

# Notes on Frequency Estimation in Data Streams

In (one of) the data streaming model(s), the data is a sequence of arrivals  $a_1, a_2, \dots, a_m$  of the form  $a_j = (i, v)$  where  $i$  is the identity of the item and belongs to the domain  $\{1, \dots, n\}$ , and  $v$  is the change in the frequency of the item: if  $v \geq 1$  then the meaning is  $v$  additions of item  $i$  and if  $v \leq -1$  then the meaning is  $v$  deletions of item  $i$ . The goal is to compute some function while using space that is sublinear in the length of the stream. This is relevant both when data is literally obtained as a long stream of signals, where the stream is too long to keep in memory, and when the data resides on some external device and reading it in one pass is much more efficient than allowing random access. A natural special case is that  $v = +1$  for every elements. In this case the stream is simply a sequence of items (with repetitions)  $a_j = i$  for  $i \in \{1, \dots, n\}$ .

One of the first problems that was studied in this model (with the special case of single additions), is computing frequency moments. Namely, let  $m_i = |\{j : a_j = i\}|$  denote the number of occurrences of  $i$  in the stream. Then for each  $k \geq 0$  we define

$$F_k = \sum_{i=1}^n (m_i)^k. \quad (1)$$

In particular,  $F_1$  equals  $m$ , the *length* of the sequence,  $F_0$  is the number of *distinct* elements appearing in the sequence (since if  $m_i > 0$  then  $m_i^0 = 1$  and if  $m_i = 0$  then  $m_i^0 = 0$ ), and  $F_2$  is the *repeat rate* or *Gini's index of homogeneity* needed in order to compute the *surprise index* of the sequence. Finally, for  $k = \infty$  we define

$$F_\infty^* = \max_{1 \leq i \leq n} m_i \quad (2)$$

Given an approximation parameter  $\epsilon$  and a security parameter  $\delta$ , the algorithm should compute an estimate  $\hat{F}_k$  such that the probability that  $|\hat{F}_k - F_k| > \epsilon F_k$  is at most  $\delta$ .

What is known?

1. There is a lower bound of  $n^{1-\frac{2}{k}}$  (for constant  $\epsilon$  and  $\delta$ ), which in particular means that for  $k \geq 3$  the lower bound is of the form  $n^\alpha$  for constant  $\alpha$  (that approaches 1 when  $k$  increases).
2. There is a (recent) upper bound whose dependence on  $n$  is  $\tilde{O}\left(n^{1-\frac{2}{k}}\right)$ , so that it roughly matches the lower bound (the exact expression is  $O\left(\frac{k^2 \log(1/\delta)}{\epsilon^{2+4/k}} n^{1-\frac{2}{k}} \log^2 m(\log m + \log n)\right)$ ).
3. For the special case of  $k = 1$ , clearly the exact value of  $F_k = m$  can be computed using space  $\log m$ . To get an estimate,  $O(\log \log m + \log(1/\epsilon))$  bits suffice.
4. For the special case of  $k = 0$  it is possible to compute an estimate that is within a factor  $1/c$  and a factor  $c$  of  $F_0$  with probability at least  $1 - 2/c$ , where  $c > 2$ , using  $O(\log n)$  bits.

5. For the special case of  $k = 2$  it suffices to use  $O\left(\frac{\log(1/\delta)}{\epsilon^2}(\log n + \log m)\right)$ .
6. Estimating  $F_\infty^*$  requires space  $\Omega(n)$  for  $m = O(n)$  and constant  $\epsilon$  and  $\delta$ .
7. Randomness is crucial: for  $k \neq 1$ , every algorithm that computes an estimate of  $F_k$  with constant  $\epsilon$  must use  $\Omega(n)$  space.

We shall discuss the original result of Alon et. al. whose dependence on  $n$  is  $\tilde{O}\left(n^{1-\frac{1}{k}}\right)$  (to be precise:  $O\left(\frac{k \log(1/\delta)}{\epsilon^2} n^{1-\frac{1}{k}}(\log n + \log m)\right)$ ). If time permits we will talk about some of the special cases.

