Enumeration

ECE 341: Random Processes

Lecture #3

note: All lecture notes, homework sets, and solutions are posted on www.BisonAcademy.com

Enumeration

Probability is defined as the number of times an event occurs as the number of trials goes to infinity.

This leads to the previous lecture

- Monte Carlo experiments

A second method is enumeration

- Assume all outcomes have equal probability
- The *exact* probability of an event is then

$$p(x) = \left(\frac{\text{number of ways to obtain outcome x}}{\text{total number of possible outcomes}}\right)$$

Enumeration is a brute-force technique.

- In some cases, it works very well.
- In other cases, there are simply too many possible outcomes

Topics in This Lecture

Solve the same problems we solved using Monte-Carlo techniques:

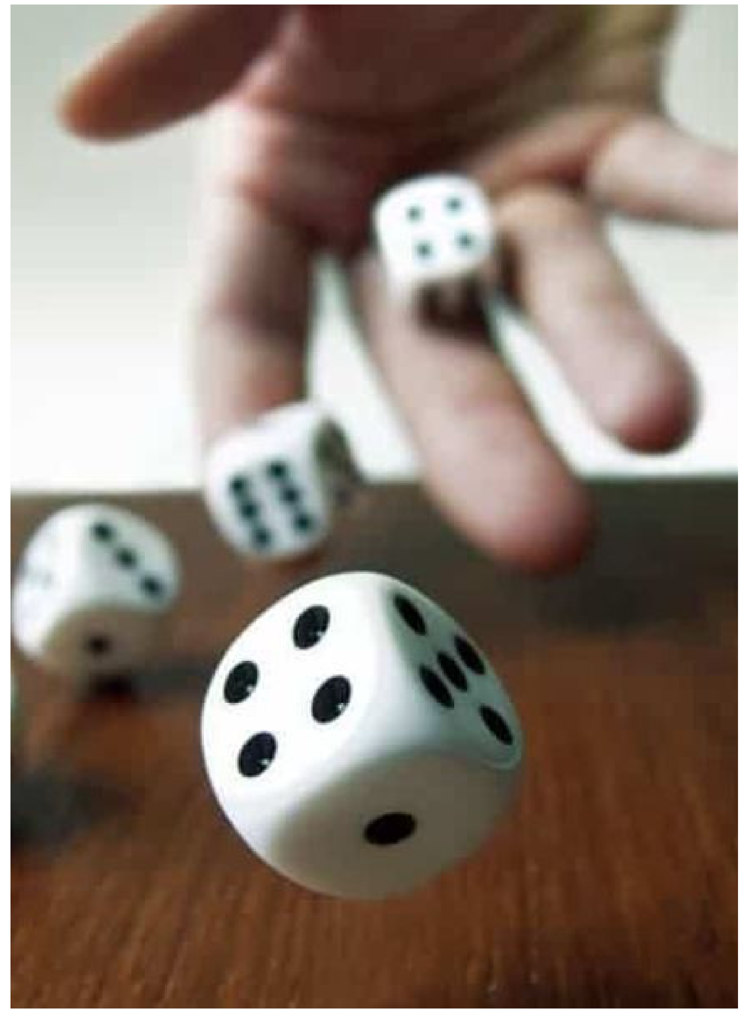
- Probability of rolling a 1 on a 6-sided die
- Probability distribution of the max(d4, d6)
- Max of (d4, d6) vs. d6
- 5-Game Match (tree diagram),
- Rolling 6-Dice
- Drawing a full-house or 3-of-a-kind in poker

Case 1: Rolling a single 6-sided die (d6)

What's the probability of rolling a 1 on a 6-sided die?

Previous lecture with 1 million rolls

- Trial 1: 166,219 times
- Trial 2: 167,090 times
- Trial 3: 166,969 times



From these results,

$$p \approx 0.166 \quad \text{(ish)}$$

That's one problem with Monte Carlo simulations

- Results are approximate
- You *can* place a bound on the actual probability
  - Student-t test
  - Future topic

Case 1: Enumeration

Assume all outcomes have equal probability.

List all possible outcomes

- {1, 2, 3, 5, 6}
- N = 6

Count how many are successes

- {1}
- M = 1

Hence, the probability of rolling a one is

$$p = \frac{M}{N} = 1/6$$

Note

- This matches up with Monte Carlo experiments
- The answer is exact

Case 2: A = max(d4, d6)

Player A rolls two dice

- { d4, d6 }

A's score is the maximum of the two

What is the probability A scores 1..6 points?

