6.5420 Randomness and Computation March 5th, 2026

Lecture 8: Pairwise Independence

Lecturer: Ronitt Rubinfeld *Scribe: Eric Zhan*

1 Overview

Recall the Max-Cut problem:

Problem 1. *Given a graph $G = (V, E)$, find a partition of $V = S \sqcup T$ to maximize $|\{(u, v) \in E | u \in S, v \in T\}|$; that is, maximize the number of edges crossing the cut.*

Note that there is a simple randomized algorithm to get $|E|/2$: for each $v \in V$, flip a coin and assign it to either S or T at random.

The naive way to derandomize this algorithm would be to use **enumeration**: simply run it on every possible random string r_n (the number of randomized bits required by our algorithm). In Max-Cut, this is equivalent to brute forcing every single possible assignment of vertices to S or T and picking the largest one. Here, this would require checking $2^{|V|}$ cases, which is obviously too slow.

How can we do better?

Suppose we have a randomized algorithm $\mathcal{A}$. At a high level, our approach is to create a "randomness generator" that takes m truly random bits (*short random seed*) and generates n semi-random bits that are good enough to run our algorithm on, while not changing the expected value of the output of $\mathcal{A}$. Then, for each possible seed $b_1 \dots b_m$, we feed the output of our randomness generator $r_1 \dots r_n$ into our randomized algorithm (be it Max-Cut or another algorithm). Finally, we take the best output from all of the runs of our algorithm (enumeration). Note that by the definition of expectation, at least one of these runs must output a value at least $\mathbb{E}[\mathcal{A}]$, so we will have successfully derandomized $\mathcal{A}$.

The running time of such an algorithm would be $2^m \cdot T(\mathcal{A})$, where $T(\mathcal{A})$ is the running time of $\mathcal{A}$. The goal is to show that it is possible to find a suitable randomness generator with $m \sim \log n$.

2 Pairwise Independence

Definition 2. *Given a probability domain T, let $t = |T|$. Pick $x_1 \dots x_n$ such that $x_i \in T$. Then, under the uniform distribution,*

1. *$\{x_i\}$ are **totally independent** if for every possible choice $b_1 \dots b_n \in T^n$,*

$$Pr[(x_1 \dots x_n) = (b_1 \dots b_n)] = \frac{1}{t^n}.$$

2. *$\{x_i\}$ are **pairwise independent** if $\forall i \neq j$, $b_i, b_j \in T^2$,*

$$Pr[(x_i, x_j) = (b_i, b_j)] = \frac{1}{t^2}.$$

Remark. Note that the above definition can be easily generalized into k-wise independence, which constrains every possible set of k vertices instead of just pairs.

Example 3. *Consider three random bits r_1, r_2, r_3. Distribution A, which is totally independent, picks one out of the 8 assignments to r_1, r_2, r_3 uniformly at random. Distribution B picks out of the four assignments $(0,0,0), (0,1,1), (1,0,1), (1,1,0)$. (In other words, x_3 is the logical XOR of x_1 and x_2.)*

Observe that in example 3, distribution B is pairwise independent: for each possible pair r_i, r_j, B chooses from $(0,0), (0,1), (1,0), (1,1)$ uniformly at random. So any possible pair shows up with probability $1/4$. However, B is *not* totally independent, since $\Pr[r_1 r_2 r_3 = 000] = 1/4 \neq 1/8$.

3 Derandomizing Max-Cut

Consider a graph of 3 vertices v_1, v_2, v_3, which edges between (v_1, v_2) and (v_2, v_3). There are eight possible cuts:

1. $S = \{v_1, v_2, v_3\}$ and $S = \emptyset$ gives a cut value of 0.
2. S $= \{v_1, v_2\}, \{v_2, v_3\}, \{v_1\}, \{v_3\}$ gives a cut value of 1.
3. S $= \{v_1, v_3\}, \{v_2\}$ gives a cut value of 2.

