1 Connection to random restrictions and Expected Bias

Assume $g : \{-1, 1\}^m \rightarrow \{-1, 1\}$ is $1 - \epsilon$-hard for size $s$. Further assume $g$ is balanced, i.e., $\mathbb{E}[g] = 0$. Although one can get rid of the last assumption, we will use it for the sake of simplicity.

Let $H$ be a $\gamma$-hard-core set for $g$ against size $s' = \Omega(\gamma^2 \log \frac{1}{\gamma \epsilon})$ of cardinality $\epsilon 2^m$ s.t. $g$ is balanced on $H$. Notice, $g$ is also balanced on $\bar{H}$.

Let $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$.

Suppose a circuit $C$ of size at most $s''$ tries to compute $f(g(x^1), \ldots, g(x^n))$ on uniformly drawn inputs $x^1, \ldots, x^n$.

Is $x^1, \ldots, x^n \in H$?

- Conditioned on $x^i \in H$, $g(x^i)$ is computationally indistinguishable (up to $\gamma$) from a random bit to $C$.

- Conditioned on $x^i \notin H$, $g(x^i)$ is still a uniformly distributed random bit, but $C$ might exactly know $g(x^i)$. (Remember, the example from last lecture.)

The point is, think of $(g(x^1), \ldots, g(x^n))$ as a random restriction $(y, \bar{I})$ with $*$-probability $\epsilon$. The best thing $C$ could do is look at $f_{y \rightarrow I}$ and output the more common value.

**Definition 1.1** The expected bias of $f$ at $\epsilon$ is

$$\text{ExpBias}_\epsilon(f) = \mathbb{E}[|\hat{f}_{y \rightarrow I}(\emptyset)|]$$

where the expectation is over random restrictions $(y, \bar{I})$ with $*$-probability $\epsilon$.

**Theorem 1.2** Let $g : \{-1, 1\}^m \rightarrow \{-1, 1\}$ which is $1 - \epsilon$-hard for size $s$, assume $g$ is balanced and let $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$. Let $\gamma > 0$. Then, $f \otimes g$ is $\frac{1}{2} + \frac{1}{2}\text{ExpBias}_\epsilon(f) + \gamma$-hard for circuits of size

$$s'' = \Omega \left( \frac{\gamma^2 \log \frac{1}{\gamma \epsilon}}{n} \right) \cdot s.$$  

**Proof:**[Sketch] We will not prove this result, although the proof is not too difficult. It follows from a hybrid argument. This is why we lose the factor of $n$ in $s''$. □

Since $\text{ExpBias}_\epsilon(\chi_{[n]}) = (1 - \epsilon)^n$, we get Yao’s XOR Lemma as a corollary at least in the case of balanced $g$.  

Corollary 1.3 (Yao’s XOR Lemma for balanced functions)

Remark 1.4 The theorem is essentially tight.

Proposition 1.5
\[ S_{1-\epsilon}(f) \leq \text{ExpBias}_\epsilon(f) \leq \sqrt{S_{1-\epsilon}(f)} \]

Proof:
\[ \mathbb{E}_{y,I}[\widehat{f}_{y-I}(\emptyset)^2] \leq \mathbb{E}_{y,I}[|\widehat{f}_{y-I}(\emptyset)|] \leq \sqrt{\mathbb{E}_{y,I}[\widehat{f}_{y-I}(\emptyset)^2]} \]

The following proposition concludes our proof. □

Proposition 1.6
\[ \mathbb{E}_{y,I}[\widehat{f}_{y-I}(\emptyset)^2] = S_{1-\epsilon}(f) \]

Proof:
\[ \mathbb{E}_{y,I}[\widehat{f}_{y-I}(\emptyset)^2] = \mathbb{E}_{I,y}[\mathbb{E}[F_\emptyset(y)^2]] = \mathbb{E}_{I}[\sum_{S \subseteq I} \hat{f}(S)^2] = \sum_{S} \hat{f}(S)^2 \Pr[S \subseteq \bar{I}] = \sum_{S} (1-\epsilon)^{|S|} \hat{f}(S)^2 \]

□

2 Very noise sensitive monotone functions

Our goal is now clear. We want to find very noise sensitive monotone functions.

Definition 2.1 The noise sensitivity of \( f \) at \( \epsilon \in [0, 1/2] \), denoted \( \text{NS}_\epsilon(f) \), is

\[ \Pr_{x,y=N_\epsilon(x)}[f(x) \neq f(y)] \]

where \( y = N_\epsilon(x) \) means that \( y \) is formed by flipping each bit of \( x \) independently with probability \( \epsilon \).
Proposition 2.2
\[ NS_\epsilon(f) = \frac{1}{2} - \frac{1}{2}S_{1-2\epsilon}(f) = \frac{1}{2} - \frac{1}{2} \sum_S (1 - 2\epsilon)^{|S|} \hat{f}(S)^2 \]

