by $\sigma = \{a_1, \ldots, a_m\}$ of length m, where each element of the stream is drawn from the universe $[n] := \{1, \ldots, n\}$, that is $\forall i \in [m]$we have, $a_i \in [n]$. This data stream also implicitly defines the frequency vector over the elements of the stream $\mathbf{f} = \langle f_1, \ldots, f_n \rangle$, where f_i denotes the occurrence of the element a_i in the stream σ. Further, we denote $||\mathbf{f}||_1$ as the ℓ_1 norm of the frequency vector $\mathbf{f}$. We consider the *turnstile model* to illustrate the Count-Min-Sketch/Count-Sketch algorithms. In this model, an item $a_j \in \sigma$ arrives in the following tuple format (j, c), and upon arrival of the item, the following update operation is performed $f_j \leftarrow f_j + c$.

Definition 3 (2-Universal Hashing Cormen et al. [2001])**.** *A randomized function $h : [n] \mapsto [k]$ is 2-universal if $\forall i, j \in [n]$ with $i \neq j$, we have the following property for any $z_1, z_2 \in [k]$, we have*

$$\Pr[h(i) = z_1 \text{ and } h(j) = z_2] = \frac{1}{k^2}.$$

A simple universal hash function example would be, for random numbers a and b and a prime number $p \geq k$, compute: $h(x) = (ax + b \mod p) \mod k$.

Fact 4 (Facts from probability theory)**.** *Let X, Y, X_i, and Y_i are the random variables and a, b, a_i, and b_i are the constants. Then, we have the following:*

$$\mathbb{E}[aX] = a\mathbb{E}[X].$$
$$\text{Var}[aX] = a^2\text{Var}[X], \text{and } \text{Var}[a+X] = \text{Var}[X].$$
$$\text{Var}(\sum_{i\in[n]} X_i) = \sum_{i\in[n]} \text{Var}(X_i) + \sum_{i\neq j, i,j\in[n]} \text{Cov}[X_i, X_j].$$
$$\text{Cov}[aX, Y] = a\text{Cov}[X, Y]; \text{Cov}[a+X, Y] = \text{Cov}[X, Y].$$
$$\text{Cov}[\sum_{i\in[n]} a_i X_i, \sum_{i\in[m]} b_i Y_i] = \sum_{i\in[n]} \sum_{j\in[m]} a_i b_j \text{Cov}[X_i, Y_j].$$

2.1 REVISITING COUNT-MIN-SKETCH Cormode and Muthukrishnan [2005]

The Count-Min Sketch suggests a space efficient data structure which answers the (approximate) frequency of the query element from the stream. It is a two dimensional array $t \times k$ of counters. We discuss it in Algorithm 1.

Initialize: $C[1, \ldots, t][1, \ldots, k] \leftarrow \vec{0}$;
Choose t independent hash function $h_1, \ldots h_t : [n] \mapsto k$, each from a 2-universal family;
Process(j, c):
for $i = 1$ *to* t **do**
  $C[i][h_i(j)] \leftarrow C[i][h_i(j)] + c$;
end
Output: On query of item a, report its estimated frequency $\hat{f}_a = \min_{1\leq i\leq t} C[i][h_i(a)]$.

Algorithm 1: Count-Min Sketch Cormode and Muthukrishnan [2005]

We now give an analysis on the guarantee offered by Count-Min-Sketch. For a query item a, the counter $C[i][h_i(a)]$ gives an overestimate of the actual frequency f_a of the item a. Therefore, $f_a \leq \hat{f}_a$, where $\hat{f}_a$ is the estimate of the frequency outputted by the Count-Min-Sketch algorithm. For a fixed query item a, we analyse its estimate using one hash function say $h(.)$. Let the random variable X denote the estimate of the frequency of query item a using the hash function $h(.)$. For $j \in [n]$, let Y_j denote the indicator of the event "$h(j) = h(a)$", in particular we have, $Y_a = 1$. Also, note that j makes a contribution to the counter *iff* $Y_j = 1$, and then it add f_j to this counter. Thus,

$$\begin{aligned} X &= \sum_{j\in[n]} f_j Y_j = f_a Y_a + \sum_{j\in[n]/\{a\}} f_j Y_j \\ &= f_a + \sum_{j\in[n]/\{a\}} f_j Y_{j.} \end{aligned} \tag{8}$$

We state the guarantee on the expectation and variance of the estimate obtained from Count-Min-Sketch algorithm in Theorem 5. We defer its proof to the appendix.

Theorem 5 (Adapted from the results of Cormode and Muthukrishnan [2005])**.** *The frequency estimate of the Count-Min Sketch algorithm captured by variable X, mentioned in Equation* (8)*, has the following properties:*

$$\mathbb{E}[X] = f_a + \frac{||\mathbf{f}||_1 - f_a}{k},$$
$$\text{Var}[X] = \frac{||\mathbf{f}||_2^2 - f_a^2}{k}\left(1 - \frac{1}{k}\right)$$

Discrimilar: We adapt a similar notation and writing style as of Chakrabarti [2020] to describe Algorithms 1, 2 and their proof of correctness stated in Theorem 5, 6.

