NUS Internship Diary

Aaryan Gupta

1 March 1, 2022

I have read the LICS paper once and have made notes at https://drive.google.com/file/d/1P8Y13nWnIs2JyaI42XXHB6zz4ZrWwhCx/view?usp=sharing. I have understood the gist of the paper. I haven't quite figured out a few details, especially the proofs. Here is the list of things left for me to understand-

- understanding the proof of bounding the dispersion index using Rashtchian's work. Will I have to read that paper?
- proving the Rennes hashing family to be concentrated after bounding the dispersion index
- analysis of the algorithm i.e. calculation of thresh and iter for given probabilistic bounds (reading the approxMC4 paper might help)

I have looked over some of the code but unfortunately I could not spend a lot of time on it due to my exams. Not a lot of progress has been done on the coding front. In the last couple of meets, Yash conveyed two major ideas on which approxMC5 can be improved upon-

- Instead of varying the hashing probabilities pi with i, we fix that to a constant. This retains theoretical boundaries because of this step carried out in the paper- $q(w,m) = \prod_{j=1}^{m}\left(\frac{1}{2}+\frac{1}{2}(1-2p_j)^w\right) \leq \left(\frac{1}{2}+\frac{1}{2}(1-2p_m)^w\right)^m$. This, however, would lead to better experimental results.
- Use a regular run of approxMC4 to precalulate qs with some probabiity in the first step. Use this in the rest of the approxMC5 iterations.

I have thought a bit about each of these ideas but they seem a little confusing to tackle. I will try and bring them up in the meeting as I have been having some trouble with the second idea.

Question: Variance $\frac{\sigma^2[\mathsf{Cnt}_{\langle S,m\rangle}]}{\mathsf{E}[\mathsf{Cnt}_{\langle S,m\rangle}]}$ is written as a summation from w=1 to n, but after using Rastachian's inequality it becomes from w=1 to ℓ, where ℓ is $\lceil(log(S))\rceil$. Why is it so?

Solution: The variance is derived as originally

$$\sigma^2[\mathsf{Cnt}_{\langle S,m\rangle}] = 2^{-m}\sum_{w=0}^{n} c_S(w)r(w,m)$$

Rastachian showed that this expression is equivalent to

$$\frac{\sigma^2[\mathsf{Cnt}_{\langle S,m\rangle}]}{E[\mathsf{Cnt}_{\langle S,m\rangle}]} = \sum_{w=1}^{\ell} c_S(w)r(w,m)$$

where $\ell = \lceil m + \log_2(k)\rceil$= $\lceil(log(S))\rceil$.

Substituting the bounds for $c_S(w)$, we get
$\frac{\sigma^2[\mathsf{Cnt}_{\langle S,m\rangle}]}{\mathsf{E}[\mathsf{Cnt}_{\langle S,m\rangle}]} \leq \sum_{w=0}^{\ell} 2\cdot\left(\frac{8e\sqrt{n\cdot\ell}}{w}\right)^w r(w,m)$ with $\ell = \lceil\log|S|\rceil$.

Edit: The original paper seems to be incorrect. The summation should vary from w=1 to 2ℓ. As the set S is taken to be left-compressed and down, we see that for a n-bit string with ℓ 1's present in the set has maximum hamming distance 2ℓ from another member in the set. Example- for $n = 6$ and $\ell = 3$ look at 10101 and 010101. The distance is $2\ell = 6$ not $\ell = 3$.

2 March 10, 2022

The two tasks I received for this week were:

Task 1: We are going to run a simplified version of ApproxMC for input set=S, number of iterations = 1, and the threshold = 1. You are given a parameter δ, and your algorithm outputs m. Provide $m1, m2$ such that with probability at least 1-δ, $(m - m1) <= \lceil log(|S|) <= (m + m2)$.

Solution: We assume $|S|$ to be a power of two. Otherwise, we will have to deal with edge cases where k is close to a power of two. The simplifies ApproxMC uses 2-universal hash functions. For 2-universal hash functions, the following lower bound inequality holds for any real number β:

1. $\mathsf{Pr}\left[\mathsf{Cnt}_{\langle S,m\rangle} \leq \beta \mathsf{E}\left[\mathsf{Cnt}_{\langle S,m\rangle}\right]\right] \leq \frac{1}{1+(1-\beta)^2 \mathsf{E}\left[\mathsf{Cnt}_{\langle S,m\rangle}\right]}$
2. $\mathsf{Pr}[0 \leq \frac{\beta|S|}{2^m}] \leq \frac{1}{1+(1-\beta)^2\frac{|S|}{2^m}}$
3. $\lim_{\beta\to 0}\mathsf{Pr}[\ 0 \leq \frac{\beta|S|}{2^m}] \leq \frac{1}{1+\frac{|S|}{2^m}}$

From this inequality, we set $\frac{\delta}{2} = \frac{1}{1+\frac{|S|}{2^m}}$.
Upon simplifying we get $m = \lfloor log(|S|)\rfloor - \log(\frac{2-\delta}{\delta})$
In turn we have $m_2 = \log(\frac{2-\delta}{\delta})$.

For the upper bound inequality, we start out with Cantelli's inequality:

1. $\mathsf{Pr}[\mathsf{Cnt}_{\langle S,m\rangle} - \mathsf{E}[\mathsf{Cnt}_{\langle S,m\rangle}] \geq \lambda] \leq \frac{1}{1+\frac{\lambda^2}{\mathsf{E}[\mathsf{Cnt}_{\langle S,m\rangle}]}}$
2. $\mathsf{Pr}[1 - \frac{|S|}{2^m}] \geq \lambda] \leq \frac{1}{1+\frac{2^m\lambda^2}{|S|}}$

From this we see that λ can be at most $\frac{1}{2}$. Substituting this in the equation and setting limit to $\frac{\delta}{2}$, we get $\frac{\delta}{2} = \frac{1}{1+\frac{2^m}{4|S|}}$.
Solving this, we get
$m = \lfloor log(|S|)\rfloor + \log(\frac{4(2-\delta)}{\delta})$.
In turn we have $m_1 = \log(\frac{4(2-\delta)}{\delta})$

Task 2: Modify the code of approxmc so that the algorithm runs with the full-sparse hash matrixes instead of hash rennes. Hash rennes is constructed with p_i values(for row i) which are encoded in a large array. Hash sparse will use a single value of p_i for every row. Let that be an input parameter for the algo for now.

Solution: In constants.cpp I noticed that the probability values are decided by the string sparseprobvalues. I set all the values to 0.1 instead of the original values. After this, I made a string instead of the original values with a user input constant_hashing_probability given at the command line. Unfortunately this gives an exception of "Too many variables given in input" which seemed to be generated by a try catch block somewhere. I will try to remove this try-catch block so that the code works fine.

Edit: I should work to keep the input restrained to only the CNF file. Yash suggested to take in the probability through a –flag in the command line. To add this, I will follow the keyword epsilon for the –epsilon flag.

3 March 20, 2022

Task: Write a program that takes in a CNF ϕ and a value δ and returns m_1, m_2 such that the $m_1 < log_2(|R_\phi|) < m_2$ with probability $1 - \frac{\delta}{2}$.

Solution: