CS395T Pseudorandomness and Combinatorial Constructions
Lecture 5, January 30, 2001
Lecturer: Prof. David Zuckerman, Scribe: Jean-Philippe Martin

1 Pairwise Independence

This lecture focuses on the concept of *pairwise independence* and its implications for pseudo-random generators. We will start with a motivating example.

1.1 MAX CUT

Definition 1 *A* cut *of a graph $G = (V, E)$ is a partition of the vertices into two sets, S and $\bar{S}$. The* size *of a cut is the number of edges which have one end in S and the other in $\bar{S}$.*

The goal of the MAX CUT problem is, for a given graph G, to identify the cut of maximal size. This is an **NP**-complete problem, so we are only going to approximate the answer.

As a side note, there exists a greedy algorithm that performs relatively well: for each node n in turn, put n in S if, of the edges incident to n, more are connected to nodes already in $\bar{S}$ than to S. Otherwise, put n in $\bar{S}$.

1.2 Randomized Algorithm

Let $G = (V, E)$ be a graph. We present a randomized algorithm for the MAX-CUT of G. This is no better than the greedy algorithm, but it provides a good illustration for the use of pairwise independent variables.

S and $\bar{S}$ are initially empty.

For each vertex $v \in V$, put v in S with probability $\frac{1}{2}$ (otherwise, put it in $\bar{S}$).

$$X_e = \begin{cases} 1 & \text{if edge } e \text{ is in the cut} \\ 0 & \text{otherwise} \end{cases}$$

$$X = \sum_e X_e$$

Note that $EX = \sum E[X_e]$, regardless of whether the X_e are independent or not.

The expected performance of this algorithm is

$$E[\text{size(cut)}] = \sum_{e \in E} Pr[e \in \text{cut}] = \frac{1}{2}|E|$$

To implement this algorithm, we need $n = |V|$ random bits. Can we do with less? It turns out that yes, because we only used the fact that any pair of vertices are independent, which is weaker than the full randomness assumption.

Definition 2 *The random variables $Y_1, Y_2, ..., Y_n$ are* pairwise independent *if*

$$\forall i \neq j \quad Y_i \textit{ and } Y_j \textit{ are independent}$$

Definition 3 *The random variables X and Y are independent if*

$$\forall a, b \quad \Pr[X = a \wedge Y = b] = \Pr[X = a] \Pr[Y = b]$$

It is very important to note that pairwise independence *does not* imply mutual independence.

1.3 Small Pairwise Independent Probability Spaces

In this construction, we start from a seed of r random bits and produce $2^r - 1$ bits which are pairwise independent.

Suppose $n = 2^r - 1$ (for convenience) and choose a uniformly at random from all r-bit strings ($a \in_U \{0,1\}^r$).

For $i \in \{0,1\}^r - \{0\}^r$, let

$$Y_i = a \cdot i \pmod 2 = \sum_j a_j i_j \pmod 2 = \bigoplus_j a_j i_j$$

The $2^r - 1$ bits Y_i are pairwise independent.

Proof:

- The resulting bits are uniformly distributed:
  $\Pr[Y_i = 1] = \frac{1}{2}$

This can be seen by the following reasoning: i is a given non-zero string, so it contains at least one bit l such that $i_l \neq 0$. If we flip the bit a_l that corresponds to this position in the random a vector, then the resulting Y_i changes as well. Since a is uniform, then both values for Y_i are equally likely.

- The resulting bits have the following property:
  $\forall i \neq j \ \Pr[Y_i \oplus Y_j = 1] = \frac{1}{2}$

The argument here is similar. Since $i \neq j$ there exists a bit l where they differ, i.e. $i_l \oplus j_l = 1$. Let us rewrite the original expression to take advantage of this fact.

$$Y_i \oplus Y_j = \bigoplus_m a_m i_m \ \oplus \ \bigoplus_n a_n j_n = \bigoplus_m a_m (i_m \oplus j_m)$$

The expression above shows that if the value of a_l changes, then the whole of $Y_i \oplus Y_j$ changes as well. Since a is uniform, it follows that the probability for the whole expression to be 1 has to be $\frac{1}{2}$.

- The resulting bits are pairwise independent

Let us assign some probability p to the event $Y_i = 0 \wedge Y_j = 1$. Because Y_i is uniform, it follows that $\Pr[Y_i = 0 \wedge Y_j = 0] = 1/2 - p$ (as shown in the table below). Since Y_j is uniform, it is also true that $\Pr[Y_j = 0 \wedge Y_i = 1] = p$.

