The Cooper Union Department of Electrical Engineering
Prof. Fred L. Fontaine
ECE300 Communication Theory
Problem Set II: Decision Theory
September 18, 2023

1. **MAP vs ML**

   Here you will study MAP and ML decision rules using urns. All should be coded up in MATLAB so you can readily change the numbers for testing. In the first part, you will set up the model and determine the MAP and ML decision rules. In the second part you will use a random number generator to simulate results. The only random number generators you are allowed to use are: randi, rand, randn, randperm. [One of these will be the easiest for this problem] Do not try to literally simulate balls in an urn! You just want to generate events with appropriate probability!

   Urn I contains r_1 red balls and b_1 blue balls, and Urn II contains r_2 red balls and b_2 blue balls. You pick a ball from urn I at random and place it into urn II. Then you pick a ball from urn II and based on its color you make a decision on the color of the ball drawn from urn I. Let us use notation R_I, R_{II}, B_I, B_{II} to denote the various *events:* R_I is selecting a red ball from urn I, R_{II} selecting a red ball from urn II, and similarly for B_I, B_{II}.

   (a) Write MATLAB code that, given b_1, r_1, b_2, r_2, computes the prior, likelihood functions, and posterior distributions.

   (b) A decision rule can be represented as a vector of length two, i.e., [decision if Blue from II, decision if Red from II], with the value 1 for Blue from I and 2 for Red from I. For example, the vector $[2, 1]$ would mean decide 2=Red from Urn I if you get Blue from urn II, and decide 1=Blue from Urn I if you get Red from Urn II. With this, write code to generate the MAP and ML decision vectors. **Note:** In case of a tie, your code should work, but it doesn't matter if it assigns the decision to 1 or 2. That is, you can use either $<$ or $\leq$, as long as the program doesn't crash if there is a tie!

   (c) Write code to compute the probability of error associated with any decision vector.

   (d) Take three cases:

   - Case 1: $b_1 = 1$, $r_1 = 9$, and then $b_2 = 1$, $r_2 = 9$.
   - Case II: $b_1 = 4$, $r_1 = 6$, and then $b_2 = 1$, $r_2 = 9$.
   - Case III: $b_1 = 1$, $r_1 = 9$ and then $b_2 = 4$, $r_2 = 6$.

   For each case, generate the decision vectors and compute the associated (theoretical) probability of error for the MAP and ML decision rules.

   (e) Should the ML decision depend on the distribution of balls in Urn I? Look at the decision rules your algorithm determined- is this consistent?

   (f) In general, could the MAP decision depend on the distribution of balls in Urn I? Look at the decision rules your algorithm determined- is this consistent?

(g) Now simulate running the experiment $N = 10^5$ times, for each of the three Cases above. Each time you draw balls from the urns, match the decision made against the correct value, and keep track of the number of errors so that at the end you can estimate the overall probability of error. (You don't need to store all the decisions in memory, just enough so you can compute the probability of error). Compare the values of probability of error in the experiment with the theoretical values you computed before.