# CMPT 210: Probability and Computing

Lecture 18

Sharan Vaswani

November 12, 2024

# Recap

- **Expectation**: For a r.v. $R$, $\mathbb{E}[R] = \sum_{x \in \text{Range}(R)} x \Pr[R = x]$.
- **Conditional Expectation**: For an event $A$ and r.v. $R$, $\mathbb{E}[R|A] := \sum_{x \in \text{Range}(R)} x \Pr[R = x|A]$.
- **Law of Total Expectation**: If $R$ is a random variable $\mathcal{S} \to V$ and events $A_1, A_2, \ldots A_n$ form a partition of the sample space i.e. for all $i$, $j$, $A_i \cap A_j = \emptyset$ and $A_1 \cup A_2 \cup \ldots \cup A_n = \mathcal{S}$, then, $\mathbb{E}[R] = \sum_i \mathbb{E}[R|A_i] \Pr[A_i]$.
- **Pairwise Independence**: R.v's $R_1, R_2, R_3, \ldots R_n$ are *pairwise* independent iff for *any* pair $R_i$ and $R_j$, for *all* $x \in \text{Range}(R_i)$ and $y \in \text{Range}(R_j)$, events $[R_i = x]$ and $[R_j = y]$ are pairwise independent implying that $\Pr[(R_i = x) \cap (R_j = y)] = \Pr[R_i = x] \Pr[R_j = y]$.

# Independence of random variables

- Similar to events, random variables $R_1, R_2, \ldots, R_n$ are mutually independent if for *all* $x_1 \in \text{Range}(R_1), x_2 \in \text{Range}(R_2), \ldots, x_n \in \text{Range}(R_n)$, events $[R_1 = x_1], [R_2 = x_2], \ldots [R_n = x_n]$ are mutually independent.

**Mutual Independence of events**: A set of events is said to be mutually independent if the probability of each event in the set is the same no matter which subset of events has occurred. For events $E_1, E_2$ and $E_3$ to be mutually independent, all the following equalities should hold:

$$\Pr[E_1 \cap E_2] = \Pr[E_1]\Pr[E_2] \quad \Pr[E_1 \cap E_3] = \Pr[E_1]\Pr[E_3]$$
$$\Pr[E_2 \cap E_3] = \Pr[E_2]\Pr[E_3] \quad \Pr[E_1 \cap E_2 \cap E_3] = \Pr[E_1]\Pr[E_2]\Pr[E_3].$$

Alternatively, (i) $\forall i$ and $j \neq i$, $\Pr[E_i|E_j] = \Pr[E_i]$ and (ii) $\forall i$ and $j, k \neq i$, $\Pr[E_i|E_j \cap E_k] = \Pr[E_i]$.

- For 2 r.v's $R_1$ and $R_2$, mutual independence and pairwise independence are equivalent.
- For more than 2 r.v's $R_1, R_2, \ldots, R_n$, mutual independence implies pairwise independence.

# Independence - Examples

Suppose there is a dinner party where $n$ people check in their coats. The coats are mixed up during dinner, so that afterward each person receives a random coat i.e. a person gets their own coat with probability $\frac{1}{n}$.

**Q**: If $G_i$ is the indicator r.v. that person $i$ gets their own coat, are the random variables $G_1, G_2, \ldots G_n$ mutually independent? No. Since if $G_1 = G_2 = \ldots G_{n-1} = 1$, then, $\Pr[G_n = 1 | (G_1 = 1 \cap G_2 = 1 \cap \ldots \cap G_{n-1} = 1)] = 1 \neq \frac{1}{n} = \Pr[G_n = 1]$. Conditioning on $(G_1, G_2, \ldots, G_{n-1})$ changes $\Pr[G_n]$, and hence the r.v's are not independent. Notice that we used the linearity of expectation even though these r.v's are not mutually independent.

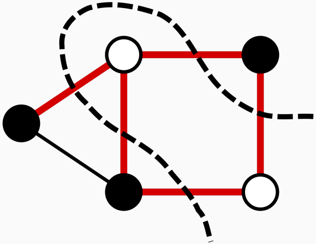
Q: Are the random variables $G_1, G_2, \ldots G_n$ pairwise independent? Ans: No.
$\Pr[G_1 = 1 | G_2 = 1] = \frac{1}{n-1} \neq \Pr[G_1 = 1]$

**Q**: What is the expected number of people who get their own coat? Let $G$ be the number of people that get their own coat. We wish to compute $\mathbb{E}[G]$. Define $G_i$ to be the indicator r.v. that person $i$ gets their own coat. Observe that $G = G_1 + G_2 + \ldots + G_n$ and by linearity of expectation $\mathbb{E}[G] = \mathbb{E}[G_1] + \mathbb{E}[G_2] + \ldots + \mathbb{E}[G_n]$. For each $i$, $\mathbb{E}[G_i] = \Pr[G_i] = \frac{1}{n}$. Hence, $\mathbb{E}[G] = 1$ meaning that on average one person will correctly receive their coat.

