CS 880: Pseudorandomness and Derandomization 5/1/2013

Lecture 27: Pseudorandomness for Half-spaces

Instructors: Holger Dell and Dieter van Melkebeek Scribe: Alexi Brooks

DRAFT

A half-space is defined as a function $H_{v,\theta} : \{-1,1\}^n \to \{-1,1\}$ for some vector $v \in \mathbb{R}^n$ and some constant threshold value $\theta \in \mathbb{R}$. For any vector $x \in \mathbb{R}^n$, we define $H_{v,\theta}(x) = \text{sign}(\langle v, x\rangle - \theta)$. In other words, we report 1 if the projection of x onto v is greater than θ, and we report -1 if the projection is less than θ. This construction does not define the behavior of $H(x)$ in a border case where $\langle v, x\rangle = \theta$, but we will not require any particular behavior in this situation.

Because we are only interested in the sign, we may freely alter the magnitude of v (and scale θ to match). The following arguments will assume that $\|v\|_2 = 1$.

1 The Sandwiching Technique

We are interested in finding a pseudorandom generator for half-spaces. One natural possibility is the INW generator.[1] As we learned previously, for branching programs of width w and length n, we can use the INW generator to achieve an error of within ϵ using a seed length of $\log n \log \frac{nw}{\epsilon}$. In order to use this generator, however, we will need to find a way to translate half-spaces into branching programs.

1.1 Step 1: Partial Sums

Recall the structure of branching programs, where each node resides in a layer and has out-degree 2. Each of those outgoing edges forms a transition to the next layer of the program and correlates with setting a particular variable. An unrestricted branching program may refer to the same variable multiple times on a particular path, although the variable may only be set once. All variable determination in half-spaces is independent, due to the nature of the inner product of vectors, so we may use a restricted form of branching program in which each variable only appears once on any given path in the program. We may also set an ordering of the variables on each path, leading to a one-to-one/onto relation between the n positions in the half-space vector v and the n layers in the branching program. At each layer of the branching program, we define the following partial sum:

$$\sum_{j=1}^{i} x_j v_j$$

If, at $i = n$, this sum is greater than θ, we are in an accepting state. (Note that we have chosen to break ties in the direction of not accepting.) Unfortunately, the width w grows exponentially in n. This means we will need a seed length of at least n in order to generate a half-space with n dimensions.

[1] See lecture 20.

1.2 Step 2: Rounding

In the previous step, we were stymied by the width of the branching program. Here we will attempt to decrease the width without substantially affecting the function of the program. We will do so by rounding, combining nodes whose expectations are similar.

Due to the nature of branching programs, we can exactly calculate the probability of acceptance at every node in the program. At a given layer i, we arrange the nodes according to their probability of acceptance. The basic plan will be to subdivide the layer into intervals of fixed size, and to keep at most one vertex in each interval. We will choose an interval size of ϵ/n.

If we call our original branching program B, we will be interested in two derivative branching programs: B_{up} keeps the node with the highest probability of acceptance in each interval. B_{down} keeps the node with the lowest probability of acceptance. In each case, the outgoing edges from an interval are defined by the outgoing edges of the node we keep. So if a particular interval X contains a node x with outgoing edges to nodes in intervals Y and Z (and we keep x), then the new branching program will have edges $X \to Y$ and $X \to Z$. Because we round at every layer, the width is bounded by n/ϵ. We can thus use the INW generator for B_{up} or B_{down} with a seed length of roughly $\log^2 n$.

1.3 Step 3: The Sandwich

It should be clear that for any v, θ, x it is true that $B_{\text{up}} \geq B$ and $B_{\text{down}} \leq B$. This follows directly from the way B_{up} and B_{down} are defined. We make the following further claim:

$$\Pr[B_{\text{up}}(U) = 1] - \Pr[B_{\text{down}}(U) = 1] \leq 2\epsilon$$

We use "$= 1$" as a stand-in for "accepts'.

Proof. We address the case of B_{up}. B_{down} is a symmetric case; the same arguments will suffice there. Transforming a particular execution of B into one of B_{up}, we see that at any given layer transition, we alter the partial sum by at most the width of the interval, ϵ/n. That is, if we would have chosen a node with a probability of acceptance a, and we instead choose the node in the interval with the maximum acceptance probability b. The difference $b - a \leq \epsilon/n$. The overall error generated throughout the execution is then bounded thus:

$$\Pr[B_{\text{up}}(U) = 1] - \Pr[B(U) = 1] \leq n * \epsilon/n = \epsilon$$

Combined, we get the desired result of

$$\Pr[B_{\text{up}}(U) = 1] - \Pr[B_{\text{down}}(U) = 1] \leq 2\epsilon$$ □

We are not quite done here. We must still prove that any PRG which works for B_{up} and B_{down} will also work for B. In particular, we want to prove this fact for the INW generator.

