/tikz/,/tikz/graphs/
conversions/canvas coordinate/.code=1 , conversions/coordinate/.code=1

Computational Probability and Randomized Algorithms

Raveesh Gupta

March 2019

Chapter 1

Basic Problems of Expectation

1.1 Aces

We choose 10 cards at random from a standard deck of 52 cards. Find expected value of number of aces.

The probability of the chosen card being an ace is $\frac{1}{13}$. Let X be a random variable denoting the number of aces from the event of the cards chosen randomly. Let Y_i be a random variable denoting 1 if the chosen card is an ace and 0 else int he i^{th} event of choosing a card from the deck. The $event_i$ is defined as choosing a card at random and then putting it back and shuffling the deck. In this problem the order of choosing a card doesn't matter i.e. the events are independent.

$$\mathrm{E}[X] = \sum_{i=1}^{10} \mathrm{E}[Y_i] \textbf{By Linearity of Expectation}$$
$$\mathrm{E}[Y_i] = 1 * \Pr[Y_i = 1] + 0 * \Pr[Y_i = 0] = \frac{1}{13}$$
$$\mathrm{E}[X] = \frac{10}{13}$$

1.2 Inflation

The price of a T.V is 1000$. Each of the next N days, the prices goes up by 5$ or 10$ (each with probability 50%). Find expected value of the final price.

Let Z be a random variable defining the price of the T.V after N days of

inflation. Let X be random variable defining total inflation after N days. Let Y_i be an indicator random variable defining the inflation in price on the i^{th} day. We have to find the (By Linearity of Expectation)

$$\begin{aligned}
\mathrm{E}[Z] &= \mathrm{E}[1000 + X] \\
&= 1000 + \mathrm{E}[X] \\
&= 1000 + \sum_{i=1}^{N} \mathrm{E}[Y_i] \\
&= 1000 + \sum_{i=1}^{N} 1 * (\frac{5}{2} + \frac{10}{2}) + 0 * (\frac{5}{2} + \frac{10}{2}) \\
&= 1000 + N * (7.5)
\end{aligned}$$

1.3 Max of 2

You roll a 6-sided die twice. Find expected value of the bigger of two scores.

	1	**2**	**3**	**4**	**5**	**6**
1	1	2	3	4	5	6
2	2	2	3	4	5	6
3	3	3	3	4	5	6
4	4	4	4	4	5	6
5	5	5	5	5	5	6
6	6	6	6	6	6	6

Each of the cell above has the probability $p = \frac{1}{36}$
Let X be a random variable defining maximum value of dice to the event of role of 2 dice.

Let expected value of bigger score $= \mathrm{E}[X]$

$$= \sum_{\textbf{all possible answers(X = r)}} \Pr[r] * r$$

$$= \frac{1}{36} * (1 * 1 + 2 * 3 + 3 * 5 + ..6 * 11)$$

$$\mathrm{E}[X] = \sum_{i=1}^{6} i * (\frac{i}{6})^2 - (\frac{i-1}{6})^2$$

1.4 Max of N

You roll a m-sided die N times. Find expected value of the bigger of N scores.

Enumerate the maximum number, the distribution will be a n-dimensional super-cube with m-length-side. Each layer will be a large cube minus a smaller cube. So we have:

$$\sum_{i=1}^{m} i * ((\frac{i}{m})^N - (\frac{i-1}{m})^N)$$

Calculate in may cause overflow, we could move the divisor into the sum and calculate (i/m)n instead.

1.5 Birthdays

You teach informatics in a class with 20 students. When someone has a birthday, you must let the whole class to play games instead of learning algorithms and using Excel. Maybe some students have birthday on the same day, so there would be fewer than 20 wasted days during a year? Find EV of the number of days when at least one student has birthday. Bonus/fact: Birthday paradox.

Lets solve some easier questions.

Let p be the probability that a randomly chosen day is not a person birthday. By intuition it is $\frac{364}{365}$, then p^k be the probability that a randomly chosen day is not a birthday for k people. Then $(1 - p^k)$ is the probability that a randomly chosen day will have at least one birthday out of k people. .. (1)

Our original question is the expected number of days when at least one student has birthday.

Let X be a random variable defining the number of days when at least one student has birthday. We can distribute X using a second random variable Y_i , where Y_i is an indicator random variable that defines 1 if at-least one person has a birthday on i^{th} day else 0.

$$X = Y_1 + Y_2....Y_{365}$$
$$Y_i = 1 * (\Pr[Y = i]) + 0 * (1 - \Pr[Y = i])$$

Since every day has an equal chance of having at-least one person having birthday, then

$$\begin{aligned} \mathrm{E}[X] &= 365 * \Pr[Y = i] \\ &= 365 * (1 - p^k) \end{aligned}$$

1.6 First Head.

Find expected value of the number of coin tosses until you get a head.

