EE526 Homework 3

Kevin Mack, Clarkson University 03/08/2018

Question 3:

In this question we consider the detection problem:

$$H_0 : \mathbf{X} = \mathbf{N},$$
$$H_1 : \mathbf{Y} = \mathbf{S} + \mathbf{N}$$

where $\mathbf{S} = [2 - 1]^T$ is a known signal vector and $\mathbf{N} \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_N)$ with

$$\mathbf{C}_N = \begin{bmatrix} \frac{7}{2} & -\frac{3}{2} \\ -\frac{3}{2} & \frac{7}{2} \end{bmatrix}$$

Decorrelation

Here we find a unitary Decorrelating Matrix $\mathbf{U}$ for $\mathbf{N}$. We decide to decorrelate the noise due to the fact that it simplifies the scalar threshold value calculation (which must be computed in every loop). The decorrelation is performed through an eigenvalue decomposition. The new problem can be stated as,

$$H_0 : \mathbf{Y} = \mathbf{M},$$
$$H_1 : \mathbf{Y} = \mathbf{R} + \mathbf{M},$$

where $\mathbf{M} = \mathbf{UN}$, and $\mathbf{R} = \mathbf{US}$, and $\mathbf{U}$ is the unitary decorrelation matrix defined by,

$$\mathbf{C}_M = \mathbf{U}\mathbf{C}_N\mathbf{U}^T.$$

With this problem, we will use the Neyman-Pearson Detector and generate an ROC curve. The results are given in Figure 1, and the code to generate the figure is given in Listing 1 below.

(e) For $P_{fa} = 0.1$, $P_d = 0.4$.

(f) The Bayesian detector with equal priors is given by $\rho = \mathbf{R}^T\mathbf{C}_M\mathbf{R}$.

Which yields $\rightarrow H_1 = \{x > 1.15\}$.

Listing 1: Matlab Code –

```
%% EE526 Homework 3 Question 3
% Do Monte-Carlo analysis and plot ROC curve for detector

```

```
S = [2 -1]‘; % original signal vector
C_N = [7/2 -3/2; -3/2 7/2]; % original Covariance matrix

% find decorrelated matrix
[U, D, W] = eig(C_N);

C_M = U*C_N*U‘;
C_M_i = inv(C_M);

% set threshold for np detector
rho_min = -10.0;
rho_max = 10.0;
rho_step = 0.1;

rho = rho_min:rho_step:rho_max;

% number of monte-carlo simulations
trials = 1e5;

%% do false positives first
fp = zeros(size(rho,2), trials);
T_fp = zeros(size(rho,2),trials);

for i = 1:size(rho,2)
    % generate new noise for every set of rho and every trial
    N = mvnrnd([0 0], C_M, trials);

    % find decorrelated matrix
    [U, D, W] = eig(C_N);

    % decorrelate
    M = U*N(1:end,:)‘;
    R = U*S;

    % restate problem without cross-terms (decorrelated noise variance)
    % in this case, no signal is present (just noise)
    X = N‘;
    Y = M;

    for j = 1:trials
        T_fp(i,j) = 1./R‘*C_M_i*Y(:,j);
        if T_fp(i,j) > rho(i)
            fp(i,j) = 1;
        end
    end
end

```

```
%% do true positives now
tp = zeros(size(rho,2), trials);
T_tp = zeros(size(rho,2),trials);
for i = 1:size(rho, 2)
    N = mvnrnd(S', C_M, trials);

    % find decorrelated matrix
    [U, D, W] = eig(C_N);

    % decorrelate
    M = U*N(1:end,:)';
    R = U*S;

    % restate problem without cross-terms (decorrelated noise variance)
    % this case has the signal plus noise
    X = S + N';
    Y = R + M;

    for j = 1:trials
        T_tp(i,j) = 1./R'*C_M_i*Y(:,j);
        if T_tp(i,j) > rho(i)
            tp(i,j) = 1;
        end
    end
end

% calculate true and false positive rates
fp_rate = fp*ones(trials, 1)./trials;
tp_rate = tp*ones(trials, 1)./trials;

markersize = 8;
figure
hold all
grid on
grid minor
plot(fp_rate, tp_rate)
plot(0.1, 0.425, 'r*', 'markersize', markersize, 'MarkerFaceColor', 'r')
plot(0.1927, 0.5908, 'go', 'markersize', markersize, 'MarkerFaceColor', 'g')
xlabel('False Positive Rate')
ylabel('True Positive Rate')
title('ROC Curve')
legend('ROC Curve', 'P_{fa} = 0.1' ,'Bayesian Detector')
```

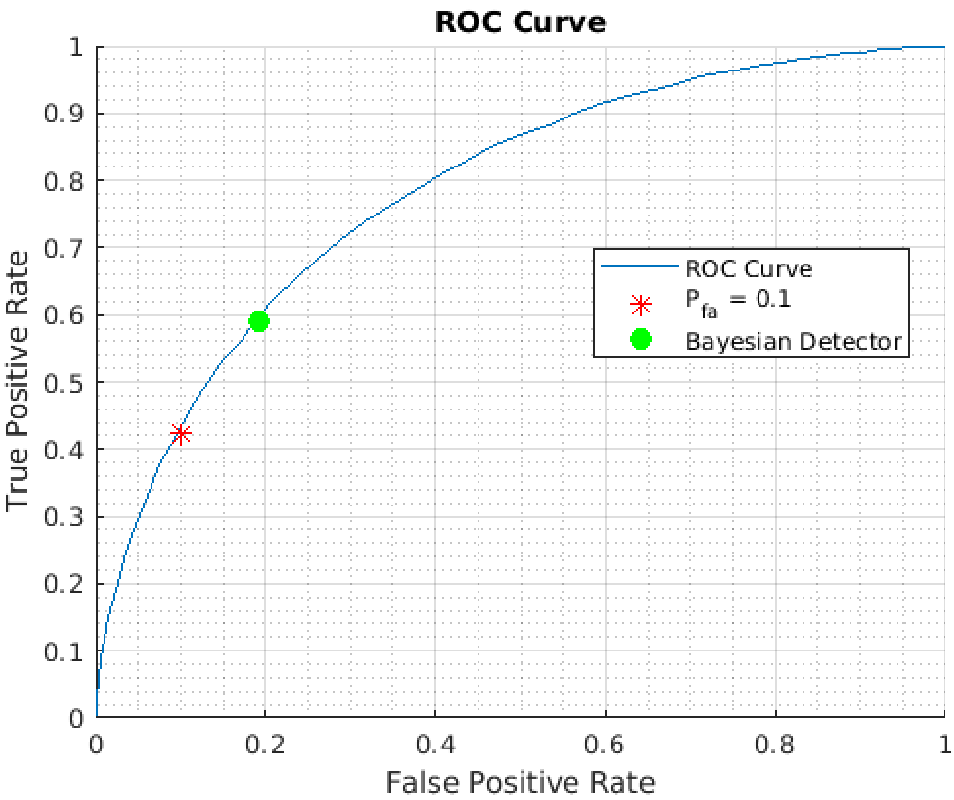

Figure 1: This graph depicts the ROC curve for the detector given in Question 3. The points marked on the graph are the threshold point for the Bayesian Decision Rule assuming equal priors, and the red asterisk marks the point where $P_{fa} = 0.1$. The curve is the result of a Monte-Carlo simulation with $1e^6$ simulations.