

## Lecture 4: Count Sketches

Lecturer: Edo Liberty

**Warning:** This note may contain typos and other inaccuracies which are usually discussed during class. Please do not cite this note as a reliable source. If you find mistakes, please inform me.

Here we will define a data structure called a CountSketch. It will allow us to estimate the number of times the most frequent elements in a stream appeared.

We denote the elements by  $o_1, \dots, o_m$  having each appeared  $n_1 \geq \dots \geq n_m$  (the names of the elements are ordered according to their frequency). Before describing the CountSketch structure, let us first analyze one of its building blocks. For lack of a more creative name, we will call it  $B$ .  $B$  is an array of length  $b$  which is associated with two hash functions:  $h : o \rightarrow [1, \dots, b]$  and  $s : o \rightarrow [-1, 1]$ .

We define two functions for  $B$  one for adding elements into it.

1. define  $Add(o)$ :
2.  $B[h(o)] = B[h(o)] + s(o)$ .

and one for returning an estimate for  $n_i$  given  $o_i$

1. define  $Query(o)$ :
2. return  $B[h(o)]s(o)$ .

In order to compute the expectation of  $B[h(o)]s(o)$  we need to define the “inverse” of  $h$ . Let  $h^{-1}(o_i) = \{o_j | h(o_j) = h(o_i)\}$ . In words,  $h^{-1}(o_i)$  is the set of all elements for  $h(o_i) = h(o_j)$ . Since each element in  $o_j \in h^{-1}(o_i)$  is encountered exactly  $n_j$  times and for each of those  $s(o_j)$  is added to  $B[h(o)]$  we have that  $B[h(o_i)] = \sum_{o_j \in h^{-1}(o_i)} n_j s(o_j)$ .

$$E[B[h(o_i)]s(o_i)] = \sum_{o_j \in h^{-1}(o_i)} n_j s(o_j) s(o_i) = n_i + \sum_{o_j \in h^{-1}(o_i), o_i \neq o_j} n_j s(o_j) s(o_i) = n_i$$

First, we see that if  $b > 8k$  we have that  $|h^{-1}(o_i) \cap \{o_1, \dots, o_k\}| = 0$  with probability at least  $7/8$ . In other words, the element  $o_i$  does not map under  $h$  to the same cell in  $B$  with any of the top  $k$  frequency items. We will define  $h_{>k}^{-1} = h^{-1}(o_i) \cap \{o_{k+1}, \dots, o_m\}$ . We will assume from this point on that  $h^{-1}(o_i) \subset \{o_{k+1}, \dots, o_m\}$  or in other words that  $h_{>k}^{-1} = h^{-1}(o_i)$ .

Now, let us bound the variance of  $B[h(o_i)]s(o_i)$ .

$$\begin{aligned} \text{Var}(B[h(o_i)]s(o_i)) &\leq E[B[h(o_i)]^2 s(o_i)^2] \\ &= E[(\sum_{o_j \in h_{>k}^{-1}(o_i)} n_j s(o_j))(\sum_{o_{j'} \in h_{>k}^{-1}(o_i)} n_{j'} s(o_{j'}))] \\ &= E_h \sum_{o_j \in h_{>k}^{-1}(o_i)} \sum_{o_{j'} \in h_{>k}^{-1}(o_i)} E_s[n_j n_{j'} s(o_j) s(o_{j'})] \\ &= E_h \sum_{o_j \in h_{>k}^{-1}(o_i)} n_j^2 \\ &= \sum_{j=k+1}^m n_j^2 / b \end{aligned}$$

Note that we have both an expectation over the choice of the hash function  $s$  and over the hash function  $h$ .

Using this bound on the variance of  $B[h(o_i)]s(o_i)$  and Chebyshev's inequality we attain that:

$$\Pr \left[ |B[h(o_i)]s(o_i) - n_i| > \sqrt{8 \sum_{j=k+1}^m n_j^2/b} \right] \leq 1/8$$

However, note that we also demanded that none of the top  $k$  elements map to the same cell as  $o_i$  which only happened with probability  $7/8$ . Using the union bound on these two events we get:

$$\Pr [|\hat{n}_i - n_i| \leq \gamma] \geq 3/4$$

where we denote  $\hat{n}_i = B[h(o_i)]s(o_i)$  and  $\gamma = \sqrt{8 \sum_{j=k+1}^m n_j^2/b}$ .

Note that this happens for every elements individually only with constant probability. We would like to get that this holds with probability  $1 - \delta$  for all elements simultaneously. We do that by repeating this entire structure  $t$  times creating the CountSketch  $B_1, \dots, B_t$ . When inserting an element we insert it into all  $t$  arrays  $B_i$  and above. When querying the CountSketch we return  $query(o_i) = Median(\hat{n}_i^1, \dots, \hat{n}_i^t)$  where  $\hat{n}_i^\ell$  is the estimator  $\hat{n}_i$  from  $B_\ell$ .

Because  $\Pr [|\hat{n}_i - n_i| \leq \gamma] \geq 3/4$  we get from Chernoff's inequality that at least half the values  $\hat{n}_i^\ell$  will be such that  $|\hat{n}_i^\ell - n_i| \leq \gamma$  (including the median) for all  $m$  elements with probability at least  $1 - \delta$  for  $t \in O(\log(m/\delta))$ .

The only thing left to do is set the correct value for  $b$  (the length of  $B$ ). We will demand that  $\gamma \leq \epsilon n_k$ . This gives  $b \geq 8 \sum_{i=k+1}^m n_i^2 / \epsilon^2 n_k^2$ . Therefore, for  $t = O(\log(m/\delta))$  and  $b \geq 8 \max(k, \frac{\sum_{i=k+1}^m n_i^2}{\epsilon^2 n_k^2})$  with probability at least  $1 - \delta$  for each element in the stream  $|\hat{n}_i - n_i| \leq \epsilon n_k$ .

The algorithm for finding the most frequent items is therefore to go over the stream and keep a CountSketch of all the elements seen this far. When we process an element, we also estimate it's frequency  $\hat{n}$  and keep the top  $k$  most frequent items in estimated frequencies. These are guaranteed to contain all elements  $o_i$  for which  $n_i > (1 + 2\epsilon)n_k$  and not to contain any element  $o_i$  for which  $n_i < (1 - 2\epsilon)n_k$ .