Case 2 (cont'd)

Step 1: List out all possibilities

- There are 24 possibilities (N = 24)

(1,1)	(1,2)	(1,3)	(1,4)	(1,5)	(1.6)
(2,1)	(2,2)	(2,3)	(2,4)	(2,5)	(2,6)
(3,1)	(3,2)	(3,3)	(3,4)	(3,5)	(3,6)
(4,1)	(4,2)	(4,3)	(4,4)	(4,5)	(4,6)

Step 2: List all possible ways to get each outcome:

- 1: (1,1)
- 2: (2,1), (2,2), (2,3)
- 3: (3,1), (3,2), (3,3), (2,3), (1,3)
- 4: (1,4), (2,4), (3,4), (4,4), (4,3), (4,2), (4,1)
- 5: (1,5), (2,5), (3,5), (4,5)
- 6: (1,6), (2,6), (3,6), (4,6)

Case 2: (cont'd)

The odds are then

- The number of ways to get each outcome
- Divided by the total number of possible outcomes

x (Score)	Number of Ways to Get this Score	p(x)
1	1	1/24
2	3	3/24
3	5	5/24
4	7	7/24
5	4	4/24
6	4	4/24

p(x) is termed *the probability density function*

Case 3: max(d4, d6) vs. d6 (conditional probability)

A third example is this:

- A rolls a d4 and a d6
  - Takes the higher score
- B rolls a d6

Highest score wins

- B wins on ties.

What is the probability that A will win this game?

- Monte Carlo: $p \approx 0.486$

Case 3: (cont'd)

Problem: Too many possible outcomes

- N = 4 * 6 * 6 = 144

Solution: Use conditional probabilities

- Divide-and-conquer technique
- Split this into six smaller problems

p(A) =

- p(A|B=1) p(B=1) +
- p(A|B=2) p(B=2) +
- p(A|B=3) p(B=3) +
- p(A|B=4) p(B=4) +
- p(A|B=5) p(B=5) +
- p(A|B=6) p(B=6)

Case 3: (cont'd)

B = 1:

- A has to score 2 or higher to win.
- There are 23 ways for A to score 2 or more points, meaning

$p(A|B=1) = \frac{23}{24}$ $\qquad$ $p(B=1) = \frac{1}{6}$

B = 2:

- A has to score 3 or higher to win.
- There are 20 ways for A to score 3 or more points, meaning

$p(A|B=2) = \frac{20}{24}$ $\qquad$ $p(B=2) = \frac{1}{6}$

B = 3:

- A has to score 4 or higher to win.
- There are 15 ways for A to score 4 or more points, meaning

$p(A|B=3) = \frac{15}{24}$ $\qquad$ $p(B=3) = \frac{1}{6}$

Case 3 (cont'd)

B = 4:

- A has to score 5 or higher to win.
- There are 8 ways for A to score 5 or more points, meaning

$p(A|B=4) = \frac{8}{24}$ $\qquad$ $p(B=4) = \frac{1}{6}$

B = 5:

- A has to score 6 or higher to win.
- There are 4 ways for A to score 6 or more points, meaning

$p(A|B=5) = \frac{4}{24}$ $\qquad$ $p(B=5) = \frac{1}{6}$

B = 6: A loses

$p(A|B=6) = 0$ $\qquad$ $p(B=6) = \frac{1}{6}$

Case 3 (cont'd)

Therefore, the probability that A wins is

$$p(A) = \left(\frac{23}{24}\right)\left(\frac{1}{6}\right) + \left(\frac{20}{24}\right)\left(\frac{1}{6}\right) + \left(\frac{15}{24}\right)\left(\frac{1}{6}\right) + \left(\frac{8}{24}\right)\left(\frac{1}{6}\right) + \left(\frac{4}{24}\right)\left(\frac{1}{6}\right) + \left(\frac{0}{24}\right)\left(\frac{1}{6}\right)$$

$$p(A) = \left(\frac{70}{144}\right) = 0.486111$$

This matches up with the Monte-Carlo simulations

- except that this answer is exact.

Case 4: 5-Game Match (Tree Analysis)

- Similar to Baseball or NBA finals

A and B are playing a match

- A has a 60% chance of winning any given game.
- Match consists of 5 games
- Whoever wins the most games wins the match

What is the chance that A wins the match?

