ECE 581 Homework 7

Shengxin Qian

1. Problem 5-1

Given that we have a known signal image S and an unknown image X to be determined. According to the algorithm in Hmwk2, we could transform two matrixes in to a single variable z.

$$z = \frac{1}{\sigma_n^2} \sum_{i=1}^{K} \sum_{j=1}^{K} x_{i,j}\, s_{i,j}$$

$s_{i,j}$ is a single element of S, $x_{i,j}$ is a single element of X

z is our observation variable and according to the result in Hmwk2,

$$f(z|H_0) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{z^2}{2\sigma^2}}, z \in (-\infty, +\infty)$$

$$f(z|H_1) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(z-\mu)^2}{2\sigma^2}}, z \in (-\infty, +\infty)$$

$$\sigma^2 = \mu = \sum_{i=1}^{K} \sum_{j=1}^{K} \frac{s_{i,j}^2}{\sigma_n^2}$$

So, the likelihood ratio supposed to be $\lambda(z) = \frac{f(z|H_1)}{f(z|H_0)}$, the Ln likelihood ratio supposed to be:

$$\ln(\lambda) = z - 0.5 * \mu.$$

According to the transformation of z, we could get the pdf of $\ln(\lambda)$:

$$f(\ln(\lambda)|H_0) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(z+0.5\mu)^2}{2\sigma^2}}, z \in (-\infty, +\infty)$$

$$f(\ln(\lambda)|H_1) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(z-0.5\mu)^2}{2\sigma^2}}, z \in (-\infty, +\infty)$$

According to the equations above, we could calculate the theoretical ROC.

