

CS 177, Solutions to Homework 1

Applications of Probability in Computer Science: Winter 2007

January 19, 2007

Problem 1: (Sample Spaces: Review Chapter 1.2)

For each of the problems below define the sample space S . State whether S is finite, countably infinite, or uncountably infinite. If S is finite, calculate $|S|$, the number of elements in the set S .

1. Toss a coin 5 times and count the number of heads.

SOLUTION:

Finite. Sample space can be written as:

$$S = \{0,1,2,3,4,5\}$$

$$|S| = 6$$

2. Toss a coin until the first “heads” shows up and count the number of tosses of “tails” that occur before this happens.

SOLUTION:

$$S = \{0,1,2,3,\dots\}$$

Countably infinite.

3. Roll 4 dice and compute their sum.

SOLUTION: Finite

$$S = \{4,5,6 \dots 24\}$$

$$|S| = 21$$

4. Roll 2 dice and compute their product.

SOLUTION: Finite. Sample space can be written as:

$$S = \{1,2,3,4,5,6,8,9,10,12,15,16,18,20,24,25,30,36\}$$

$$|S| = 18$$

5. Measure the time (rounded to integer minutes) that it takes for a disk to fail from the time it is installed.

SOLUTION:

$$S = \{0,1,2,\dots\}$$

countably infinite

6. Measure the number of unique customers (rounded to integer millions) with US addresses that visit the Amazon.com Web site in any given 24-hour period. “Unique” here means that if a person visits the site more than once during the time period they are only counted once and not multiple times.

SOLUTION:

$$S = \{0,1,2,\dots,300\}$$

finite, $|S| = 300$ (this is based on the assumption that there are 300 million people resident in the United States—any number around 300 would have been fine).

Problem 2: (Events: Review Chapter 1.2)

Two chess players A and B decide to play each other in 5 online chess games. Let A_k, B_k, D_k represent the events that A wins game k , B wins game k , or D wins game k , respectively, where $k = 1, \dots, 5$.

Describe the following events in terms of A_k, B_k , and D_k . For example, the event that A wins game 1 and B wins game 2 would be described as $A_1 \cap B_2$.

Note that there was a typo in the above problem: it should have said that D_k represents the outcome where game k is a draw.

1. A wins at least 1 of the first 3 games.

SOLUTION: $A_1 \cup A_2 \cup A_3$

2. B does not win any of the first 3 games

SOLUTION: $B_1^c \cap B_2^c \cap B_3^c$ (note that this is not the same as losing the first 3 games because draws can occur)

3. None of the 5 games end in a draw.

SOLUTION: $D_1^c \cap D_2^c \cap D_3^c \cap D_4^c \cap D_5^c$

4. There is 1 draw (and no more) over the 5 games.

SOLUTION: $\{D_1 \cap D_2^c \cap D_3^c \cap D_4^c \cap D_5^c\} \cup \{D_1^c \cap D_2 \cap D_3^c \cap D_4^c \cap D_5^c\} \cup \{D_1^c \cap D_2^c \cap D_3 \cap D_4^c \cap D_5^c\} \cup \{D_1^c \cap D_2^c \cap D_3^c \cap D_4 \cap D_5^c\} \cup \{D_1^c \cap D_2^c \cap D_3^c \cap D_4^c \cap D_5\}$

5. A wins 2 games in a row at least once during the series of 5 games. If A wins more than 3 or more games in a row, this counts as winning 2 games in a row, so include this in your definition of the event.

SOLUTION: $\{A_1 \cap A_2\} \cup \{A_2 \cap A_3\} \cup \{A_3 \cap A_4\} \cup \{A_4 \cap A_5\}$

Problem 3:

Consider a communications network where each packet travels over 2 different subnetworks to get from one computer to another. In each subnetwork the packet can take 1, 2, 3, 4, or 5 hops, where each of the 5 possibilities is equally likely. The number of hops in one network does not depend in any way on the number of hops taken in the other network (i.e., they are independent). Let X be the random variable corresponding to the total number of hops taken by a packet.

