Intelligent Learning and Analysis Systems SS19, Exercise Sheet 1

by Andreas Hene, Niklas Mertens, Richard Palme.
Tutor: Maximilian Thiessen, Group 3

April 26, 2019

5. The ϕ-HH Problem: Lower Bounds

Suppose $\Sigma_1 = \Sigma_2$. Since $S_1 \neq S_2$, there is an $x \in [n]$ such that w.l.o.g. $x \in S_1, x \notin S_2$.

Now let σ_1 be the concatenation of $\langle S_1 \rangle$ and $\langle x \rangle$, and let σ_2 be the concatenation of $\langle S_2 \rangle$ and $\langle x \rangle$. Since the stream lengths of σ_1 and σ_2 are $m + 1$, we might have more than m' bits at our disposal, but this is not important.

Let $\phi = \frac{2}{m+1}$. Then every item which appears at least twice in σ_1 or σ_2 is a ϕ-heavy hitter. Since S_1, S_2 are sets, there is no item that appears twice in $\langle S_1 \rangle$ or $\langle S_2 \rangle$.

Since $x \in S_1$ but $x \notin S_2$, x is a heavy hitter of σ_1, but x is not a heavy hitter of σ_2. But before we processed the last element of the streams, the respective states of the storages Σ_1 and Σ_2 were equal by assumption. And since the last elements of both σ_1 and σ_2 are identical, the algorithm should have identical output for both σ_1 and σ_2. But this is not the case. So we contradicted our assumption, which means Σ_1 and Σ_2 have to be different.

2. Identifying the Majority

Let $\sigma = \langle a_1, \ldots, a_m \rangle$ and $a_i \in [n]$.

Algorithm 1 Algorithm for finding the majority of σ, if it exists

```
c ← 0
for all i ∈ [m] do
    if c = 0 then
        s ← a_i
        c ← 1
    end if
    if s = a_i then
        c += 1
    else
        c −= 1
    end if
end for
return s
```

Let $A(\sigma)$ be the output of the algorithm.

Claim: $A(\sigma)$ is the majority of σ, if it exists.
Proof: Let k be the number of times c is set to 0 in the for loop. We prove the claim by induction over k.

Base case:
Let $k = 0$. Since c is never set to 0 during the loop, we know that s never changes after iteration 1, i.e. $A(\sigma) = a_1$. Also, c increases at least one time more often than it decreases, so $A(\sigma) = a_1$ has to be the majority of σ.

Step case:
Let $k > 0$. Suppose the claim is true for all streams for which c is set to 0 at most $k-1$ times in the for loop. Let d be the majority of σ. Let j be the first iteration where c is set to 0. Side note: When c is set to 0 in an iteration j', j' has to be even. d can occur at most $\frac{j}{2}$ times in $\tau \coloneqq \langle a_1, \ldots, a_j \rangle$, otherwise d would be the majority of τ and c couldn't be set to 0 in iteration j. Consequently d will occur at least

$$\left\lfloor \frac{m}{2} \right\rfloor + 1 - \frac{j}{2} = \left\lfloor \frac{m-j}{2} \right\rfloor + 1$$

times in $\sigma' = \langle a_{j+1}, \ldots, a_m \rangle$. Because σ' is a stream of length $m - j$, d is the majority of σ'. Since the number of zeros in σ' is smaller than k, we can apply the inductive hypothesis to see that $A(\sigma') = d$. Since in iteration j of computing $A(\sigma)$, c is set to 0, applying the algorithm to σ is the same as applying it to τ and then to σ'. Hence, $A(\sigma) = d$.

∎