

Lecture 13: Extractors from Hashing

October 28, 2004

Scribe: Gholamreza Haffari

1 Existence of Extractors

Last time we proved a theorem which indicates there is no *universal* seedless extractors. That is, for any function $\text{Ext} : \{0, 1\}^n \rightarrow \{0, 1\}$ there is a flat $(n-1)$ -source X such that $\text{Ext}(X)$ is constant. However, for any k -source there does exist a seedless extractor.

Theorem 1 $\forall k \leq n$, and for any flat k -source X , there exists a (seedless) $\text{Ext} : \{0, 1\}^n \rightarrow \{0, 1\}^m$ with $m = k - 2 \log \frac{1}{\epsilon} - O(1)$ such that $\text{Ext}(X)$ is ϵ -close to U_m .

Notation: In the following proof we set: $M = 2^m$, $[M] = \{0, \dots, M-1\}$ and $\{0, 1\}^m$ to be the set of all m -bit binary strings.

Proof: Take a random function $\text{Ext} : \{0, 1\}^n \rightarrow \{0, 1\}^m$. To show $\text{Ext}(X)$ is close to uniform we have to show:

$$\Delta(\text{Ext}(X), U_m) \leq \epsilon$$

we need to show $\forall T \subseteq [M]$,

$$\Pr[\text{Ext}(X) \subseteq T - \mu(T)] \leq \epsilon$$

here $\mu(T)$ is the expected fraction of random variables that have value 1. By Chernoff, the left side of the above inequality is upperbounded by $e^{-\Omega(\epsilon^2 K)}$. So:

$$\Pr[\exists \text{ some } T \text{ s.t. the inequality is violated}] \leq (\#T\text{'s})e^{-\Omega(\epsilon^2 K)} \leq 2^M e^{-\Omega(\epsilon^2 K)} \stackrel{?}{<} 1$$

We can choose the parameter M good enough to guarantee the last inequality:

$$2^M < e^{\alpha \epsilon^2 K} = 2^{(\log e) \alpha \epsilon^2 K}$$

$$\iff M < \beta \epsilon^2 K$$

This implies $m = k - 2 \log \frac{1}{\epsilon} - O(1)$. ■

Definition 2 (Nisan & Zuckerman) $\text{Ext} : \{0, 1\}^n \times \{0, 1\}^d \rightarrow \{0, 1\}^m$ is a (k, ϵ) -extractor if for any k -source X on $\{0, 1\}^n$, $\text{Ext}(X, U_d)$ is ϵ -close to uniform distribution.

Our goal is to minimize m .

Theorem 3 $\forall k \leq n, \epsilon > 0$, there exists a (k, ϵ) -extractor $\text{Ext} : \{0, 1\}^n \rightarrow \{0, 1\}^m$ with $m = k + d - 2 \log \frac{1}{\epsilon} - O(1)$ and $d = \log(n - k) + 2 \log \frac{1}{\epsilon} + O(1)$.

These bounds on m and d are tight up to a constant.

Proof: The proof is arguing by counting. Consider a fixed k -source X . Choose Ext at random.

$$\begin{aligned} \Pr[\text{Ext fails}] &\leq (\#\text{flat sources}) \cdot \Pr[\text{Ext fails for } X] \\ &\leq \binom{N}{K} 2^{\Omega(KD\epsilon^2)} \leq \left(\frac{Ne}{K}\right)^K 2^{\Omega(KD\epsilon^2)} \end{aligned}$$

We want to upperbound the latter expression with 1. This happens when $m = k + d - 2 \log \frac{1}{\epsilon} - O(1)$ and $D\epsilon^2 \geq 2 \log \left(\frac{Ne}{K}\right) = c(n - k) + c'$ for constants c, c' . This implies $d = \log(n - k) + 2 \log \frac{1}{\epsilon} + O(1)$. ■

2 Simulating BPP Using Weak Sources

Take any **BPP** algorithm $A(x, r)$, where x is an input, r is a random string. Suppose we do not have a source of perfectly random bits. Instead, we want to use a k -source Y . Our idea is to compute $A(X, \text{Ext}(Y, U_d))$ by the method of taking *majority vote* over all possible seeds. That is to compute:

$$\begin{aligned} &A(X, \text{Ext}(Y, s_1)) \\ &\quad \vdots \\ &A(X, \text{Ext}(Y, s_{2^d})) \end{aligned}$$

and then take the majority vote. Why does it work?

Suppose A has error probability less than γ . Also assume that Y, U_d are such that $A(X, \text{Ext}(Y, U_d))$ is ϵ -close to uniform U_m , where $m = |r|$. The error probability of $A(X, \text{Ext}(Y, U_d))$ is at most $\gamma + \epsilon$. So by Markov style of reasoning, the probability that the majority decision to be wrong, is at most $2(\gamma + \epsilon)$. We conclude that for our purpose of simulating **BPP** algorithms, it is enough to have access to a weak source of l -bit string with $l^{\Omega(1)}$ min-entropy.

3 Extractors from Hashing

A universal *hash* family $H = \{h\}$, where $h : \{0, 1\}^n \rightarrow \{0, 1\}^l$ and $l \leq n$ is such that:

- $\forall a \in \{0, 1\}^n$, for a random function h , $h(a)$ is uniform U_l .
- for any $a \neq b \in \{0, 1\}^n$,

$$\forall x, y \in \{0, 1\}^l, \Pr[h(a) = x \wedge h(b) = y] = \frac{1}{L^2}$$

$$(L = 2^l)$$

Theorem 4 (Leftover Hash Lemma) *If H is a universal hash family from n bits to l bits, where $l = k - 2 \log \frac{1}{\epsilon}$, then $\text{Ext}(x, h) = (h, h(x))$ is a (k, ϵ) -extractor.*

We can show this hash function with an $l \times n$ matrix (which is large). The other way is to use affine functions which is better since we need only to determine coefficients a, b . ($h_{a,b}(x) = ax + b$)

Here we have $m = k + d - 2 \log \frac{1}{\epsilon} - O(1)$ (optimal), but $|h| = O(n)$ ("bad").

Proof: We should show:

$$\begin{aligned} \Delta((H, H(X)), U_d \times U_l) &\leq \epsilon \\ \iff \frac{1}{2} \|((H, H(X)) - U_d \times U_l)\|_1 &\leq \epsilon \\ \iff \frac{1}{2} \sqrt{DL} \|((H, H(X)) - U_d \times U_l)\|_2 &\leq \epsilon \end{aligned}$$

(where the last implication is by Cauchy-Schwarz). Now look at the equation:

$$\|((H, H(X)) - U_d \times U_l)\|^2 = \text{Col}((H, H(X))) - 2\text{Col}(U_d \times U_l) + \text{Col}(U_d \times U_l)$$

On the other hand:

$$\text{Col}(U_d \times U_l) = \frac{1}{DL}$$

and,

$$\begin{aligned} \text{Col}(H, H(X)) &= \Pr_{h, h' \in H, x, x' \in X} [(h, h(x)) = (h', h'(x'))] \\ &= \Pr[h = h'] \cdot \Pr[h(x) = h(x')] \\ &= \text{Col}(H) \cdot (\Pr[x = x'] \cdot \Pr[h(x) = h(x') | x \neq x']) \\ &\leq \frac{1}{D} \left(\frac{1}{K} + \frac{1}{L} \right) \end{aligned}$$

To see why the last inequality holds, recall that $\log \frac{1}{\text{Col}(X)} = H_2 \geq H_\infty \geq k$. Taking $l = k - 2 \log \frac{1}{\epsilon}$ we get:

$$\text{Col}(H, H(X)) = \frac{1 + \epsilon^2}{DL}$$

So we come to the conclusion that:

$$\|((H, H(X)) - U_d \times U_l)\|^2 \leq \frac{1 + \epsilon^2}{DL} - \frac{1}{DL} = \frac{\epsilon^2}{DL}$$

Hence:

$$\Delta((H, H(X)), U_d \times U_l) \leq \frac{1}{2} \sqrt{DL} \cdot \sqrt{\frac{\epsilon^2}{DL}} = \frac{\epsilon}{2}$$

■