1. What is the probability mass function (pmf) for X ?

SOLUTION: $P(x) = \{$
1/25, $x = 2$
2/25, $x = 3$
3/25, $x = 4$
4/25, $x = 5$
5/25, $x = 6$
4/25, $x = 7$
3/25, $x = 8$
2/25, $x = 9$
1/25, $x = 10$
 $\}$

2. Say each hop causes a delay of 1 millisecond. What is the expected delay for a packet in traversing the two subnetworks? (i.e., what is $E(X)$?)

SOLUTION:

$$E(x) = \sum_{i=2}^{10} x_i * p(x_i) = 2\frac{1}{25} + 3\frac{2}{25} + \dots = 6$$

Problem 4

Say you have chosen a 6-character password for a particular computer account, where English letters or the digits 0 through 9 are allowed in each position, and that is not case-sensitive (upper and lower letters are treated the same way), e.g., `ez1ad2`.

1. If someone guesses randomly at your password, what is the probability that they guess it correctly on their first guess?

SOLUTION: There are 36 possibilities for each position. So the probability of guessing correctly in any one position is $\frac{1}{36}$. The guesses can be viewed as independent trials, so the probability of guessing correctly 6 times in a row is $1/(36^6) = 4.5910^{-10}$

2. What if they write a computer program that randomly selects 1 million (different) passwords and tries them all—what is the probability that none of these passwords match the correct password?

SOLUTION: One way to think about this problem is to look at it as follows: say the person trying to guess the password first randomly selected k passwords (all different), out of n total possible passwords. And then you came along and independently selected a password (randomly) out of the n total possible passwords (and you had no knowledge of what the k selected passwords were). What is the probability that you selected one of the k that were already selected? It should be obvious that it there is probability $p = \frac{k}{n}$ of this happening. And so the probability of your password not matching any of the k selected ones is $1 - p = 1 - \frac{k}{n}$. The problem you are asked to solve is exactly the same as this one. The order in which you select your password and the other person selects their k passwords does not matter, since you are both acting independently. If we plug in $k = 10^6$ and $n = 36^6$, we get the solution that $1 - p = 0.9995$.

This is just one way to think about the problem—you can also think of it in the order in which it was stated, i.e., that you first select your password and then the “guesser” selects k randomly—these 2 processes are exactly the same and give the same answer, you should convince yourself that this is the case.

A different version of this problem would be if you were asked to calculate the same probability but now “guesser” randomly selected a password without keeping track of what was previously guessed, so that repetitions are allowed. We can analyze this as follows. The probability of none of these passwords matching is the probability that the guess is incorrect on each one of the 1 million tries. The probability of guessing incorrectly on 1 try is $1 - p = 1 - 4.5910^{-10}$ (from part 1). The probability of this happening 1 million times (independent trials) is $(1 - p)^{10^6}$, which works out to be 0.9995. This happens to be the same answer as the previous case, where the passwords were all different—this is because the number of passwords out of 1 million that will be guessed more than once is relatively small (compared to 1 million), so the difference between the 2 approaches is not numerically significant here. But in general one could of course get quite different answers.

3. Say that your password really is `ez1ad2` and that the person guessing knows (somehow) that your password has vowels in the first and 4th positions, non-vowel letters in the 2nd and 5th positions, and digits in the 3rd and 6th positions. Answer parts 1 and 2 of this problem again, but now assuming that the person guessing has this extra information.

SOLUTION: (this solution counts “y” as a vowel—you could also solve for the more usual case of 5 vowels where y is not considered to be a vowel)

Total letters: $|26|$

Total vowels: $|6| = \{a, e, o, u, y, i\}$

part 1:

$$1/(6 * 20 * 10 * 6 * 20 * 10) = 6.94 \times 10^{-7}$$

part 2:

We now have that $n = 6 * 20 * 10 * 6 * 20 * 10 = 1440000$ (far fewer possible passwords than before). Using the same reasoning as before, we have $p = 1 - \frac{k}{n} = 0.3056$ which is now much lower, i.e., the probability they will guess the password is much higher (at about 0.7) because they have much more information (there are fewer possible passwords). If we assumed that the guesses were not necessarily all different, then we would have $P(\text{non match}) = (1 - p)^n = (1 - 6.9410^{-7})^{10^6} = 0.4996$. So now the probability of the password being guessed has dropped to about 0.5, which tells us that (in this case, unlike earlier) keeping track of the guesses and making sure each one is different, leads to an increase of a correct match from about 0.5 to 0.7.

