5. Pairwise independence

5.1 Introduction

In the last lecture we saw two ways to derandomize. One was the technique of conditional expectation which works in cases where we can compute the conditional expectation efficiently. The second was the method of pairwise independence. In this lecture we explore pairwise independence further and define pairwise independent hash families.

5.2 Pairwise independence

A set of random variables $R_1, R_2, \ldots, R_t$ is pairwise independent if each R_i is unbiased, and for any two R_i, R_j, R_i and R_j are independent. Notice that if the random variables are pairwise independent, then $\Pr[R_u = 1 \wedge R_v = 0] = \Pr[R_u = 1]\Pr[R_v = 0 \mid R_v = 1] = \Pr[R_u = 1]\Pr[R_v = 0]$. Therefore it is sufficient for us to use pairwise independent random variables in the MAX-CUT algorithm. The interesting thing is that to construct n pairwise independent bits, it is sufficient to have $O(\log n)$ independent bits. We will see the construction now.

Let $b_1, b_2, \ldots, b_k$ be k binary unbiased random bits. For each $S \subset [k]$ where S is non-empty, let $R_S = \oplus_{i \in S} b_i$. Notice that R_S is unbiased (why?). Also, for $S \neq T$, $R_S = R_{S \cap T} \oplus R_{S \setminus T}$ and $R_T = R_{S \cap T} \oplus R_{T \setminus S}$. Now $R_{S \cap T}$ is common in both R_S and R_T and is unbiased. Similary, at least one of $R_{S \setminus T}$ or $R_{T \setminus S}$ is non-empty and unbiased. Therefore, R_S and R_T are independent.

5.3 Pairwise independent hash families

In many applications in computer science, it is necessary to get pairwise independent functions. For instance we might want hash functions $H : \{0,1\}^n \to \{0,1\}^m$, i.e. 2^n pairwise independent random variables, each over the set 2^m. If we were naively use the construction in the previous section, and repeat for each of the m bits, then we need nm initial randomness. To get 2^n pairwise independent random bits, we need n truly random bits. Now for each of the m bits we can use this to get the 2^n random variables over the set $\{0,1\}^m$. But, we can do much better here. We will look at such constructions of more general pairwise independent hash functions now.

Definition 5.1 (Pairwise independent hash family)**.** *A set $\mathcal{H} = \{h : [N] \to [M]\}$ of functions is a pairwise independent hash family if it satisfies the following two conditions: For every $x_1 \neq x_2 \in [N]$ and $y_1, y_2 \in [M]$, $\Pr_{h \in_r \mathcal{H}}[H(x_1) = y_1 \wedge H(x_2) = y_2] = \frac{1}{M^2}$.*

Suppose we take the set of all functions from $[N]$ to $[M]$. Then is it a pairwise independent hash family?

Lemma 5.2. *The set of all functions from $[N]$ to $[M]$ is a pairwise independent hash family.*

Proof. Let $x_1 \neq x_2 \in [N]$ and let $y_1, y_2 \in [M]$. The number of functions $h : [N] \to [M]$ such that $h(x_1) = y_1$ and $h(x_2) = y_2$ is M^{N-2}. The total number of functions from $[N]$ to $[M]$ is M^N. Therefore $\Pr_{h \in_r \mathcal{H}}[h(x_1) = y_1 \wedge h(x_2) = y_2] = M^{N-2}/M^N = 1/M^2$. □

This is not useful for the purposes of derandomization even though it is very explicit. This is because, to represent a function from this family it require $N \log M$ bits and that is too large, say when $N = \{0,1\}^n$. Typically we require functions that can be represented with $\text{poly}(\log N, \log M)$ bits (In the previous section, we saw an explicit family that can be represented with $\log N \cdot \log M$ bits.

Before we go into the construction of pairwise independent hash families, we will take a small detour into finite fields.

5.3.1 A brief introduction to finite fields

Definition 5.3. *A field $(F, +\times)$ consists of a set F and two binary operations $+$ and $\times$ defined on F that satisfies the following conditions.*

- *$(F, +)$ is a commutative group with identity 0.*
- *$(F \setminus \{0\}, \times)$ is a commutative group with identity 1.*
- *$+$ distributed over $\times$.*

The set of real numbers $\mathbb{R}$ is a field, wherea the set of integers $\mathbb{Z}$ is not a field. We will only look at finite fields in this course. They have a lot nice structural properties that we will find useful.

