Advanced Proof-Systems ; Problem Set 1

Yosef Goren

November 16, 2022

1 Chernoff Bound and Sequential Repetition

1.1 Markov's Inequality

$$\mu = \sum_r Pr[x = r] \cdot r = \sum_{r \leq \alpha\mu} Pr[x = r] \cdot r + \sum_{r > \alpha\mu} Pr[x = r] \cdot r$$

$$\geq \sum_{r > \alpha\mu} Pr[x = r] \cdot r > \sum_{r > \alpha\mu} Pr[x = r] \cdot \alpha\mu = \alpha\mu \sum_{r > \alpha\mu} Pr[x = r] = \alpha\mu \cdot Pr[x > \alpha\mu]$$

$$\Rightarrow 1 > \alpha Pr[x > \alpha\mu] \Rightarrow Pr[x > \alpha\mu] < \frac{1}{\alpha}$$

1.2 Chernoff Bound

Let $\bar{x}$ be defined as described.
Auxilary Calculations:

1.
$$E(\bar{x}^2) = E((\frac{1}{n} \sum_{i \in [n]} x_i - p)^2)$$

$$= \frac{1}{n^2}(\sum_{i=j}(E((x_i - p)(x_j - p))) + \sum_{i \neq j}(E((x_i - p)(x_j - p))))$$

$$= \frac{1}{n^2}(n \cdot E((x_1 - p)^2) + n \cdot (n-1)E((x_1 - p)(x_2 - p)))$$

$$= \frac{1}{n^2}(n \cdot (E(x_1^2) - 2pE(x_1) + p^2) + n(n-1)E(x_1x_2 - p(x_1 + x_2) + p^2))$$

$$= \frac{1}{n^2}(n \cdot (p - p^2) + n(n-1)(p^2 - 2p^2 + p^2))$$

$$= \frac{p - p^2}{n}$$

2.
$$E(\bar{x}) = E(\frac{1}{n} \sum_{i \in [n]} x_i - p) = \frac{1}{n} \sum_{i \in [n]} E(x_i - p) = \frac{1}{n} \sum_{i \in [n]} 0 = 0$$

Showing the chernoff bound:
The second transision is use of Markov, the 4'th uses the auxilary calculations.

$$Pr[\bar{x} > \epsilon] \leq Pr[e^{\lambda\bar{x}} > e^{\lambda\epsilon}] \leq \frac{E(e^{\lambda\bar{x}})}{e^{\lambda\epsilon}} \leq E(\lambda + \lambda\bar{x} + (\lambda\bar{x})^2)e^{-\lambda\epsilon}$$

$$= e^{-\lambda\epsilon}(\lambda + \lambda E(\bar{x}) + \lambda^2 E(\bar{x}^2)) = e^{-\lambda\epsilon}(\lambda + \lambda 0 + \lambda^2 \frac{p - p^2}{n})$$

$$= \lambda e^{-\lambda\epsilon}(1 + \lambda \frac{p - p^2}{n})$$

Now set $\lambda = \frac{n\epsilon}{2p(1-p)}$:

$$= \frac{n\epsilon}{2p(1-p)} e^{-\frac{n\epsilon}{2p(1-p)}\epsilon}(1 + \frac{n\epsilon}{2p(1-p)} \frac{p - p^2}{n})$$

$$= \frac{n\epsilon(1 + \frac{\epsilon}{2})}{2p(1-p)} \frac{1}{e^{\frac{n\epsilon}{2p(1-p)}\epsilon}} = 2^{-\Omega(\epsilon^2 n)}$$

Thus:

$$Pr[\bar{x} > \epsilon] < 2^{-\Omega(\epsilon^2 n)}$$

1.3 Statistics Fun

Chernoff was enough fun for me.

1.4 Sequential Repetition of Interactive Proofs

For any IP, P, V and $n \in \mathbb{N}$ - define the P^n, V^n to be a protocol where P^n and V^n run n iterations of the original protocol, and where V^n accepts iff at-least half of the iterations have resulted with V accepting.

Now let P, V be an IP with soundness and completeness errors of $\frac{1}{3}$.
We would like to show that P^n, V^n is errors of $2^{-\Omega(n)}$.

For this analysis, we make a few auxilary notations. Let C_i denote the occurance that the i'th iteration resulted with V accepting. And let $C = \frac{1}{n}\sum_{i\in[n]} C_i$.
Let $p = Pr[C_1] \geq \frac{2}{3}$.

