CS 180: Homework 5

Jonathan Woong

804205763

Winter 2017

Discussion 1B

Monday 13th March, 2017

Problem 1

Consider the complete rooted ternary tree of depth n. That is take a rooted tree where the root has three children, each child has exactly 3 children and so on for n levels. So for example for $n = 1$, you have the root connected to its three children (with 4 nodes in total); for $n = 2$ you have 13 nodes in total; and more generally, the total number of nodes in the tree of depth n is exactly $1 + 3 + 3^2 + \ldots 3^n = (3^{n+1} - 1)/2$. Now, suppose that you delete each edge of the tree independently with probability $1/2$. Let X be the number of nodes which are still connected to the root after the deletion of the edges. Compute the expectation of X.

Let T be the complete ternary tree of depth n. T has exactly $(3^{n+1} - 1)$ nodes.

For each node $u \in T$,define X_u as:

$$X_u = \begin{cases} 0 & \text{if } u \text{ not connected to root} \\ \\ 1 & \text{if } u \text{ node is connected to root} \end{cases}$$

$X = X_1 + \cdots + X_{(3^{n+1}-1)} = \sum_{u \in T} X_u$

Let $n = 0$ have 1 level, $n = 1$ have 2 levels, $n = 2$ have 3 levels, etc.

CLAIM: For any node at level i, $Pr[X_i = 1] = 1/2^{i-1}$.

PROOF BY INDUCTION:

BASE: $n = 1$, $Pr[u$ at level $i = 2$ is connected to root$] = Pr[X_2 = 1] = 1/2^{2-1} = 1/2 =$ true because the probability of an edge being deleted is $1/2$, so the probability of an edge not being deleted is $1/2$.

INDUCTION: $n + 1$ depth means the leaves are at level $i = n + 2$.

$Pr[$leaf u is connected to root$] = Pr[X_{n+2} = 1] = 1/2^{(n+2)-1} = 1/2^{n+1}$

$n + 1$ depth implies that the path from the root to the leaf u is length $n + 1$.

If each edge has a $1/2$ probability of not being deleted, then the probability of u being connected to the root means that each edge in the path from the root to u must not be deleted. Since the deletion of any edge is independent, the probability of a path of length $n + 1$ existing from the root to u is the product of the probabilities that each edge is not deleted, giving the total probability of $1/2^{n+1}$. This is consistent with the calculation above.

$\therefore$ The claim is true by induction.

For a node u at level i: $E[X_u] = Pr[X_u = 1] = 1/2^{i-1}$.

By linearity of expectation: $E[X] = E[\sum_{u \in T} X_u] = \sum_{u \in T} E[X_u] = \sum_{u \in T} 1/2^{i-1}$.

For any T of depth n, there are exactly 3^{i-1} nodes at level i.

$E[X] = \sum_{i=1}^{n} 3^{i-1}/2^{i-1} = \sum_{i=1}^{n} (3/2)^{i-1} = 2((3/2)^n - 1)$

Problem 2

Call a sequence of coin tosses "monotone" if the sequence never changes from Heads to Tails when parsed left to right. For example, the sequences $TTTHHHH$, $TTTT$ are monotone whereas $TTTHHHT$ is not. Consider a sequence of n coin tosses of a fair coin. For integer $k > 0$, let Y_k be the number of monotone sub-sequences of length k in the n coin tosses. Compute the expecation of Y_k (with proof).

$Y_k =\#$ of monotone sub-sequences of length k in n coin tosses

For any k:

$$Y_k = \begin{cases} 1 & \text{if } Y_k \text{ is monotone} \\ 0 & \text{otherwise} \end{cases}$$

To compute $Pr[Y_k]$ we look at a few examples and derive a general formula:

$Pr[Y_3 = 1] = Pr[Y_3 = \{HHH\}] + Pr[Y_3 = \{THH\}] + Pr[Y_3 = \{TTH\}] + Pr[Y_3 = \{TTT\}] = 4(1/2)^3$

$Pr[Y_4 = 1] = Pr[Y_4 = \{HHHH\}]+Pr[Y_4 = \{THHH\}]+Pr[Y_4 = \{TTHH\}]+Pr[Y_4 = \{TTTH\}]+Pr[Y_4 = \{TTTT\}] = 5(1/2)^4$

$Pr[Y_5 = 1] = Pr[Y_5 = \{HHHHH\}]+Pr[Y_5 = \{THHHH\}]+Pr[Y_5 = \{TTHHH\}]+Pr[Y_5 = \{TTTHH\}]+ Pr[Y_5 = \{TTTTH\}] + Pr[Y_5 = \{TTTTT\}] = 6(1/2)^5$

So $Pr[Y_k = 1] = (k + 1) * (1/2)^k$

Let $X = \{x_1, \dots, x_n\}$ be the sequence of n coin tosses.