Problem 5

You are writing code that controls the behavior of a very simple agent in a computer game. At each time-step in the game, the agent makes a decision to stay still (with probability 0.6), to walk (with probability 0.3), or to run (with probability 0.1). This is a very simple agent and it does not keep track from one time-step to the next of what it did before—so it is “memoryless.” Answer the following questions about the agent’s behavior:

1. What is the probability that the agent stands, walks, and then runs, in that order, in 3 consecutive time-steps?

SOLUTION: The events at each time step are independent so we can multiply the probabilities together: $P(\text{stands}) \times P(\text{walk}) \times P(\text{run}) = 0.6 * 0.3 * 0.1 = 0.018$

2. What is the probability that the agent stands, walks, and runs, *but in any order*, in 3 consecutive time-steps?

SOLUTION:

6 combinations ($3 \times 2 \times 1$) are possible, each with the same probability we calculated in part 1 (since order does not matter), and thus we get $6 \times 0.018 = 0.108$.

3. What is the probability that over 5 time-steps the agent will not move (will not run or walk) in any of the 5 time-steps?

SOLUTION:

This is the same as the probability that the agent will stand in each of the 5 time-steps, so we have

$$P(\text{stand}) * P(\text{stand}) * P(\text{stand}) * P(\text{stand}) * P(\text{stand}) = 0.6^5 = 0.0778$$

4. What is the probability that over 10 time-steps the agent will walk or run at least once? (hint: it is easiest to solve this problem by defining a situation that is the opposite of this).

SOLUTION:

This is the same as 1 minus the probability that the agent does not walk or run at all in 10 time-steps (you should convince yourself that these 2 different events sum to 1, i.e., that they are mutually exclusive and exhaustive). The probability of not walking or running in 10 time-steps is $0.6^{10} = 0.006$ and 1 minus this is 0.994.

Problem 6

In the board game Risk, a player A can “invade” the territory of another player B on the board and the outcome of a “battle” is determined by each player rolling a certain number of dice. Here we consider a simplified version of this problem. A and B each roll a single die (at the same time). If the number on A’s die is greater than that on B’s die then A wins the battle. If the number on A’s is less than or equal to that on B’s die, then B wins the battle. What is the probability that the attacker A will win a battle using these rules? Show clearly how you arrived at your answer.

SOLUTION:

B rolls 1: $P(A_{win}) = 1/6 * 5/6 = 5/36$

B rolls 2: $P(A_{win}) = 1/6 * 4/6 = 4/36$

B rolls 3: $P(A_{win}) = 1/6 * 3/6 = 3/36$

B rolls 4: $P(A_{win}) = 1/6 * 2/6 = 2/36$

B rolls 5: $P(A_{win}) = 1/6 * 1/6 = 1/36$

TOTAL: $P(A_{win}) = 5/36 + 4/36 + 3/36 + 2/36 + 1/36 = 15/36 = 0.4166$

Problem 7 (MATLAB)

Write a MATLAB function to plot the binomial distribution (or probability mass function) for each of the following examples:

- $n = 20, p = 0.1$
- $n = 20, p = 0.75$
- $n = 1000, p = 0.5$
- $n = 1000, p = 0.9$

You should end up generating 4 different graphs in total. Comment briefly (a few sentences) on the differences between the 4 plots.

You will need to write the MATLAB function `binomial_pmf.m`. A template is available on the class Web site which contains most of the function, but you will need to fill in certain details.

