CAS CS 599 B: Mathematical Methods for TCS

Lecturer: Mark Bun Spring 2022

Lecture Notes 8: Intro to Pseudorandomness, Bounded Independence

Reading.

- Vadhan, Pseudorandomness, §2.1-2.3, 3.5

Pseudorandomness is the study of objects that behave as if though they were random, despite being constructed using little or no randomness. Some examples:

- Prime numbers. There's nothing random about them, but in many respects they appear as though they were strewn about randomly within the natural numbers. For example, the Siegel-Walfisz Theorem implies that 25% of primes end in each of the digits $1, 3, 7, 9$, which is what you would expect if they were distributed randomly. This perspective was essential in Green and Tao's 2004 proof that the primes contain arbitrarily long arithmetic progressions.
- Pseudorandom distributions. The uniform distribution on $\{-1, 1\}^n$ is specified by a sample space of size 2^n. Many randomized algorithms do not need this degree of randomness, and can in fact be simulated using distributions with much smaller sample spaces. When these sample spaces are small enough, they can be efficiently enumerated over, removing the need for randomness entirely. This is the idea behind derandomization of algorithms using pseudorandom generators.
- Expander graphs. Expanders are sparse graphs whose connectivity properties make them look like random graphs. For instance, the destination of a short random walk on an expander will mimic a uniformly random vertex.

1 Examples of Randomness in Computation

Primality Testing. Most primaility tests used in practice are *probabilistic* tests, in that they make random decisions and may output the wrong answer with some probability. For example, the Miller-Rabin test is a poly-time algorithm with the following guarantee. On input a prime number, it always outputs "prime." But on input a composite number, it incorrectly reports "prime" with probability up to $1/4$. In complexity parlance, this places the primality problem in the class $\mathbf{coRP}$.

The AKS primaility testing algorithm is deterministic and poly-time, but slower in theory and practice.

Polynomial Identity Testing. Given two degree-d polynomials p, q over a field $\mathbb{F}$, is it the case that $p \equiv q$ as formal polynomials? There is a simple randomized algorithm for this problem. Let $S \subseteq \mathbb{F}$ be any set of size at least $2d$. Sample a point $x_0 \overset{\text{R}}{\leftarrow} S$ uniformly at random. If $p(x_0) = q(x_0)$ output "$p \equiv q$"; otherwise, output "$p \not\equiv q$". The algorithm always succeeds when $p \equiv q$, while the Schwartz-Zippel Lemma below shows that if $p \not\equiv q$, it fails with probability at most $d/|S| \leq 1/2$.

Lemma 1. *If f is a nonzero polynomial over a field $\mathbb{F}$, and $S \subseteq \mathbb{F}$ is finite, then*

$$\Pr_{x \leftarrow S}[f(x) = 0] \leq \frac{\deg(f)}{|S|}.$$

When one takes appropriate care to define how polynomials are given to algorithms as input, this algorithm shows that polynomial identity testing (PIT) is in $\mathbf{coRP}$. It is an important open problem to determine whether PIT also has an efficient deterministic algorithm.

MAX-CUT. Given an undirected graph $G = (V = [n], E)$ and subset $C \subseteq V$, define $\mathrm{cut}(C) = \{(i,j) \mid i \in C, j \notin C\}$. The MAX-CUT problem is to find a set $C \subseteq V$ maximizing $\mathrm{cut}(C)$.

This is a classic $\mathbf{NP}$-hard problem, but there is a simple randomized algorithm that approximates the maximum cut to within a factor of $1/2$. Just choose C at random by placing each vertex $u \in V$ into C independently with probability $1/2$. By linearity of expectation,

$$\mathbb{E}[|\,\mathrm{cut}(C)] = \sum_{e \in E} \Pr[e \text{ is cut}] = \frac{|E|}{2}.$$

Since the maximum cut in any graph has size at most $|E|$, this serves as an expected $1/2$-approximation.

2 Derandomized MAX-CUT via Pairwise Independence

Our randomized approximation algorithm for MAX-CUT was stated using independent random placements of vertices into C. But we didn't need the full brunt of uniform randomness. Letting r_i be the ± 1-indicator for whether vertex i is included in C, all we needed was

$$\forall i \neq j \qquad \frac{1}{2} = \Pr[(i,j) \text{ is cut}] = \Pr[r_i \neq r_j].$$

The motivates the following definition:

Definition 2. A sequence of random variables $r_1, \ldots, r_n \in \{-1,1\}^n$ are *pairwise independent* if for all $i \neq j$, the pair (r_i, r_j) is uniform over $\{-1,1\}^2$.

Recall that generating n truly uniform bits requires a sample space of size 2^n. We can n pairwise independent bits much more efficiently using a sample space of size $O(n)$.

Proposition 3. *Let $x_1, \ldots, x_k$ be i.i.d. uniform bits. For each subset $S \subseteq [k]$, with $S \neq \emptyset$, let*

$$r_S = \chi_S(x_1, \ldots, x_k).$$

Then the $2^k - 1$ random variables $(r_S)_{S \neq \emptyset}$ are pairwise independent.

Thus, we can generate $n = 2k - 1$ pairwise independent bits using $k = \log(n+1)$ uniform bits, which can be sampled using a sample space of size $2^k = n + 1$.

Proof. It suffices to show that

1. $\mathbb{E}[r_i] = 0$ (the bits are unbiased), and

2. $\mathbb{E}[r_i r_j] = 0$ (every pair of bits is independent).

