Juliette Woodrow
CS 109

Section #6

Section 6

1 Beta Sum Warmup

What is the distribution of the sum of 100 IID Betas? Let X be the sum:

$$X = \sum_{i=1}^{100} X_i \quad \text{where each } X_i \sim \text{Beta}(a = 3, b = 4)$$

Note the expectation and variance of a Beta:

$$E[X_i] = \frac{a}{a+b} \qquad \text{Var}(X_i) = \frac{ab}{(a+b)^2(a+b+1)} \qquad \text{Where } X_i \sim \text{Beta}(a, b)$$

2 Timing Attack

In this problem we will see how to crack a password in linear time by measuring how long the password check takes to execute (see code below).

```
# An insecure string comparison
def does_password_match(guess, password):
   n_guess = len(guess)
   n_password = len(password)
   if n_guess != n_password:
      return False                 # 4 lines executed to get here
   for i in range(n_guess):
      if guess[i] != password[i]:
         return False              # 6 + 2i lines executed to get here
   return True                     # 5 + 2n lines executed to get here
```

Assume that our server takes T ms to execute any line in the code where $T \sim N(\mu = 5, \sigma^2 = 0.5)$ milliseconds. The amount of time taken to execute a line is always independent of other lines.

On our site, all passwords only use lower case letters and are between 5 and 10 letters long, inclusive. A hacker is trying to crack the root password which is "gobayes" by carefully measuring how long the code takes to tell her that her guesses are incorrect.

a. What is the distribution for the time it takes to execute k lines of code?

b. First, the hacker needs to find the length of the password. What is the probability that the time taken to check a guess of correct length (when the server executes 6 lines) is longer than the time taken to check a guess of an incorrect length (when the server only executes 4 lines)? Assume the first letter of the guess does not match the password's first letter. Hint: $P(A > B) = P(A - B > 0)$.

c. (Optional) Now that our hacker knows the length of the password, to get the actual string, she will try to determine one letter at a time, starting with the first letter. To start, the hacker tries the string "aaaaaaa" and sees that it takes 27ms. Based on this timing, how much more probable is it that first character did not match (server executes 6 lines) than the first character did match (server executes 8 lines)? Assume that all letters in the alphabet are equally likely to be the first letter.

d. (Optional) If it takes the hacker 6 guesses to find the length of the password, and 26 guesses per letter to crack the password string, how many attempts does she need to crack our password, "gobayes"? Yikes!

3 Binary Tree

Consider the following function for constructing binary trees:

```
def random_binary_tree(p):
    """
    Returns a dictionary representing a random binary tree structure.
    The dictionary can have two keys, "left" and "right".
    """
    if random_bernoulli(p):   # returns true with probability p
        new_node = {}  # initialize one new node
        new_node["left"] = random_binary_tree(p)
        new_node["right"] = random_binary_tree(p)
        return new_node
    else:
        return None
```

The `if` branch is taken with probability p (and the `else` branch with probability $1 - p$). A tree with no nodes is represented by `None`; so a tree node with no left child has `None` for the `left` field (and the same for the right child).

(a) Let X be the number of nodes in a tree returned by `random_binary_tree(p)`. You can assume $0 < p < 0.5$. What is $E[X]$, in terms of p?

(b) (Optional) Why did we need to assume that p is less than 0.5?