For the cases with $n = 1000$ the binomial coefficients will be impractical to calculate directly. You can use the following approximation to the binomial which is accurate for large n :

$$p(i) \approx \frac{1}{\sqrt{2\pi np(1-p)}} e^{-(i-np)^2/(2np(1-p))}$$

We have provided a function called `binopmf_approx.m` on the class Web site that uses this approximation and that you can use in place of the exact function (`binomial_pmf.m`) for the cases where $n = 1000$. Note that you may still encounter underflow, in the sense that some of the probabilities will be smaller than the smallest number that your computer can represent—if so, for the purposes of plotting it is ok to assume that these probabilities are effectively zero since they will not show up on the plot anyway.

For plotting the graphs, here are some MATLAB commands that will generate the probability mass function for $n = 20$ and plot the graph (you can use this once you have written your (`binomial_pmf.m` function)).

```
% first set the parameter values that we need:
n=20; p = 0.1;
% next create a vector of length n+1 containing all integers from 0 to n
ivalues = 0:1:n;
% now generate a vector y, where each value corresponds to p(i), i=0 to n
% (note how this is done as a vector operation, avoiding a "for loop")
y = binomial_pmf(ivalues,n,p);
bar(y); % use a bar plot to show the values of the binomial pmf
xlabel('i','FontSize',14);
ylabel('probability(i)','FontSize',14);
title('Binomial Probability Mass Function: n=20, p = 0.1','FontSize',14);
```

You should familiarize yourself with the functions being called above to make sure you know how they work, e.g., type `help bar` in MATLAB to find out what it does.

For large values of n , e.g., $n = 1000$ the barplot will be very dense, so I recommend you instead use a “line plot”, i.e., you can use:

```
plot(ivalues,y) instead of bar(y)
```

Solution:

1. For $n = 20, p = 0.1$ the probability mass function (pmf) is centered around $np = 2$ and is quite broad. When we change the value of p to 0.75, the center of the pmf shifts to $np = 15$ and it is still quite broad (high variance, since n is relatively small).
2. When n is increased to 1000, and p changes to 0.5 we now get a much narrower pmf (lower variance) because of the increase in n , and the pmf is now symmetric about $np = 500$.
3. When we change p to 0.9, we shift the center of the pmf to $np = 900$.

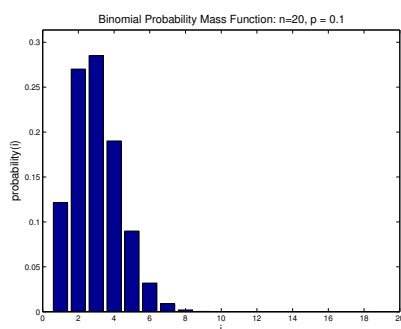


Figure 1: A bar graph of the binomial distribution with $n = 20$ and $p = 0.1$.

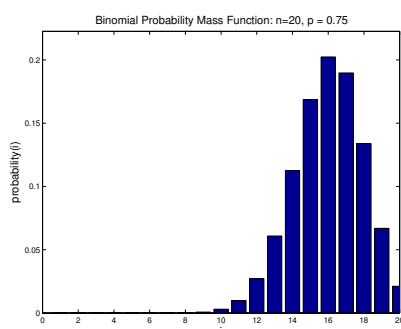


Figure 2: A graph of the binomial distribution with $n = 20$ and $p = 0.75$.

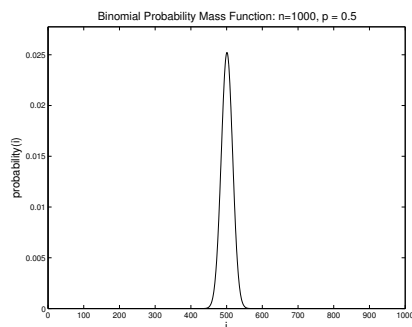


Figure 3: A graph of the binomial distribution with $n = 1000$ and $p = 0.5$.

