Algorithm Design 21/22

Hands On 8 - Count-Min Sketch

Federico Ramacciotti

1 Problem

Consider the counters $F[i]$ for $1 \leq i \leq n$, where n is the number of items in the stream of any length. At any time, we know that $||F||$ is the total number of items (with repetitions) seen so far, where each $F[i]$ contains how many times item i has been so far. We saw that CM-sketches provide a FPTAS $F'[i]$ such that $F[i] \leq F'[i] \leq F[i] + \varepsilon ||F||$, where the latter inequality holds with probability at least $1 - \delta$.

Consider now a range query (a, b), where we want $F_{ab} = \sum_{a \leq i \leq b} F[i]$. Show how to adapt CM-sketch so that a FPTAS F'_{ab} is provided:

- Baseline is $\sum_{a \leq i \leq b} F'[i]$, but this has drawbacks as both time and error grows with $b - a + 1$.
- Consider how to maintain counters for just the sums when $b - a + 1$ is any power of 2 (less or equal to n):
  - Can we now answer quickly also when $b - a + 1$ is not a power of two?
  - Can we reduce the number of these power-of-2 intervals from $n \log n$ to $2n$?
  - Can we bound the error with a certain probability? *Suggestion*: it does not suffices to say that it is at most δ the probability of error of each individual counter; while each counter is still the actual wanted value plus the residual as before, it is better to consider the sum V of these wanted values and the sum X of these residuals, and apply Markov's inequality to V and X rather than on the individual counters.

2 Solution

2.1 Baseline solution

The baseline solution uses the general Count-Min Sketch of the stream and check the entire interval $[a, b]$ for every query. This solution requires $b - a + 1$ time, so to reduce it to logarithmic time check the solution below.

2.2 Better solution

In a better solution we can store the counters for $\log n$ positions. For example, at position 0, we store the counters for the intervals $[0, 2], [0, 4], [0, 8]$ and so on. As we can see, the space used to store all the counters for all the n positions in the array is $n \log n$. We can reduce the number of counters using a different concept of intervals, as explained below.

2.3 Best solution

The best solution uses ranges with pairs of elements, creating a sort of tree upon the array F. For example, in an array with length 8, we compute the sum of the pairs $[0, 1], [2, 3], [4, 5], [6, 7]$ at the first level, then the sum of the intervals $[0, 3], [4, 7]$ doing the sum of the lower level pairs, and so on, with the last level being the sum of the whole array. The height of the tree is therefore $O(\log n)$.

In this way the error is no more linear in $b - a + 1$, as we walk only the tree in its height in $O(\log n)$ and not the whole interval every time. To store the ranges (namely, the levels of the tree) we use a Count-Min Sketch for each one of them; thus we use $O(\log n)$ sketches for a total of

$O(r * c * \log n)$ space. Updating an element recursively updates the smaller ranges, with the last range being the complete Count-Min Sketch of the actual stream. Querying an interval takes the biggest interval from the first element and reduces the interval iteratively, e.g. with an array of 8 elements and the query $[3, 7]$, we take the first interval $[3, 4]$, then $[5, 6]$ and finally $[7, 7]$ and sum them up.

This solution reduces the error because using only one Count-Min Sketch for the stream and the counters increases the error probability (more elements to map in the same space). We can also use less space by using only one sketch when we get to $\sqrt{n}$ counters, since it is useless for the space to use a new Count-Min Sketch when we have for example 2 or 4 counters.

2.3.1 Error bounding with Markov's inequality

$$Pr[X \geq V] < \frac{E[X]}{V}$$

$$V \ ? = \sum F[i]$$

$$X \ ? = \sum F'[i]$$

Errore e' la somma degli intervalli che hanno una collisione con l'intervallo in questione
Ragionare su un unico CMS