**Questions?**

# Max Cut

- **Aim**: Given a graph $G = (\mathcal{V}, \mathcal{E})$, partition the graph's vertices into two complementary sets $\mathcal{S}$ and $\mathcal{T}$, such that the number of edges between the set $\mathcal{S}$ and the set $\mathcal{T}$ is as large as possible.
- Max Cut has applications to VLSI circuit design.



In this example, the set $\mathcal{U}$ consists of nodes colored black. $|\mathcal{U}| = 3$ and $|\delta(\mathcal{U})| = 5$.

**Formal objective**: Find a set $\mathcal{U} \subseteq \mathcal{V}$ of vertices that solve the following problem:

$$\max_{\mathcal{U} \subseteq \mathcal{V}} |\delta(\mathcal{U})| \text{ where } \delta(\mathcal{U}) := \{(u, v) \in \mathcal{E} | u \in \mathcal{U} \text{ and } v \notin \mathcal{U}\}$$

$\delta(\mathcal{U})$ is the "cut" corresponding to the set $\mathcal{U}$ and the aim is to find the cut with the largest size.

# Max Cut

- Max Cut is NP-hard (Karp, 1972), meaning that there is no polynomial (in $|\mathcal{E}|$) time algorithm that solves Max Cut exactly.
- We want to find an approximate solution $\mathcal{U}$ such that, if OPT is the size of the optimal cut, then, $|\delta(\mathcal{U})| \geq \alpha \, OPT$ where $\alpha \in (0, 1)$ is the multiplicative approximation factor.
- Randomized algorithm that guarantees an approximate solution with $\alpha = \frac{1}{2}$ with probability close to 1 (Erdos, 1967).
- Complicated algorithm with $\alpha = 0.878$. (Goemans and Williamson, 1995).
- Under some technical conditions, no efficient algorithm has $\alpha > 0.878$ (Khot et al, 2004).

We will use Erdos' randomized algorithm and prove the result in expectation. We wish to prove that for $\mathcal{U}$ returned by Erdos' algorithm,

$$\mathbb{E}[|\delta(\mathcal{U})|] \geq \frac{1}{2} OPT$$

**Algorithm**: Select $\mathcal{U}$ to be a random subset of $\mathcal{V}$ i.e. for each vertex $v$, choose $v$ to be in the set $\mathcal{U}$ independently with probability $\frac{1}{2}$ (do not even look at the edges!).

# Max Cut

**Claim**: For Erdos' algorithm, $\mathbb{E}[|\delta(\mathcal{U})|] \geq \frac{1}{2} OPT$.

**Proof**: For each edge $(u, v) \in \mathcal{E}$, let $X_{u,v}$ be the indicator random variable equal to 1 iff $(u, v)$ is in the cut, i.e. the event $E_{u,v} = \{(u, v) \in \delta(\mathcal{U})\}$ happens.

$$\mathbb{E}[|\delta(\mathcal{U})|] = \mathbb{E}\left[\sum_{(u,v)\in\mathcal{E}} X_{u,v}\right] = \sum_{(u,v)\in\mathcal{E}} \mathbb{E}\left[X_{u,v}\right] = \sum_{(u,v)\in\mathcal{E}} \Pr[E_{u,v}]$$

(Linearity of expectation, and Expectation of indicator r.v's.)

$$\Pr[E_{u,v}] = \Pr[(u, v) \in \delta(\mathcal{U})] = \Pr\left[(u \in \mathcal{U} \cap v \notin \mathcal{U}) \cup (u \notin \mathcal{U} \cap v \in \mathcal{U})\right]$$

$$= \Pr\left[(u \in \mathcal{U} \cap v \notin \mathcal{U})\right] + \Pr\left[(u \notin \mathcal{U} \cap v \in \mathcal{U})\right] \quad \text{(Union rule for mutually exclusive events)}$$

$$\Pr[E_{u,v}] = \Pr[u \in \mathcal{U}]\ \Pr[v \notin \mathcal{U}] + \Pr[u \notin \mathcal{U}]\ \Pr[v \in \mathcal{U}] = \frac{1}{2}\frac{1}{2} + \frac{1}{2}\frac{1}{2} = \frac{1}{2}.$$

(Independent events)

$$\implies \mathbb{E}[|\delta(\mathcal{U})|] = \sum_{(u,v)\in\mathcal{E}} \Pr[E_{u,v}] = \frac{|\mathcal{E}|}{2} \geq \frac{\mathrm{OPT}}{2}.$$

