ECS 20 — Lecture 20 — Fall 2013 —5 Dec 2013

Phil Rogaway

Today:

- A last bit of probability
- "Review session"
- Closing comments
- You grade me

Announcements:

- eval.ucdavis.edu ← for course evaluations
- special office hours online
- Final: you may not leave during the last 30 mins.

Basic definitions / theory

Review:

Def: A (finite) **probability space** (S,P) is

- a finite set S (*the sample space*) and
- a function P: $S \to [0,1]$ (the *probability measure*) such that that

$$\sum_{x \in S} P(x) = 1$$ (alternative notation: ω, μ, Ω for x, P, S)

In general, whenever you hear *probability* make sure that you are clear **what** is the probability space is: what is the sample space and what is the probability measure on it.

An **outcome** is a point in S.

Def: Let (S, P) be a probability space.
An **event** is a subset of S.

Def: Let A be an event of probability space (S, P).

$$P(A) = \sum_{a \in A} P(a)$$ (I'm used to using Pr, will probability slip)

The probability of event A. By convention, $P(\emptyset)=0$

Def: The **uniform** distribution is the one where $P(a) = 1/|S|$ —ie, all points are equiprobable.
Def: Events A and B are **independent** if $P(A \cap B) = P(A)\,P(B)$.
Def: if $B \neq \emptyset$ then $P(A|B) = P(A \cap B) / P(B)$

Prop: $P(A) = P(A|B)P(B) + P(A|B^c)P(B^c)$

Def: A **random variable** is a function $X\colon S \to \mathbf{R}$ from the sample space to the reals.

Def: $\mathbf{E}[X] = \sum_{s \in S} P(s)\, X(s)$ // expected value of X ("average value")

-Prop: $\mathbf{E}(X+Y) = \mathbf{E}(X) + \mathbf{E}(Y)$ // expectation is linear.

Eg 1: Alice rolls a die.
What's do you expect the square of her roll to be?

```
could be 1 ....  could be a 36!

Definition: a RV is a function from X: S -> \R

Definition: E[X] = \sum X(s) P(s)
                     s


So, in this problem,
   E[X] = 1(1/6) + 2^2(1/6) + 3^2(1/6) + ... + 6^2(1/6)
        = 1/6(1+4+9+15+25+36)
        = 91/6
        \approx 15.2

Exercise: Repeat, supposing she rolls a *pair* of dice:
```

Eg 2: Subway

When Pablo, a CS grad student at MIT, leaves his office late at night, he wanders to the subway and takes the first train North or South:

```
                             Girlfriend's home
                                 /|\
                                  |
                                  |
                                  |
         Laboratory -------> subway stop
                                  |
                                  |
                                  |
                                 \|/
                             Student's home
```

There are trains every 10 mins, both N and S. Yet during the last 31 days, Pablo only has gone home only 3 times, and this seems to be about typical, in his experience. Explain what is going on and compute Pablo's average wait time for a train? Assume trains arrive at the station on 1-minute intervals.

Example :

```
Northbound    Southbound

              11:00
                      1 min
    11:01


                      9 min



              11:10
                      1 min
    11:11


                      9 min



              11:20
    11:21

Let X = Wait time

    (1/10) (0.5 min)  + (9/10) ( 4.5 mins)
   = 0.05 + 4.05 mins
   = 4.1 mins
```

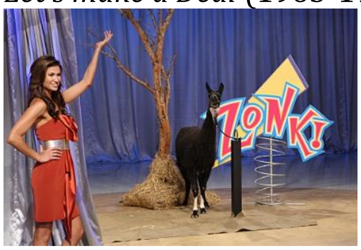
How should the trains be staggered to minimize Pablo's wait time?

```
               11:00
     11:05
               11:10
     11:05
               11:20

Average wait time will be 2.5 mins
```

Ex 3: Monty Hall Problem

Let's make a Deal (1963-1968)




```
========     ========     ========
|      |     |      |     |      |
| bad  |     | bad  |     | good |
|      |     |      |     |      |
========     ========     ========
   1            2             3
```

A good prize is hidden behind a random curtain/door (junk, "zonk", behind the other two). You choose an arbitrary door. The host opens one of the unselected doors that does NOT contain the good prize. Should you switch to the other door?

```
   loc of good prize       which door to open if host must choose
S =     {1, 2, 3}     ×    {0, 1}
                                            not relevant
WIN = get good prize
Strategy STICK: choose door 1 and stick with it:  P (WIN)=1/3
Strategy SWITCH: choose door 1 and then switch (always) when offered a
chance.

(1,0)  (2,0)   (3,0)        Second bit doesn't matter.
Lose    Win     Win
(1,1)  (2,1)   (3,1)
Lose    Win     Win

P(Win) = 4/6 = 2/3

Or:  P(Win) =    P(Win | initialCorrect) Pr(initialCorrect)
              + P(Win | initialIncorrect) Pr(initialIncorrect)

            =    0 + 1 (2/3) = 2/3

Or just choose door 1
lose  win  win
```

Eg 4: Smaller / Bigger game

Alice uniformly chooses two distinct numbers between1 and 10 and announces the **first**. Bob guesses if the second is **smaller** or **bigger** than the first How should Bob play optimally—and, if he does so, what is his chance to win?

As usual, start by figuring out the sample space
$S = \{(i,j) \in \{1..10\}^2 : i \neq j\}$
Alice announces any of

```
   1  2  3  4  5  6  7  8  9  10
```

Bob's strategy:

```
- If Alice announces 1,2,3,4,5  guess SMALLER
- If Alice announces 6,7,8,9,10 guess BIGGER

P(Win) = P(Win| AliceAnswers1) P(AliceAnswers1) +
         P(Win| AliceAnswers2) P(AliceAnswers2) +
         ..
         P(Win| AliceAnswers10) P(AliceAnswers10)


       = (1/10) (P(Win | AliceAnswers1)  + ... + P(Win | AliceAnswers10)
       = (1/10) (9/9 + 8/9 + 7/9 + 6/9 + 5/9 +
                 5/9 + 6/9 + 7/9 + 8/9 + 9/9)
       = (1/10) (7*10/9)             numbers clearly average 7
       = 70/90
       = 7/9 ≈ 78%
```