CMPT 210: Probability and Computing

Lecture 16

Sharan Vaswani

March 9, 2023

Recap

Expectation/mean of a random variable R is denoted by $\mathbb{E}[R]$ and "summarizes" its distribution. Formally, $\mathbb{E}[R] := \sum_{\omega \in \mathcal{S}} \Pr[\omega]\, R[\omega]$

Alternate definition of expectation: $\mathbb{E}[R] = \sum_{x \in \text{Range}(R)} x \Pr[R = x]$.

Linearity of Expectation: For n random variables $R_1, R_2, \ldots, R_n$ and constants $a_1, a_2, \ldots, a_n$, $\mathbb{E}\left[\sum_{i=1}^{n} a_i R_i\right] = \sum_{i=1}^{n} a_i\, \mathbb{E}[R_i]$.

Conditional Expectation: For random variable R, the expected value of R conditioned on an event A is given by:

$$\mathbb{E}[R|A] = \sum_{x \in \text{Range}(R)} x \Pr[R = x|A]$$

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, then,

$$\mathbb{E}[R] = \sum_{i} \mathbb{E}[R|A_i] \Pr[A_i]$$

Expectation - Examples

For a random variable $X : \mathcal{S} \to V$ and a function $g : V \to \mathbb{R}$, we define $\mathbb{E}[g(X)]$ as follows:

$$\mathbb{E}[g(X)] := \sum_{x \in \text{Range}(X)} g(x) \Pr[X = x]$$

If $g(x) = x$ for all $x \in \text{Range}(X)$, then $\mathbb{E}[g(X)] = \mathbb{E}[X]$.

Q: For a standard dice, if X is the r.v. corresponding to the number that comes up on the dice, compute $\mathbb{E}[X^2]$ and $(\mathbb{E}[X])^2$

For a standard dice, $X \sim \text{Uniform}(\{1, 2, 3, 4, 5, 6\})$ and hence,

$$\mathbb{E}[X^2] = \sum_{x \in \{1,2,3,4,5,6\}} x^2 \Pr[X = x] = \frac{1}{6}\left[1^2 + 2^2 + \ldots + 6^2\right] = \frac{91}{6}$$

$$(\mathbb{E}[X])^2 = \left(\sum_{x \in \{1,2,3,4,5,6\}} x \Pr[X = x]\right)^2 = \left(\frac{1}{6}\left[1 + 2 + \ldots + 6\right]\right)^2 = \frac{49}{4}$$

Randomized Quick Select

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

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.

Randomized Quick Select – Analysis

In order to upper-bound the total number of comparisons, we use the Lemma with an induction on n. Recall that we need to prove that Randomized Quick Select requires fewer than $8n$ comparisons in expectation.

Base case: If $n = 1$, then we require $0 < 8(1)$ comparisons. Hence the base case is satisfied.

Inductive Step: Assume that for all $m < n$,
$\mathbb{E}[\text{Total number of comparisons for size } m \text{ array}] < 8\,m$.

$$
\begin{aligned}
&\mathbb{E}[\text{Total number of comparisons for size } n \text{ array}] \\
&= \mathbb{E}[(n-1) + \text{Total number of comparisons in child sub-problem}] \\
&= (n-1) + \mathbb{E}[\text{Total number of comparisons in child sub-problem}] && \text{(Linearity of expectation)} \\
&< (n-1) + 8\,\mathbb{E}[|\text{Child}|] && \text{(Induction hypothesis)} \\
&< (n-1) + 8\frac{7n}{8} < 8n. && \text{(Lemma)}
\end{aligned}
$$

Hence, for any $k \in [n]$, on average, Randomized Quick Select requires fewer than $8n$ comparisons, even though it might require $O(n^2)$ comparisons in the worst-case.

Questions?

Independence of random variables

We define two random variables R_1 and R_2 to be independent if for *all* $x_1 \in \text{Range}(R_1)$ and $x_2 \in \text{Range}(R_2)$, events $[R_1 = x_1]$ and $[R_2 = x_2]$ are independent. More formally, we require,

$$\Pr[(R_1 = x_1) \cap (R_2 = x_2)] = \Pr[(R_1 = x_1)] \Pr[(R_2 = x_2)]$$

Q: Suppose we toss three independent, unbiased coins. Let C be r.v. equal to the number of heads that appear and M be the r.v. that is equal to 1 if all the coins match. Are random variables C and M independent?

$\text{Range}(C) = \{0, 1, 2, 3\}$ and $\text{Range}(M) = \{0, 1\}$. $\Pr[C = 3] = \frac{1}{8}$ and $\Pr[M = 1] = \frac{1}{4}$. $\Pr[(C = 3) \cap (M = 1)] = \frac{1}{8} \neq \frac{1}{32} = \Pr[C = 3] \Pr[M = 1]$. Hence, C and M are not independent.

Independence - Examples

Q: If H_1 is the indicator r.v. equal to one if the first toss is a heads, are H_1 and M independent?
$\Pr[H_1 = 1] = \Pr[H_1 = 0] = \frac{1}{2}$, $\Pr[M = 1] = \frac{1}{4}$, $\Pr[M = 0] = \frac{3}{4}$.
$\Pr[H_1 = 0 \cap M = 1] = \Pr[\{TTT\}] = \frac{1}{8} = \Pr[H_1 = 0] \Pr[M = 1]$.
$\Pr[H_1 = 1 \cap M = 1] = \Pr[\{HHH\}] = \frac{1}{8} = \Pr[H_1 = 1] \Pr[M = 1]$.
$\Pr[H_1 = 0 \cap M = 0] = \Pr[\{THH, THT, TTH\}] = \frac{3}{8} = \Pr[H_1 = 0] \Pr[M = 0]$.
$\Pr[H_1 = 1 \cap M = 0] = \Pr[\{HHT, HTH, HTT\}] = \frac{3}{8} = \Pr[H_1 = 1] \Pr[M = 0]$.
Hence, H_1 and M are independent.