Initialize: $C[1,\ldots,t][1,\ldots,k] \leftarrow \vec{0}$;
Choose t independent random hash functions $h_1,\ldots,h_t : [n] \mapsto [k]$, each from a 2-universal family;
Choose t independent random hash functions $g_1,\ldots,g_t : [n] \mapsto \{-1,+1\}$ each from a 2-universal family;
Process(j,c):
for $i = 1$ *to* t **do**
  $C[i][h_i(j)] \leftarrow C[i][h_i(j)] + cg_i(j)$;
end
Output: On query of item a, report its estimated frequency $\hat{f}_a = \text{median}_{i\in[t]} g_i(a)C[i][h_i(a)]$.

Algorithm 2: Count-Sketch Charikar et al. [2004].

2.2 REVISITING COUNT-SKETCH

The Count-Sketch Charikar et al. [2004] suggests a space efficient data structure which answer the (approximate) frequency of the query element from the stream. Similar to Count-Min-Sketch, the Count-Sketch is a two dimensional array $t \times k$ of counters. We discuss this in Algorithm 2.

For a fixed query item a, we analyse the estimate of Count-Sketch using one pair of hash functions $h(.)$, and $g(.)$. Let the random variable X which denotes the corresponding output $\hat{f}_a$. For each item $j \in [n]$, let Y_j denote the indicator of the event $h(j) = h(a)$. Notice that an item j contributes to $C[h(a)]$ *iff* $h(j) = h(a)$, and the amount of the contribution is its frequency f_j times the random sign $g(j)$. Thus,

$$\begin{aligned} X &= g(a)\sum_{j=1}^{n} f_j g(j) Y_j. \\ &= g(a)^2 f_a Y_a + \sum_{j\in[n]/\{a\}}^{n} f_j g(a) g(j) Y_j. \qquad (9) \\ &= f_a + g(a)\sum_{j\in[n]/\{a\}}^{n} f_j g(j) Y_j. \qquad (10) \end{aligned}$$

For each $j \in [n]/\{a\}$ we have the following two equalities, which we will repeatedly use.

$$\begin{aligned} &\mathbb{E}[g(j)] = 0, \text{ and} \\ &\mathbb{E}[Y_j^2] = \mathbb{E}[Y_j] = \Pr[h(j) = h(a)] = 1/k. \qquad (11) \end{aligned}$$

Equation (11) holds as $g(.)$ is from 2-universal family and can take sign between $\{-1,+1\}$ each with probability $1/2$. Equation (11) holds since g and h are independent. We have

$$\mathbb{E}[g(j)Y_j] = \mathbb{E}[g(j)]\mathbb{E}[Y_j] = 0 \times \mathbb{E}[Y_j] = 0. \qquad (12)$$

We state the guarantee on the expectation and variance of the estimate obtained from Count-Sketch algorithm in Theorem 6. We defer its proof to the appendix.

Theorem 6 (Adapted from the results of Charikar et al. [2004])**.** *The frequency estimate of the Count-Sketch algorithm captured by variable X, mentioned in Equation* (10)*, has the following properties:*

$$\begin{aligned} \mathbb{E}[X] &= f_a, \\ \text{Var}[X] &= \frac{||\mathbf{f}||_2^2 - f_a^2}{k}. \end{aligned}$$

3 ANALYSIS

We would like to emphasize that the frequency estimation in the Count-Min-Sketch algorithm (done *via* the random variable X, Equation (8)) has been repeated t times, and the overall minimum was taken as the final estimate. Similarly, in the Count-Sketch algorithm, the frequency estimation step done *via* random variable X (Equation (9)) was repeated t times, and the median of all the estimates was taken. If we set $t = \log 1/\delta$, then the estimated frequency is well concentrated around its actual value with probability at least $1-\delta$. However, for both these algorithms in order to have a fair comparison, we perform the variance reduction analysis for one estimate only. Of course, one can repeat this step several times and can take minimum/median of the estimates for Count-Min-Sketch and Count-Sketch, respectively, and could possibly give further improvements.

3.1 PROOF OF THEOREM 1:

Proof. Recall that for Count-Min Sketch, we define the indicator random variable Y_j of the event $h(j) = h(a)$, for the query point a. We now define our control variate random variable as follows:

$$\begin{aligned} Z &= \sum_{j\in[n]} Y_j. \qquad (13) \\ &= Y_a + \sum_{j\in[n]/\{a\}} Y_j = 1 + \sum_{j\in[n]/\{a\}} Y_j. \end{aligned}$$

In the following analysis we will repeatedly use the two equalities:

$$\mathbb{E}[Y_j] = 1/k, \quad \mathbb{E}[Y_j Y_l] = 1/k^2, \quad \mathbb{E}[Y_j^2] = \mathbb{E}[Y_j].$$

We calculate the expected cost of random variable Z

$$\begin{aligned} \mathbb{E}[Z] &= \mathbb{E}[1 + \sum_{j\in[n]/\{a\}} Y_j] = 1 + \mathbb{E}[\sum_{j\in[n]/\{a\}} Y_j]. \\ &= 1 + \frac{n-1}{k}. \qquad (14) \end{aligned}$$

We calculate the variance of the random variable Z.

