COMP 150PP Programming Problems

September 14, 2016

Our designs aren't quite ready, but here's an advance look at programming problems I'll ask you to work on:

- Before we tackle these problems in earnest, I think we probably will spend one more class period on design. But if you want to start thinking about them early, here they are.
- The problems are intended for you to work on together, even outside of class.
- I suspect that everybody learns more and has more fun if we can find a way to discuss the problems on Piazza. But I'm not sure how to manage those discussions so as to avoid spoilers. I may ask you to try to put spoilers in notes labeled SPOILER in the title.

The problem-solving task

Simple distribution and probability questions

A. What distribution of integers results from throwing a single d6? What about a d12?

B. Given that you have a d6 and a d12, what's the probability of throwing a total of 11?

C. You're given a `P D` (distribution over dice) representing the ability to draw one die at random from a bag. You draw two dice. (Assume you are drawing dice "with replacement", so that you can draw two by doing the exact same probabilistic draw twice.)

- What's the probability of drawing a d6 and a d12?
- What's the probability that exactly one of the two dice is a d20?

D. You draw two dice, throw them, and total the numbers. What's the ("joint") probability of drawing a d6 and a d12 and throwing 11?

E. You draw two dice and throw them. The numbers total 11. Given that you know the total of 11, what's the ("conditional") probability that you drew a d6 and a d12?

Dice and coins

F. You throw a d6 to get number N, then toss that number of coins. What's the probability of observing exactly three heads?

G. You throw a d6 to get number N, then toss that number of coins. What's the probability of observing at least three heads?

H. You throw a d6 to get number N, then toss that number of coins. The coins land in a row. What's the probability that the row contains three *consecutive* heads?

I. I throw a d6 to get number N, then toss that number of coins. I don't tell you what N is, but I do tell you that exactly three of the coins came up heads. What's the probability that $N = 3$?

J. I throw a d6 to get number N, then toss that number of coins. I tell you that $N > 4$. What's the probability that exactly three of the coins come up heads?

K. I throw a d6 to get number N, then toss that number of coins. I don't tell you what N is, but I do tell you that exactly three of the coins came up heads. What's the probability that $N = 3$?

L. I throw a d6 to get number N, then toss that number of coins. I don't tell you what N is, but I do tell you that exactly *one* of the coins came up heads. What's the most likely value of~N?

M. I throw a d6 to get number N, then toss that number of coins. I don't tell you what N is, but I do tell you that exactly three of the coins came up heads. What's the posterior distribution of~N?

Tally-sheet questions

In all these questions, you throw a tally sheet (draw two dice, throw 30 times, mark left column for totals less than 16, the middle column for totals of exactly 16, and the right column for totals of more than 16).

N. What's the probability that you have drawn two d12's and put 3 marks in the right column of the tally sheet?

O. What's the probability that all marks are in the left column?

P. On average, how many marks can you expect to put in the right column of the tally sheet?

Gambling questions

Probability was invented by gamblers for gamblers. Let me describe the *tally-sheet game*:

> The tally-sheet game has two players: the *thrower* and the *guesser*.
>
> - The thrower creates a tally sheet by drawing two dice and throwing them 30 times using the procedure you know.
> - The guesser sees the tally sheet but not the dice and has three chances to try to guess the dice.
>
> It's not gambling unless you play for money. Here are the payouts:
>
> - If the guesser gets the right dice on the first try, the guesser wins one dollar ($1.00) from the thrower.
> - If the guesser gets the right dice on the second try, the guesser wins fifty cents ($0.50) from the thrower.
> - If the guesser gets the right dice on the third try, the guesser wins a quarter ($0.25) from the thrower.
> - If the guesser fails to get the right dice in three tries, the guesser loses and pays a dime ($0.10) to the thrower.

Write code to answer the following questions about the tally-sheet game:[1]

Q. Who is likely to win money: the guesser or the thrower?

R. You have your choice of playing guesser or thrower for a series of 1000 games. You choose whichever is more likely to win. How much do you expect to win?

S. You're playing thrower, and I ask if you'd be willing to change the rules so that instead of throwing 30 times, you throw 50 times. Assuming we're going to play 1000 games, what's the minimum amount you should insist on charging me for this privilege?

[1] In all these questions, please assume that the cost of throwing dice and the cost of computing answers are both zero. In other words, the questions should consider only the payouts from the game; all actions in the real world are free.

T. As guesser, I hate it that I can never tell the difference between d6+d8 and d6+d6. I want the option to change the rules so that the middle column corresponds to 13, not 16. (The left and right columns change correspondingly.) Assuming I'm going to play 1000 games as guesser, what's the most I should be willing to pay for the privilege of exercising this option?

U. Suppose that in addition to the original dice, I add 12 d4's (four-sided dice) that I kept out of the original experiment. Now putting the middle at 16 is even less helpful. Where should the split be to maximize the expected return for the guesser? (Assume the guesser always picks the most likely guess in the posterior distribution, which is the *maximum a posteriori* choice.)

The coding task

In the language of your choice, in teams of your choice, implement probability distributions in a way that enables you to *run* code that answers the questions above. Answer at least one question from each group. *Send me your solutions* by noon on Sunday, September 25 so we can have Code Show and Tell on Monday, September 26.

Here is some advice:

- Prefer a language that has a static, polymorphic type system. Standard ML (from COMP 105) or Haskell are entirely adequate to this task. Other reasonable choices include F#, OCaml, Scala, and (possibly) Java with generics. You could even consider C++ with templates, especially if you're a good C++ programmer and you use libgc.
- No matter what language you go with, choose an immutable representation of probability distributions.
- Keep your eye on the types of intermediate results—you have *got* to keep the state space small, or your computations won't terminate in your lifetime. The size of the state space is simply the number of values in a distribution that have nonzero probability.
- Be careful to keep the size of your representation proportional to the size of the state space. If you're very naive about how you write your functions, you risk allowing your representation to get exponentially large in the number of function applications, even if the state space is small.