Pr	$Y_j = 0$	$Y_j = 1$
$Y_i = 0$	$1/2 - p$	p
$Y_i = 1$	p	$1/2 - p$

We know that $\Pr[Y_i \oplus Y_j = 1] = 1/2$. The table shows us that $\Pr[Y_i \oplus Y_j = 1] = 2p$. Therefore, $p = 1/4$ and we see that each cell in the table has probability $1/4$. So, Y_i and Y_j are independent. $\square$

We can therefore run the MAX CUT algorithm using only $r = \lceil \log_2(n+1) \rceil$ random bits, or derandomize it entirely by cycling over all $2^r = n + 1$ elements of the probability space.

1.4 Another Construction

This construction builds q different random values $Y_i \in \mathbb{F}_q$ (q is a prime or a power of a prime, $\mathbb{F}_q$ is a field) from 2 random values in the field.

1. Choose $a, b \in_U \mathbb{F}_q$.
2. For $i \in \mathbb{F}_q$, let $Y_i = ai + b$ (where $+$ and $\cdot$ are operations in $\mathbb{F}_q$). The Y_i are pairwise independent.

size(probability space) $= q^2$ ($q \geq n$).

Proof: To prove this result, it is sufficient to see that the following matrix defines a bijection between pairs (a, b) and pairs (Y_i, Y_j).

$$\begin{bmatrix} Y_i \\ Y_j \end{bmatrix} = \begin{bmatrix} i & 1 \\ j & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}$$

det$= i - j \neq 0$ so the matrix is invertible. For $x, y \in \mathbb{F}_q$, let

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} i & 1 \\ j & 1 \end{bmatrix}^{-1} \begin{bmatrix} x \\ y \end{bmatrix}$$

We can now conclude:

$\forall i \neq j \quad \Pr[Y_i = x \text{ and } Y_j = y] = \Pr[a = x' \text{ and } b = y'] = \Pr[a = x'] \Pr[b = y'] = q^{-2}$

$\square$

The resulting space has size q^2.

1.5 Generalization to k-wise independence

k-wise independence is a generalization of pairwise independence: instead of requiring that any pair of values are independent, we require that all variables in any subset of size k be independent.

The same construction can be used in this case:

1. Pick $a_0, a_1, ..., a_{k-1} \in \mathbb{F}_q$
2. $Y_i = a_0 + a_1 i + a_2 i^2 + a_3 i^3 + ... + a_{k-1} i^{k-1}$

The size of this space is q^k, $q \geq n$.

Proof: The proof flows exactly as before. This time, the matrix is a van der Monde matrix, which is invertible. For example:

$$\vec{Y} = \begin{bmatrix} 1 & i_1 & i_1^2 \\ 1 & i_2 & i_2^2 \\ 1 & i_3 & i_3^2 \end{bmatrix} \vec{a}$$

$\square$

While this construction is optimal for random variables in $\mathbb{F}_q$, there are different constructions for k-wise independent $\{0,1\}$ random variables that give probability spaces of size $O(n^{\lfloor k/2 \rfloor})$.

2 Amplifying the success probability

It is possible to amplify the success probability of a **BPP** algorithm by running it several times on independent random strings. In this section we show how to amplify the success probability of an algorithm using fewer random bits than this naive approach.

We suppose that we are working with a **BPP** algorithm that uses r random bits and gives the following answer when presented with a member of the language:
$\Pr[\text{“}yes''] \geq 2/3, \quad \Pr[\text{“}no''] \leq 1/3$

2.1 Pairwise Independent Bits

This section describes how to amplify the success probability using pairwise independence. This requires only D pairwise independent r-bit strings.

1. Choose $Y_1 ... Y_D$ pairwise independently from $\{0,1\}^r$ (we use a field of size $q = 2^r$, so the size of the pairwise space $= 2^{2r}$ and we use $2r$ independent random bits).
2. Let Z_i = output of the algorithm when using the random bits of Y_i ($Z_i \in \{0,1\}$).
3. $Z = \sum_{i=1}^{D} Z_i$
4. Output 1 if $Z > D/2$ and 0 otherwise.