Fact 5.4. *For any field F, $(F, \times)$ is a cyclic group.*

Fact 5.5. *For every prime p, $\mathbb{Z}_p$ is a field under addition and multiplication modulo p. Also, $\mathbb{Z}_m$ is not a field for any composite m.*

Fact 5.6. *For every prime p, and natural number n, there is a field with p^n elements. We will denote this field by $\mathbb{F}_{p^n}$ since for a fixed p and n this field is unique up to isomorphism.*

For a field $\mathbb{F}$, the *characteristic* of the field is the least integer n such that adding the multiplicative identity n times gives 0 (the additive identity). All finite fields have a number as its characteristic. It is also important for us to see the construction of finite fields.

Construction of finite fields of characteristic p: Let n be an integer, and $\mathbb{F}_p$ the finite field with p elements. Let $\mathbb{F}_p[X]$ denote the set of polynomials where the coefficients come from the field $\mathbb{F}_p$. Let $Q(X)$ be an *irreducible* polynomial of degree n over $\mathbb{F}_p$. Then $\mathbb{F}_p/Q(X)$ is the finite field of size p^n. The field $\mathbb{F}_p/Q(X)$ consists of polynomials of degree at most n over $\mathbb{F}_p$ with the addition and multiplication modulo the polynomial $Q(X)$. For any field $\mathbb{F}_p$ and an integer n, there exists an irreducible polynomial in $\mathbb{F}_p[X]$. Moreover, it can efficiently computed in time $\text{poly}(p, k)$. For now we will look at the field $\mathbb{F}_2$ which contains the elements 0 and 1.

Example 5.7. *The field $\mathbb{F}_4$ is obtained as follows. Consider the set of polynomial $\mathbb{F}_2[X]$. You can check that X^2+X+1 is irreducible over $\mathbb{F}_2$. In fact, X^2+X+1 is the only degree two irreducible polynomial over $\mathbb{F}_2$. So, $\mathbb{F}_4 = \mathbb{F}_2/(X^2+X+1)$. Another way to look at $\mathbb{F}_4$ is as a* vector space *over the field $\mathbb{F}_2$. In fact $\mathbb{F}_4$ consists of the following four polynomials, $0, 1$ (it contains $\mathbb{F}_2$ as a sub-field), X and $X+1$. This can also be thought of as the following tuples $0 \to (0,0)$, $1 \to (0,1)$, $X \to (1,0)$ and $X+1 \to (1,1)$. For X and $X+1$ in $\mathbb{F}_4$, the multiplication is given by $X(X+1) mod (X^2+X+1) = 1$. Therefore, the multiplication of the vectors $(1,0)$ and $(1,1)$ in $\mathbb{F}_4$ gives $(0,1)$.*

5.3.2 An explicit hash function family

Definition 5.8. *Let $\mathbb{F}$ be any finite field. For $a, b \in \mathbb{F}$, let $h_{a,b} : \mathbb{F} \to \mathbb{F}$ be defined as $h_{a,b}(x) = ax + b$. Let $\mathcal{H} = \{h_{a,b} \mid a, b \in \mathbb{F}\}$.*

Observe that the hash family has $|\mathbb{F}|^2$ many elements and hence can be represented by $2 \log |\mathbb{F}|$ bits.

Lemma 5.9. *The family $\mathcal{H}$ of functions is a pairwise independent hash family.*

Proof. Let $x_1 \neq x_2 \in \mathbb{F}$, and $y_1, y_2 \in \mathbb{F}$. Let a and b be two unknowns such that we want $y_1 = ax_1 + b$ and $y_2 = ax_2 + b$. Therefore $y_1 - y_2 = a(x_1 - x_2)$. Since $\mathbb{F}$ is a field, there is exactly one value of a that satisfies this. Fixing that value for a, there is exactly one value for b. □

In many applications we want a hash family from the set $\{0,1\}^n$ to $\{0,1\}^m$ where $m < n$. So we consider hash functions from $\mathbb{F}_{2^n} \to \mathbb{F}_{2^m}$ constructed as follows: Let $a \in \{0,1\}^n$ and $b \in \{0,1\}^m$, $h_{a,b}(x) = (ax)|_m + b$ where $y|_m$ is truncating the bit strings to m bits. Here, we think of a and x as elements of the field $\mathbb{F}_{2^n}$, perform the multiplication in the field $\mathbb{F}_{2^n}$ polynomials and map them back to binary strings. This hash family can be represented by $m + n$ bits.