CS 177, Homework 1

Applications of Probability in Computer Science: Winter 2007

Due Date: Tuesday January 16th, in class

Reading for Week 1

Your reading this week will review material that you covered (for the most part) in Math 67:

- Chapter 1: pages 1 through 24. Review basic notions of sample spaces, events, and probability axioms.
- Chapter 2.1 and 2.2 on discrete random variables.
- Section 2.4 (expected value) pages 97 to 102, and Section 2.4.2 (variance) pages 106 through 109
- Section 2.5: pages 113–117 on Indicators and the Binomial distribution (you can skip example 2.5.2 for now), and Section 2.5.3 on the geometric distribution.

MATLAB Tutorials for Week 1

Take a tutorial session on MATLAB (you will probably need to spend at least 2 hours on this). See the class Web page for information on where to find MATLAB. Specifically, you should go through the `Getting Started` section (to bring this up, type `doc` at the command line in MATLAB), and focus on the following items:

- `Matrices and Arrays`
- `Graphics`: specifically, Overview of MATLAB Plotting, and Printing Graphics.
- `Programming`: specifically, Flow Control, and the sections on Scripts, Functions, Vectorization, Preallocation, under Scripts and Functions.
- `Desktop Tools` and `Development Environment`.

It is important that as you go through this tutorial material that you try out the commands in MATLAB to see how they work—so you should have MATLAB up and running while you go through the tutorial material and interactively try out commands as you go. We will be using MATLAB in many of the assignments, so time spent learning how to use MATLAB in the first week of class will pay off later on.

Instructions for Handing in Homework Solutions

Hand in hardcopies of solutions to all problems in class. If the problem requires MATLAB code, please append your MATLAB code to the back of your solutions. Your code should be clearly documented. If graphs are requested, include them with your solutions—clearly label your graphs and make sure it is clear which graph goes with which problem solution.

Important: for MATLAB functions that you are asked to write, please submit them electronically by the time of class on the due date to the appropriate homework folder for CS 177 under EEE. (So for Homework 1 there are 2 MATLAB functions that you need to submit). Note that MATLAB is an interpreted language where functions are interpreted at run time (not compiled) so you just submit your "source code" (the .m files) and we can run them directly.

Please staple all your solutions and mark your name clearly on the front page.

Problems

IMPORTANT: In all problems below clearly show how you arrived at your answer. Solutions that do not clearly show intermediate work will lose points.

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.
2. Toss a coin until the first "heads" shows up and count the number of tosses of "tails" that occur before this happens.
3. Roll 4 dice and compute their sum.
4. Roll 2 dice and compute their product.
5. Measure the time (rounded to integer minutes) that it takes for a disk to fail from the time it is installed.
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.

.

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 $k = 1, \ldots, 5$.

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 $A_1 \cap B_2$.

1. A wins at least 1 of the first 3 games.
2. B does not win any of the first 3 games.
3. None of the 5 games end in a draw.
4. There is 1 draw (and no more) over the 5 games.
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.

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?
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 $E[X]$?)

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?
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?
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 vowels, 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.

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

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 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.

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)
```

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.

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:

1. What is the probability that exactly 1000 of the original 10000 drives will be returned?
2. What is the probability that 1000 or more of the original 10000 drives will be returned?
3. What is the probability that 1100 or more of the original 10000 drives will be returned?
4. What is the probability that 900 or less of the original 10000 drives will be returned?

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.