$$\text{Var}[Z] = \text{Var}[1 + \sum_{j\in[n]/\{a\}} Y_j] = \text{Var}[\sum_{j\in[n]/\{a\}} Y_j]. \qquad (15)$$

$$= \sum_{j\in[n]/\{a\}} \mathrm{Var}[Y_j] + \sum_{l\neq j, l, j\in[n]/\{a\}} \mathrm{Cov}[Y_j, Y_l]. \quad (16)$$

$$= \sum_{j\in[n]/\{a\}} \left(\mathbb{E}[Y_j^2] - \mathbb{E}[Y_j]^2\right) + \\ + \sum_{l\neq j, l, j\in[n]/\{a\}} \left(\mathbb{E}[Y_j Y_l] - \mathbb{E}[Y_j]\mathbb{E}[Y_l]\right).$$

$$= \sum_{j\in[n]/\{a\}} \left(\mathbb{E}[Y_j] - \mathbb{E}[Y_j]^2\right) + \\ + \sum_{l\neq j, l, j\in[n]/\{a\}} \left(\mathbb{E}[Y_j Y_l] - \mathbb{E}[Y_j]\mathbb{E}[Y_l]\right).$$

$$= \sum_{j\in[n]/\{a\}} \left(\frac{1}{k} - \frac{1}{k^2}\right) + \sum_{l\neq j, l, j\in[n]/\{a\}} \left(\frac{1}{k^2} - \frac{1}{k^2}\right).$$

$$= \frac{(n-1)}{k}\left(1 - \frac{1}{k}\right). \quad (17)$$

Equations (15) and (16) hold due to Fact 4. We now calculate the covariance between the term X and Z.

$$\mathrm{Cov}[X, Z] = \mathrm{Cov}\left(f_a + \sum_{j\in[n]/\{a\}} f_j Y_j, 1 + \sum_{j\in[n]/\{a\}} Y_j\right).$$

$$= \mathrm{Cov}\left(\sum_{j\in[n]/\{a\}} f_j Y_j, \sum_{j\in[n]/\{a\}} Y_j\right). \quad (18)$$

$$= \sum_{i\in[n]/\{a\}} \sum_{j\in[n]/\{a\}} f_i \mathrm{Cov}\left(Y_i, Y_j\right). \quad (19)$$

$$= \sum_{i\in[n]/\{a\}} \sum_{j\in[n]/\{a\}} f_i \left(\mathbb{E}[Y_i Y_j] - \mathbb{E}[Y_i]\mathbb{E}[Y_j]\right).$$

$$= \sum_{i\in[n]/\{a\}} f_i \left(\mathbb{E}[Y_i Y_i] - \mathbb{E}[Y_i]\mathbb{E}[Y_i]\right) + \\ + \sum_{i\neq j, i, j\in[n]/\{a\}} f_i \left(\mathbb{E}[Y_i Y_j] - \mathbb{E}[Y_i]\mathbb{E}[Y_j]\right).$$

$$= \sum_{i\in[n]/\{a\}} f_i \left(\mathbb{E}[Y_i] - \mathbb{E}[Y_i]^2\right) + \\ + \sum_{i\neq j, i, j\in[n]/\{a\}} f_i \left(\mathbb{E}[Y_i Y_j] - \mathbb{E}[Y_i]\mathbb{E}[Y_j]\right).$$

$$= \sum_{i\in[n]/\{a\}} f_i \left(\frac{1}{k} - \frac{1}{k^2}\right) + \sum_{i\neq j, i, j\in[n]/\{a\}} f_i \left(\frac{1}{k^2} - \frac{1}{k^2}\right).$$

$$= \frac{||\mathbf{f}||_1 - f_a}{k}\left(1 - \frac{1}{k}\right). \quad (20)$$

Equations (18) and (19) hold due to Fact 4. Equations (20) and (17) give the control variate coefficient $\hat{c}$ as follows:

$$\hat{c} = -\frac{\mathrm{Cov}[X, Z]}{\mathrm{Var}[Z]} = -\frac{||\mathbf{f}||_1 - f_a}{n-1}. \quad (21)$$

Equations (20), (17) give the variance reduction as follows:

$$\text{Variance Reduction} = \frac{\mathrm{Cov}[X, Z]^2}{\mathrm{Var}[Z]}.$$

$$= \frac{(||\mathbf{f}||_1 - f_a)^2}{(n-1)k}\left(1 - \frac{1}{k}\right). \quad (22)$$

□

In the following corollary, using Chebyshev's inequality, we show that the sketch size of our CV estimator is much smaller than that of the vanilla CMS estimator while simultaneously offering the same concentration guarantee. We illustrate a pictorial comparison on this in Figure 2.