To start off, we will examine the soundness error; Let $x \in L$.
Note how in this case, we can esaly see how C_i, C_j are i.i.d for any choice of i, j since these are just repetitions of the same protocol between P and V.
Thus we can use chernoff bound on C:

$$Pr[(P^n, V^n)(x) = 1] = Pr[C > \frac{1}{2}]$$

$$<_{\texttt{chernoff}} 2^{-\Omega(\frac{1}{2^2}n)} = 2^{-\Omega(n)}$$

As for the soundness, since I did not manage to find a formal proof, I will attempt to give a coinvincing (proof by intimidation?) argument.

A nice place to start would be to consider the proof for *completness*; since the soundness and completness errors are the same, why can't we use the same proof? Indeed we can symmetrically define C_i' to be the oppesite of C_i, and have C_1 happend with the same probability as C_1' in oppisite cases ($x \in L$ as opposed to $x \notin L$).
The only problem with this is that the prover P^{n*} does not have to run the protocol P in each iteration (unline P^n from the completness case), and thus we cannot simply assume that each itearation of running (P_i^*, V) will be independent of other iterations.

The reason this - intuitively - is similar to something we have seen in the lectures, the best prover P^* strategy would be to run the protocol independently in each iteration; since he has to ability to simulate whatever 'conclusions' he might have had from prior iterations.
Another way to think about it is that the prover should not have the ability to 'deduce' anything from any prior iterations, since whatever happend in them - he could esaly simulate himself (since the verifer has no secretes and the prover is not computationally bounded).
Q.E.D

■

2 Perfect Soundness

2.1 $NP \subseteq IP^{PS}$

Let $L \in NP$. There there is an NP relation R_L and a corresponding TM R_L to dicide R_L.

Define the following IP, on input x, (P, V) runs as follows:

- P finds y s.t. $|y| = poly(|x|) \wedge (x, y) \in R_L$ and sends it.
- V gets some y' and accepts iff $M_L(x, y')$ accepts.

Proof of correctness:

- Perfect Soundness:
  Assume towards contrediction that for $x \notin L$, $(P^*, V)(x) = 1$, thus V got some y' s.t. $M_L(x, y') = 1$, but this means (by def. of NP) that $(x, y') \in R_L$ which means $x \in L$. Contrediction.
- Completeness:
  If indeed $x \in L$, then $\exists y : (x, y) \in R_L \wedge |y| = poly(|x|)$, thus P will sends it and V will run $M_L(x, y)$, get 1 and accept.

2.2 $IP^{PS} \subseteq NP$

Let $L \in IP^{PS}$. By def. $\exists P,V \in PPT$ s.t. the scheme is IP with perfect soundness.
Now define the following NP relation R_L:

$$R_L = \{(x,y) \mid x \in L \wedge y \texttt{ is the transcript of the run } (P,V)(x)\}$$

Now define the following TM M_L to decide R_L:
On input (x,y) run as follows:

- M_L simulates the run of V; whenever coins are tossed, M_L uses the transcript to toss the same coins. Whenever input is received from P, it uses the transcript to get the input. If indeed this simulation matches the actions of V (at every step the simulated V acts the same as in transcript both yield 'accept') - then M_L accepts (and rejects otherwise).

For any $(x,y) \in R_L$, $M_L(x,y)$ accepts, since y really is a transcript of $(P,V)(x)$ and so M_L gets the same run as the real iteractive proof - thus it accepts.
For any $(x,y) \notin R_L$, assume towards contrediction that $M_L(x,y)$ accepts, thus y is a transcript of some run of $(P,V)(x)$, but this means that $x \in L$, Contrediction.

So we get that indeed M_L dicides R_L, it is also easy to see how the size of these transcripts has to be polynomial due to V's polynomial limitation in the definition of IP.

To sum up, we have shown an NP relation R_L for L, meaning it is indeed in NP.

3 Naive Set Lower Bound

3.1

First, we define some notations:
For each iteration i of the lonely verifier, we define C_i to be the occurance that the sampled u_i i within S.
We define C to be the sum of all C_i's.
Also, define $r := \frac{|S|}{|U|}$ - meaning the ratio between elements within U which are in S.

Now, consider the case where $|S| > t$:

$$Pr[V \texttt{ accepts}] = Pr[C > \frac{t}{2|U|}]$$

$$= Pr[C - r > \frac{t}{2|U|} - r] = Pr[C - \frac{S}{U} > \frac{t - 2|S|}{2|U|}]$$

$$<_{\texttt{chernoff},\epsilon=\frac{t-2|S|}{2|U|}} 2^{-(\frac{t-2|S|}{2|U|})^2 k} = 0.99 \Rightarrow -(\frac{|S|}{2|U|})^2 k = \log(0.99)$$

$$\Rightarrow k \geq (\log(0.99))(\frac{2|U|}{|S|})^2$$

Since the chernoff bound is aysymptotic, the exact constant here does not matter, and we only care that k has to be $\Omega(\frac{|U|^2}{|S|^2})$.