Assume first that the length of the sequence,  $m$ , is known in advance. This assumption is removed later. Let  $s_1 = \frac{8}{\epsilon^2} k n^{1-\frac{1}{k}}$  and  $s_2 = 2 \log(1/\delta)$ . The algorithm computes  $s_2$  random variables,  $Y_1, \dots, Y_{s_2}$  and outputs their median (this is a standard technique to go from a constant probability of deviating by more than some allowed deviation, to only an  $\delta$  probability that this event occurs so the interesting part is in defining and analyzing the behavior of the  $Y_t$ 's).

Each  $Y_t$  is the average of  $s_1$  random variables,  $X_{t,j}$  where  $1 \leq j \leq s_1$ . The  $X_{t,j}$ 's are independent, identically distributed random variables. In order to explain how each  $X_{t,j} = X$  is distributed, we introduce some notation. For each  $p \in \{1, \dots, m\}$ , let

$$r(p) = \left| \{q : q \geq p, a_q = a_p\} \right| \quad (3)$$

denote the number of occurrences of  $\ell = a_p$  among the elements in the sequence that follow  $a_p$ , including  $a_p$  (so that  $r(p) \geq 1$ ). Next define

$$R_k(p) = m \cdot \left( (r(p))^k - (r(p) - 1)^k \right) \quad (4)$$

Each variable  $X_{t,j} = X$  is determined (independently) by selecting an index  $p \in \{1, \dots, m\}$  uniformly at random and letting  $X = R_k(p)$ . Note that in order to compute  $r(p)$  and hence  $X = R_k(p)$  it suffices to use  $\log m$  bits to select  $p$  and count up to  $p$ , and then it suffices to maintain the  $\log n$  bits representing  $a_p$  and the  $\log m$  bits representing  $r(p)$  and the  $\log m$  bits representing  $R_k(p)$ . By definition of  $X$  (recall that  $m_i = \{j : a_j = i\}$ ),

$$\text{Exp}[X] = \frac{1}{m} \cdot \sum_{j=1}^m R_k(j) \quad (5)$$

$$= \sum_{j=1}^m \left( (r(j))^k - (r(j) - 1)^k \right) \quad (6)$$

$$= \sum_{i=1}^n \left( ((m_i)^k - (m_i - 1)^k) + ((m_i - 1)^k - (m_i - 2)^k) + \dots + (2^k - 1^k) + (1^k - 0^k) \right) \quad (7)$$

$$= \sum_{i=1}^k (m_i)^k = F_k. \quad (8)$$

Thus we have an unbiased estimator of  $F_k$ . What remains to be done is to bound the deviation of the average of the  $X_{t,j}$ 's from this correct expected value. (The  $X_{t,j}$ 's are independent, so we could

apply Chernoff. However, their range is very big so we wouldn't get a very good bound.) To this end we bound the variance  $\text{Var}[X] = \text{Exp}[X^2] - \text{Exp}^2[X]$  and apply Chebishev:

$$\Pr[|X - \text{Exp}[X]| \geq t \cdot \text{Var}^{1/2}[X]] \leq \frac{1}{t^2}$$

so that

$$\Pr[|X - \text{Exp}[X]| \geq T] \leq \frac{\text{Var}[X]}{T^2}$$

In order to bound  $\text{Exp}[X^2]$  we shall use the following inequality, which holds for any pair of numbers  $a, b$  such that  $a > b > 0$ :

$$a^k - b^k = (a - b)(a^{k-1} + a^{k-2}b + \dots + ab^{k-2} + b^{k-1}) \quad (9)$$

$$\leq (a - b)ka^{k-1} \quad (10)$$

(You may be familiar with the special case of  $(a^2 - b^2) = (a - b)(a + b)$ .) We use this inequality with  $a = b + 1$ , so that  $a^k - (a - 1)^k \leq ka^{k-1}$ , and get:

