Hands On 8: Count-min sketch: range queries

Algorithm Design

Ferraro Domenico
559813

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 suffice 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

The baseline solution, as already said, consists of computing an estimate for each item. To compute the range query, sum the estimates such that $F'_{ab} = \sum_{a \leq i \leq b} F'[i]$. This has drawbacks as both time and error grows linearly to the size of the range, which is $b - a + 1$.

Introducing estimates for ranges

To assure that the error does not increase linearly to the size of the range but logarithmically, as suggested, we can maintain counters for the sums of ranges which length is a power of 2. The idea is to make estimations for ranges and not for each single item. To compute the range query from a to

b, we find out the minimum number of non-overlapping ranges and we sum up their estimations. A way to build the range is the following: for each $i \leq n$, we have all the ranges $[i,\ i + 2^y - 1]$ such that $i + 2^y - 1 \leq n$ for $y > 0$. In other words, for each starting position i we have all the possible ranges with a power of 2 length, up to the end. The number of these possible ranges is at most $\log n$, so the total number of ranges is indeed $O(n \log n)$.

Introducing dyadic ranges

To use a smaller number of ranges we can make use of dyadic ranges. A range $[a, b]$ can be split into $2 \log n$ non-overlapping ranges of length a power of 2 called dyadic ranges. The set of dyadic ranges can be precomputed and has a size of $2n$. The following is a visual representation of the set of dyadic ranges given $n = 8$ items:

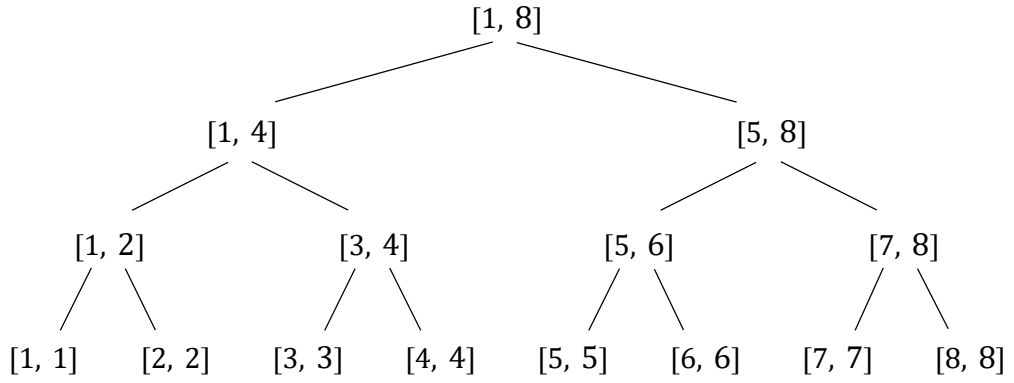

Figure 1: *Dyadic tree: visual representation of a set of dyadic ranges. The root represents the original interval; each node represents a dyadic range. Given a node, its left child represents the first half of the range, while the right child represents the second half. By splitting in a half, we reach the leaves, which represent single elements (ranges of length 1).*

Multiple CM-Sketches to achieve logarithmic error

The idea is to use $\log n$ CM-Sketches, one for each "level" of dyadic ranges. Each sketch shares the same hash functions with the same parameters. With this approach we still have one sketch to estimate the number of times each element occurs (the baseline approach), but we also have sketches that estimate the number of elements in dyadic ranges.

When it is required to update the range, we find its minimal set of non-overlapping dyadic ranges that cover the range, and we update their counter estimation. When we need to compute the range query, we also find such set of non-overlapping dyadic ranges and we sum up their estimates. The error grows logarithmically because of the minimum number of non-overlapping dyadic ranges, which is $2 \log n$. Given a range query for a range $[a, b]$, we can bound the error:

$$F_{ab} \leq F'_{ab} \leq F_{ab} + 2\epsilon \log n \, \|F\|$$

2.1 Error analysis

Let's call D the set of dyadic ranges:

$$D = \{[1,n],[1,n/2],[n/2+1,n],\dots,[n-1,n-1],[n,n]\}$$

Let $D'_{[a,b]} \subset D$ be the minimal set of non-overlapping dyadic ranges that cover the range $[a,b]$. As we know $\left|D'_{[a,b]}\right| \leq 2\log n$. The range query is the sum of all the estimates of such dyadic ranges:

$$range\_query(a,b) = \sum_{[c,d]\in D'_{[a,b]}} F'_{[c,d]}$$

Expected error of a dyadic range

Let's now focus on the approximated counter for a dyadic range. Let's have a look at a fixed hash function j of the CM-sketch which estimates the dyadic range $[a,b] \in D$. First thing first, let's define an indicator variable which is equal to 1 if the j^{th} hash function has collisions between two different dyadic ranges $[a,b] \neq [c,d]$:

$$I_{jabcd} = \begin{cases} 1, & if\ h_j(a,b) = h_j(c,d) \wedge [a,b] \neq [c,d] \\ 0, & otherwise \end{cases}$$

With that indicator variable we can define the residual Y_{jab} for a dyadic range $[a,b]$ and a hash function j:

$$Y_{jab} = \sum_{\substack{[c,d]\in D \wedge a,b\neq c,d \\ |a-b|=|c-d|}} I_{jabcd} F_{[c,d]}$$

We can analyze the expected error of a dyadic range $[a,b]$ using the expectation of Y_{jab}.

$$\begin{aligned} E\left[Y_{jab}\right] &= E\left[\sum_{\substack{[c,d]\in D \wedge a,b\neq c,d \\ |a-b|=|c-d|}} I_{jabcd} F_{[c,d]}\right] \\ &= \sum_{\substack{[c,d]\in D \wedge a,b\neq c,d \\ |a-b|=|c-d|}} E\left[I_{jabcd} F_{[c,d]}\right] \\ &= \sum_{\substack{[c,d]\in D \wedge a,b\neq c,d \\ |a-b|=|c-d|}} Pr\left[I_{jabcd} = 1\right] * F_{[c,d]} \\ &\leq \sum_{\substack{[c,d]\in D \wedge a,b\neq c,d \\ |a-b|=|c-d|}} \frac{\epsilon}{e} * F_{[c,d]} \\ &= \frac{\epsilon}{e} \sum_{\substack{[c,d]\in D \wedge a,b\neq c,d \\ |a-b|=|c-d|}} F_{[c,d]} \leq \frac{\epsilon}{e}\|F\| \end{aligned}$$

Finding the expected overall error

We can then define the overall error term X_{jab} for the queried range $[a, b]$ and its expected value as

$$X_{jab} = \sum_{[c,d] \in D'_{[a,b]}} Y_{jcd}$$

$$E[X_{jab}] = E\left[\sum_{[c,d] \in D'_{[a,b]}} Y_{jcd}\right] = \sum_{[c,d] \in D'_{[a,b]}} E[Y_{jcd}] \leq \sum_{[c,d] \in D'_{[a,b]}} \frac{\epsilon}{e}\|F\| = 2\log n \frac{\epsilon}{e}\|F\|$$

The estimate counter for the queried range $[a, b]$, given a fixed hash function j, is the following

$$F'_{[a,b]} = F_{[a,b]} + X_{jab}$$

Compute the probability that $F'_{[a,b]} \geq F_{[a,b]} + 2\log n\, \epsilon\|F\|$

We can use Markov's inequality and we can also do it on d hash functions:

$$\begin{aligned}
\Pr[\forall j \in [d]: F'_{[a,b]} \geq F_{[a,b]} + 2\log n\, \epsilon\|F\|] &= \prod_{j=1}^{d} \Pr[F_{[a,b]} + X_{jab} \geq F_{[a,b]} + 2\log n\, \epsilon\|F\|] \\
&= \prod_{j=1}^{d} \Pr[X_{jab} \geq 2\log n\, \epsilon\|F\|] \\
&\leq \prod_{j=1}^{d} \frac{E[X_{jab}]}{2\log n\, \epsilon\|F\|} \\
&\leq \prod_{j=1}^{d} \frac{2\log n \frac{\epsilon}{e}\|F\|}{2\log n\, \epsilon\|F\|} \\
&= \prod_{j=1}^{d} \frac{1}{e} = \left(\frac{1}{e}\right)^{d}
\end{aligned}$$