1.4 Step 4: Proving the Generator

We want a proof which shows that any PRG which is ϵ-pseudorandom for B_{up} and B_{down} will also be ϵ-pseudorandom for B. We will show a constant approximation, where any PRG which is ϵ-pseudorandom for B_{up} and B_{down} will be 3ϵ-pseudorandom for B.

Proof. Let $D = G(U^n)$ be the distribution of size n produced by our candidate pseudorandom generator. Note that n here is not the seed length; the seed length is whatever seed length the generator G requires in order to produce an *output* length of n.

We wish to show a bound on the quantity

$$|\Pr[B(D) = 1] - \Pr[B(U) = 1]|$$

We will consider the symmetric bounds separately to avoid dealing with the absolute value signs. The argument is the same in each case, so we will only show the positive case here. By the definitions of B_{up} and B_{down}, we have the following inequality:

$$\Pr[B(D) = 1] - \Pr[B(U) = 1] \leq \Pr[B_{\text{up}}(D) = 1] - \Pr[B_{\text{down}}(U) = 1]$$

Because the distribution is ϵ-pseudorandom for the rounded programs, we can substitute the following:

$$\Pr[B(D) = 1] - \Pr[B(U) = 1] \leq \Pr[B_{\text{up}}(U) = 1] + \epsilon - \Pr[B_{\text{down}}(U) = 1]$$

Finally, we can draw the argument in Step 3 to simplify the right-hand side:

$$\Pr[B(D) = 1] - \Pr[B(U) = 1] \leq 3\epsilon$$

□

1.5 Result

The INW generator will work for half-spaces, with a required seed length of $\mathcal{O}(\log n \log \frac{n}{\epsilon})$.

2 The Invariance Principle

The core idea in this part of the lecture is that, when a distribution accumulates random variables, it usually approaches a Gaussian distribution.

2.1 Central Limit Theorem

Let $X_1, X_2, \ldots$ be identical random variables with a mean of 0 and a variance of (σ^2) for some finite σ. Let $S_n = \frac{x_1 + \ldots + x_n}{n}$. As $n \to \infty$, $S_n \to N(0, \sigma^2)$, where N is the Normal distribution with a mean of 0 and a variance of σ^2.

If you think back to the previous section on half-spaces, you can see an immediate application. As the length of the v, x vectors grows, we expect $\langle v, x \rangle \to N(0, \sigma^2)$. Unfortunately, the Central Limit Theorem tells us nothing about the convergence rate.

2.2 Barry-Esseen Theorem

Let $Y_1, \ldots, Y_n$ be independent (not necessarily identical) random variables with the following properties:

1. $\mathrm{E}[Y_i] = 0 \forall i$
2. $\sum \mathrm{E}[Y_i^2] = \sigma^2 \forall i$. (Equivalently, $\sum \mathrm{Var}(Y_i) = \sigma^2$.)

3. $\sum \mathrm{E}[Y_i^4] \leq \gamma$

Let $S_n = \frac{Y_1 + \ldots + Y_n}{\sigma}$. (Note that the denominator in this case is not n, but σ.) Then the "infinity" distance

$$d_\infty(S_n, N(0,1)) \leq \frac{\sqrt{\gamma}}{\sigma^2}$$

The infinity distance is defined as the supremum over $t \in \mathbb{R}$ of the difference in probabilities. It allows for the possibility that the maximum value might not properly exist, instead rising to unbounded height at either a limiting approach to a point or simply as t grows large. It is otherwise identical to statistical distance.

Given this bound, we may say that for some "nice" half-spaces, projections onto v will be close to a Normal distribution.

2.3 Corollary of the Barry-Esseen Theorem

Take $v \in \mathbb{R}^n$ with $\|v\|_2 \leq 1$ and $\|v\|_\infty \leq \epsilon$. (The latter inequality is to the "infinity norm", defined as $\|x\|_\infty = \max_i |x_i|$.) Then $d_\infty(\langle v, U \rangle, N(0,1)) \leq \epsilon$.

Proof. Draw x_i uniformly at random from $\{-1, 1\}$ and let $Y_i = v_i x_i$. Then $\langle v, U_n \rangle = \sum_{i=1}^n Y_i$. If the Barry-Esseen Theorem holds for this case, then we may simply apply it and have our desired result. We may check each required property of the theorem, and find that each holds:

1. $\mathrm{E}[Y_i] = v_i \mathrm{E}[X_i] = 0$
2. $\sum \mathrm{E}[Y_i^2] = \sum w_i^2 = \|w\|_2^2 = 1$
3. $\sum \mathrm{E}[Y_i^4] = \sum w_i^4 = \sum w_i^2 w_i^2 \leq \sum w_i^2 \epsilon^2 = \epsilon^2$

As described above, we apply the Barry-Esseen Theorem with $\sigma^2 = 1$ and $\gamma = \epsilon^2$, giving us

$$d_\infty(\langle v, U \rangle, N(0,1)) \leq \epsilon$$ □