Corollary 7. *If we set $k = 3/\varepsilon^2$ in Algorithm 1, then for a query item a with actual frequency f_a, its estimated frequency $\hat{f}_a$ outputted by the algorithm satisfies the following:*

$$\Pr\left[\left|\hat{f}_a - \left(f_a + \frac{||\mathbf{f}||_1 - f_a}{k}\right)\right| \geq \varepsilon\sqrt{||\mathbf{f}||_2^2 - f_a^2}\right] \leq \frac{1}{3}.$$

Further, if we set $k = \frac{3}{\varepsilon^2}\cdot\left(\frac{(n-1)(||\mathbf{f}||_2^2 - f_a^2) - (||\mathbf{f}||_1 - f_a)^2}{(n-1)\left(||\mathbf{f}||_2^2 - f_a^2\right)}\right)$ as the sketch size of our control variate estimate, then for the query item a, its estimated frequency $\tilde{f}_a$ by our CV estimate satisfies the following:

$$\Pr\left[\left|\tilde{f}_a - \left(f_a + \frac{||\mathbf{f}||_1 - f_a}{k}\right)\right| \geq \varepsilon\sqrt{||\mathbf{f}||_2^2 - f_a^2}\right] \leq \frac{1}{3}.$$

How to compute Z for Count-Min-Sketch: Recall that for a query item a, the CV estimate requires computing the quantity $Z = \sum_{j\in[n]} Y_j$, where Y_j is the indicator of the event $h(j) = h(a)$, and h is a 2-universal hash function $h : [n] \mapsto [k]$. As we know the universe $[n]$, we can maintain a count-array of size k that keeps a count on the number of elements $j \in [n]$ that fall in each bin $\in [k]$ under the mapping of $h(.)$. Note that this count-array, only requires the value of n (size of the universe), is independent of the data stream σ, and can be easily computed during the preprocessing. We use this count-array to compute Z: for query item $a \in [n]$ all we need to check is the count of bin $h(a) \in [k]$.

3.2 PROOF OF THEOREM 2:

Proof. Recall that for Count-Sketch, we define the indicator random variable Y_j of the event $h(j) = h(a)$, for the query point a. We now define our control variate random variable as follows:

$$Z = g(a)\sum_{j\in[n]} g(j)Y_j. \quad (23)$$

$$= g(a)^2 Y_a + \sum_{j\in[n]/\{a\}} g(j)Y_j = 1 + \sum_{j\in[n]/\{a\}} g(j)Y_j.$$

In the following analysis we repeatedly use the following:

$$g(j)^2 = 1, \quad \mathbb{E}[g(j)] = 0, \text{ and}$$
$$\mathbb{E}[Y_j^2] = \mathbb{E}[Y_j] = \Pr[h(j) = h(a)] = 1/k.$$

We calculate the expectation of the term Z

$$\mathbb{E}[Z] = \mathbb{E}[1 + \sum_{j\in[n]/\{a\}} g(j)Y_j].$$

$$= 1 + \sum_{j\in[n]/\{a\}} \mathbb{E}[g(j)Y_j].$$
$$= 1 + 0 = 1. \tag{24}$$

We calculate the variance of the random variable Z

$$\mathrm{Var}[Z] = \mathrm{Var}\left(1 + \sum_{j\in[n]/\{a\}} g(j)Y_j\right).$$
$$= \mathrm{Var}\left(\sum_{j\in[n]/\{a\}} g(j)Y_j\right).$$
$$= \sum_{j\in[n]/\{a\}} \mathrm{Var}[g(j)Y_j] + \sum_{l\neq j,l,j\in[n]/\{a\}} \mathrm{Cov}[g(j)Y_j, g(l)Y_l]. \tag{25}$$
$$= \sum_{j\in[n]/\{a\}} \left(\mathbb{E}[g(j)^2Y_j^2] - \mathbb{E}[g(j)Y_j]^2\right) +$$
$$+ \sum_{l\neq j,l,j\in[n]/\{a\}} \left(\mathbb{E}[g(j)g(l)Y_jY_l] - \mathbb{E}[g(j)Y_j]\mathbb{E}[g(l)Y_l]\right).$$
$$= \sum_{j\in[n]/\{a\}} (\mathbb{E}[Y_j] - 0) + 0 - 0.$$
$$= \frac{(n-1)}{k}. \tag{26}$$

We calculate the covariance between the random variables X and Z that is $\mathrm{Cov}[X,Z]$

$$= \mathrm{Cov}\left[\left(g(a)\sum_{j\in[n]} f_jg(j)Y_j\right), \left(g(a)\sum_{j\in[n]} g(j)Y_j\right)\right].$$
$$= \mathrm{Cov}\left[\left(f_a + g(a)\sum_{j\in[n]/\{a\}} f_jg(j)Y_j\right), \left(g(a) + \sum_{j\in[n]/\{a\}} g(j)Y_j\right)\right].$$
$$= \mathrm{Cov}\left[\left(\sum_{j\in[n]/\{a\}} f_jg(j)Y_j\right), \left(\sum_{j\in[n]/\{a\}} g(j)Y_j\right)\right]. \tag{27}$$
$$= \sum_{i,j\in[n]/\{a\}} f_i\mathrm{Cov}\left(g(i)Y_i, g(j)Y_j\right). \tag{28}$$
$$= \sum_{i,j\in[n]/\{a\}} f_i\left(\mathbb{E}[g(i)g(j)Y_iY_j] - \mathbb{E}[g(i)Y_i]\mathbb{E}[g(j)Y_j]\right).$$
$$= \sum_{i\in[n]/\{a\}} f_i\left(\mathbb{E}[g(i)g(i)Y_iY_i] - \mathbb{E}[g(i)Y_i]\mathbb{E}[g(i)Y_i]\right) +$$
$$+ \sum_{i\neq j,i,j\in[n]/\{a\}} f_i\left(\mathbb{E}[g(i)g(j)Y_iY_j] - \mathbb{E}[g(i)Y_i]\mathbb{E}[g(j)Y_j]\right).$$
$$= \sum_{i\in[n]/\{a\}} f_i\left(\mathbb{E}[Y_i] - \mathbb{E}[g(i)Y_i]^2\right) +$$
$$+ \sum_{i\neq j,i,j\in[n]/\{a\}} f_i\left(\mathbb{E}[g(i)g(j)Y_iY_j] - \mathbb{E}[g(i)Y_i]\mathbb{E}[g(j)Y_j]\right).$$
$$= \sum_{i\in[n]/\{a\}} f_i\mathbb{E}[Y_i] - 0 + 0 - 0.$$
$$= \frac{||\mathbf{f}||_1 - f_a}{k}. \tag{29}$$

