Thresholding-based decision with a single sample

Objective

Simulate a binary constant-signal detection system based on thresholding, and characterize the system via the Receiver Operating Characteristic.

Theoretical aspects

Decision with a single sample

Consider a binary message encoded with two constant levels 0 and A, affected by white noise. Taking a single sample of the signal we obtain a value r:

$$r = s + n,$$

where s is the true signal value (0 or A), and n is the sample of noise.

The receiver decides what is the true signal by comparing the sample r with a threshold T, whose value depends on the specific decision criterion used:

- Maximum Likelihood
- Minimum error probability
- etc.

For one decision, there can be **four outcomes**:

- **correct rejection**: signal is 0, detection is 0
- **false alarm**: signal is 0, detection is 1
- **miss**: signal is 1, detection is 0
- **hit** (correct detection): signal is 1, detection is 1

The Receiver Operating Characteristic (ROC) curve is the plot of the probability P(hit) against the probability P(false alarm), for all possible values of T.

Matlab snippets and code

Generating random binary data

In Matlab, we can generate a vector randomly filled with 0's and 1's in the following way:

- We use the function `rand()` to generate a vector with random floating point numbers between 0 and 1
- We compare the vector with some constant. The comparison result will be 0's and 1's, which are placed randomly.
- If we compare with 0.5, we get an equal amount of 0's and 1's (equal probability).
- In general, if we compare with some threshold $p \in [0, 1]$ we get 0's and 1's with probability p and $1 - p$, respectively.

In [ ]:
```
% Generate a random vector with 25\% 0's and 70\% 1's
v = rand(1,10000) > 0.25;
```

Counting values of 1 from a binary vector

If we have a vector containing only 0's and 1's, we can count the number of 1's by simply summing the vector.

In [ ]:
```
% Count how many 1's are in the vector v generated above
count = sum(v);
```

Exercises

1. Simulate threshold-based detection with a single sample, as follows:

   - Generate a vector of 100000 values 0 or A, with equal probability (consider $A = 5$)
   - Add over it a random noise with normal distribution $\mathcal{N}(0, \sigma^2 = 1)$
   - Pick a value of T between 0 and A, and compare each element with T to decide which sample is logical 0 or logical 1 (A)
   - Compare the decision result with the true original vector, and count how many correct detections and how may false alarms have been
   - Estimate P(hit) and P(false alarm) by dividing the above numbers to the size of the vector

1. Wrap the above code into a function `[phit, pfa] = myThreshDet(T)` that returns the two probabilities for a given `T`. Draw the ROC by running the function for 100 values of T uniformly spaced between 0 and A, and plotting the resulting vector `phit` against `pfa`.

1. Repeat the same simulation for two samples per bit:

   - double the values of the starting vector, making two consecutive 0 or A values, e.g.
   $$[00AA00AAAA00AA\dots]$$
   - the decision now uses **the average value** of the two consecutive samples of a bit
   - plot the ROC and compare with the first one. Which is better?

Final questions

1. Suppose we make we start with a vector with much fewer 1's than 0's. Should we increase or decrease the threshold T?
2. In a practical scenario, what could be a disadvantage of using 2 samples for detection, compared to just 1?