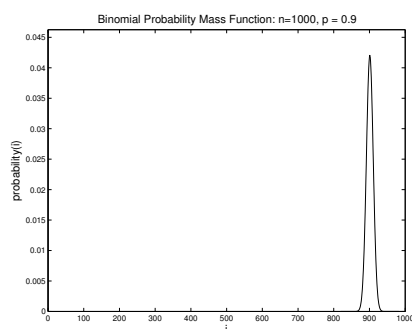


Figure 4: A graph of the binomial distribution with $n = 1000$ and $p = 0.9$.

Problem 8 (MATLAB)

As a followup to the previous question, you will now write a function called `binomial_cdf.m` that calculates the cumulative distribution function (CDF) for the binomial. This function should be called as:

```
y = binomial_cdf(n,p);
```

where y contains the values of the binomial cdf evaluated at all values from 0 to n (i.e., it will be a vector of length $n + 1$). To generate the CDF you will need to first get the values of the pmf at all values from 0 to n by calling the function `binomial_pmf.m` from within your `binomial_cdf.m` function. If $np > 5$ or $n(1 - p) > 5$ your code can call the approximation version instead, `binopmf_approx.m`.

Then you generate partial cumulative sums of the *pmf*, namely, all values summed from 0 to i , where i ranges from 0 to n . You can do this directly with a for-loop or use the MATLAB function `cumsum.m` to help do it for you (the MATLAB function will be much faster than a direct for-loop).

Now use the results from your function to plot graphs of the binomial CDF for the same 4 pairs of n and p values used in the previous problem.

Solution: See figures of the CDF function. The CDFs match what we would expect to see from the corresponding pmf figures earlier.

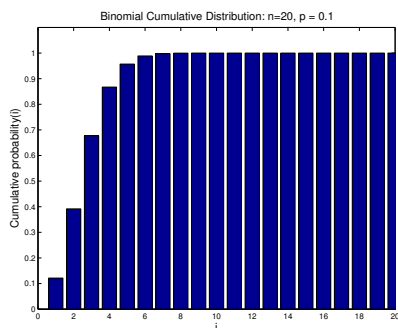


Figure 5: A bar graph of the CDF for the binomial distribution with $n = 20$ and $p = 0.1$.

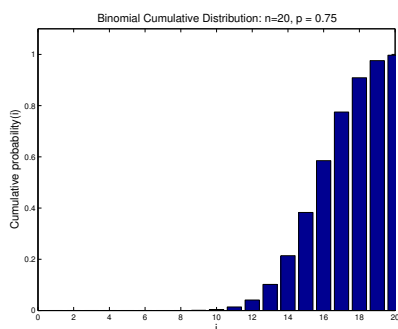


Figure 6: A graph of the CDF for the binomial distribution with $n = 20$ and $p = 0.75$.

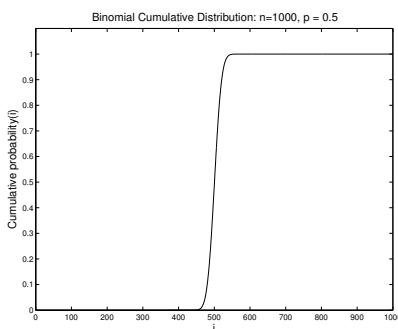


Figure 7: A graph of the CDF for the binomial distribution with $n = 1000$ and $p = 0.5$.

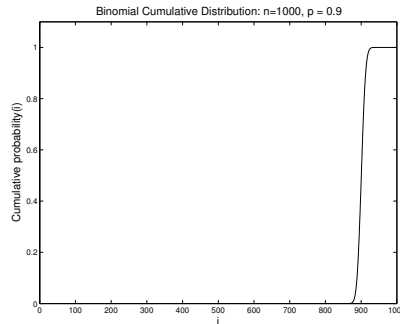


Figure 8: A graph of the CDF for the binomial distribution with $n = 1000$ and $p = 0.9$.