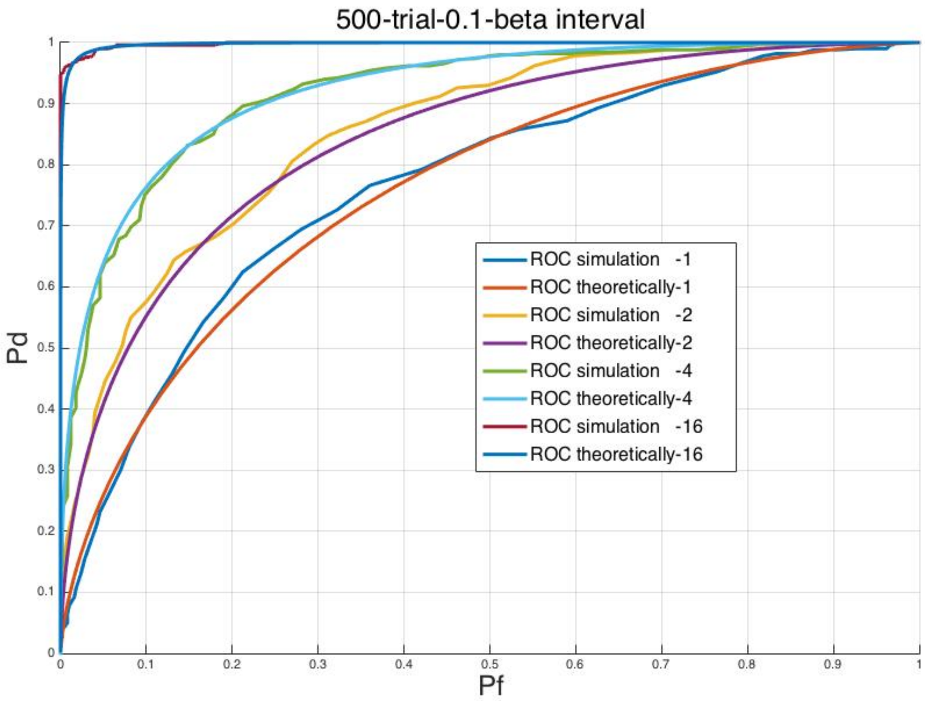
2. Problem 5-2

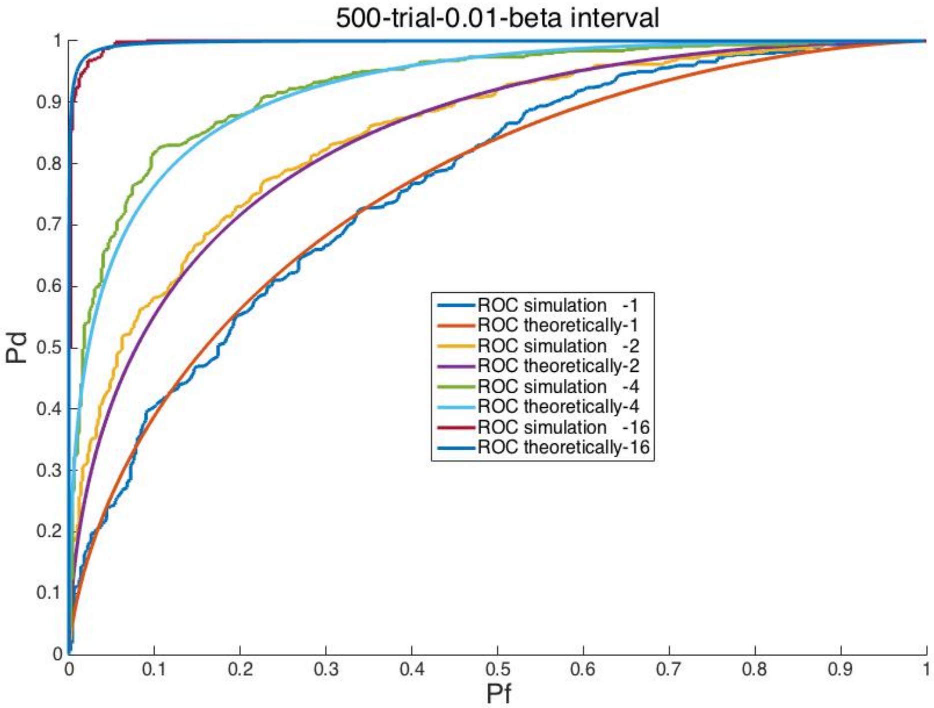
The brute-force computer simulation in this problem is to generate 500 test images with only noise (H_0) and 500 test images with noise and signal (H_1) and calculate the observation value z of each image. Because the noise and signal is statistically independent. More than that, noise is according to N(0,1) distribution. So, the value $z|H_1$, $z|H_0$ is also according to the pdf above. 500 trials could sketchily describe the distribution of value z under each hypothesis with the number of values of variable z in each interval. With these values of z, we could get numbers of values $\ln(\lambda)$ which are also according to the pdf above.

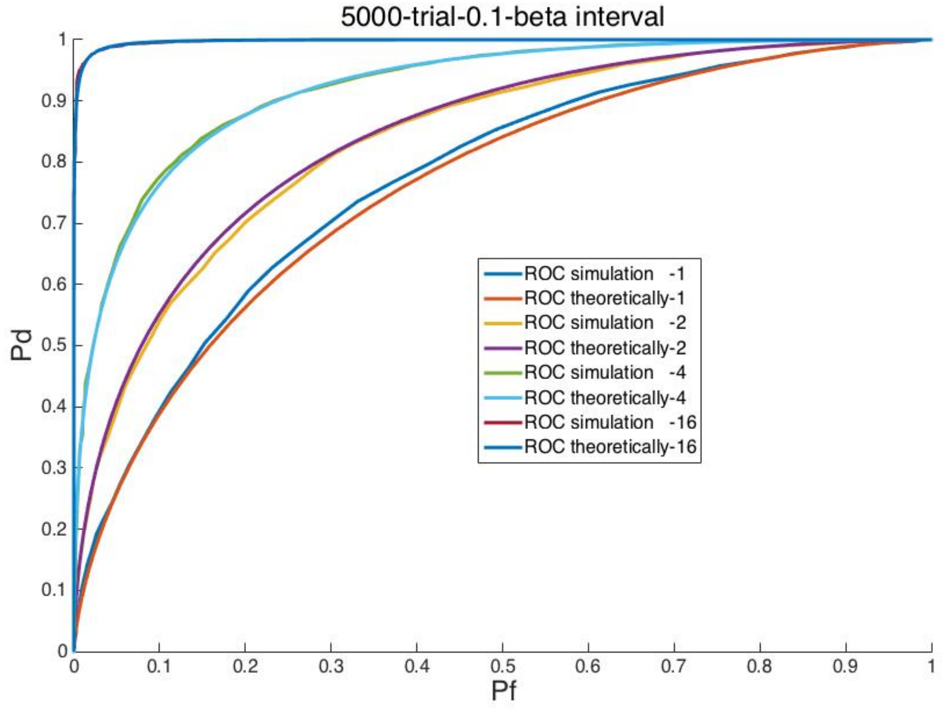
So, for brute-force computer simulation, P_d=the number of values of $\ln(\lambda)\,|H_1 > \beta$, β is threshold. P_f=the number of values of $\ln(\lambda)|H_0 > \beta$, β is threshold. At last, connect different pairs (P_d, P_f) with different β, which is so called ROC curve.

