7 Probability Trees and Bayes

7.1 Bob's Beautiful Boxes

Bob has two beautiful boxes of balls. Box A contains 2 green balls and 7 red balls. Box B contains 4 green balls and 3 red balls. Bob flips a coin to randomly select a box. Then he randomly selects one ball from that box. If the selected ball is red, what is the probability that it was chosen from Box A?

Let's build up to this step by step.

a. These balls are not equally likely to be selected. Why not? Which are more likely and which less likely?

b. Let's define some events.

- A: Bob selects Box A
- B: Bob selects Box B
- R: Bob selects a red ball

c. Let's do some inventory. Which of these probabilities can we easily determine? Which is our main question? How can you use the items on the list that you know to figure out the values you don't know? (Feel free to add additional probabilities to the inventory if that is helpful.)

$$p(A) \qquad p(B) \qquad p(R)$$

$$p(A \cap B) \qquad p(A \cup B) \qquad p(A \cap R) \qquad p(B \cap R)$$

$$p(A \mid R) \qquad p(R \mid A) \qquad p(B \mid R) \qquad p(R \mid B)$$

7.2 Breast cancer screening

Here is some information about breast cancer screening. (Note: These percentages are approximate, and very difficult to estimate.)

- American Cancer Society estimates that about 1.7% of women have breast cancer. http://www.cancer.org/cancer/cancerbasics/cancer-prevalence
- Susan G. Komen For The Cure Foundation states that mammography correctly identifies about 78% of women who truly have breast cancer. http://ww5.komen.org/BreastCancer/AccuracyofMammograms.html
- An article published in 2003 suggests that up to 10% of all mammograms are false positive. http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1360940

If a mammogram yields a positive result, what is the probability that patient has cancer?

7.3 Disease Testing

A test for a rare medical condition that affects 1 person in 10,000 has the following properties: If a person is healthy, it correctly diagnoses this 98% of the time. If a person is diseased, it correctly diagnoses this 99% of the time. If you take the test and it comes back positive (ie, the test says you have the disease), what is the probability that you have the disease?

Hint: Use the inventory method. Useful events include

- H: person is healthy
- D: person is diseased
- $+$: test is positive (indicates disease)
- $-$: test is negative (indicates healthy)

7.4 Random Primality Testing

Miller's test for primality of a number n uses arithmetic mod n and a base b with $1 < b < n$. For a given b, the test either determines that n is composite or it is inconclusive. It can be shown that for any composite number, the test is conclusive and correct for at least 3/4 of the possible values of b. For a prime number, the test is always inconclusive. This leads to an interesting randomized test:

```
string PrimeTest(n, attempts) {
  for (i = 0; i < attempts; i++) {
    select b at random with 1 < b < n
    if (MillersTest(n, b) ==  "composite") {
      return("composite (for sure)")
    }
    return("prime (well, probably)")
  }
}
```

a. If n is prime, what is the probability that `PrimeTest(n, 5)` returns `"prime (well, probably)"`?

b. If n is composite, what is the probability that `PrimeTest(n, 5)` returns `"prime (well, probably)"`?

c. If n is composite, what is the probability that `PrimeTest(n, 10)` returns `"prime (well, probably)"`?

d. If n is composite, what is the probability that `PrimeTest(n, 20)` returns `"prime (well, probably)"`?

Note: For RSA cryptography purposes, once we get two large probably prime numbers, we can try encoding and decoding a message. If a composite number sneaks through the randomized test, decoding won't work and we can try again with a different pair of probably prime numbers.

7.5 Hashing

A hash function h assigns one of m storage locations (an index in an array, for example) to each key k. Typically there are many more keys than storage locations, but only a much smaller number of keys will occur in a particular application. For example, if the keys are social security numbers, then there are approximately 350 active social security numbers (keys), but a give application may have data on only a very small fraction of these people. If $h(k_1) = h(k_2)$, then we say there is a collision (because we would like to put both keys into the same storage location and have to figure out some work-around).

Good hash functions have the property that if a key is selected at random, then the probability of being mapped to a specified storage location is approximately $1/m$, where m is the number of storage locations. (So each storage location is roughly equally likely to be selected.) For the questions below, assume that the probability is exactly $1/m$. That should be a good approximation for good hash functions.

a. A hash function is mapping to 20 storage locations. What is the probability that we can place 4 keys no collisions? (That is, what is the probability that the first four hash values will all be different.)

b. A hash function is mapping to 50 storage locations. What is the probability that we can place 10 keys no collisions? (That is, what is the probability that the first ten hash values will all be different.)

c. If a hash function is mapping to 300 storage locations, what is the probability that we can place 25 keys no collisions?

d. Make up a similar problem of your own with a different number of storage locations and a different number of keys to place.

e. This is probably too big for your calculator, but if m is 1 million, then the probability of placing 1178 keys without any collisions is $\approx 1/2$.

f. Explain why these results indicate that hash tables will need to deal with collisions if they are going to be memory efficient.