CS761 Derandomization and Pseudorandomness **2022-23 Sem I**

Lecture 3: 09/08/22

Scribe: Raj Aryan Agrawal *Lecturer: Rohit Gurjar*

Pairwise Independence

Definition 3.1 (Pairwise Independence)**.** *Let $X_1, X_2, \ldots, X_n$ be random variables such that for any i, j $i \neq j$, we have*

$$\forall \alpha, \beta \ Pr(X_i = \alpha | X_j = \beta) = Pr(X_i = \alpha)$$

Or we can write it as

$$Pr(X_i = \alpha, X_j = \beta) = Pr(X_i = \alpha) Pr(X_j = \beta)$$

This can be extended to k-wise independence.

Notation: We will use $X \in_R S$ to denote that the random variable X is chosen uniformly randomly from S.

Claim 3.2. *Choose $X_1 \in_R \{0,1\}$ and $X_2 \in_R \{0,1\}$ and set $X_3 = X_1 \oplus X_2$ (where $\oplus$ is the XOR operation). Then the variables X_1, X_2, X_3 are pairwise independent.*

To see the claim verify that $\Pr(X_3 = \alpha | X_i = \beta) = 1/2$ for $i = 1, 2$ and for any $\alpha, \beta \in \{0,1\}$.

Max Cut Problem

Problem 3.3 (Max Cut Problem)**.** *Given a graph $G = (V, E)$, the max cut problem is to find a partition of $V = A \cup B$ such that*

$$|\{(v_1, v_2) \in E \mid v_1 \in A, v_2 \in B\}|$$

is maximized.

That is, we need to find a cut with the maximum number of cut edges. The problem is known to be NP-hard, so we don't expect an efficient algorithm that gives an exact answer. Let us see a simple randomized algorithm that gives a good approximation.

Randomized (1/2)-Approximation Algorithm

Algorithm 3.4. *For each vertex $v \in V$, put it in set S with probability $1/2$, independently. Then the set of cross edges is*

$$\delta(S) = \{(u, v) | u \in S, v \notin S\}.$$

The partition is then S and $V - S$.

Claim 3.5. *In expectation, this algorithm gives the size of the cut within 1/2 approximation of the maximum cut. That is*

$$E[|\delta(S)|] \geq \frac{MaxCut}{2}$$

Proof. For each edge $e \in E$, define

$$X_e = \begin{cases} 1 & if \ e \in \delta(S) \\ 0 & otherwise \end{cases}$$

First, let us note that $\Pr[X_e = 1] = 1/2$ because for any edge, its two endpoints will fall into different parts with probability $1/2$. Hence, $\mathbb{E}[X_e] = 1/2$. Now, the expected number of cross edges would be

$$\mathbb{E}[|\delta(S)|] = \mathbb{E}\left[\sum_{e\in E} X_e\right] = \sum_{e\in E}\mathbb{E}[X_e] = \sum_{e\in E} 1/2 = \frac{|E|}{2}$$

by using linearity of expectation. Since $MaxCut \leq |E|$, we have

$$\mathbb{E}[|\delta(S)|] \geq \frac{MaxCut}{2}$$

□

Question. In the above analysis, have we really used the fact that the vertices were put in S *independently* of each other? The answer is no. Linearity of expectation certainly does not need any kind of independence assumption. So, the only crucial part is to find the probability $\Pr[X_e = 1] = 1/2$. This probability will be $1/2$, as long as the two endpoints fall in S randomly and *independently* of each other. In other words, we only need that any two vertices are independent of each other. We do not need complete independence. And that's exactly pairwise independence.

Here is a modified algorithm.

Algorithm 3.6. *Generate n pairwise independent random bits, say $b_1, b_2, \ldots, b_n$. Put the ith vertex in S if and only if $b_i = 1$.*

The advantage is that to generate n pairwise independent bits, we need much fewer random bits than n. Let's see one such generation procedure.

Generating pairwise independent bits

The idea is inspired from the XOR example seen above.

Algorithm 3.7. *Take k independent random bits $b_1, b_2, \ldots, b_k$, we can generate $2^k - 1$ random variables defined as, $\forall S \subseteq \{1, 2, \ldots, k\}$, $S \neq \phi$, define*

$$b_S = \bigoplus_{i\in S} b_i$$

The correctness of the construction can be seen by the below 2 propositions

Proposition 3.8. *$Pr[b_S = 1] = \frac{1}{2}$ $\forall S \subseteq \{1, 2, \ldots, k\}$.*

Proposition 3.9. *For any $S \neq T$ and for any $\alpha, \beta \in \{0, 1\}$, $Pr[b_S = \alpha \mid b_T = \beta] = \frac{1}{2}$*

For the latter, we can argue that if we fix all bits of T, we still have some bits in S which are randomly set, which gives probability of $1/2$. This argument only works when $S \not\subseteq T$. If $S \subset T$, then we will argue that b_T is independent of b_S, which is equivalent to saying that b_S is independent of b_T.

Thus our algorithm can run using only $\log |V|$ number of random bits. Now, we will use this fact crucially to make the algorithm completely deterministic. The idea is that any randomized algorithm using k bits can be simulated deterministically with a 2^k blow up in the running time.