What is the probability that this algorithm errs?
$EZ \geq \frac{2}{3}D$ for 'yes' input
$EZ \leq \frac{1}{3}D$ for 'no' input
$\Pr[\text{algorithm errs}] \leq \Pr[|Z - EZ| \geq \frac{1}{6}D]$

Theorem 1 *Chebychev's Inequality*

$$\Pr[|Z - EZ| \geq l\sigma] = \Pr[(Z - EZ)^2 \geq l^2\sigma^2] \leq \frac{1}{l^2}$$

The variance σ^2 *of* $Z \doteq E[(Z - EZ)^2] \doteq E[Z^2] - (EZ)^2$

Proposition 1 *The variance of the sum of pairwise independent variables is equal to the sum of the variances. Formally, for $Z_i ... Z_n$ pairwise independent,*

$$\sigma^2(\sum Z_i) = \sum \sigma^2(Z_i)$$

Proof:

$$E\left[\left(\sum Z_i\right)^2\right] = E\left[\sum_i Z_i^2\right] + 2E\left[\sum_{i<j} Z_i Z_j\right]$$

$$= \sum EZ_i^2 + 2\sum_{i<j}[(EZ_i)(EZ_j)] \text{ (because we are only considering pairs of variables)}$$

$$\left(E\left[\sum Z_i\right]\right)^2 = \sum EZ_i^2 + 2\sum_{i<j}[(EZ_i)(EZ_j)]$$

So σ^2 *of* $\sum Z_i =$ *(by definition)* $E(\sum Z_i)^2 - (E(\sum Z_i))^2 = \sum EZ_i^2 - \sum(EZ_i)^2 = \sum \sigma^2(Z_i)$ $\square$

We are now ready to compute the variance of our Z function and use Chebychev's inequality to derive an upper bound to the probability of error.

$$Z_i = \begin{cases} 1 \text{ with probability } p \\ 0 \text{ otherwise} \end{cases}$$

$$\sigma^2(Z_i) = E(Z_i^2) - (EZ_i)^2 = p - p^2 = p(1-p) \leq \frac{1}{3} \cdot \frac{2}{3} = \frac{2}{9}$$

$\sigma^2(Z) \leq \frac{2}{9}D$. $\sigma \leq \frac{\sqrt{2}}{3}\sqrt{D}$.
So

$$\Pr[|Z - EZ| \geq \frac{1}{6}D] = \Pr[|Z - EZ| \geq \underbrace{\frac{\sqrt{2}}{3}\sqrt{D}}_{\geq\sigma} \cdot \underbrace{\frac{1}{2\sqrt{2}}\sqrt{D}}_{\leq l}] \leq \frac{1}{l^2} \leq \frac{8}{D}$$

3 Probability Spaces Satisfying Exact Constraints

In our discussion of pseudo-random generators, we showed the existence of a PRG. If there are N constraints and we tolerate an error ϵ then there exists a probability space of size $O(\log(N)/\epsilon^2)$. k-wise independence has fewer constraints (a constraint is of the form $\Pr[event] = \beta$). If we have n variables $\in [q]$ then the number of constraints $\leq \binom{n}{k}q^k$. $\binom{n}{k}$ is the number of sets of k variables; for each of these we have a constraint per combination of their values, so q^k constraints. In this case, however, we are seeking an error $\epsilon = 0$.

Theorem 2 (Koller-Megiddo) *If there are N consistent constraints then there exists a probability space of size $\leq N$ (a set of constraints is consistent if there exists a probability space (such as the uniform distribution) satisfying these constraints).*

Proof: Domain $\Omega = [q]^n$ in our example (this is the total domain over which all the random variables are chosen). Each constraint defines a subset of Ω. We seek some probability function $p : \Omega \mapsto [0, 1]$ s.t. for each constraining subset S_i,

$$\sum_{w \in S_i} p(w) = a_i \quad \text{for some } a_i$$

We can express this as a linear system:

$$\begin{bmatrix} 1_{S_1} \\ 1_{S_2} \\ 1_{S_3} \\ \cdots \\ \\ \end{bmatrix} \begin{bmatrix} \\ \\ p \\ \\ \\ \end{bmatrix} = \begin{bmatrix} \\ a \\ \\ \end{bmatrix}$$

$$|\Omega|$$

Consistency ensures that there exists a non-negative solution $p \geq 0$. We then use a theorem in linear programming to show that there exists a basic non-negative solution (i.e. the columns where $p_j \neq 0$ are independent). Therefore, the number of independent columns $\leq$ number of constraints. Note also that the size of the probability space is equal to the number of locations in p that are non-zero. Hence, the size of the probability space $\leq$ number of constraints.

$\square$

Koller-Megiddo also present a constructive argument for independence constraints $\Pr[Y_1 = a_1 \wedge Y_2 = a_2 \wedge ...] = \Pr[Y_1 = a_1] \Pr[Y_2 = a_2]....$