Equation (27) and (28) hold due to Fact 4. Equations (29), and (26) give the control variate coefficient $\hat{c}$ is

$$\hat{c} = -\frac{\mathrm{Cov}[X,Z]}{\mathrm{Var}[Z]} = -\frac{||\mathbf{f}||_1 - f_a}{n-1}. \tag{30}$$

Equations (29), and (26) give the variance reduction term is the following

$$\text{Variance Reduction} = \frac{\mathrm{Cov}[X,Z]^2}{\mathrm{Var}[Z]} = \frac{(||\mathbf{f}||_1 - f_a)^2}{(n-1)k}. \tag{31}$$

□

In the following corollary, using Chebyshev's inequality, we show that the sketch size of our CV estimator is much smaller than that of the vanilla CS estimator while simultaneously offering the same concentration guarantee. We illustrate a pictorial comparison on this in Figure 2.

Corollary 8. *If we set $k = 3/\varepsilon^2$ in Algorithm 2, then for a query item a with actual frequency f_a, its estimated frequency $\hat{f}_a$ outputted by the algorithm satisfies the following:*

$$\Pr\left[|\hat{f}_a - f_a| \geq \varepsilon\sqrt{||\mathbf{f}||_2^2 - f_a^2}\right] \leq \frac{1}{3}.$$

Further, if we set $k = \frac{3}{\varepsilon^2}\cdot\left(\frac{(n-1)(||\mathbf{f}||_2^2 - f_a^2) - (||\mathbf{f}||_1 - f_a)^2}{(n-1)(||\mathbf{f}||_2^2 - f_a^2)}\right)$ in the sketch size of our control variate estimate, then for the query item a, its estimated frequency $\tilde{f}_a$ by our CV estimate satisfies the following:

$$\Pr\left[|\tilde{f}_a - f_a| \geq \varepsilon\sqrt{||\mathbf{f}||_2^2 - f_a^2}\right] \leq \frac{1}{3}.$$

How to compute Z for Count-Sketch: Recall that for a query item a, our CV estimate requires computing the quantity $Z = g(a)\sum_{j\in[n]} g(j)Y_j$, where Y_j is the indicator of the event $h(j) = h(a)$, h and g are 2-universal hash functions such that $h : [n] \mapsto [k]$ and $g : [n] \mapsto \{-1, +1\}$. Similar to CMS, we can maintain a count-array of size k that keeps a count on the number of elements $j \in [n]$ that fall in each bin $\in [k]$. Note that this count-array, only requires the value of n, is independent of the data stream σ, and can be easily computed during the preprocessing. For a query item a, we can use this count-array to compute Z: all we need to check is the count of bin $h(a) \in [k]$ and multiply it with $g(a)$.

How to choose a good control-variate function: For higher variance reduction one should choose the control variate such that it has a low variance, and simultaneously also has high co-variance with the random variable which is used to measure the estimate. Further, in order to ensure the applicability of the approach, the control variate, its expected value, variance, and co-variance with original random variable should be easily computed/estimated from the given dataset. We don't claim that the control variates used in this

work are the best possible for the task. It is conceivable that one might be able to come up with a better control variate for the purpose of frequency estimation. Our work leaves the question of investigating the best control variate for frequency estimation problems as an intriguing open question.

4 EXPERIMENTS

Datasets: We use two datasets for our experiments – `synthetic`, and `password-frequency` dataset. In the synthetic dataset, we generate a stream of 100000 integers such that each number is randomly sampled between 0 and 100. We consider each number as an item of the stream. The `password-frequency` dataset consists of three entries – the password, its hashcode, and the frequency of the password. For each password, we generate a unique integer hashcode using the classical `Rabin fingerprint` string hashing algorithm Rabin [1981], and we consider this hashcode as an item of the stream. The dataset consists of 683039 distinct passwords. The link of the dataset is available here rob [a], and we combine all the files into one and make it available here rob [b].

