COMP 480/580 — Probabilistic Algorithms and Data Structure Feb 12, 2018

Lecture 10

Lecturer: Anshumali Shrivastava *Scribe By: Jacob Burger*

1 Count Sketch

1.1 Problem and Goal

The problem faced in this algorithm is the same as the problem from Count Min Sketch. At some time t, we have (i, δ_i) such that in some array C, we go to position i and increment by δ_i. C therefore represents an array

C_1	C_2	C_3	.	.	.	C_n

such that at time t we have $C_i \leftarrow C_i + \delta_i$.

However, we are only allowed $<< O(n)$ memory (sublinear) "sketch" S data structure. Our goal is to get the count of component C_i after a given time of said increments. If this can be done, you can estimate the most frequent hitter (i.e. the most visited website in a system of cached website accesses).

1.2 Framework

The framework we are working within can be seen as a set of two functions:

$$void\ Update(S, i, \delta_i) \text{ and } float\ Query(S, i)$$

The update function simply updates the "sketch" (array) S at position S_i and time t while the query function is used to check the value of S_i at said time t. This framework applies both to the algorithm discussed last class for Count Min Sketch (CMS) as well as the algorithm for Count Sketch (CS).

There are several problems with CMS, the first is that CMS cannot handle negative data. Because each entry is iterated by a positive count and the bias is a positive one, negative inputs would alter said bias and completely unravel the algorithm. Its error is also estimated via a Markov approximation, making it less precise than CS, which is estimated with a chebyshev approximation.

1.3 Algorithm

Assume you have two functions, $h(i)$ and $g(i)$:

$$h : [1 \dots N] \to [1 \dots R]$$

$$g : [1 \dots N] \to \{-1, 1\}$$

We can thus see that function h is a generic hash function that maps from 1 to R, and g is a function that maps uniformly to the outputs -1 and 1, thus $E(g(i)) = 0$. Furthermore, $g(x)$ is not entirely a random coin toss, however, for $g(i) = g(i)$, (the result is always consistent for a given i).

Given these functions, the function $Update(S, i, \delta_i)$ becomes:

$$S[h(i)] \leftarrow S[h(i)] + \delta_i * g(i)$$

The retrieval function $Query(S, i)$ then becomes simply:

$$return\ S_i * g(i)$$

1.4 Proofs and Results

Our goal is to prove that the update function produces a reasonable estimator for C_i, similar to the way we did for CMS.

With the update and query functions in mind, we can calculate the expected value for object i (an estimator for C_i), using some of the techniques from CMS. The estimator for $S(h(i))$, for instance, can be formulated as the same formula for $S(h(i))$ for CMS, simply adding the multiplication of $g(j)$ and $g(i)$ as follows:

$$S(h(i)) = \textstyle\sum_j (I_{h(j)=h(i)} * C_j * g(j)) * g(i)$$

$$= (C_i * g(i)) + \textstyle\sum_j (I_{h(j)=h(i)} * C_j * g(j))$$

In addition, we can see that:

$$g(i) * S(h(i)) = (C_i * g(i)^2) + \textstyle\sum_j (I_{h(j)=h(i)} * C_j * g(j) * g(i))$$

Again from our calculations from last class on CMS, we know the expected valuesthe indicator function is as follows:

$$E(I_{h(j)=h(i)_{i \neq j}}) = \tfrac{1}{R}$$

Following intuition,

$$E(g(i)) = 0$$

as the expected value of an even distribution is the sum of the values divided by n values $(\frac{(-1+1)}{2} = 0)$. We can also see through intution that

$$E(g(i)^2) = 1$$

as $g(i)^2 \rightarrow \{1\}$. Knowing these three expectations, we wish to calculate $E(g(i) * S(h(i)))$, which is the estimator for C_i:

$$E(g(i) * S(h(i))) = C_i * E(g(i)^2) + \textstyle\sum_j (E(I_{h(j)=h(i)}) * C_j * E(g(j)) * E(g(i)))$$