# Questions?

# Randomized Quick Select

**Aim**: Given an array $A$ of $n$ distinct numbers, return the $k^{th}$ smallest element in $A$ for $k \in [1, n]$.

**Algorithm** Randomized Quick Select

```
function QuickSelect(A, k)
If Length(A) = 1, return A[1].
Select p ∈ A uniformly at random.
Construct sets Left := {x ∈ A | x < p} and Right := {x ∈ A | x > p}.
r = |Left| + 1 {Element p is the r^th smallest element in A.}
if k = r then
    return p
else if k < r then
    QuickSelect(Left, k)
else
    QuickSelect(Right, k − r)
end if
```

# Randomized Quick Select

*Example*: If $A = \{2, 7, 0, 1, 3\}$ and we wish to find the $2^{nd}$ smallest element meaning that $k = 2$. According to the algorithm, $p \sim \text{Uniform}(A)$. Say $p = 3$.

Then after step 1, Left $= \{2, 0, 1\}$ and Right $= \{7\}$. $r := |\text{Left}| + 1 = 3 + 1 = 4$. Since $r > k$, we recurse on the left-hand side by calling the algorithm on $\{2, 0, 1\}$ with $k = 2$.

$p \sim \text{Uniform}(\{2, 0, 1\})$. Say $p = 1$. After step 2, Left $= \{0\}$ and Right $= \{2\}$.
$r := |\text{Left}| + 1 = 1 + 1 = 2$. Since $r = k$, we terminate the recursion and return $p = 1$ as the second-smallest element in $A$.

Q: Run the algorithm if $p = 0$ in the first step? Ans: Left $= \{\}$ and Right $= \{2, 7, 1, 3\}$. Hence $r = 1 < k = 2$. Hence we will recurse on the right-hand side by calling the algorithm on $\{2, 7, 1, 3\}$ with $k = 1$.

Q: Run the algorithm if $p = 1$ in the first step? Ans: Left $= \{0\}$ and Right $= \{2, 7, 3\}$. Hence $r = 1 + 1 = 2$. Hence we will return the pivot element $p = 1$.

# Randomized Quick Select – Analysis

**Alternate way**: Sort the elements in $A$ and return the $k^{th}$ element in the sorted list. Uses $O(n\log(n))$ comparisons.

Q: Can Randomized Quick Select do better – what is the maximum number of comparisons required by Randomized Quick Select in the worst-case? Ans: $O(n^2)$ when $k = n$ and the pivots are chosen in increasing order.

• In the worst case, Randomized Quick Select is worse than the naive strategy of sorting and returning the $k^{th}$ element. What about the average (over the pivot selection) case?

**Claim**: For any array $A$ with $n$ distinct elements, and for any $k \in [n]$, Randomized Quick Select performs fewer than $8n$ comparisons in expectation.

In order to prove this claim, we will need to prove the following lemma.

# Randomized Quick Select – Analysis

**Lemma**: The child sub-problem's array (either Left or Right) after the partitioning (in Line 4 of the algorithm) has expected size smaller than $\frac{7n}{8}$.

*Proof*: Define a "good" event $\mathcal{E}$ that the randomly chosen pivot splits the array roughly in half.

Formally, if $n$ is the length of the array, then $\mathcal{E}$ is the event that $r \in \left(\frac{n}{4}, \frac{3n}{4}\right]$ (for simplicity, let us assume that $n$ is divisible by 4.) Since $p$ is chosen uniformly at random, $\Pr[\mathcal{E}] = \frac{3n/4 - n/4}{n} = \frac{1}{2}$.

Recall that $|\text{Left}| = r - 1$ and $|\text{Right}| = n - r$. Hence if event $\mathcal{E}$ happens, then $|\text{Left}| < \frac{3n}{4}$ and $|\text{Right}| < \frac{3n}{4}$. Hence, $|\text{Child}| < \frac{3n}{4}$. If event $\mathcal{E}$ does not happen, in the worst-case, $|\text{Child}| < n$. By using the law of total expectation,

$$\begin{aligned}\mathbb{E}[|\text{Child}|] &= \mathbb{E}[|\text{Child}| \,|\mathcal{E}] \Pr[\mathcal{E}] + \mathbb{E}[|\text{Child}| \,|\mathcal{E}^c] \Pr[\mathcal{E}^c] \\ &< \frac{3n}{4}\frac{1}{2} + (n)\frac{1}{2} = \frac{7n}{8}.\end{aligned}$$

- Hence on average, the size of the child sub-problem is smaller than $\frac{7n}{8}$, proving the lemma.

# Questions?