An algorithm sampling r_1, r_2, r_3 will give an expected cut value of 1. However, note that if we use distribution B instead, we still get a cut value of 1!

Note that for *any* graph, we can run our randomized Max-Cut algorithm by sampling from a pairwise independent distribution of random bits instead of uniform. Indeed,

$$\mathbb{E}[\text{cut size}] = \mathbb{E}[\sum_{(u,v)\in E} \mathbf{1}((u,v) \text{ crosses cut})] = \sum_{(u,v)\in E} \mathbb{E}[\mathbf{1}(r_u r_v \in (01, 10))] = \frac{|E|}{4} + \frac{|E|}{4} = \frac{|E|}{2}$$

since each of the indicator variables only depends on two random bits r_u and r_v.

3.1 Randomness Generator for Bits

Begin by choosing m truly random bits $b_1 \dots b_m$. For all possible subsets $S \subseteq [m]$ with $S \neq \emptyset$, define

$$C_S = \bigoplus_{i\in S} b_i = \left(\sum_{i\in S} b_i\right) \mod 2.$$

Claim 4. *The variables C_S are pairwise independent.*

The proof of the above claim is left as a homework problem.

Note that to generate n pairwise independent bits, we only need $O(\log n)$ truly random bits, so we are done! We have showed that we can derandomize Max-Cut.

3.2 Randomness Generator for Integers

How can we generate random integers from $0 \dots q-1$ for some prime q?

First approach: generate each bit randomly and independently using the above construction. Each bit requires $\log q$ bits of randomness, and we need $\log q$ bits, so the overall number of random bits grows with $O(\log^2 q)$.

Better approach:

- Pick $a, b \in \mathbb{Z}_q$ uniformly at random.

- For each $x \in \mathbb{Z}_q$, set $r_x = ax + b \mod q$.
- Return $r_1 r_2 \dots r_q$.

Interpretation: we are picking a random *hash function* $h_{ab} : \mathbb{Z}_q \to \mathbb{Z}_q$ from a *family* of functions $\mathcal{H} = \{h_{ab} | a, b \in \mathbb{Z}_q\}$.

Definition 5. *A family of hash functions $\mathcal{H}$ is pairwise independent (or strongly 2-universal) if, when choosing $h \in_{\mathcal{U}} \mathcal{H}$ with $h : [N] \to [M]$,*

- *$\forall x \in [N]$, $h(x) \in_{\mathcal{U}} [M]$.*
- *$\forall x_i \neq x_j \in [N]$, $h(x_1), h(x_2) \in_{\mathcal{U}} [M]^2$.*

Remark. Note that in Definition 5, we are sampling over *hash functions* h. In other words, for every *fixed* $x_1, x_2 \in [N]^2$ and $y_1, y_2 \in [M]^2$, over all $h \in \mathcal{H}$, exactly $|\mathcal{H}|/M^2$ of these hash functions satisfy $h(x_1) = y_1$ and $h(x_2) = y_2$.

Remark. Definition 5 only makes sense for a *family* of functions. A single hash function h obviously cannot be pairwise independent.

Claim 6. *The family $\mathcal{H} = \{h_{ab} | a, b \in \mathbb{Z}_q\}$ from above is pairwise independent.*

Proof. For any $x \neq w$, c, and d, we want to show

$$\Pr[h_{ab}(x) = c \cap h_{ab}(w) = d] = \frac{1}{q^2}.$$

We can rewrite this condition as

$$\begin{pmatrix} x & 1 \\ w & 1 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} c \\ d \end{pmatrix}.$$

Since $w \neq x$, the 2×2 matrix on the left is invertible. Additionally, since we are working in the field $\mathbb{Z}_q$, we have unique modular inverses. Thus, there is exactly one solution to a, b that satisfies the equation.

There are q^2 total possible values for a, b, so the probability that the condition holds is $1/q^2$, as desired. □