Let $S = \{s_1 < \dots < s_k\}$ be a set of size k containing integer positions in X where the subsequence $x_{s_1} \dots x_{s_k}$ is monotone.

Y_k is then the count of all possible combinations of monotone sequences of length k: $Y_k = \sum_S Y_S$

This gives $E[Y_S] = 1 * Pr[Y_S = 1] = (k + 1) * (1/2)^k$ for a single instance of S.

By linearity of expectation: $E[Y_k] = E[\sum_S Y_S]$

$= \sum_S E[Y_S]$

$= \sum_{i=1}^{\binom{n}{k}} (k + 1) * (1/2)^k$

$= [(k + 1) * (1/2)^k][\sum_{i=1}^{\binom{n}{k}} 1]$

$E[Y_k] = \binom{n}{k}(k+1)*(1/2)^k$ since there are exactly $\binom{n}{k}$ possible subsequences of length k in a string of length n.

Problem 3

Define a random graph $G = (V, E)$ as follows: the vertex set of G is $V = \{1, \ldots, n\}$. Now for each pair $\{i, j\}$ with $i \neq j \in [n]$, add the pair $\{i, j\}$ to E with probability $1/2$ independent of the choice for every other pair. Let X_5 be the number of independent sets of size 5 in the graph G. Compute the expectation of X_5.

Let $I \in G$ be a set of vertices of size 5.

For any I:

$$X_I = \begin{cases} 1 & \text{if } X_I \text{ is an independent set} \\ 0 & \text{otherwise} \end{cases}$$

There are exactly $\binom{5}{2} = 10$ possible pairs in a set of 5 vertices.

In order for X_I to be an independent set, all 10 possible pairs must not be added to E. Since the probability of adding a pair $\{i, j\}$ to E is $1/2$, the probability of not adding $\{i, j\}$ to E is also $1/2$.

This gives $E[X_I] = 1 * Pr[X_I = 1] = (1/2^{10})$.

X_5 is then the count of all possible I that satisfy $X_I = 1$: $X_5 = \sum_I X_I$.

By linearity of expectation: $E[X_5] = E[\sum_I X_I] = \sum_I E[X_I]$

$= \sum_{i=1}^{\binom{n}{5}} (1/2^{10})$

$= [(1/2^{10})] \sum_{i=1}^{\binom{n}{5}} 1$

$= \binom{n}{5}(1/2^{10})$ since there are exactly $\binom{n}{5}$ possible instances of I in a string of length n.

Problem 4

Let A be an array of n positive integers and $W = (w_1, \ldots, w_n)$ be a list of positive integers where we view w_i as the *weight* of a_i. For an integer $k \geq 1$, let WSELECT(A, k) be the largest element a in A such that the total weight of items not larger than a in A is at most k. That is, WSELECT(A, k) is defined to be the largest element a in A such that $\sum_{i:a_i \leq a} w_i \leq k$. If there is no such element in A, then we set WSELECT$(A, k) = 0$.

Give a randomized algorithm which given as input a list of n positive integers A, the associated weights W, and an integer $k \geq 1$ finds WSELECT(A, k). Assume that the elements of A are distinct.

WSELECT(A, k):

1. Pick an element a from A uniformly at random. The weight of that item is w_a.
2. If $|A| = 1$, RETURN a.
3. For each element $W[i]$:

   If $W[i] < w_a$, add $W[i]$ to W_-.

   Else if $W[i] > w_a$, add $W[i]$ to $W+$.
4. Let $S = (\sum_{i \in W_-} w_i) + w_a$:

   If $S > k$, let $B = W_-$ and RETURN WSELECT(B, k).

   Else if $S < k$, let $B = W+$ and RETURN WSELECT$(B, k - S)$.

   Else if $S = k$, return a.

Problem 5

Suppose you are given n apples and n oranges. The apples are all of different weights and all the oranges have different weights. However, for each apple there is a corresponding orange of the same weight and vice versa. You are also given a weighing machine that will only compare apples and oranges. Your goal is to use this machine to pair up each apple with the orange of the same weight with as few uses of the machine as possible.

Give a randomized algorithm to determining the pairing.