Methodology: Let X be the random variable denoting the estimate obtained from the Count-Min-Sketch and Count-Sketch algorithms. Then the updated estimate proposed by our algorithm is $X + c(Z - \mathbb{E}[Z])$, where c is the control variate correction, and Z is the control variate random variable. The optimum value of c, for both Count-Min-Sketch and Count-Sketch given by $\hat{c}$ turns out to be

$$\hat{c} = -\frac{\text{Cov}[X,Z]}{\text{Var}[Z]} = -\frac{||\mathbf{f}||_1 - f_a}{n-1}$$

(see Equations (21) and (30)). For Count-Min-Sketch recall that control variate random variable $Z = \sum_{j\in[n]} Y_j$ (see Equation (13)). This value can be easily computed by examining the hash function $h(.)$. Further, the expected value of Z is $\mathbb{E}[Z] = 1 + (n-1)/k$ from Equation (14).

For Count-Sketch the control variate random variable $Z = g(a)\sum_{j\in[n]} g(j)Y_j$ from Equation (23). This value can be easily computed by examining the hash functions $h(.)$ and $g(.)$. Further, the expected value of Z is $\mathbb{E}[Z] = 1$ due to Equation (24).

For both Count-Min-Sketch and Count-Sketch our new estimate is given by the following expression our estimate $= X + \hat{c}(Z - \mathbb{E}[Z])$. We can compute the respective values of Z and $\mathbb{E}[Z]$ of Count-Min-Sketch and Count-Sketch mentioned above. Finally, in the above estimate we require the value of $||\mathbf{f}||_1$ and f_a. The value of $||\mathbf{f}||_1$ is the length of the stream σ and its value is m. In order to compute the f_a , we use the estimate obtained by the vanilla version of Count-Min-Sketch and Count-Sketch, respectively, as a proxy.

Evaluation Metric: We evaluate the performance of our approach with *vanilla* Count-Min-Sketch and Count-Sketch algorithms on root-mean-square-error (RMSE) measure. To do so, for the given data stream, we first create its sketches using Count-Min-Sketch and Count-Sketch methods. For every item in the stream, we estimate its frequency using vanilla Count-Min-Sketch, Count-Sketch, and our methods. We compute the square of the difference between the estimated frequency and the ground truth frequency. We repeat this for all the distinct items available in the stream, add all such numbers obtained from squared difference, compute their mean, and then compute the square root. We report this as RMSE. Note that lower RMSE indicates that our estimator closely approximates the ground truth frequency. We repeat this for several values of k (size of sketch) and report the corresponding values of RMSE.

Insight: We run our experiments on the datasets and summarise our results in Figures 3 and 4 for Count-Min-Sketch and Count-Sketch, respectively. In comparison with both methods, for every value of k, our methods report lower RMSE – that is closer to the ground-truth frequency. The running time of our method is almost the same as the corresponding baseline methods. We observed this pattern for both the datasets.

5 CONCLUSION

We consider the problem of frequency estimation in a large stream data, and show variance reduction in the estimates of the classical algorithms – Count-Min-Sketch Cormode and Muthukrishnan [2005] and Count-Sketch Charikar et al. [2004] – for the task. Our technique relies on the classical Control-Variate trick Lavenberg and Welch [1981] used for reducing variance in Monte-Carlo simulation. We present a theoretical analysis of our proposal and complement it with experiments on synthetic and real-world datasets. We notice that our estimate outputs lower variance as compared to their respective variance, at the cost of little computational overhead. Our work leaves the possibility of several open questions and research directions – improving the variance reduction shown in this work by choosing a better control variate random variable, how to choose a good control variate estimator for the given task, exploring the applicability of control variate trick for reducing the variance of other fundamental randomized algorithms. Recently, Hsu *et. al* Hsu et al. [2019] suggests learning based frequency estimators. They propose a new class of algorithms that learn distribution of the items in the data stream and use them to improve its frequency estimates. An interesting research direction is to combine this with our method and compe up with improved frequency estimators.

Finally, as Count-Min-Sketch Cormode and Muthukrishnan [2005] and Count-Sketch Charikar et al. [2004] have been widely used in a variety of applications such as *kernel den-*

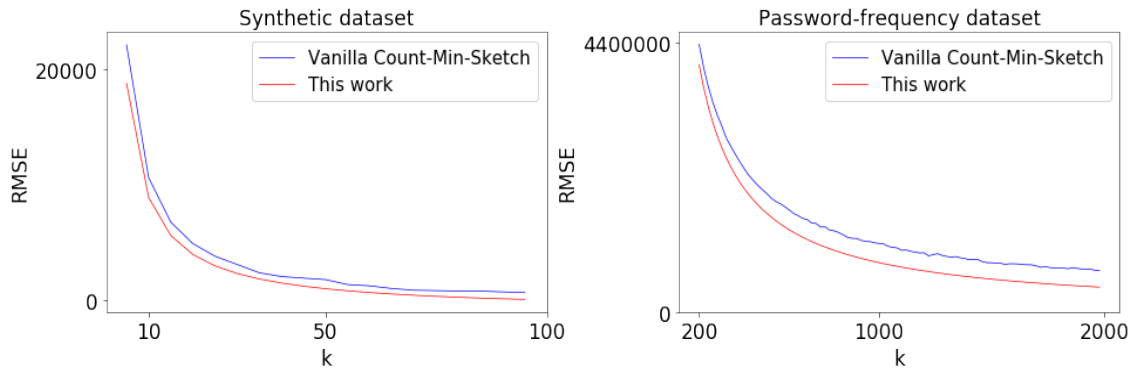

Figure 3: Comparison between *vanilla* Count-Min-Sketch and our estimate obtained *via* Control variate correction on RMSE measure for Synthetic and Password-frequency datasets. A lower value of RMSE is an indication of better performance.