Problem 9 (MATLAB)

A company (e.g., Dell) sells 10000 hard-drives of a particular type in a year. Each drive has a probability of 0.9 of failing within the first year of operation. Each failed drive is returned and the company must supply the customer with a new drive. Answer the following questions using results from the MATLAB functions in previous 2 questions:

COMMENT: Note that this problem should have stated that there was a $p = 0.1$ probability of failure of each drive in the first year of operation (not 0.9)—a probability of 0.9 of failure would be unrealistically high. The problem was graded using the value of 0.9 but the solution below uses $p = 0.1$. The same methods can be used to arrive at the answers irrespective of the value of p .

It should also be clear that the number of drives that fail and are returned is a binomial random variable X , where $X \sim \text{Bin}(n, p)$, where $n = 10000$ and $p = 0.1$. Thus, we can use the MATLAB code from the previous 2 problems to calculate the solutions below.

1. What is the probability that exactly 1000 of the original 10000 drives will be returned?
SOLUTION: This is the probability that exactly 1000 drives will fail, i.e., that we will have 1000 “successes” (where success is defined here as a disk drive failing) in the binomial model. Thus,

$$P(X = 1000) = \binom{10000}{1000} p^{1000} (1-p)^{9000} = 0.133$$

2. What is the probability that 1000 or more of the original 10000 drives will be returned?
SOLUTION: Recall that $\text{CDF}(x_1)$ is defined as $P(X \leq x_1) = 1 - P(X > x_1)$. So $P(X > x_1) = 1 - \text{CDF}(x_1)$. It follows that we want to compute $P(X > 999) = 1 - \text{CDF}(999)$. Using the MATLAB code from the previous problem this evaluates to 0.5199 (which makes sense, since $E[X] = np = 1000$ so we expect about 1/2 the probability mass to be at or above the mean value).
3. What is the probability that 1100 or more of the original 10000 drives will be returned?
SOLUTION: Using the same reasoning as in the last part, this is $P(X > 1099) = 1 -$

$\text{CDF}(1099) \approx 0.0005$. Note that while there is about a 50% chance that 1000 or more disks will be returned, there is only a 0.05% chance that this number will be greater than or equal to 1100. So we could confidently put 1100 disks in storage for returns and be 99.95% sure that this would be enough for the whole year.

4. What is the probability that 900 or less of the original 10000 drives will be returned?
SOLUTION: here we can just use the CDF, i.e., $\text{CDF}(900) \approx 0.0004$. Note that combining this result with the result of the previous part, we now know that there is a 99.9% chance that between 900 and 1100 disks will be returned during the year.

5. The engineers at the company would like to switch to a new type of hard-drive for the following year. They plan to again sell 10000 such drives with the same warranty policy. They can control the manufacturing of the disk to make it more reliable by using more expensive components—so there is a tradeoff between higher reliability and cost. They would like to find the largest possible value of p (least-expensive) such that the probability of 500 or more disks being returned is less than 2×10^{-7} . Use the MATLAB functions to find the value of p that has this property—you only need to find the value of p with 2 decimal points, e.g, 0.85, 0.91, etc.

SOLUTION: The idea here is that we make p (the probability of disk failure) smaller, we increase the cost. So we would like to find the value of p that is as large as possible (as cheap as possible) and still satisfies the requirement. Given $X \sim \text{Bin}(10000, p)$ we want to find a parameter p such that $P(X > 499) < 2 \times 10^{-7}$ or equivalently that $1 - \text{CDF}(499) < 2 \times 10^{-7}$. It turns out, using trial and error and the CDF MATLAB function from the previous problem, that $p = 0.04$ gives a probability of 2.5×10^{-7} —so a p value slightly smaller than 0.04 will work, e.g., $p = 0.039$.

Interestingly, note that if we were to use the same disks as in the original problem, with $p = 0.1$ the probability of 500 or more disks being returned is essentially 1, i.e., it is virtually certain. By reducing the probability of failure from 10% ($p = 0.1$) to about 4% ($p = 0.04$) we drastically reduce the probability of 500 or more disks (down to about 2×10^{-7})—a relatively small change in p has a very large effect!