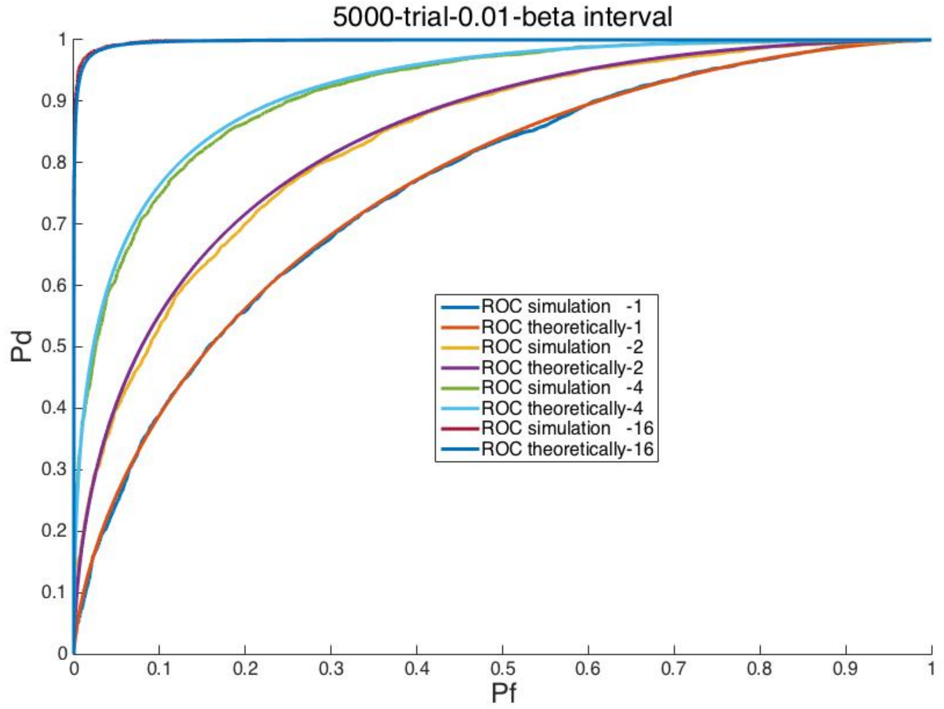
For theoretical result, ROC curve is also generated by different pairs of (P_d, P_f)

$$P_d = \int_{\beta}^{+\infty} f(\ln(\lambda)|H_1) \ , \ P_f = \int_{\beta}^{+\infty} f(\ln(\lambda)|H_0)$$









As we could see in four pictures above, the number before 'trial' means the number of trials, the number before 'beta interval' means the interval of different β.

● Why is ROC closer to the upper-left corner with higher $\frac{E_s}{\sigma_n^2}$

Of course we could understand $\frac{E_s}{\sigma_n^2}$ is a kind of way to describe SNR. In the picture, we could see the ROC curve under different SNR of input images. **The higher the SNR is, the closer to the upper-left corner the ROC curve is.**

As we all know SNR is a way to describe the sensitivity of two different hypothesizes. The higher the SNR is, the higher the sensitivity is, the higher the P_d is for each specific P_f.

