CS 744: Pseudorandomness Generators

Lecture 13: Amplification of Linear Guesses

February 4, 2015

The following algorithm for generating a pseudorandom source from a one-way permutation follows from the work of Goldreich-Levin 89 and Levin 93. Note that we will work in the contrapositive form. [1]

Suppose $G : \Sigma^R \to \Sigma$ is a function such that $G(z)$ is a good guess for $\oplus_{i=1}^{R} x_i z_i$ for a fixed $x \in \Sigma^R$, then we describe an algorithm below which outputs *a short list* of elements from Σ^R which contains x with very high probability. In this sense, the bit-valued function G's success can be amplified to invert x.

The idea is to take k strings at random from Σ^R, and compute their G values. Since they are uniformly picked at random, they will be oriented fairly uniformly in the vector-space of R-length bit vectors. The sum of the inner products of these vectors with x (mod 2) can be used indicate which "side" of the vector space x lies on. If we collect this orientation information for all the R co-ordinates, we will have a good information about x itself.

To begin, the success of G is defined as

$$s_G = E_{z \sim \Sigma^R} \left[(-1)^{G(z) + \sum_{i=1}^{R} x_i z_i} \right].$$

This is the sum of the correct guesses minus the sum of the incorrect guesses.

1 Algorithm

Consider the following algorithm.

1. Construct a random bit matrix $B_{k \times R}$.

[1] This treatment is from Knuth v2. 3e.

2. Compute, for every $b \in \Sigma^k$,

$$
\begin{aligned}
h_1(b) &= \sum_{c\neq 0, c\in\Sigma^k} (-1)^{b\cdot c + G(cB+e_1)} \\
h_2(b) &= \sum_{c\neq 0, c\in\Sigma^k} (-1)^{b\cdot c + G(cB+e_1)} \\
&\dots \\
h_R(b) &= \sum_{c\neq 0, c\in\Sigma^k} (-1)^{b\cdot c + G(cB+e_R)},
\end{aligned}
$$

where $e_1, e_2, \dots, e_R$ are the unit vectors.

3. For each string b, output the R-long string

$$x(b) = [h_1(b) < 0][h_2(b) < 0] \dots [h_R(b) < 0],$$

where for any proposition p, $[p] = 1$ if y is true, and 0 otherwise.

2 Inverting 0^R

We show that $x = 0^R$ will probably be output whenever G is a good approximation to the constant function 0. Suppose $s_G = E((-1)^{G(z)})$ is positive, and k is sufficiently large, it suffices to prove that

$$\sum_{c\neq 0, c\in\Sigma^k} (-1)^{G(cB+e_i)} > 0$$

for all e_i, $1 \le i \le R$, with probability greater than 1/2.

Claim For a fixed $c \in \Sigma^k$, the string $d = cB$ is uniformly distributed. Since B is a random matrix, the probability that a bit in d is 0 is equal to that it is 1.

Claim Moreover, cB and $c'B$ are pairwise independent. When $c \neq c'$, the probability of a particular result (d, d') occuring as $(cB, c'B)$ is $\frac{1}{2^{2R}}$. (Why?)

Hence, for any e_i, by Chebyshev's inequality, the probability that s_G is negative is at most $1/((2^k - 1)(s_G)^2)$. Hence by the union bound, the prbability that $x(0)$ is non-zero is at most $R/((2^k - 1)(s_G)^2)$. If k is sufficiently large, then this error is small.

3 Inverting an arbitrary vector in Σ^R

Now we show that it is possible to invert any vector $x \in \Sigma^R$ using the above algorithm, if we are able to invert 0^R. First, consider the function $H(z) = (G(z) + z_j) \mod 2 = G((x + e_j) + z)$. The last equality says that it is possible to get the effect of complementing the j^{th} bit of x by adding the j^{th} bit of z to the output of $G(z)$.

Consider the above algorithm executed with H instead of G. Then step 2 computes

$$
\begin{aligned}
h_1(b) &= \sum_{c\neq 0, c\in\Sigma^k} (-1)^{b\cdot c + G(cB+e_1)+(cB+e_1)e_j} \\
h_2(b) &= \sum_{c\neq 0, c\in\Sigma^k} (-1)^{b\cdot c + G(cB+e_1)+(cB+e_2)e_j} \\
&\ldots \\
h_R(b) &= \sum_{c\neq 0, c\in\Sigma^k} (-1)^{b\cdot c + G(cB+e_R)+(cB+e_R)e_j},
\end{aligned}
$$