Institut for Matematik og Datalogi
Syddansk Universitet

November 14, 2023
JBJ

DM551/MM851 – Fall 2023 – Weekly Note 10

Prerequisite test to get access to exam

The test is available in itslearning and from the home page so please spend 5-10 minutes in the near future to get it done (the deadline is November 30 at 18:00 but please do it soon).

Stuff covered in week 46
This is a prediction as the Thurday lecture is still in the future.

- Cormen sections 5.1-5.3
- I gave more examples of the usefulness of flows based on the notes on Weekly note 9.
- I showed how to solve the exercises on flows in the last 4 bullets from Weekly note 9
- If there is time I will also cover Kleinberg and Tardos 13.6 and Cormen 11.3.3

Lectures in week 47

- Kleinberg and Tardos 13.6 and Cormen 11.3.3 (if it was not covered in Week 46).
- Cormen Section 11.5 on perfect hashing
- I will tell you about a very useful probabilistic datastructure called *count-min-sketch* (see notes below) which uses hashing to answer questions about streams of data and many other things. For some material on this see the links on the bottom of the homepage. You can find much more via Google and I will also write a bit about it soon.
- Some further stuff on Hashing.
- If there is time, I show how to solve Problem 5, January 2010.

Exercises in week 47

- Cormen Section 5.3: 5.3-4, 5.3-5, 5.3-7
- Cormen Problem 5-2
- January 2013 Problems 3
- Kleinberg-Tardos Problems 13.9, 13.10 and 13.13.

Notes on Count-min-sketch

Let S be a (possibly very long or even infinite) stream of data. Our goal is to estimate frequencies of elements that occour often in S.
Let b, ℓ be integers to be determined later, let $\mathcal{H}$ be a universal family of hash functions from the universe U that contains all possible elements which may occur in the stream to the set of integers $\{1, 2, \ldots, b\}$ and let $h_1, h_2, \ldots, h_\ell$ be distinct members from $\mathcal{H}$ chosen randomly. Below we say that the functions h_i above are **universal** by which we mean that they are randomly chosen from the universal family $\mathcal{H}$. Using these hash functions we build an $\ell \times b$ array M of counters. Initially $M_{i,j} = 0$ for all $i \in [\ell], j \in [b]$.

We process the stream (as it arrives) by proceeding as follows with the current element x from S: For every $i \in [\ell]$ we increase the counter $M_{i,h_i(x)}$ by one. So each new element of S increases the value of exactly ℓ entries of M.

Assume now that we have processed the first n elements of the stream S and denote by S_n the ordered sequence consisting of the first n elements in S (for example if
$S = A, B, A, C, B, A, D, B, A, B, C, A, B, C, D, ...$, then $S_7 = A, B, A, C, B, A, D$).

What can we say about the frequencies, f_x, of those elements x that occur at least once in S_n based on just our array of counters?

Let x be a fixed element occuring in S_n and let us first see what the counter $M_{i,h_i(x)}$ actually counts. For convenience we denote $M_{i,h_i(x)}$ by $Z_{i,x}$. Note that this is a random variable because h_i is randomly chosen from $\mathcal{H}$. Let the indicator variable $I_{i,x}$ be defined as follows (on the elements of S_n)

$$I_{i,x}(y) = \begin{cases} 1 & \text{if } h_i(x) = h_i(y) \\ 0 & \text{otherwise} \end{cases}$$

Now we can express $Z_{i,x}$ as follows

$$Z_{i,x} = f_x + \sum_{\{y \in S_n | y \neq x\}} f_y \cdot I_{i,x}(y)$$

It follows that $Z_{i,x}$ is an **upper bound** on f_x (it counts all occurences of x and of every element $y \neq x$ which is hashed to the same cell as x by h_i). Let us determine the expected value of $Z_{i,x}$. For this we use that $p(I_{i,x}(y) = 1) \leq \frac{1}{b}$ as h_i is universal. We will also use that the sum of the frequencies of all elements occuring at least once in S_n is n.

$$
\begin{aligned}
E[Z_{i,x}] &= E[f_x + \sum_{\{y \in S_n | y \neq x\}} f_y \cdot I_{i,x}(y)] \\
&= E[f_x] + E[\sum_{\{y \in S_n | y \neq x\}} f_y \cdot I_{i,x}(y)] \\
&= f_x + \sum_{\{y \in S_n | y \neq x\}} f_y \cdot E[I_{i,x}(y)] \\
&\leq f_x + \sum_{\{y \in S_n | y \neq x\}} f_y \cdot \frac{1}{b} \\
&= f_x + \frac{1}{b} \sum_{\{y \in S_n | y \neq x\}} f_y \\
&\leq f_x + \frac{n}{b} \qquad (1)
\end{aligned}
$$

So, in expectation, the count $Z_{i,z}$ for f_x is off by at most $\frac{n}{b}$. Since n can be huge and we use only a fixed set of b counters, we cannot expect a better estimate than something depending on n. Recall that Markov's inequality implies that the probability that a random variable is at least twice its expectation is at most $1/2$. Hence (considering the random variable $Z_{i,x} - f_x$) we get from (1) that

$$p[Z_{i,x} - f_x \geq \frac{2n}{b}] \leq 1/2 \qquad (2)$$

Of course the same analysis hold for all values $i \in [\ell]$ so setting $\hat{f}_x = \min_{i \in [\ell]} Z_{i,x}$ we get an upper bound for f_x and since the hash functions $h_1, h_2, \ldots, h_\ell$ are independent of each other[1] we get from (2) that

$$p[\hat{f}_x - f_x \geq \frac{2n}{b}] \leq (1/2)^\ell \qquad (3)$$

Now suppose that we are given values ϵ and δ where we want the probability that our estimate $\hat{f}_x$ is off by more that ϵn is at most δ. Using the calculations above, we can calculate useful values of b and ℓ based on ϵ and δ: It follows from (3) that if we take $b = \frac{2}{\epsilon}$ and $\ell = \log_2(\frac{1}{\delta})$, then

$$p[\hat{f}_x - f_x \geq \epsilon n] \leq \delta \qquad (4)$$

Hence, using $b \cdot \ell = \frac{2}{\epsilon} \cdot \log_2(\frac{1}{\delta})$ counters (the size of the array M) we can achieve the desired accuracies **independently** of n which could be huge. For example, supposse we want to estimate the frequency of those elements that occur with a frequency of at least 1 % and

[1] Actually here we need a stronger property of $\mathcal{H}$ to guarantee that (3) holds, namely that $\mathcal{H}$ is a so-called **strongly universal** family of hash functions, but we will not define this here for simplicity.

we want this estimate to be off by at least 0.01% with probability at most $\frac{1}{1000}$. Taking $\epsilon = 10^{-4}$ and $\delta = 10^{-3}$ we get that $p[\hat{f}_x - f_x \geq \frac{n}{10000}] \leq \frac{1}{1000}$ will hold if we use 10 different hash functions, each hashing to $\{1, \ldots, 20000\}$ so in total we need only 200000 counters to achieve this accuracy!