● Why isn't ROC simulation curve equal to ROC theoretically

Because the higher the number of trials is, the closer to the odds the frequency of $\ln(\lambda)$ in some region is. When the number of trials is infinite, the frequency is equal to odds. That is also why the ROC simulation curve is closer to ROC theoretically comparing with 500 trials and 5000 trials.

There is another factor to affect the error between ROC simulation and ROC theoretically which is the interval of β when we drew the ROC curve. The interval of β is resolution of our graph. The smaller the β is, the higher the resolution is. That is why the ROC simulation curve is closer to ROC theoretically comparing with 0.1 beta interval and 0.01 beta interval.

● Why isn't ROC simulation curve smooth

Because when the number of trials is not big enough, the distribution of $\ln(\lambda)$ is not very uniformly. That is why a small β increase would change P_d or P_f a lot. Comparing with 500 trials and 5000 trials, the latter graph is much more smooth. On the other hand, the distribution of $\ln(\lambda)$ is more uniformly with a larger β interval. That is why the curve with 0.1 β interval is more smooth than 0.01 β interval.

Hmwk7_force_simulation.m

```
%================================================================
%               ECE 581 Hmwk7
%               Shengxin Qian
%================================================================
clear
set(figure,'NumberTitle','off','Name','abc');
for Es=[1,2,4,16]                             %keep variance of noise=1 and set Es=1,2,4,16
    s=signal_image_generator(1024,Es);            %generate signal image according to required Es

    sum_varianceH0=sum_variance_H0(s,1024,1);      %calculate the required variance(H0) for algorithm in Hmwk2
    sum_varianceH1=sum_variance_H1(s,1024,1);      %calculate the required variance(H1) for algorithm in Hmwk2
    sum_meanH1=sum_mean_H1(s,1024,1);             %calculate the required mean(H1) for algorithm in Hmwk2

    trial_number=5000;                         %the experiment time
    z_H0=normrnd(0,sqrt(sum_varianceH0),1,trial_number);%generate the output(H0) of algorithm in Hmwk2
    z_H1=normrnd(sum_meanH1,sqrt(sum_varianceH1),1,trial_number);%generate the output(H1) of algorithm in Hmwk2

    lnlambda_H0=z_H0-0.5*sum_meanH1;              %the lnlambda_H0 output according to the problem 5-1
    lnlambda_H1=z_H1-0.5*sum_meanH1;              %the lnlambda_H1 output according to the problem 5-1

    index=1;
    for beta=min(lnlambda_H0):0.01:max(lnlambda_H1)
        Pf(index)=sum(lnlambda_H0>=beta);         %Pf in experiment
        Pd(index)=sum(lnlambda_H1>=beta);         %Pd in experiment
        Pft(index)=1-normcdf(beta,-0.5*sum_meanH1,sqrt(sum_varianceH0));%Pf theoretically
        Pdt(index)=1-normcdf(beta,0.5*sum_meanH1,sqrt(sum_varianceH1)); %Pd theoretically
        index=index+1;
    end

    hold on
    plot(Pf/trial_number,Pd/trial_number);
    plot(Pft,Pdt);
end
```

signal_image_generator.m

```matlab
function y=signal_image_generator(n,Es)
   s=sqrt(Es/n);
   y=eye(n)*s;
```

sum_mean_H1.m

```matlab
function y=sum_mean_H1(s,n,single_variance)
   y=0;
   for i=1:n
       for j=1:n
             y=y+s(i,j)^2;
       end
   end
   y=y/single_variance;
```

sum_variance_H0.m

```matlab
function y=sum_variance_H0(s,n,single_variance)
   y=0;
   for i=1:n
       for j=1:n
             y=y+s(i,j)^2;
       end
   end
   y=y/single_variance;
```

sum_variance_H1.m

```matlab
function y=sum_variance_H1(s,n,single_variance)
   y=0;
   for i=1:n
       for j=1:n
             y=y+s(i,j)^2;
       end
   end
   y=y/single_variance;
```