Monte Carlo results:

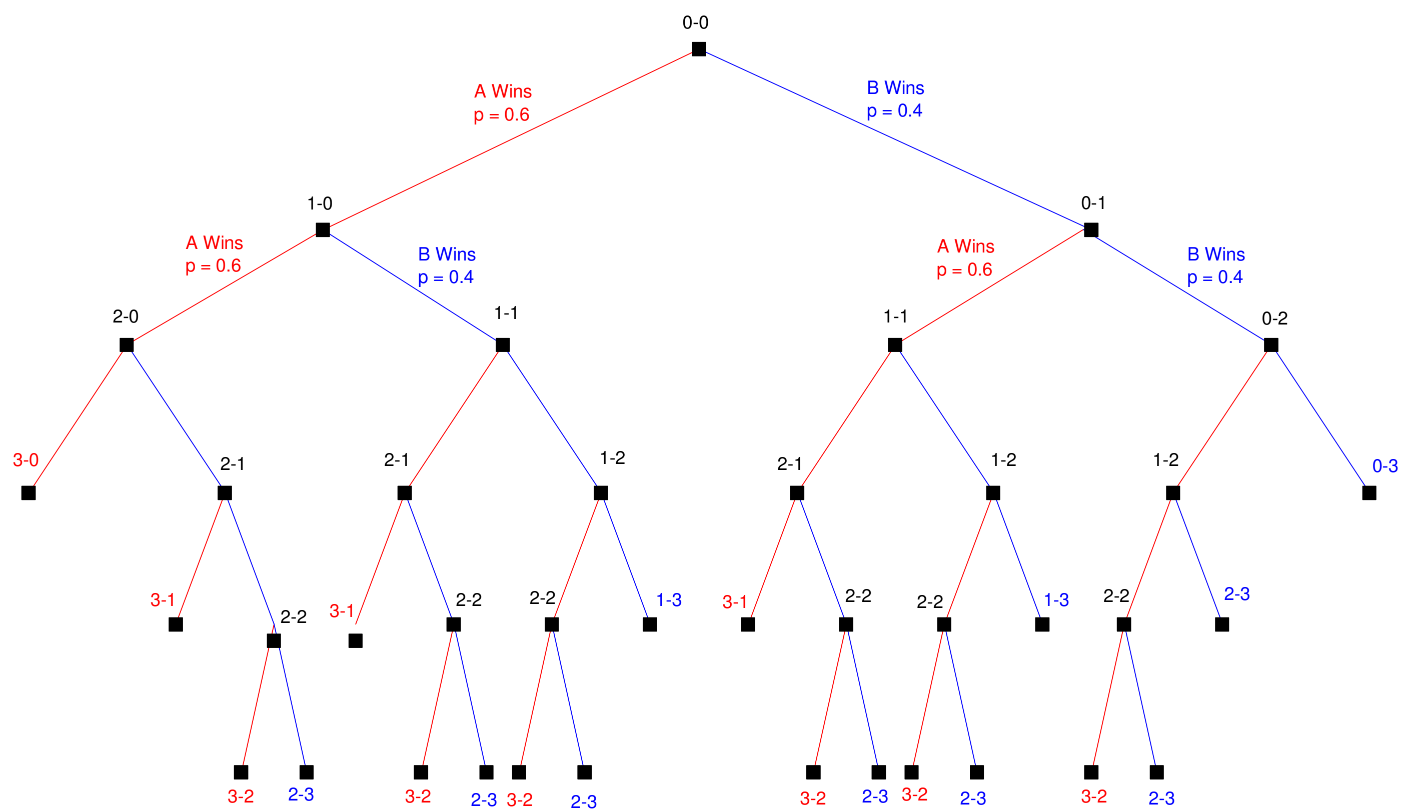
- p = 0.682 (ish)

Case 4: Tree Diagram

Each game has two possible outcomes:

- A wins (p = 0.6)
- B wins (p = 0.4)

List out all ways the series can proceed:



Case 4 (cont'd)

Count the number of ways A wins:

- 1 outcome ends 3-0
- 3 outcomes end in 3-1
- 6 outcomes end in 3-2

The odds of A winning the match are

$$p(A) = 1 \cdot p^3 + 3 \cdot p^3 q + 6 \cdot p^3 q^2$$

$$p(A) = 0.68256$$

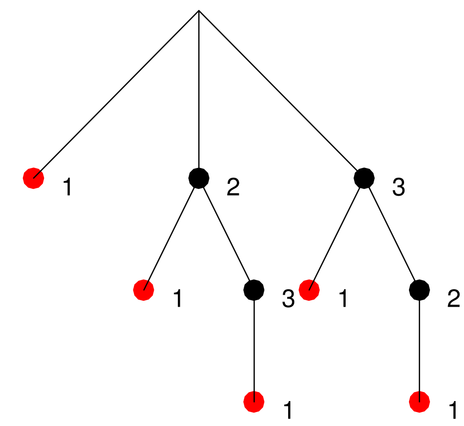
Note

- This matches up with Monte Carlo
- The answer is *exact*

Sidelight: Sampling With and Without Replacement

Tree diagrams for finite series

- First to win 3 games
- In a bin of 3 marbles (2 black, one red)
  - Pick one marble
  - Stop if it's red
  - If it's not red, leave it our and repeat