Let $A = \{a_1, \ldots, a_n\}$ be the set of apples. The weight of apple a_i is w_{a_i}.

Let $R = \{r_1, \ldots, r_n\}$ be the set of oranges. The weight of orange r_i is w_{r_i}.

$PAIR(A, R)$:

1. If $A \cup R = \emptyset$: RETURN $\emptyset$.

Else: pick an apple a from A uniformly at random. Pick an orange r from R uniformly at random.

2. Remove a from A and remove r from R.

3. If $|A| = 0$ and $|R| = 0$: RETURN (a, r).

4. For each apple $a_i \in A$:

   If $w_r > w_{a_i}$: add a_i to A_-.

   Else if $w_r < w_{a_i}$: add a_i to $A+$.

   Else if $w_r = w_{a_i}$: set $q = (a_i, r)$.

5. For each orange $r_i \in R$:

   If $w_a > w_{r_i}$: add r_i to R_-.

   Else if $w_a < w_{r_i}$: add r_i to $R+$.

   Else if $w_a = w_{r_i}$: set $p = (a, r_i)$.

6. Let $s = \begin{cases} (a, r) & \text{if } w_a = w_r \\ p \cup q & \text{otherwise} \end{cases}$

7. RETURN $s \cup \begin{cases} PAIR(A+, R_-) & \text{if } A_-, R+ = \emptyset \text{ and } A+, R_- \neq \emptyset \\ PAIR(A_-, R+) & \text{if } A+, R_- = \emptyset \text{ and } A_-, R+ \neq \emptyset \\ PAIR(A_-, R_-) \cup PAIR(A+, R+) & \text{otherwise} \end{cases}$

An example problem:

1.

A	8	4	3	5	7	1	6	2
R	2	4	7	3	5	8	1	6

2.

$A : a = 5$	8	4	3	7	1	6	2
$R : r = 4$	2	7	3	5	8	1	6

4. | A_{-} | 3 | 1 | 2 | | $p = (a, r_4) = (5_a, 5_r)$ | | $A+$ | 8 | 7 | 6 |

5. | R_{-} | 2 | 3 | 1 | | $q = (a_2, r) = (4_a, 4_r)$ | | $R+$ | 7 | 8 | 6 |

6. RETURN $PAIR(A_{-}, R_{-}) \cup PAIR(A+, R+) \cup p \cup q$.

$= \text{RETURN } [(1_a, 1_r) \cup (2_a, 2_r) \cup (3_a, 3_r)] \cup [(6_a, 6_r) \cup (7_a, 7_r) \cup (8_a, 8_r)] \cup (5_a, 5_r) \cup (4_a, 4_r)$.

$= \text{RETURN } \{(1_a, 1_r), (2_a, 2_r), (3_a, 3_r), (4_a, 4_r), (5_a, 5_r), (6_a, 6_r), (7_a, 7_r), (8_a, 8_r)\}$.

1_{-}. $PAIR(A_{-}, R_{-}) =$

A	3	1	2
R	2	3	1

2_{-}.

$A : a = 3$	1	2
$R : r = 2$	3	1

4_{-}. | A_{-} | 1 | | $p = (a, r_1) = (3_a, 3_r)$ | | $A+$ | $\emptyset$ |

5_{-}. | R_{-} | 1 | | $q = (a_2, r) = (2_a, 2_r)$ | | $R+$ | $\emptyset$ |

6_{-}. RETURN $PAIR(A_{-}, R_{-}) \cup PAIR(A+, R+) \cup p \cup q$

$= \text{RETURN } (1_a, 1_r) \cup \emptyset \cup (3_a, 3_r) \cup (2_a, 2_r)$

$= \text{RETURN } \{(1_a, 1_r), (2_a, 2_r), (3_a, 3_r)\}$

1_{+}. $PAIR(A+, R+) =$

A	8	7	6
R	7	8	6

2_{+}.

$A : a = 8$	7	6
$R : r = 6$	7	8

4_{+}. | A_{-} | $\emptyset$ | | $p = (a, r_2) = (8_a, 8_r)$ | | $A+$ | 7 |

5_{+}. | R_{-} | 7 | | $q = (a_2, r) = (6_a, 6_r)$ | | $R+$ | $\emptyset$ |

6+. RETURN $PAIR(A+, R_{-}) \cup p \cup q$

$= \text{RETURN } (7_a, 7_r) \cup (8_a, 8_r) \cup (6_a, 6_r)$

$= \text{RETURN } \{(6_a, 6_r), (7_a, 7_r), (8_a, 8_r)\}$