**INFO 2950 Lecture 20 bonus clarifications (2023-11-06)**

1. **Why do we double the Urn A rows for "Bayes intuition"? (Lec 20 Slide 43)**

Instead of representing the actual items in the urns, now we're just taking our "priors" from our first draw (a pumpkin) and recognizing that, given the information we have, Urn A is twice as likely as Urn B (because there are twice as many pumpkins in Urn A as Urn B).

Before, we represented our distribution of likelihoods as drawing from one of two urns (A or B). We can now re-represent our distribution of likelihoods intuitively such that Urn A is twice as likely, by basically saying that we are drawing from one of three urns (A, B, or A again). If we're drawing from one of these three urns (A, A, or B) the probability of Urn A is built in to be twice as likely as Urn B (which is our prior). Then, looking at the items inside of our urns A, A, and B, we can calculate the distributions of getting each item accordingly.

2. **Why do we add 14 to the denominator when doing Laplace smoothing? (Lec 20 Slide 157)**

Before, we only looked at the probabilities for the words in our test string ("a very close game") and ignored the other words that occurred in our training data. Let's look at a more comprehensive table of counts if we include *all the words* in our training data. The below table has 14 rows.

| Word | # in Sports | # in Not Sports | Laplace Corrected Counts in Sports (2nd column + 1) | Laplace Corrected Counts in Not Sports (3rd column + 1) |
|---|---|---|---|---|
| "a" | 2 | 1 | 3 | 2 |
| "very" | 1 | 0 | 2 | 1 |
| "close" | 0 | 1 | 1 | 2 |
| "game" | 2 | 0 | 3 | 1 |
| "great" | 1 | 0 | 2 | 1 |
| "the" | 0 | 1 | 1 | 2 |
| "election" | 0 | 2 | 1 | 3 |
| "was" | 0 | 2 | 1 | 3 |
| "over" | 0 | 1 | 1 | 2 |
| "clean" | 2 | 0 | 3 | 1 |
| "match" | 1 | 0 | 2 | 1 |
| "but" | 1 | 0 | 2 | 1 |
| "forgettable" | 1 | 0 | 2 | 1 |
| "it" | 0 | 1 | 1 | 2 |

Now, if we want to calculate the Laplace-corrected Pr(Words|Sports=1), we want to divide the probability of a specific word by the total # words in Sports (i.e., sum up column 4).
The total # words in sports = [the sum of column 2] + [the # distinct words] = 11 + 14 = [the sum of column 4] = 25. So, the Pr("a"|Sports=1) = 3 / 25 (i.e., that row of column 4 divided by the sum of column 4).

If we want to calculate the Laplace-corrected Pr(Words|Sports=0), we want to divide the probability of a specific word by the total # words in Not-Sports (i.e., sum up column 5).
The total # words in not-sports = [the sum of column 3] + [the # distinct words] = 9 + 14 = [the sum of column 5] = 23. So, the Pr("a"|Sports=0) = 2 / 23 (i.e., that row of column 5 divided by the sum of column 5).

### 3. Why do we use Log probabilities re: monotonicity? (Lec 20 Slide 90)

We want to compare whether the probability of, for example, being a Sports text is greater than being a Not-sports text.

If Pr(Sports=1|text) > Pr(Sports=0|text), then Log[Pr(Sports=1|text)] > Log[Pr(Sports=0|text)], always.

If Pr(Sports=1|text) < Pr(Sports=0|text), then Log[Pr(Sports=1|text)] < Log[Pr(Sports=0|text)], always.

If Pr(Sports=1|text) = Pr(Sports=0|text), then Log[Pr(Sports=1|text)] = Log[Pr(Sports=0|text)], always.

Taking the log of probabilities will always preserve the ordering of which probabilities are bigger and which probabilities are smaller. This is because logarithms are monotonic functions!

This is useful because, at the end of the day, what we care about is which Sports value gives us the highest probability for Pr(Sports|text) – i.e., the "argmax." If Pr(Sports=1|text)>Pr(Sports=0|text), then we want to classify the text with a Sports tag because Sports=1 was the Y that gave us the highest Pr(Sports=Y|text). We can come to the exact same conclusion by finding that Log[Pr(Sports=1|text)] > Log[Pr(Sports=0|text)].

Why use logarithms at all instead of just comparing the original probabilities? Because we're less likely to end up with Python-induced 0's when multiplying together probabilities that are very very small.