Let p be the probability of getting a head and f be the probability of not getting a head. Let X be the number of events (tosses) to get a head.

$$\begin{aligned} \mathrm{E}[X] &= 1 * p + 2 * f * p...i * f^{i-1} * p....\infty \\ f * (\mathrm{E}[X]) &= 1 * f * p + 2 * f^2 * p...i * f^i * p..\infty \\ &\textbf{Subtracting above equations} \\ \rightarrow (1 - f)\,\mathrm{E}[X] &= 1 * p + 1 * f * p + ..1 * f^i * p....\infty \\ \rightarrow \mathrm{E}[X] &= 1 + f + f^2...f^i..\infty \\ \rightarrow &= \frac{1}{1 - f} \\ \rightarrow \mathrm{E}[X] &= \frac{1}{p} \end{aligned}$$

Since $p = 1/2 \rightarrow \mathrm{E}[X] = 2$.

1.7 First two Heads

Find expected value of the number of coin tosses until you get heads two times in total.

Let consider R be the expected number of tosses until the occurrence of the first head. Let p be the probability of getting a head and f be the probability of not getting a head. Let X be the number of events (tosses) to get 2 heads in total.

Lets prove lemma ...(1) that $p + f * p + f^2 * p...\infty = 1$.

$$x = p + f * p + f^2 * p...\infty \tag{1.1}$$
$$\frac{x}{p} = 1 + f + f^2..\infty \tag{1.2}$$
$$\frac{x}{p} = \frac{1}{(1-f)} \tag{1.3}$$
$$x = 1 \tag{1.4}$$
$$\tag{1.5}$$

$$\begin{aligned}
\mathrm{E}[X] &= (1+R) * p + (2+R) * f * p...(i+R) * f^{i-1} * p....\infty \\
&= R * (p + f * p + f^2 * p...\infty) + (p + 2 * f * p + 3 * f^2 * p..\infty) \\
&= R(1) + R \textbf{ From lemma (1) and last question} \\
&= 2 * R \\
&= 4 \text{ tosses}
\end{aligned}$$

Another interesting and useful way to think of this problem is to first calculate the expected number of tosses to get the 1^{st} head and then calculate the expected number of tosses to get the 2^{nd} head, since both events are independent and similar we can say both values are equal and we can distribute. Linearity of expectation gives us $R + R = 2 * R$. Similarly think about case of N heads.

1.8 First 2 heads in a row

Find expected value of the number of coin tosses until you get heads two times in a row.

Let consider R be the expected number of tosses until the occurrence of the first head. Let p be the probability of getting a head and f be the probability of not getting a head. Let X be the number of events (tosses) to get 2 heads in

a row.

$$\begin{aligned}
\mathrm{E}[X] &= (1+R)*p + (2+R+R)*f*p ... (1+R)i*f^{i-1}*p .... \infty \\
&= (1+R)\sum_{i=1}^{\infty} i*f^{i-1}*p \\
&= (1+R)*(\textbf{ expected number of tosses to get first head}) \\
&= (1+R)*R \\
&= 6 \text{ tosses}
\end{aligned}$$