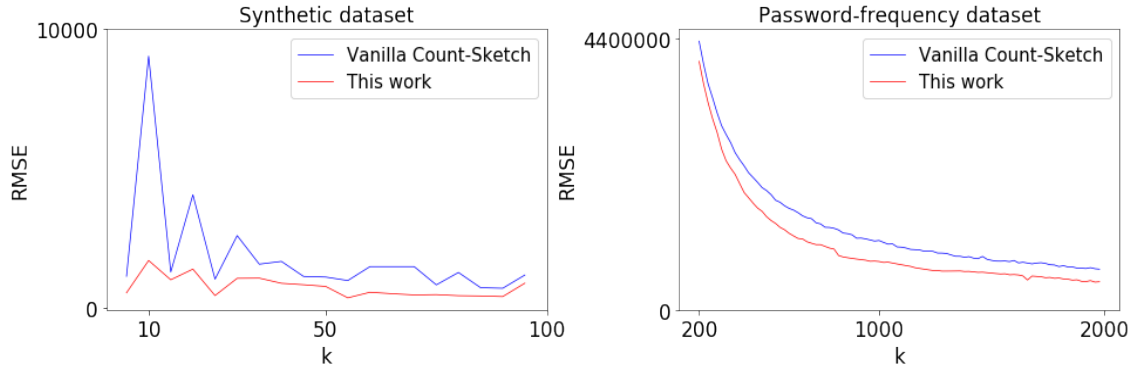

Figure 4: Comparison between *vanilla* Count-Sketch and our estimate obtained *via* Control variate correction on RMSE measure for Synthetic and Password-frequency datasets.

sity estimation Coleman and Shrivastava [2020], *compressing gradient optimizers* Spring et al. [2019], *extreme classification* Talukdar and Cohen [2014], Tai et al. [2018], *low-rank approximation* Clarkson and Woodruff [2013], *compressed matrix multiplication* Pagh [2013], *sketching polynomial kernel* Pham and Pagh [2013], *large scale feature selection* Aghazadeh et al. [2018], *anomaly detection* Luo and Shrivastava [2018], *sparse recovery* Gilbert and Indyk [2010], *clustering* Indyk [2004], *computing synopsis of a massive dataset* Cormode et al. [2012] (also see references therein, and CMS) to name a few. We hope that our proposal can potentially benefit these applications by providing a more accurate estimate of the frequency count.

References

`https://sites.google.com/site/countminsketch/`.

`https://github.com/robinske/password-data`, a.

`https://tinyurl.com/y8vu8dkz`, b.

Amirali Aghazadeh, Ryan Spring, Daniel LeJeune, Gautam Dasarathy, Anshumali Shrivastava, and Richard G. Baraniuk. MISSION: ultra large-scale feature selection using count-sketches. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pages 80–88, 2018. URL `http://proceedings.mlr.press/v80/aghazadeh18a.html`.

Amit Chakrabarti. Data stream algorithms lecture notes, July 2020. URL `https://www.cs.dartmouth.edu/~ac/Teach/data-streams-lecnotes.pdf`.

Moses Charikar, Kevin C. Chen, and Martin Farach-Colton. Finding frequent items in data streams. *Theor. Comput. Sci.*, 312(1):3–15, 2004. doi: 10.1016/S0304-3975(03)00400-6. URL `https://doi.org/10.1016/S0304-3975(03)00400-6`.

Kenneth L. Clarkson and David P. Woodruff. Low rank approximation and regression in input sparsity time. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing*

Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013, pages 81–90. ACM, 2013. doi: 10.1145/2488608.2488620. URL https://doi.org/10.1145/2488608.2488620.

Benjamin Coleman and Anshumali Shrivastava. Sublinear RACE sketches for approximate kernel density estimation on streaming data. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 1739–1749, 2020. doi: 10.1145/3366423.3380244. URL https://doi.org/10.1145/3366423.3380244.

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. The MIT Press, 2 edition, 2001.

Graham Cormode and S. Muthukrishnan. An improved data stream summary: the count-min sketch and its applications. *J. Algorithms*, 55(1):58–75, 2005. doi: 10.1016/j.jalgor.2003.12.001. URL https://doi.org/10.1016/j.jalgor.2003.12.001.

Graham Cormode, Minos N. Garofalakis, Peter J. Haas, and Chris Jermaine. Synopses for massive data: Samples, histograms, wavelets, sketches. *Found. Trends Databases*, 4(1-3):1–294, 2012. doi: 10.1561/1900000004. URL https://doi.org/10.1561/1900000004.