While this looks messy at first, we can eliminate the entire right hand side knowing that $E(g(i)) = 0$. This then reduces to:

$$E(g(i) * S(h(i))) = C_i * E(g(i)^2)$$

But we know $E(g(i)^2) = 1$, thus this is simply our final result:

$$E(g(i) * S(h(i))) = C_i$$

$$E(\hat{C_i}) = C_i$$

We now know that we have a reasonable estimator for C_i. But lets look back to CMS. For CMS we wanted to know the error was greater than a certain value with a constant probability. To due this they used some argument, the minimum sketch, to boost this, and estimated using a Markov approximation of bounds. Unfortunately, due to the fact that the random variable in CS is not negative, we cannot use Markov. Furthermore, we want an interval. Sounds like Chebyshev's.

The calculation of Chebyshev's inequality only requires us to calculate the variance, which is done as follows (quick reminder on the variance formula):

$$Var(X) = E[(X - E(X))^2]$$

$$Var(\hat{C_i}) = E[(g(i) * S(h(i) - C_i)^2]$$

$$= E[(C_i * g(i)^2 + \textstyle\sum_j I_j * C_j * g(j) * g(i) - C_i)^2]$$

$$= E[(\textstyle\sum_j I_j * C_j * g(j) * g(i))^2]$$

$$= E[\textstyle\sum_j I_j^2 * C_j^2 * g(j)^2 * g(i)^2 + \sum_{j \neq i, k \neq i} I_j I_k * C_j C_k * g(j) g(k) * g(i)^2]$$

Similar to earlier, we see the right half of this expectation sum go away as $E(g(i)) = 0$. Furthermore, we know the square of an identity function is just the identity function and $E(g(i)^2) = 1$. Thus we have:

$$Var(\hat{C_i}) = E[\textstyle\sum_j I_j * C_j^2]$$

$$= \tfrac{1}{R} \textstyle\sum_{j \neq i} C_j^2 = \frac{||C||_2^2}{R}$$

The final step is to plug this into Chebyshev's inequality. Recall that Chebyshev's inequality tells us that error greater than some k times the standard deviation is $< \frac{1}{k^2}$. This means that for a k of $\sqrt{3}$ we have:

$$Pr(|X - E(X)| > k\sigma) < \tfrac{1}{k^2}\text{,or}$$

$$Pr(Error > \sqrt{\tfrac{3}{R}}||C||_2) < \tfrac{1}{3}$$

This meets our criteria for obtaining error greater than some value with a constant probability.

The final step is to boost this probability, only rather than using the minimum to counter the positive bias of CMS, we will use the median. The update function becomes:

$$for\ j\ in\ 1 \to d;\ S_j(h_j(i)) \leftarrow S_j(h_j(i)) + \delta_i * g_j(i)$$

Which uses a seperate h and g function for every row.

The query function becomes:

$$return\ median_j[S_j(h_j(i)) * g_j(i)]$$

Which returns the median of the column at i over the varius j's.

Overall the final results can be summarized in the following discoveries by using this median and our formulas above:

$$C_i - \frac{3}{\sqrt{R}}||C||_2 \leq \hat{C}_i \leq C_i + \frac{3}{\sqrt{R}}||C||_2$$

Giving us a bound on our estimator. Finally, the ideal depth within this is:

$$d = log(\tfrac{1}{\delta})$$

1.5 Conclusion

Overall, we can see the math from Count Sketch does not differ that drastically from that of Count Min Sketch. The primary difference is the use of a sign function $g(i)$ that removes the positive bias on the value, thus reducing the need to take a minimum. Furthermore, you are creating a negative random variable, thus losing the ability to use Markov's inequality as an estimator for error. We therefore used Chebyshev's inequality combined with the median to counter the bias introduced from the sign function. The last step was simply computing a function to optimize the depth and estimate the error on the estimator.