Deterministic Algorithm

Algorithm 3.10. *Enumerate over all choices of $(b_1, b_2, \ldots, b_k) \in \{0, 1\}^{\log |V|}$, and for each choice, and assign the vertices according to the random bits $\{b_T\}$ $\forall T \subseteq \{0, 1\}^{\log |V|}$.*

That is, we have $2^{\log |V|} = |V|$ many choices for $(b_1, b_2, \ldots, b_k)$ and thus, we have the same number of partitions. The guarantee we have is that the expectation of the cut-set size over these $|V|$ many partitions is at least $|E|/2$. In particular, one of these partitions will have cut-set size at least $|E|/2$. Note that this deterministic algorithm does not even look at the input graph. It is just going over a pre-determined small set of partitions and one of them is guaranteed to be large enough.

The following example will be helpful. Suppose $|V| = 8$. Then we will take three bits b_1, b_2, b_3, and go over all 8 possibilities for them. For each fixing of b_1, b_2, b_3, we will generate the following 8 bits:

$$0, b_1, b_2, b_3, b_1 \oplus b_2, b_2 \oplus b_3, b_1 \oplus b_3, b_1 \oplus b_2 \oplus b_3.$$

These eight bits will determine which vertex goes to which part. The following table describes different partitions of vertices we will go over. The vertices are indexed as $v_0, v_1, \ldots, v_7$.

v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_7	
0	b_1	b_2	b_3	$b_1 \oplus b_2$	$b_2 \oplus b_3$	$b_1 \oplus b_3$	$b_1 \oplus b_2 \oplus b_3$	Partition
0	0	0	0	0	0	0	0	$\{v_0, v_1, v_2, v_3, v_4, v_5, v_6, v_7\} \cup \phi$
0	1	0	0	1	0	1	1	$\{v_0, v_2, v_3, v_5\} \cup \{v_1, v_4, v_6, v_7\}$
0	0	1	0	1	1	0	1	$\{v_0, v_1, v_3, v_6\} \cup \{v_2, v_4, v_5, v_7\}$
0	0	0	1	0	1	1	1	$\{v_0, v_1, v_2, v_4\} \cup \{v_3, v_5, v_6, v_7\}$
0	1	1	0	0	1	1	0	$\{v_0, v_3, v_4, v_7\} \cup \{v_1, v_2, v_5, v_6\}$
0	0	1	1	1	0	1	0	$\{v_0, v_1, v_5, v_7\} \cup \{v_2, v_3, v_4, v_6\}$
0	1	0	1	1	1	0	0	$\{v_0, v_2, v_6, v_7\} \cup \{v_1, v_3, v_4, v_5\}$
0	1	1	1	0	0	0	1	$\{v_0, v_3, v_4, v_5\} \cup \{v_1, v_2, v_3, v_7\}$

Our guarantee is that no matter what the graph is on 8 vertices, one of the above 8 partitions will have at least $|E|/2$ cut edges.

We have seen that using $\log n$ independent random bits, we can generate n pairwise independent bits. Is this optimal? Can we do the same using a small number of random bits? It turns out that $\log n$ is indeed the optimal.

Proposition 3.11. *For any sequence of n pairwise independent random bits, the sample space size must be at least n.*

We will prove this in the next lecture.

Pairwise Independent Variables

Suppose we want to construct random variables which are not bits, but have a larger sample space. Can we still generate pairwise independent random variables. It will be convenient to see the sample space as a finite field. We will construct n random variables $X_1, X_2, \ldots, X_n \in \mathbb{F}$ for a finite field $\mathbb{F}$ and $n = |\mathbb{F}|$. We want that,

$$\forall \alpha \in \mathbb{F}, \forall i \ Pr(X_i = \alpha) = \frac{1}{|\mathbb{F}|}$$

and

$$\forall \alpha, \beta \in \mathbb{F}, \forall i, j \ Pr(X_i = \alpha, X_j = \beta) = \frac{1}{|\mathbb{F}|^2}$$

Claim 3.12. *Choose k independent random values $b_1, b_2, \ldots, b_k \in_R \mathbb{F}$. Define for any $S \in \{1, 2, \ldots, k\}$*

$$b_S = \sum_{i \in S} b_i$$

The summation is over the field. Th random variables $\{b_S : S \subseteq \{1, 2, \ldots, k\}, S \neq \phi\}$ are pairwise independent.

The argument for this is the same as before. Remember that we had used $\oplus$ for bits, which is simply the addition operation in $GF(2)$. Thus, our seed length is roughly $\log n \cdot \log |\mathbb{F}|$. Also, it is not clear if we can generalize this construction to k-wise independence. We will discuss a different construction which easily generalizes to k-wise independence and also has a better seed length.

Claim 3.13. *Randomly choose $a, b \in_R F$ and then generate the following random variables*

$$\{az + b : z \in \mathbb{F}\}.$$

These $|\mathbb{F}|$ random variables are pairwise independent.

To generalize to k-wise independence, we would use degree $k-1$ polynomial. We will discuss in the next lecture.