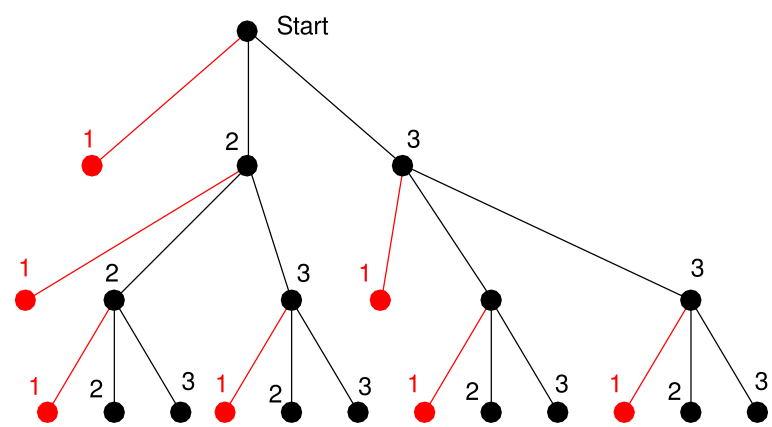


Tree diagrams do not work for infinite series

- First to win *by* 3 games
- In a bin of 3 marbles (2 black, one red)
  - Pick one marble
  - Stop if it's red
  - If it's not red, replace the marble
  - Repeat

For the latter, we need a different tool

- Markov chains
- Future topic



Enumeration with Matlab

Enumeration is a brute-force solution

- Go through every possible outcome
- Count how many of them were successes

With Matlab, you can write programs to grind out all possibilities using nested for-loops.

Case 5: Farkle (6d6)

Suppose you roll six 6-sided dice (6d6)

What are the odds of rolling

- Two triples (xxx yyy)?
- One triple (xxx aab or xxx abc)?

From Monte-Carlo

- Two triples: 6337 in 1,000,000 rolls
- One triple: 308,026 in 1,000,000 rolls

Case 5: Number of Rolls

There are 46,656 ways to roll 6d6.

- The first die has six possibilities
- The second die also has six possibilities
- etc

$$N = 6^6 = 46,656$$

That seems like a large number, but it's no problem for Matlab.

Case 5 Nested For-Loops

Start by going through every possible outcome

- Nested for-loops
- 46,656 different outcomes

```
for d1 = 1:6
   for d2 = 1:6
      for d3 = 1:6
         for d4 = 1:6
            for d5 = 1:6
               for d6 = 1:6
                  Roll = [d1, d2, d3, d4, d5, d6];
               end
            end
         end
      end
   end
end
```

Case 5: Determine the frequency

Once you roll the dice

- Find the frequency of each number
- Sort in decreasing order

```
Roll = [d1, d2, d3, d4, d5, d6];
F = zeros(1,6);
for i=1:6
   F(i) = sum(Roll == i);
end
F = sort(F, 'descend')

[Roll]
[F]
```

script window

Example:

- Roll = {2, 5, 2, 2, 5, 6}
- F(1) = 3
  - There are three 2's
- F(2) = 2
  - There are two 5's
  - Next highest frequency
- F(3) = 1
  - There is one 6

```
Roll = 2  5  2  2  5  6

F = 3  1  1  0  0  0
```

command window

Case 5: Determine the roll type

Once you know F(), you can determine the type of hand

```
    if( (F(1) == 3)*(F(2)==3))
        Pair33 = Pair33 + 1;
        end
    if( (F(1) == 3)*(F(2)<3))
        Pair3 = Pair3 + 1;
        end
```

By counting, you'll know the total number of hands that result in two and one triples.

Net Code

- Nested for-loops
- Goes through all combinations

```
Pair33 = 0;
Pair3 = 0;
N = 0;
for d1 = 1:6
  for d2 = 1:6
    for d3 = 1:6
      for d4 = 1:6
        for d5 = 1:6
          for d6 = 1:6
             Dice = [d1,d2,d3,d4,d5,d6];
             N = N + 1;
             F = zeros(1,6);
             for i=1:6
               F(i) = sum(Dice == i);
             end
             F = sort(F, 'descend');
             if( (F(1) == 3)*(F(2) == 3))
               Pair33 = Pair33 + 1;
             end
             if( (F(1) == 3)*(F(2) < 3))
               Pair3 = Pair3 + 1;
             end
           end
         end
       end
     end
   end
end
```

Case 5: Results

The results are

- Pair33 =   300
- Pair3 =   14400
- N   =   46656
- Elapsed time is 1.972013 seconds.