Similarly, we can see the other constrain yeild similar results:
Assume $|S| < \frac{t}{100}$.
The first transition summurizes the above steps and uses chernoff bound again.

$$Pr[C > \frac{t}{2|U|}] < 2^{-(\frac{t-2|S|}{2|U|})^2 k} = 0.01 \Rightarrow \cdots \Rightarrow k = \Omega(\frac{|U|^2}{|S|^2})$$

Both constains yield the same requirement on k, so to sum up, $k\Omega(\frac{|U|^2}{|S|^2})$.

3.2

As we have seen, k has to be proportional to $\frac{|U|}{|S|}$, hence in is not polynomial w.r. to $log(|U|)$, and using it would break the complexity constraints on the verifier of the interactive proof.

4 Matrix Multiplication

4.1 Naive Algorithm

We can deduce an algorithm from the definition of matrix multiplication. Since $\forall i,j : (A \cdot B)_{i,j} = \langle A_{i,:}, B_{:,j} \rangle = \sum_{k=1}^{n} A_{i,k} \cdot B_{k,j}$, we can write the following algorithm:

```
1    C = zeros(n,n)
2    for i=1 to n:
3    for j=1 to n:
4        for k=1 to n:
5            C[i,j] += A[i,k] * B[k,j]
```

4.2 Random Subset Sum Principle

Lemma:
For any $x \in \mathbb{F}^n \neq 0$ and uniformly sampled $\in \mathbb{F}^n$, $\langle x, y \rangle$ is uniformly distributed over $\mathbb{F}^n$.
Proof of lemma: Since $x \neq 0$, it must have at least one non-zero entry - denote it with x_i. We can see that:

$$\langle x, y \rangle = \sum_{j=1}^{n} x_j y_j = x_i y_i + \sum_{j \neq i} x_j y_j := x_i y_i + \alpha \qquad (*)$$

Note how α is just a constant (with respect to i) value in $\mathbb{F}^n$.
We have also seen in the lecture that for any non-zero $a \in \mathbb{F}^n$, and uniformly sampled $b \in \mathbb{F}^n$ - $a \cdot b$ is distributed uniformly - meaning so is $x_i \cdot y_i$ in (*).
Moreover, for any non-zero $a \in \mathbb{F}^n$, and uniformly sampled $b \in \mathbb{F}^n$ - $a + b$ is also uniformly distributed.
Meaning that in (*), $x_i \cdot y_i + \alpha$ is uniformly distributed, hence - so is $\langle x, y \rangle$.

Proof of claim:
Let $x \in \mathbb{F}^n \setminus \{0\}$. From the lemma, we know that $\langle x, y \rangle$ is uniformly distributed over $\mathbb{F}^n$, thus the probability that $\langle x, y \rangle = 0$ is $\frac{|\{0\}|}{|\mathbb{F}^n|} = \frac{1}{|\mathbb{F}^n|}$.

4.3 Randomized Verification of Matrix Multiplication

Algorithm P:

```
1    On input A,B,C
2    sample x uniformly from F^n
3    d := B*x
4    e := A*d
5    c := C*x
6    return e=c
```

Now it is easy to see how the complexity of this algorithm is $O(n^2)$, since in lines 3,4,5 we have operations of $O(n^2)$ complexity, and the rest of the operations seen can be done in linear complexity.

As for the correctness:

- Completness - If $A \cdot B = C$, then $Pr[P(A, B, C) = 1] = 1$:
  When indeed $A \cdot B = C$, then multipling both sides by x will yield the same value, meaning $e = c$ and $P(A, B, C) = 1$.

- Soundness - If $A \cdot B \neq C$, then $Pr[P(A, B, C) = 1] \leq \frac{1}{2}$:
  The conditions for the algorithm to accept are:

  $(AB)x = Cx \Leftrightarrow \forall i \in [n] : (AB)_{i,:}x_i = C_{i,:}x_i \Leftrightarrow \forall i \in [n] : ((AB)_{i,:} - C_{i,:})x_i = 0$

  Now since $A \cdot B \neq C$, there must be at least one k such that:

  $$(AB)_{k,:} - C_{k,:} \neq 0 \Rightarrow Pr[(AB)_{k,:} - C_{k,:} \cdot x_k] \leq \frac{1}{|\mathbb{F}^n|}$$

  So we can see:

  $$Pr[\forall i \in [n] : ((AB)_{i,:} - C_{i,:})x_i = 0] \leq Pr[((AB)_{i,:} - C_{i,:})x_i] \leq \frac{1}{|\mathbb{F}^n|}$$

  It seems fair to assume $|\mathbb{F}^n| \geq 2$, but just in case, we can handle $\mathbb{F} = \{e\}$ by checking for this case spesifically.