$$\begin{aligned} \text{Exp}[X^2] &= \frac{1}{m} \cdot \sum_{i=1}^n \sum_{r=1}^{m_i} \left( m \cdot (r^k - (r-1)^k) \right)^2 \\ &\leq m \cdot \sum_{i=1}^n \sum_{r=1}^{m_i} k \cdot r^{k-1} \cdot ((r^k - (r-1)^k)^k) \\ &= m \cdot k \cdot \sum_{i=1}^n \left( (m_i)^{2k-1} - (m_i)^{k-1}(m_i - 1)^k + (m_i - 1)^{2k-1} - (m_i - 1)^{k-1}(m_i - 2)^k + \dots + 2^{2k-1} \right) \\ &\leq m \cdot k \cdot \sum_{i=1}^n m_i^{2k-1} \\ &= m \cdot k \cdot F_{2k-1} = k \cdot F_1 \cdot F_{2k-1} \end{aligned}$$

It can be shown (and is given as an exercise) that

$$F_1 \cdot F_{2k-1} \leq n^{1-1/k} \cdot (F_k)^2 \quad (16)$$

where we have used the inequality  $(\frac{1}{n} \sum_{i=1}^n m_i)^k \leq \frac{1}{n} \sum_{i=1}^n m_i^k$ . Therefore,

$$\text{Var}[X] \leq \text{Exp}[X^2] \leq k \cdot F_1 \cdot F_{2k-1} \leq k \cdot n^{1-1/k} \cdot F_k^2 \quad (17)$$

and so

$$\text{Var}[Y_t] = \text{Var} \left[ \frac{1}{s_1} \sum_{j=1}^{s_1} X_{t,j} \right] = \frac{1}{s_1} \text{Var}[X] \leq \frac{k \cdot n^{1-1/k} \cdot F_k^2}{s_1} \quad (18)$$

whereas

$$\text{Exp}[Y_t] = \text{Exp} \left[ \frac{1}{s_1} \sum_{j=1}^{s_1} X_{t,j} \right] = \text{Exp}[X] = F_k \quad (19)$$

By Chebyshev's inequality

$$\Pr\left[|Y_t - F_k| > \epsilon F_k\right] \leq \frac{\text{Var}[Y_t]}{\epsilon^2 F_k^2} \leq \frac{k \cdot n^{1-1/k} \cdot F_k^2}{s_1 \cdot \epsilon^2 F_k^2} \quad (20)$$

By our choice of  $s_1 = \frac{8}{\epsilon^2} k n^{1-1/k}$  this is at most  $\frac{1}{8}$ . As mentioned before, a standard analysis transform the constant probability of small deviation of the  $Y_t$ 's to a high probability of small deviation of their median (given as an exercise).

**DEALING WITH AN UNKNOWN  $m$ .** In this case we start computing the random variable  $X$  with the assumption that  $m = 1$ , so that necessarily  $a_p = a_1$  (and we get that  $r(p) = 1$  and  $X = 1 \cdot (1^k - 0^k) = 1$ ). If indeed  $m = 1$  the process ends (note that if  $m = 1$  then  $F_k = 1$  for every  $k$ ). Otherwise, the value of  $m$  is updated to 2, and  $p = 1$  is replaced by  $p = 2$  with probability  $1/2$ . In either case,  $r(p)$  is modified accordingly. In general, after viewing the first  $t - 1$  items, there is a current choice of  $p_{t-1}$  and a corresponding value of  $r(p_{t-1})$ . If a new item arrives, the "belief" for  $m$  is changed to  $t$  and  $p_t$  is set to  $t$  with probability  $1/t$  and remains  $p_{t-1}$  with probability  $1 - 1/t$ . In the former case we have that  $r(p_t) = 1$ , and in the latter case  $r(p_t)$  is  $r(p_{t-1}) + 1$ , if  $a_t = a_{p_t}$ , and is  $r(p_{t-1})$  otherwise. As in the case that  $m$  is known, the algorithm only needs to remember  $a_{p_t}$  and  $r(p_t)$  at each step, at a cost of  $O(\log n + \log m)$  bits, and flipping a coin with bias  $1/m$  takes  $O(\log m)$  bits as well.

**ON THE RELATION BETWEEN  $m$  AND  $n$ .** If  $m = \text{poly}(n)$  then the factor of  $(\log n + \log m)$  is simply  $\log n$ . When  $m$  is very large then instead of computing  $r(p)$  exactly, we can estimate it using  $\log \log m + \log(1/\epsilon)$  bits.

## Improved Estimation of $F_2$

If we plug in  $k = 2$  in the aforementioned expression, we get a dependence on  $n$  that grows like  $\tilde{O}(\sqrt{n})$ . We next show how to get an estimate using only  $O\left(\frac{\log(1/\delta)}{\epsilon^2}(\log n + \log m)\right)$  memory bits.

We set  $s_2 = 2 \log(1/\delta)$  as before and  $s_1 = \frac{16}{\epsilon^2}$ . Here too the output is the median of  $s_1$  random variables  $Y_1, \dots, Y_{s_1}$ , where each  $Y_t$  is the average of  $X_{t,j}$  for  $j = 1, \dots, s_2$ . Each  $X_{i,j} = X$  is computed as follows.

A central idea is using a set  $V = \{v_1, \dots, v_h\}$  of vectors of length  $n$  with  $+1, -1$  entries that are *four-wise* independent. That is, for every four distinct coordinates,  $1 \leq i_1 < i_2 < i_3 < i_4 \leq n$ , and for every choice of  $\gamma_1, \dots, \gamma_4 \in \{-1, +1\}$ , exactly a  $(1/16)$ -fraction of the vectors in  $V$  have  $\gamma_j$  in their  $i_j$  coordinate for every  $j = 1, \dots, 4$ . (Note that 4-wise independence implies that for each coordinate  $i$ , half of the vectors in  $V$  have  $+1$  in the  $i$ 'th coordinate and half have  $-1$ , and it implies  $s$ -wise independence for  $s = 2$  and  $s = 3$ .) Such sets, of size only  $h = O(n^2)$ , not only exist but it is possible to compute each particular coordinate of any  $v_p$  of our choice using  $O(\log n)$  space.

To compute  $X$ , we first select  $1 \leq p \leq h$  uniformly at random (this requires  $O(\log n)$  bits of space). This determines  $v_p = (\beta_1, \dots, \beta_n)$  (where we will compute the coordinates of  $v_p$  when we need them). Let  $Z = \sum_{i=1}^n \beta_i \cdot m_i$ . Computing  $Z$  can be done in one pass using  $O(\log n + \log m)$  space: Initially,  $Z = 0$ . For each  $a_j$ ,  $j = 1, \dots, m$ , if  $\beta_{a_j} = +1$  then  $Z$  is incremented by 1, and if  $\beta_{a_j} = -1$  then it is decremented by 1. To compute each  $\beta_{a_j}$  it takes  $O(\log n)$  space, and to maintain  $Z$  it takes  $O(\log m)$  space. When the sequence terminates, we set  $X = Z^2$ .

As in the proof for general  $k$  we next compute  $\text{Exp}[X]$  and  $\text{Var}[X]$ . Before doing so, we make a few observations that follow from the fact that each  $\beta_i \in \{-1, +1\}$  and the 4-wise independence:

1. For every  $i$ ,  $\beta_i^2 = \beta_i^4 = 1$ , while  $\beta_i^3 = \beta_i$ .
2. For every  $i, j$

$$\text{Exp}[\beta_i \cdot \beta_j] = \frac{1}{4}(+1 \cdot +1) + \frac{1}{4}(+1 \cdot -1) + \frac{1}{4}(-1 \cdot +1) + \frac{1}{4}(-1 \cdot -1) = 0$$

3. Similarly, for every  $i \neq j \neq k$ ,  $\text{Exp}[\beta_i \beta_j \beta_k] = 0$  and for every  $i \neq j \neq k \neq \ell$ ,  $\text{Exp}[\beta_i \beta_j \beta_k \beta_\ell] = 0$ .

Using the first two properties:

$$\text{Exp}[X] = \text{Exp}[Z^2] = \text{Exp} \left[ \left( \sum_{i=1}^n \beta_i m_i \right)^2 \right] \quad (21)$$

$$= \text{Exp} \left[ \sum_i \sum_j \beta_i \beta_j m_i m_j \right] \quad (22)$$

$$= \sum_i (m_i)^2 \text{Exp}[\beta_i^2] + \sum_i \sum_{j \neq i} m_i m_j \text{Exp}[\beta_i \cdot \beta_j] \quad (23)$$

$$= \sum_i (m_i)^2 = F_2 \quad (24)$$

Similarly (though a bit more tediously...)

$$\text{Exp}[X^2] = \text{Exp} \left[ \left( \sum_{i=1}^n \beta_i m_i \right)^4 \right] \quad (25)$$

$$= \sum_i (m_i)^4 \text{Exp}[\beta_i^4] + 4 \sum_{i \neq j} (m_i)^3 m_j \text{Exp}[\beta_i^3 \cdot \beta_j] + 4 \sum_{i \neq j \neq k} (m_i)^2 m_j m_k \text{Exp}[\beta_i^2 \beta_j \beta_k] \\ + 6 \sum_{i \neq j} (m_i)^2 (m_j)^2 \text{Exp}[\beta_i^2 \beta_j^2] + \sum_{i \neq j \neq k \neq \ell} m_i m_j m_k m_\ell \text{Exp}[\beta_i \beta_j \beta_k \beta_\ell] \quad (26)$$

$$= \sum_i (m_i)^4 + 6 \sum_{i \neq j} (m_i)^2 (m_j)^2 \quad (27)$$

It follows that

$$\text{Var}[X] = \text{Exp}[X^2] - (\text{Exp}[X])^2 \quad (28)$$

$$= \sum_i (m_i)^4 + 6 \sum_{i \neq j} (m_i)^2 (m_j)^2 - \left( \sum_i (m_i)^2 \right)^2 \quad (29)$$

$$= \sum_i (m_i)^4 + 6 \sum_{i \neq j} (m_i)^2 (m_j)^2 - \left( \sum_i (m_i)^4 + 2 \sum_{i \neq j} (m_i)^2 (m_j)^2 \right) \quad (30)$$

$$= 4 \sum_{i \neq j} (m_i)^2 (m_j)^2 \leq 2F_2^2 \quad (31)$$

By Chebishev, for each  $1 \leq i \leq s_2$ ,

$$\Pr [|Y_i - F_2| > \epsilon F_2] \leq \frac{\text{Var}[Y_i]}{\epsilon^2 F_2^2} \leq \frac{2F_2^2}{s_1 \epsilon^2 F_2^2} = \frac{1}{8} \quad (32)$$

and we complete the argument as before.

### Estimating $F_0$ to within a constant factor

Here we'll only give the idea without the full analysis. Let  $F = GF(2^d)$  where  $d = \lceil \log n \rceil$ . We view each  $a_j$  in the sequence as an element in the field  $F$ . To compute an estimate for  $F_0$  (the number of distinct elements in the sequence), the algorithm selects  $\alpha, \beta$  uniformly at random in  $F$ . For each  $a_j$ , the algorithm computes  $z_j = z(a_j) = \alpha a_j + \beta$ , and considers the representation of  $z_j$  as a  $d$ -bit vector  $z_{j,1}, \dots, z_{j,d}$ . It then sets  $r_j = r(z_j)$  to be the largest index such that all  $r(z_j)$  rightmost bits of  $z_j$  are 0. It maintains  $R$  as the maximum over all  $r_j$ , and when the sequence terminates it outputs  $Y = 2^R$ .

The underlying idea is that for each fixed  $\ell \in F$ ,  $z(\ell)$  is uniformly distributed in  $F$  (given the choice of  $\alpha$  and  $\beta$ ). That is, for every  $\ell, \ell' \in F$ ,  $\Pr_{\alpha, \beta}[z(\ell) = \ell'] = \frac{1}{|F|}$ . and so, for any  $r$ , the probability that its  $r$  rightmost bits are 0 is  $2^{-r}$ . Now, if  $F_0 < 2^r/c$ , then by Markov, the probability that any one of the the  $F_0$  different values  $\ell$  gives  $z(\ell)$  with  $r(z(\ell)) > r$  is less than  $1/c$ . (A few more details: Let  $B$  denote the subset of elements that appear in the stream (where we want to estimate  $|B|$ ). Let  $F_r$  denote all elements in  $F$  whose  $r$  rightmost bits are 0, so that  $|F_r| = 2^{d-r}$ . For each element  $\ell \in F$ , let  $X_\ell$  be a 0/1 random variable that is 1 if and only if  $z(\ell) \in F_r$ . Now,  $\Pr[X_\ell = 1] = 2^{-r}$  so that  $\text{Exp}[\sum_{\ell \in B} X_\ell] = |B| \cdot 2^{-r}$ . If  $|B| < 2^r/c$  then This expectation is less than  $1/c$ , and so the probability that get at least 1 (i.e.,  $c$  times the expectation) is less than  $1/c$ ) For the other direction (showing that if  $F_0 > c2^r$  then the probability that none of the  $F_0$  different  $\ell$ 's give  $r(z(\ell)) > r$  is less than  $1/c$ ), requires to apply Chebishev, and to use the fact that for any pair  $\ell \neq \ell'$  the probability that  $r(z(\ell)), r(z(\ell')) \geq r$ , is  $2^{-2r}$ .

## Constructing $k$ -Wise Independent Sample Spaces

In the estimation of  $F_2$  we built on the existence of a set of  $n$ -dimensional vectors of size  $O(n^2)$  that are 4-wise independent. Here we shall show a general (but slightly weaker) construction of  $k$ -wise independent sample spaces of size  $O(n^k)$ . Here too let  $F = GF(2^d)$  where  $d = \lceil \log n \rceil$ . We shall actually construct a set of  $n^k$  vectors over  $F^n$  (if we want to get binary vectors we can take the least significant bit of each coordinate).

Let  $w_1, \dots, w_n$  denote the elements of the field  $F$ . For each choice of  $k$  elements  $c_0, \dots, c_{k-1} \in F$  we define the vector  $v^{c_0, \dots, c_{k-1}}$  as follows:  $v_i^{c_0, \dots, c_{k-1}} = \sum_{j=0}^{k-1} c_j \cdot w_i^j$ . In other words, if we define the (univariate) polynomial  $p^{c_0, \dots, c_{k-1}} = \sum_{j=0}^{k-1} c_j x^j$ , then  $v_i^{c_0, \dots, c_{k-1}} = p^{c_0, \dots, c_{k-1}}(w_i)$ . By construction there are  $n^k$  vectors and each coordinate of any given vector can be computed using  $O(\log n)$  bits. To see why we get a  $k$ -wise independent space, consider the  $n \times d$  Vandermonde matrix, where  $M_{i,j} = w_i^j$ . Then each vector  $v^{c_0, \dots, c_{k-1}}$  is the result of multiplying the matrix  $M$  with the vector  $(c_0, \dots, c_{k-1})$ . If we consider any choice of  $k$  rows, they are linearly independent, implying the desired  $k$ -wise independence.