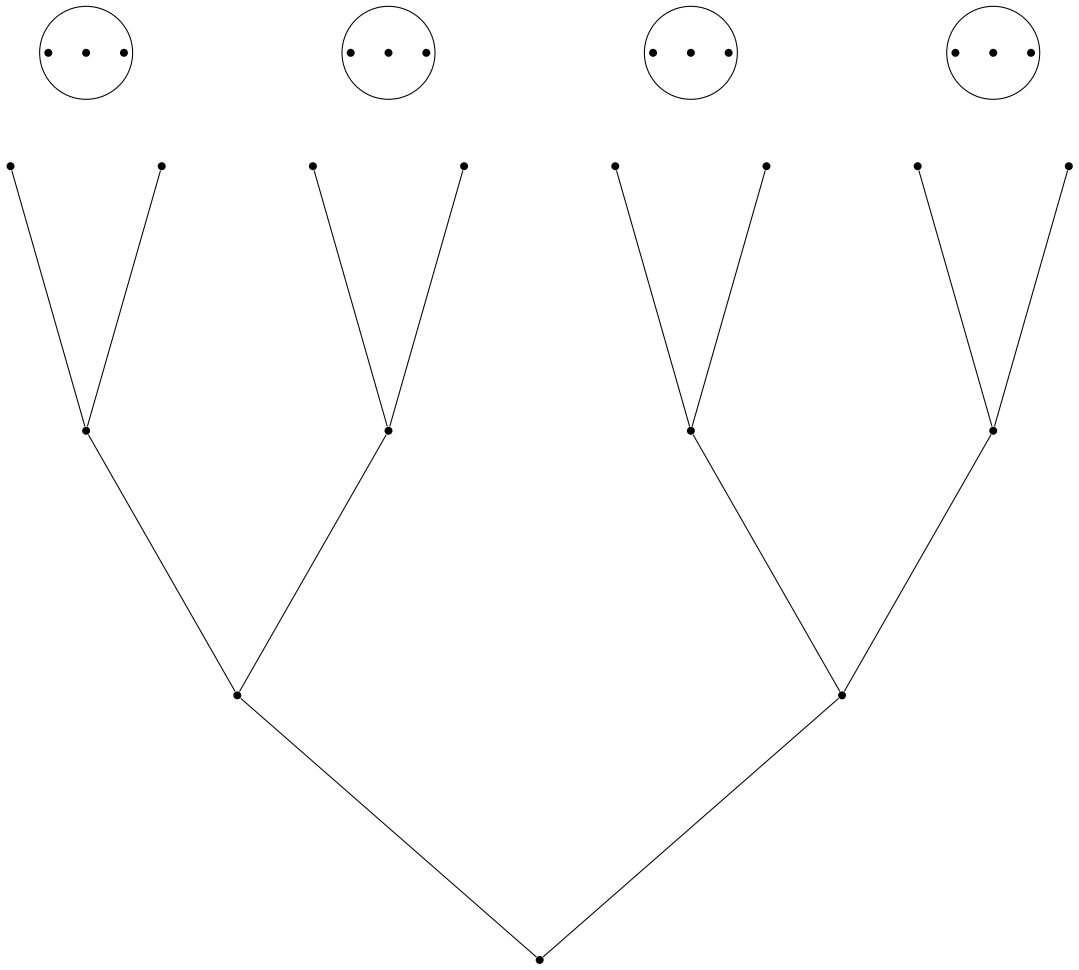
1.9 Volleyball

12 teams, including Poland, play in the volleyball tournament. Teams are divided into 4 groups, each with 3 teams. In each group, every teams plays against every other team, and then two best teams advance to the elimination stage. In case of a perfect tie, two random teams advance. The elimination stage has quarterfinals, half-finals and the final match. In every match, a random of two teams wins (50% each).

a) Find probability that Poland will win the whole tournament.

b) Find expected number of the number of matches won by Poland.

c) Find expected number of the number of matches won by Poland, assuming that they won the whole tournament. (in other words, find expected number of the number of matches won by the winner of the whole tournament).



Assumption all teams have equal chance of winning
First part is to find the probability of a Poland winning.
Let me define some basic probabilities
Probability of *Poland* advancing to the tournament stage = $\frac{2}{3}$

Probability of *Poland* winning a match in tournament = $\frac{1}{2}$

Probability of *Poland* winning the tournament = $\frac{2}{3} * \frac{1}{2} * \frac{1}{2} * \frac{1}{2} = \frac{1}{12}$

Another way to do analysis is to think that all teams have equal capabilities and equal probabilities of winning. So by symmetry of this arrangement above probability of any team winning is $\frac{1}{12}$

Second part is find the expected number of matches won by *Poland*,

Using Linearity of expectation we can distribute the the matches won by *Poland*

$$\mathrm{E}[X] = 1 + \frac{2}{3} * \frac{1}{2} * 1 + \frac{2}{3} * \frac{1}{2} * \frac{1}{2} * 1 + \frac{2}{3} * \frac{1}{2} * \frac{1}{2} * \frac{1}{2} = \frac{19}{12}$$

Another way to analyse is saying that since each match is an independent event and symmetry of this arrangement. So EV is (number of matches held in the tournament)/(number of teams) i.e
Total number of matches held $\neq \sum_{i=1}^{12}$ number of matches played by i^{th} team

BONUS AND IMPORTANT: Another interesting question is to ask the expected number of matches played by *Poland* .
That would be (*total number of matches all the teams played*)/(number of teams).

Chapter 2

Typewriter Monkey (Google Code Jam)

Given a set of K characters and a target word L and length $|S|$ of the string made out of K characters. Find the expected value of number of instances of L in S.

