Updating Probability Values for Improved Bounds

Aaryan Gupta

June 21 2022

1 How the current implementation works

Although the theory paper uses Rashtchian's bounds for upper bounding $c_S(w)$, the code calculates the upper bound using the 4-term combinatorial summation expression for $c_S(w)$. The program only uses $l = \lfloor log|S| \rfloor$. The dispersion index bound is set as $\rho = 1.1$. The expression for the dispersion index is given by

$$\frac{\sigma^2[Cnt_{(S,m)}]}{E[Cnt_{(S,m)}]} \leq \sum_{w=0}^{w=l} c_S(w).r(w,m) \leq \rho$$

where $r(w,m) = \prod_{j=1}^{j=m} \left(\frac{1}{2} + \frac{1}{2}(1-2p_j)^w \right) - \frac{1}{2^m}$.

For a fixed value of m, the p_i's are started off with a value of 0.5 and are iteratively decreased until the variance bound inequality gets violated. For each term the $c_S(w)$ calculation is done by the 4-term combinatorial expression.

Note: The $c_S(w)$ bound derived in the LICS paper has a small error. The bound should be $c_S(w) \leq \frac{w}{2} \left(\frac{8e\sqrt{nl}}{w} \right)^w$ instead of $c_S(w) \leq 2 \left(\frac{8e\sqrt{nl}}{w} \right)^w$.

2 My First Approach

I used the improved bound I derived in the summer for $l = \lfloor log|S| \rfloor$ given by

$$c_S(w) \leq \frac{w}{2} \left(\frac{4e\sqrt{nl}}{w} \right)^w$$

The $c_S(w)$ was upper bounded using this analytic expression instead of a combinatorial expression. The p_i's were iterated in the same manner.

Results: This approach was coded in the file "newvalues1.py" led to higher values of p_i in comparison to the original approach. This was undesirable and probably happened because the improved analytical bound is still weak in comparison to the weaker old combinatorial expression.

3 My Second Approach

In the improved bounds for $c_S(w)$ I derived in the summer, instead of the 4-term combinatorial expression,for a fixed $x \in S$ I improved it to

$$c_S(w) \leq \sum_{(b,a)\in U'} e_{(b,a)}(S) \leq \sum_{(b,a)\in U'} \binom{n-|x|}{a}\binom{|x|}{a}\binom{|x|}{b-a} \leq \sum_{(b,a)\in U'} \binom{n-b}{a}\binom{l}{a}\binom{l}{b-a}$$

where $U' = \{(b,a) : b \geq a, b+a = w\}$.

We calculate these $c_S(w)$ values and substitute in the sum and iteratively find the p_i values like in the original approach.

Results: This approach was coded in the file "newvalues2.py" led to slightly higher values of p_i in comparison to the original approach. This is probably happening because of some optimisation included in the original code that I have not yet incorporated into my code.

4 My Third Approach

This approach was just based on using Rashtchian's old bounds but for $l = \min\{\lfloor log|S| \rfloor, \lfloor \frac{2log|S|}{logn - loglog|S|} \rfloor\}$ instead of just logsetsize.

Results: