6.895 Randomness and Computation April 30, 2008

Lecture 22: The Leftover Hash Lemma and Explicit Extractors

Lecturer: Ronitt Rubinfeld *Scribe: Andy Drucker*

Recall the notion of a k-source:

Definition: A *k-source* X is a random variable (taking values in $\{0,1\}^n$, say, for some $n > 0$) such that, for all $x \in \{0,1\}^n$,

$$Pr[X = x] \leq 2^{-k}.$$

A random variable which is a k-source is also said to have *min-entropy* k.

Extractors are functions which take as input a k-source, and some 'pure' randomness (called the *seed*), and output nearly-uniform bits.

Definition: Let $n, m, k, d > 0$ be integers and let $\epsilon > 0$.
We say a function $Ext : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a (k,ϵ)-*seeded extractor* (or simply a (k,ϵ) *extractor*) if, for all k-sources X on n bits, we have

$$\Delta(Ext(X, U_d), U_m) < \epsilon,$$

where $\Delta(S,T)$ denotes the statistical distance between r.v.'s S and T, and where U_t denotes the uniform distribution on t-bit strings.

Why would an extractor be a useful thing to have? Arguably, it is difficult to find 'pure' randomness in nature, but we may still hope to find 'significant' randomness in the form of a random variable that is 'hard to predict' (which is precisely what is guaranteed by the definition of a k-source). If we can find 'just a little' pure randomness, namely d bits, we get an output from the extractor that can be used as input to a randomized algorithm.

To see this, consider a randomized algorithm $M(y,r)$ taking input y and m random bits r. Suppose that M computes language L with bounded-error: if $y \in L$, then for all y, $M(x,r)$ accepts with probability $> \frac{5}{6}$ over the random choice of r, while if $y \notin L, M(x,r)$ accepts with probability $< \frac{1}{6}$ over r.

Define the random function $F(r) = M(x,r)$. Suppose X is a k-source to which we have access and Ext a $(k, \frac{1}{6})$-extractor; by definition we have that

$$\Delta(Ext(X, U_d), U_m) < \frac{1}{6}.$$

Since statistical distance cannot be increased by application of a function (even a random function), we have that

$$\Delta(F(Ext(X, U_d)), F(U_m)) < \frac{1}{6},$$

so

$$|Pr[M(x, Ext(X, U_d)) = 1] - Pr[M(x, U_m)] = 1| < \frac{1}{6}.$$

It follows that M is a bounded-error probabilistic algorithm for L with completeness $\frac{2}{3}$ and soundness $\frac{1}{3}$, even when run on the extractor's output rather than on uniform randomness U_m. (Since the extractor output is almost as good as uniform randomness for practical purposes, we sometimes call it *pseudorandom*.)

If d is small, we can eliminate the need for any pure randomness in our scheme, by taking a sample $x \leftarrow X$ and computing the majority vote of $M(y, Ext(x,s))$ over all $s \in \{0,1\}^d$, although we need somewhat higher success probability in the algorithm M to conclude that we get a 'representative'

sample x with good probability. If d is logarithmic in $|y|$ and M is polynomial-time, this gives a polynomial-time scheme for computing L with X as the only source of randomness.

When we look for constructions of extractors, we have several goals. First, as usual we prefer explicit constructions. The probabilistic method gives us nearly optimal extractors, but this is unsatisfactory as our goal in building extractors is to precisely to reduce our reliance on randomness!

In terms of parameters, we would like m to be small, so that we get a lot of pseudorandom bits out of the extractor; d and k to be small, so that we don't have to put too much randomness in; ϵ to be small, so that the output is very nearly uniform; and we'd like to work with whatever values of n we need to.

As an indication of the most basic constraints, if one hopes to build a nontrivial extractor, one must have $m \leq d + k$; otherwise, defining a k-source that contains k uniformly random coordinates, one could 'stretch' $d + k$ random bits to obtain a greater number of almost-random bits, which is easily seen to be impossible.

The 'Leftover Hash Lemma' of Impagliazzo is a classic construction of extractors, based on pairwise independent function spaces. It achieves nearly optimal randomness extraction (i.e. $k + d$ is very close to m), at the cost of requiring a large seed length $d = \Omega(n)$. First, let us define pairwise independent function spaces.

Definition: A family H of functions $h : \{0,1\}^n \to \{0,1\}^l$ is *pairwise independent (p.i.)* if for all distinct $x, y \in \{0,1\}^n$ and for all $a, b \in \{0,1\}^l$,

$$Pr_h[h(x) = a \vee h(y) = b] = \frac{1}{(2^l)^2},$$

where the probability is taken over a uniform selection of h from H.

We now state and prove the main result.

Theorem (Leftover Hash Lemma): If the family H of functions $h : \{0,1\}^n \to \{0,1\}^l$ is pairwise independent, where $l = k - 2log(\frac{1}{\epsilon}) - O(1)$, then $Ext(x,h) = (h, h(x))$ is a $(k, \epsilon/2)$-extractor.

(Here we identify a function h with its index in H.)

Note: the seed length d is $log|H|$, and we assume $|H|$ is a power of 2. It is known that p.i. families H exist with $|H| = \Theta(n)$, and that this is best possible. Thus we cannot hope for logarithmic seed length by this approach. Note, however, that the output length $m = d + l$ is very nearly the input min-entropy $k + d$, when ϵ is a constant, so we extract almost all of the randomness of X.

Proof: Conceptually, the proof is broken into three steps. We define a measure of a random variable called its 'collision probability' and show that the output of Ext has low collision probability; we use this fact to show that the extractor is close to uniform in the l_2 metric; then we use a general inequality between l_2 and l_1 to conclude that the extractor is close to uniform in statistical distance.

Fix now any k-source X; we aim to show that $\Delta(Ext(X, U_d), U_m) < \epsilon/2$.

Given a random variable D, define the *collision probability*

$$Col(D) = Pr[x = y],$$

where x, y are drawn independently from D.

First, we claim that $Col(X) \leq \frac{1}{2^k}$. To see this, let p_{max} be the largest probability of any outcome of X. Then

$$Col(X) = \sum_x Pr[X = x]^2 \leq p_{max} \cdot \sum_x Pr[X = x] = p_{max} \leq \frac{1}{2^k}.$$

Let us now analyze $Col(Ext(X, H)) = Col((H, H(x)))$ where, abusing notation, H denotes a uniformly chosen $h \in H$. Let H' denote an independent copy of H, and let x, x' denote independent draws from X.

$$Col((H, H(x))) = Pr_{x,x',H,H'}[(H, H(x)) = (H', H'(x'))]$$

$$= Pr_{H,H'}[H = H'] \cdot Pr_{x,x'}[H(x) = H'(x')|H = H']$$

$$= Pr_{H,H'}[H = H'] \cdot Pr_{x,x',H}[H(x) = H(x')]$$

$$= Pr[H = H'] \cdot \Big(Pr[x = x'] + Pr[x \neq x] \cdot Pr[H(x) = H(x')|x \neq x'] \Big)$$

$$\leq \frac{1}{2^d} \cdot (\frac{1}{2^k} + Pr[H(x) = H(x')|x \neq x'])$$

(recalling that $Col(X) \leq \frac{1}{2^k}$ since X is a k-source)

$$= \frac{1}{2^d} \cdot (\frac{1}{2^k} + \frac{1}{2^l})$$

(since H is pairwise independent)

$$= \frac{1}{2^{d+l}} \cdot (\frac{1}{2^{2log(\frac{1}{\epsilon})+O(1)}} + 1)$$

$$\leq \frac{1}{2^{d+l}}(\epsilon^2 + 1).$$

We now turn to analyze the squared l_2 distance $||(H, H(x)) - U_m||_2^2$. Given any two probability distributions p, q,

$$||p - q||^2 = \sum_x (p_i - q_i)^2$$

$$= \sum_x p_x^2 + \sum_x q_x^2 - 2\sum_x p_x q_x.$$

Suppose now q is the uniform distribution $U_m = U_d \times U_l$; then the above becomes

$$\sum_x p_x^2 + \frac{1}{2^{d+l}} - 2 \cdot \frac{1}{2^{d+l}} \sum_x p_x$$

$$= Col(P) + \frac{1}{2^{d+l}} - 2 \cdot \frac{1}{2^{d+l}} = Col(P) - \frac{1}{2^{d+l}}.$$

Thus

$$||(H, H(x)) - U_m||_2^2 = Col((H, H(x))) - \frac{1}{2^{d+l}}$$

$$\leq \frac{1}{2^{d+l}}(\epsilon^2 + 1) - \frac{1}{2^{d+l}} = \frac{\epsilon^2}{2^{d+l}}$$

by our earlier work.

Using the general fact that for $v \in \mathbf{R}^K$, $||v||_1 \leq \sqrt{K}||v||_2$, we conclude

$$\Delta((H, H(x)), U_m) = \frac{1}{2}||(H, H(x)) - U_m||_1 \leq \frac{1}{2}\sqrt{2^{d+l}}||(H, H(x)) - U_m||_2$$

$$\leq \frac{1}{2}\sqrt{2^{d+l}}\sqrt{\frac{\epsilon^2}{2^{d+l}}} = \epsilon/2.$$

This completes the proof. $\diamond$

We note briefly that extractors can be viewed as bipartite graphs. Let $Ext : \{0,1\}^n \times \{0,1\}^k \rightarrow \{0,1\}^m$ be a (k, ϵ)-extractor; construct a bipartite multigraph $G_{Ext} = (U, V)$ with $U = \{0,1\}^n, V = \{0,1\}^m$, and include an edge (x, y) with for each $s \in \{0,1\}^k$ such that $Ext(x, s) = y$.

The condition that Ext is an extractor can be formulated in graph-theoretic terms as a statement about G_{Ext}, although we do not do so here. We remark, however, that the condition is similar to a bipartite version of the expansion property for graphs. The correspondence is not exact, but it is close enough that there has been a fruitful interplay between expander and extractor research.

We conclude with a table indicating major approaches to extractor construction (as of roughly 2006). The first two entries are benchmarks. For each method, we give a rough indication of seed and output lengths (in terms of n, the input length, and k, the required min-entropy), with ϵ held to a (small as desired) constant.

Method	**Seed Length** (d)	**Output Length**
optimal, nonconstructive	$log(n-k) + O(1)$	$k + d - O(1)$
needed for BPP simulation with weak randomness	$O(log(n))$	$k^{\Omega(1)}$
'spectral'	$O(n-k)$	n
p.i. hashing (this lecture)	$O(n)$	$k + d - O(1)$
'almost-p.i. hashing'	$O(log(n) + k)$	$(1+\gamma)k$, $\gamma \rightarrow 0$
'explicit 1'	$O(log(n))$	$(1+\gamma)k$, $\gamma \rightarrow 0$
'explicit 2'	$O(log^2(n))$	$k + d - O(1)$