There can be overlapping instances of L in S, find the maximum number of L possible in S. Find O the maximum overlap possible between 2 instances of L. In other words, Find the maximum prefix of L that is also a suffix of L. Define Z[i] as the maximum length of prefix of string S[$i..n$] that is also a prefix of S[1..n].
if

$$\mathrm{Z}[i] + i - 1 == n \tag{2.1}$$

then
O_{max} will be the first occurrence of condition (1.1)

Maximum number of instances possible is equal to $MAX = 1 + \frac{(|S|-|L|)}{|L|-O_{max}}$

X_i be a random variable denoting the number of instances of L in S. Goal is to find E[X_i].
Definition of

$$\mathrm{E}[X_i] = \sum_{i=1}^{MAX} X_i * \Pr[X = i]$$

where $\Pr[X = i]$ is the probability of i instances of L in S. We solve this by introducing an indicator random variable Y_i .

$$Y_i = 1 <=> L == S[i...(|L| + i - 1)] \tag{2.2}$$

E[X] can be distributed in the following way by **Linearity Of Expectation**.

$$\mathrm{E}[X] = \sum_{i=1}^{MAX-|L|+1} \mathrm{E}[Y_i] \tag{2.3}$$

Let P be the probability of $S[1...|L|] = \mathrm{L}$.

$$P = \Pi_{i=1}^{|L|} \frac{\#L[i] \epsilon K}{|K|} \tag{2.4}$$

Each $\mathrm{E}[Y_i] = 1 * P + 0 * (1 - P)$. An observation is that $\mathrm{E}[Y_i]$ is equal for every i which implies.

$$\mathrm{E}[X] = (MAX - |L| + 1) * P$$

Chapter 3

Balls into bins

Suppose we have m balls, labeled $i = 1, ..., m$ and n bins, labeled $j = 1, ..., n$. Each ball is thrown into one of the bin independently and uniformly at random. What is the expected number of balls in each bin.

We have to find $\mathrm{E}[X]$ where X_i is a random variable denoting number of balls in i_{th} bin. Define an indicator random variable $Y_{i,j}$.

$$Y_{i,j} = 1 \text{ if only if } j_{th} \text{ ball goes into } i_{th} \text{ bin} \tag{3.1}$$

Linearity of Expectation Distributes

$$\mathrm{E}[X_i] = \sum_{j=1}^{m} \mathrm{E}[Y_{i,j}] \tag{3.2}$$

$\mathrm{E}[Y_{i,j}] = 1 * \Pr[i,j] + 0 * (1 - \Pr[i,j])$. By intuition its clear that $\Pr[i,j] = \frac{1}{n}$.

$$\begin{aligned}
\mathrm{E}[X_i] &= \sum_{j=1}^{m} \mathrm{E}[Y_{i,j}] \\
&= m\,\mathrm{E}[Y_{i,j}] \\
&= m\,\Pr[i,j] \\
&= \frac{m}{n}
\end{aligned}$$

Chapter 4

Kleofáš and the n-thlon

Suppose there are m participants and n competitions . Each competition Kleofáš gets a rank $x[i]$. Overall score of a j_{th} participant is defined as

$$s[j] = \sum_{i=1}^{n} rank_{i,j} \tag{4.1}$$

where $rank_{i,j}$ is the rank of the j_{th} participant in i_{th} competition. Overall rank of j_{th} participant is defined as

$$R[j] = 1 + \sum_{i=1}^{m} (s[i] < s[j]) \tag{4.2}$$

Find the expected overall rank of Kleofáš.

Let Keofáš be the 1_{th} participant with overall score of $s[1] = \sum_{i=1}^{n} x[i] = C$. We have to Find $\mathrm{E}[R[1]]$ or $1 + \sum_{i=2}^{m} \mathrm{E}[Y_i]$ where

$$Y_i = 1 \text{ if only if } s[i] \text{ is less than } C$$

and

$$\mathrm{E}[Y_i] = 1 * Pr[Y = i] + 0 * (1 - Pr[Y = i])$$

where $Pr[Y = i]$ is the probability of $s[i]$ is less than C.

We have to use dynamic programming to find the probability of $s[i]$ being less than C.