Proof:
\[ S_{1-2\epsilon}(f) = \mathbb{E}_{x,y \sim 1-2\epsilon,x} [f(x)f(y)] \]
That is, \( x \) is drawn uniformly at random and \( y \) is a \((1 - 2\epsilon)\)-correlated copy of \( x \). But, that is equivalent to choosing \( y = N_\epsilon(x) \). So,
\[ \mathbb{E}_{x,y \sim 1-2\epsilon,x} [f(x)f(y)] = \mathbb{E}_{x,y = N_\epsilon(x)} [1 - 2\epsilon \mathbb{I}[f(x) \neq f(y)]] = 1 - 2NS_\epsilon(f) \]
\( \square \)

Theorem 2.3 If \( f : \{-1, 1\}^n \rightarrow \{-1, 1\} \) is monotone (in NP) and \( NS_{n-\alpha}(f) \geq \frac{1}{2} - n^{-\beta} \), then “NP is \( 1 - \frac{1}{\text{poly}(n)} \)-hard for poly-size circuits” implies “NP is \( \approx \frac{1}{2} + n^{-\beta/2} \)-hard for poly-size circuits”.

We can picture the following.

Proposition 2.4 \( NS_\epsilon(f) \) is a concave, increasing function of \( \epsilon \). It is 0 when \( \epsilon = 0 \) and it is \( \frac{1}{2} \) at \( \epsilon = \frac{1}{2} \).

Proof: Since \( NS_\epsilon(f) \) is a concave function of \( \epsilon \), 0 at 0, we have \( NS_\epsilon(f) \leq \epsilon NS_0'(f) \). \( \square \)

Therefore, \( NS_\epsilon(f) \leq O(\epsilon\sqrt{n}) \) if \( f \) is monotone and \( NS_\epsilon(f) < \frac{1}{4} \), if \( \epsilon < \Omega \left( \frac{1}{\sqrt{n}} \right) \).

3 Recursive Majority

Theorem 3.1
\[ NS_\epsilon(Maj_n) \leq O(\sqrt{\epsilon}) \]

Although, this theorem makes Majority a seemingly bad candidate for our purpose, we still try to recursively construct some good function starting with Majority.
\[ p(\epsilon) := NS_\epsilon(Maj_3) = \frac{3}{2} \epsilon - \frac{3}{2} \epsilon^2 + \epsilon^3 = \]

Question: What is \( NS_\epsilon \) of \( Maj_3(Maj_3(\ldots), Maj_3(\ldots), Maj_3(\ldots)) \)?

Observation 3.2 If \( f \) is balanced, then
\[ NS_\epsilon(f' \otimes f) = NS_{NS_\epsilon(f)}(f') \].
In particular,

- for small $\epsilon$, $p(p(\epsilon)) \approx \left(\frac{3}{2}\right)^2 \epsilon$, $p(p(p(\epsilon))) \approx \left(\frac{3}{2}\right)^3 \epsilon$.

- for small $\delta$, $p(p(1-\delta)) \approx \frac{1}{2} - \left(\frac{3}{4}\right)^2 \delta$.

So, define

$$Maj_3^{(k)} = Maj_3 \otimes \cdots \otimes Maj_3.$$  

We get $NS_\epsilon(Maj_3^{(k)}) = p^{(k)}(\epsilon)$. The input length is $3^k$.

**Fact 3.3** If depth $k \geq (1 + o(1))(\log_2 \left(\frac{1}{\epsilon}\right) + \log_3 \left(\frac{1}{\delta}\right))$, then $NS_\epsilon(Maj_3^{(k)}) \geq \frac{1}{2} - \delta$.

Write $n = 3^k$ for the input length. We get $NS_{\epsilon,p}(Maj_3^{(k)}) \geq \frac{1}{2} - \delta$, so long as

$$n \approx 3^{\log_3/2(1/\epsilon) + \log_3(1/\delta)} = \left(\frac{1}{\epsilon}\right)^{\log_3/2} \left(\frac{1}{\delta}\right)^{\log_3} \approx \left(\frac{1}{\epsilon}\right)^{2.71} \left(\frac{1}{\delta}\right)^{3.82}.$$  

So, if $\epsilon \geq \frac{1}{n^{1/7}}$, $\delta \leq \frac{1}{n^{1/3}}$, this holds. Finally, you get a monotone function $f$, computable in polynomial time, with $NS_{n^{-1/\epsilon}}(f) \geq \frac{1}{2} - n^{-1/8}$.

**Corollary 3.4** If $\exists L \in \text{NP (balanced)}$ which is $1 - \frac{1}{\text{poly}(n)}$-hard for size $s = \text{poly}(n)$, then $\exists L \in \text{NP}$ which is $\frac{1}{2} + \frac{1}{n^{1/10}}$-hard (infinitely often) for size $\text{poly}(n)$.

**Remark 3.5** Improvement (Healy-Vadhan-Viola): If there exists a balanced $L \in \text{NP}$, $1 - \frac{1}{\text{poly}(n)}$-hard for size $2^{\Omega(n)}$, then $\exists L \in \text{NP}$ which is $\frac{1}{2} + 2^{-\Omega(\sqrt{n})}$ hard for size $2^{\Omega(n)}$. 

4