R A Leo Elworth, Qi Wang, Pavan K Kota, C J Barberan, Benjamin Coleman, Advait Balaji, Gaurav Gupta, Richard G Baraniuk, Anshumali Shrivastava, and Todd J Treangen. To Petabytes and beyond: recent advances in probabilistic and signal processing algorithms and their application to metagenomics. *Nucleic Acids Research*, 48(10):5217–5234, 04 2020. ISSN 0305-1048. doi: 10.1093/nar/gkaa265. URL https://doi.org/10.1093/nar/gkaa265.

Cristian Estan and George Varghese. New directions in traffic measurement and accounting. In *Proceedings of the ACM SIGCOMM 2002 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, August 19-23, 2002, Pittsburgh, PA, USA*, pages 323–336, 2002. doi: 10.1145/633025.633056. URL https://doi.org/10.1145/633025.633056.

Anna C. Gilbert and Piotr Indyk. Sparse recovery using sparse matrices. *Proc. IEEE*, 98(6):937–947, 2010. doi: 10.1109/JPROC.2010.2045092. URL https://doi.org/10.1109/JPROC.2010.2045092.

Chen-Yu Hsu, Piotr Indyk, Dina Katabi, and Ali Vakilian. Learning-based frequency estimation algorithms. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019. URL https://openreview.net/forum?id=r1lohoCqY7.

Piotr Indyk. Algorithms for dynamic geometric problems over data streams. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, June 13-16, 2004*, pages 373–380, 2004. doi: 10.1145/1007352.1007413. URL https://doi.org/10.1145/1007352.1007413.

Keegan Kang. Using the multivariate normal to improve random projections. In *Intelligent Data Engineering and Automated Learning - IDEAL 2017 - 18th International Conference, Guilin, China, October 30 - November 1, 2017, Proceedings*, pages 397–405, 2017. doi: 10.1007/978-3-319-68935-7_43. URL https://doi.org/10.1007/978-3-319-68935-7_43.

Keegan Kang and Giles Hooker. Random projections with control variates. In *Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods, ICPRAM 2017, Porto, Portugal, February 24-26, 2017*, pages 138–147, 2017. doi: 10.5220/0006188801380147. URL https://doi.org/10.5220/0006188801380147.

S. Lavenberg and P. Welch. A perspective on the use of control variables to increase the efficiency of monte carlo simulations. *Management Science*, 27:322–335, 03 1981. doi: 10.1287/mnsc.27.3.322.

Chen Luo and Anshumali Shrivastava. Arrays of (locality-sensitive) count estimators (ACE): anomaly detection on the edge. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, pages 1439–1448, 2018. doi: 10.1145/3178876.3186056. URL https://doi.org/10.1145/3178876.3186056.

Samuel Madden and Michael Franklin. Fjording the stream: An architecture for queries over streaming sensor data. pages 555 – 566, 02 2002. ISBN 0-7695-1531-2. doi: 10.1109/ICDE.2002.994774.

S. Muthukrishnan. Data streams: Algorithms and applications. *Foundations and Trends in Theoretical Computer Science*, 1(2), 2005. doi: 10.1561/0400000002. URL https://doi.org/10.1561/0400000002.

Rasmus Pagh. Compressed matrix multiplication. *ACM Trans. Comput. Theory*, 5(3):9:1–9:17, 2013. doi: 10.1145/2493252.2493254. URL https://doi.org/10.1145/2493252.2493254.

Ninh Pham and Rasmus Pagh. Fast and scalable polynomial kernels via explicit feature maps. In Inderjit S. Dhillon, Yehuda Koren, Rayid Ghani, Ted E. Senator, Paul Bradley, Rajesh Parekh, Jingrui He, Robert L. Grossman, and Ramasamy Uthurusamy, editors, *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago,*

IL, USA, August 11-14, 2013, pages 239–247. ACM, 2013. doi: 10.1145/2487575.2487591. URL https://doi.org/10.1145/2487575.2487591.

Michael O. Rabin. Fingerprinting by random polynomials. 1981.

Ryan Spring, Anastasios Kyrillidis, Vijai Mohan, and Anshumali Shrivastava. Compressing gradient optimizers via count-sketches. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 5946–5955, 2019. URL http://proceedings.mlr.press/v97/spring19a.html.

Kai Sheng Tai, Vatsal Sharan, Peter Bailis, and Gregory Valiant. Sketching linear classifiers over data streams. In *Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018, Houston, TX, USA, June 10-15, 2018*, pages 757–772, 2018. doi: 10.1145/3183713.3196930. URL https://doi.org/10.1145/3183713.3196930.

Partha Pratim Talukdar and William W. Cohen. Scaling graph-based semi supervised learning to large number of labels using count-min sketch. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics, AISTATS 2014, Reykjavik, Iceland, April 22-25, 2014*, pages 940–947, 2014. URL http://proceedings.mlr.press/v33/talukdar14.html.

Yunyue Zhu and Dennis E. Shasha. Statstream: Statistical monitoring of thousands of data streams in real time. In *Proceedings of 28th International Conference on Very Large Data Bases, VLDB 2002, Hong Kong, August 20-23, 2002*, pages 358–369. Morgan Kaufmann, 2002. doi: 10.1016/B978-155860869-6/50039-1. URL http://www.vldb.org/conf/2002/S10P04.pdf.