Lets define $dp(i, X)$ is the probability to assign 2^{nd} participant in i_{th} competition such that $s[2] < C$.

$$dp(i, X) = \forall j \epsilon [1, m] \wedge j \neq x[i] => cnt = cnt + \frac{dp(i + 1, X + j)}{(m - 1)}$$

$$dp(i, >= C) = 0$$
$$dp(n+1, < C) = 1$$

Probability of each member is equal since all rankings in each competition are equi-probable. So $Pr[Y = i]$ is equal for all i $\epsilon[2, m]$.

$$\mathrm{E}[R[1]] = 1 + \sum_{i=2}^{m} \mathrm{E}[Y_i] \tag{4.3}$$
$$= 1 + \sum_{i=2}^{m} \Pr[i] \tag{4.4}$$
$$= 1 + (m-1) * dp(1, 0) \tag{4.5}$$

Chapter 5

The Game

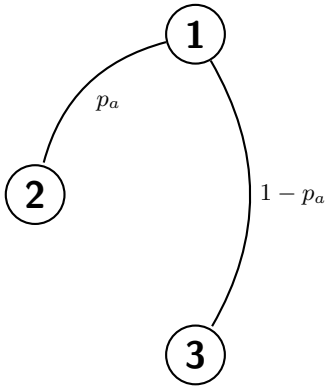
There are 2 players $player_a, player_b$ playing a combat game with initial health HP_a and HP_b and power of x_a and x_b. $player_a$ tries to hit the $player_b$ with the probability of p_a and $player_b$ tries to hit $player_a$ with probability p_b. If a player hits the other player, other player's health reduces by power of first player otherwise it reduces by 1. A player wins if the opponent's health becomes 0. Find the probability of $player_a$ of wining the game. $player_a$ starts the game.

We will look at this problem by drawing a graph of states. A state or a node is defined by the player's turn and HP_a and HP_b. Let's call this state: $dp(HP_a, HP_b, t)$ where t is 1 if its $player_a$ turn or 0 if its $player_b$ turn. $dp(HP_a, HP_b, t)$ is defined as the probability of $player_a$ wining if HP_a and HP_b are the health of a and b player and its turn of t player.

We can define the transition of this state into its children state like the following:

$dp(HP_a, HP_b, t) = (p_a) * dp(HP_a, HP_b - x_a, 1-t) + (1-p_b) * dp(HP_a, HP_b - 1, 1-t)$ **if t = 1 hence** $player_a$ **turn**

$dp(HP_a, HP_b, t) = (p_b) * dp(HP_a - x_b, HP_a, t) + (1-p_b) * dp(HP_a - 1, HP_b, 1-t)$ **if t = 0 hence** $player_b$ **turn.**



Base Cases are as follows:

$$dp(i > 0, j <= 0, t) = 1 dp(i <= 0, j > 0, t) = 0$$

This recursive solution with memoization (Using pre-computed answers in O(1)) will algorithmic complexity of $O(HP_a * HP_b)$.

5.1 The Harder Game

There are 2 players $player_a, player_b$ playing a combat game with initial health HP_a and HP_b and power of x_a and x_b. $player_a$ tries to hit the $player_b$ with the probability of p_a and $player_b$ tries to hit $player_a$ with probability p_b. If a player hits the other player, other player's health reduces by power of first player otherwise it remains the same. A player wins if the opponent's health becomes 0. Find the probability of $player_a$ of wining the game. $player_a$ starts the game.

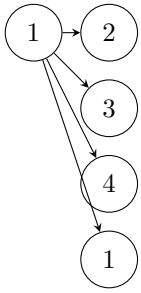
We look at this problem by technique of drawing graphs. A state or a node is defined by player's health HP_a, HP_b . Lets call this state $dp(HP_a, HP_b)$. Its defined as the probability of $player_a$ winning with health HP_a, HP_b respectively.

We can define the transition of children states as the following:

$$dp(HP_a, HP_b) = p_a * p_b * dp(HP_a - x_b, HP_b - x_a) + p_a * (1 - p_b) * dp(HP_a, HP_b - x_a) + (1 - p_a) * p_b * dp(HP_a - x_b, HP_b)$$

Notice there is a self-loop on 1. To solve it we move the repeating term to the left. $dp(HP_a, HP_b) * (1 - (1 - p_a) * (1 - p_b)) = (p_a) * (p_b) * dp(HP_a - x_b, HP_b - x_a) + (p_a) * (1 - p_b) * dp(HP_a, HP_b - x_a) + (1 - p_a) * (p_b) * dp(HP_a - x_b, HP_b)$
Base Cases are as follows:

$$dp(i > 0, j <= 0) = 1 dp(i <= 0, j > 0) = 0$$



This recursive solution with memoization (Using pre-computed answers in O(1)) will algorithmic complexity of $O(HP_a * HP_b)$.
Note if tried to solve this problem by the algorithm mention in the problem The Game, there will a loop of 2 states which is a much harder problem to solve.