There are

- 300 ways to get two triples,
- 14,000 ways to get one triple, and
- 46,656 total number of ways to roll three dice.

This took 1.97 seconds

- 3.4GHz Windows computer
- Not a problem for Matlab

Case 6: Enumeration with Card Games

Finally, let's use enumeration in poker

- 52 card deck
- Deal out 5 cards

Create nested for-loops

- Avoid duplication of cards

Go through every possible hand

- 2,598,960 total

```
N = 0;
for c1 = 1:52
  for c2 = c1+1:52
    for c3 = c2+1:52
      for c4 = c3+1:52
        for c5 = c4+1:52
          Hand = [c1,c2,c3,c4,c5] - 1;
          N = N + 1
        end
      end
    end
  end
end
[N]
```

Script Window

```
N = 25989690
```

Command Window

Case 6: Determine hand

Hand is card number

- 0 to 51

Value = card value

- 1..13
- Ace through King

Suit = Card suit

- 1..4
- Club, Diamond, Heart, Spade

Example:

- Card #1 is the 2 of clubs
- Card #7 is the 8 of clubs
- Card #9 is the 10 of clubs
- Card #22 is the 10 of diamonds
- Card #47 is the 9 of spades

```
Hand = [c1,c2,c3,c4,c5] - 1;
Value = mod(Hand, 13) + 1
Suit = floor(Hand/13) + 1
```

Script Window

```
Hand  =  1   7   9  22  47
Value =  2   8  10  10   9
Suit  =  1   1   1   2   4
```

Command Window

Case 6: Determine hand type

Once you have your hand

- Determine the frequency of each card
  - variable F()
- Sort in descending order
  - F(1) is highest frequency of cards
- Check hand type
  - 3 + 2 = full house
  - 3 + 1 = three of a kind

```
F = zeros(1,5);
for i=1:13
    F(i) = sum(Value == i);
end
F = sort(F, 'descend');
if( (F(1) == 3)*(F(2) == 2) )
    FH = FH + 1;
elseif( (F(1) == 3)*(F(2) < 2) )
    Pair3 = Pair3 + 1; end
end
```

Case 6: Resulting Matlab Code

Every possible poker hand

- Loops 2,598,960 times
- Takes 186 seconds to run

```
Pair3 = 0;
FH = 0;
N = 0;
for c1=1:52
  for c2 = c1+1:52
    for c3 = c2+1:52
      for c4 = c3+1:52
        for c5 = c4+1:52
          N = N + 1;
          Hand = [c1,c2,c3,c4,c5] - 1;
          Value = mod(Hand,13) + 1;
          Suit = floor(Hand/13) + 1;
          F = zeros(1,13);
          for n=1:13
            F(n) = sum(Value == n);
          end
          F = sort(F, 'descend');
          if ((F(1) == 3)*(F(2) == 2))
             FH = FH + 1;
          elseif ((F(1) == 3)*(F(2) < 2))
             Pair3 = Pair3 + 1;
          end
        end
      end
    end
  end
end
```

Case 6: Results

Net Result:

- 2,598,960 poker hands
- 3744 full-houses
- 54,912 three-of-a-kind

Results match with Monte Carlo

- Monte-Carlo is approximate
- Enumeration is exact

Results match with Wikipedia

- Poker has been analyzed to death

$$p(fh) = \left(\frac{3744}{2{,}598{,}960}\right) = 0.0014406$$

$$p(3ok) = \left(\frac{54{,}912}{2{,}598{,}960}\right) = 0.0211285$$

```
N =           2598960
FH =             3744
Pair3 =         54912
Elapsed time is 186.303521 seconds.
```

Summary

While Monte-Carlo simulations give you approximate probabilities, enumeration gives you exact probabilities.

Enumeration is a brute-force approach:

- You go through list out every possible outcome.

Assuming each outcome has equal probability,

$$p = \left(\frac{\text{the number of successful outcomes}}{\text{the total number of outcomes}}\right)$$

Sometimes, enumeration works well

Sometimes, enumeration doesn't work

- There are too many possible outcomes

For the latter case, we need a different tool

- Combinatorics
- Next lecture