For the first condition, $\mathbb{E}[r_S] = \mathbb{E}[\chi_S] = 0$ since S is nonempty. For the second condition, let $S \neq T$ be nonempty sets. Then $\mathbb{E}[r_S r_T] = \mathbb{E}[\chi_S \chi_T] = \mathbb{E}[\chi_{S \Delta T}] = 0$. $\square$

Pairwise independence not only lets us reduce the randomness used in our MAX-CUT approximation; it lets us completely eliminate it. The idea is to simply enumerate over the entire (small) sample space needed to generate pairwise independent r.v.'s.

Deterministic $1/2$-approximation to MAX-CUT:
On input $(V = [n], E)$:

```
1. Let $k = \lceil \log n + 1 \rceil$. For every sequence of bits $x_1, \ldots, x_k \in \{-1, 1\}^k$:
2.     Run the randomized algorithm for MAX-CUT using $(r_S = \chi_S(x))_{S \neq 0}$ as the random choices.
3. Output the largest cut constructed.
```

This works because the pairwise independent MAX-CUT algorithm guarantees that

$$\mathop{\mathbb{E}}_{x_1,\ldots,x_k}[\mathrm{cut}(S)] = \frac{|E|}{2}.$$

Hence, there must *exist* an assignment to the bits $x_1, \ldots, x_k$ yielding a cut S whose value is at least this expectation.

3 Bounded Independence

The definition of pairwise independence extends naturally to k-wise independence for larger k.

Definition 4. Random variables $r_1, \ldots, r_n \in \{-1, 1\}^n$ are k-wise independent if, for every $|S| = k$, we have that $(r_i)_{i \in S}$ is uniformly random in $\{-1, 1\}^k$, i.e., for every $t \in \{-1, 1\}^k$, we have $\Pr[(r_i)_{i \in S} = (t_1, \ldots, t_k)] = 2^{-k}$.

Proposition 5 (Bounded independence, equivalent definition for bits)**.** *Random variables* $r_1, \ldots, r_n \in \{-1, 1\}^n$ *are* k*-wise independent if and only if for every* $0 < |S| \leq k$, *we have* $\mathbb{E}[\chi_S(r_1, \ldots, r_n)] = 0$.

Proof. For the forward direction, suppose $r_1, \ldots, r_n$ are k-wise independent. Then for every $0 < |S| \leq k$.

$$\mathbb{E}[\chi_S(r)] = \mathbb{E}\left[\prod_{i \in S} r_i\right] = \prod_{i \in S} \mathbb{E}[r_i] = 0.$$

For the reverse direction, suppose $\mathbb{E}[\chi_S(r_1, \ldots, r_n)] = 0$ for every $0 < |S| \leq k$. Let $|S| = k$ and $t \in \{-1, 1\}^k$. Let

$$f_t(r) = \begin{cases} 1 & \text{if } (r_i)_{i \in S} = (t_1, \ldots, t_k) \\ 0 & \text{otherwise.} \end{cases}$$

Then

$$\begin{aligned}
\Pr[(r_i)_{i\in S} = (t_1,\ldots,t_k)] &= \mathop{\mathbb{E}}_r [f_t(r)] \\
&= \sum_{T\subseteq S} \widehat{f}_t(T) \mathop{\mathbb{E}}_r [\chi_T(r)] \\
&= \widehat{f}_t(\emptyset) \\
&= \mathop{\mathbb{E}}_{x\sim\{-1,1\}^n} [f_t(x)] \\
&= 2^{-|S|}.
\end{aligned}$$

□

Definition 6. Random variables $r_1,\ldots,r_n \in [m]$ are *k-wise independent* if for all $|S| = k$, we have that $(r_i)_{i\in S}$ is uniformly distributed in $[m]^k$.

Theorem 7 (Construction of k-wise independent r.v.s)**.** *Let $\mathbb{F}$ be a finite field. There exists an efficiently computable collection of k-wise independent random variables for $n = m = |\mathbb{F}|$ that can be sampled using $k\log|\mathbb{F}|$ uniformly random bits.*

Proof. Let $a_0,\ldots,a_{k-1}$ be uniformly random elements of $\mathbb{F}$. For each $y \in \mathbb{F}$, let

$$r_y = a_0 + a_1 y + a_2 y^2 + \cdots + a_{k-1}y^{k-1}.$$

We claim that $(r_y)_{y\in\mathbb{F}}$ is k-wise independent. To do this, we will show that for all sets $S \subseteq \mathbb{F}$ with $|S| = k$ and all $t_1,\ldots,t_k$, there is a *unique* polynomial p of degree $k-1$ such that $p(y_i) = t_i$ for all $y_i \in S$. This suffices because it implies that

$$\Pr[(r_y)_{y\in S} = (t_1,\ldots,t_k)] = \frac{1}{|\mathbb{F}|^k} = \frac{1}{m^k}.$$

We show that such a polynomial p exists by constructing it using the Lagrange interpolation formula:

$$p(y) = \sum_{i=1}^{k} t_i \cdot \prod_{j\neq i} \frac{y - y_j}{y_i - y_j}.$$

It's unique because the construction of this polynomial represents a surjective map from the $\mathbb{F}^k$ (the t's) to $\mathbb{F}_k$ (coefficients of the polynomial), which must be a bijection. □