6

(a)

**Result:**

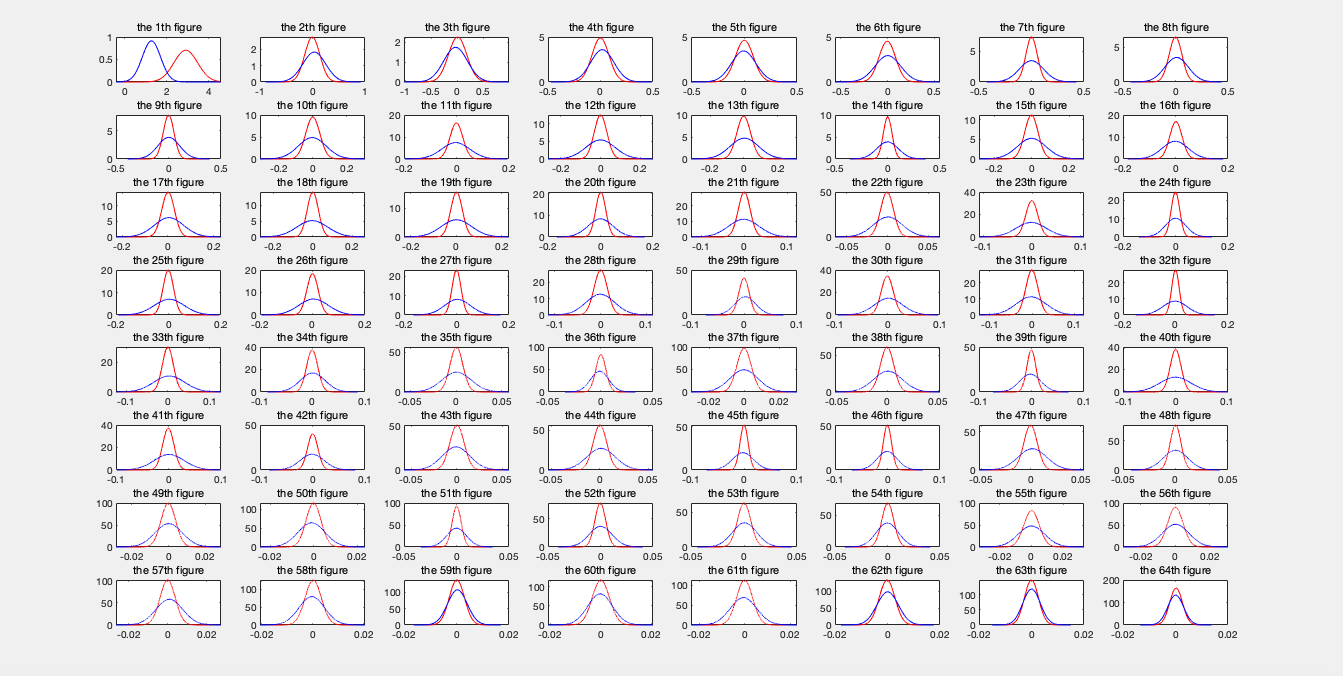
/Users/zenghailong/Desktop/ECE 271 A/homework2/homework2结果截图/屏幕快照 2018-10-30 下午8.05.07.png

The result here is the same as the result obtained from last week. Last week, I just used the number of sample for grass and cheetah, for each class’s prior probability, computing by the number of sample of related class divided by the total number of sample(the sum of every classes’ sample).

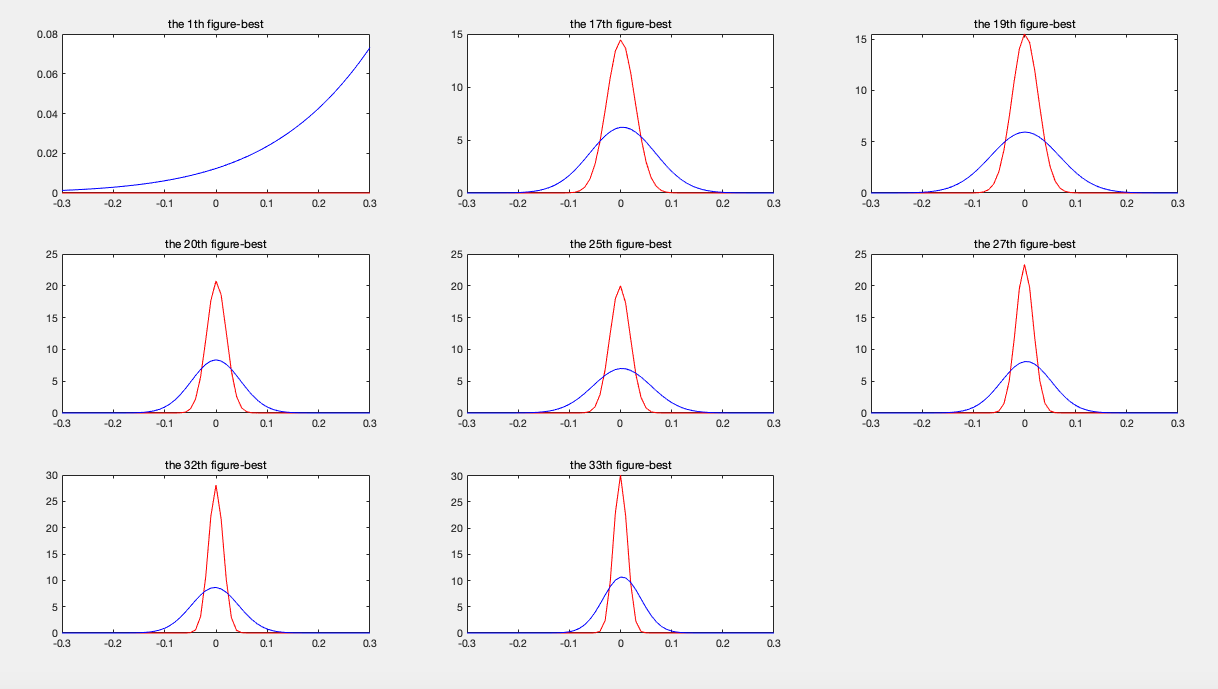
(b)

**Result:**

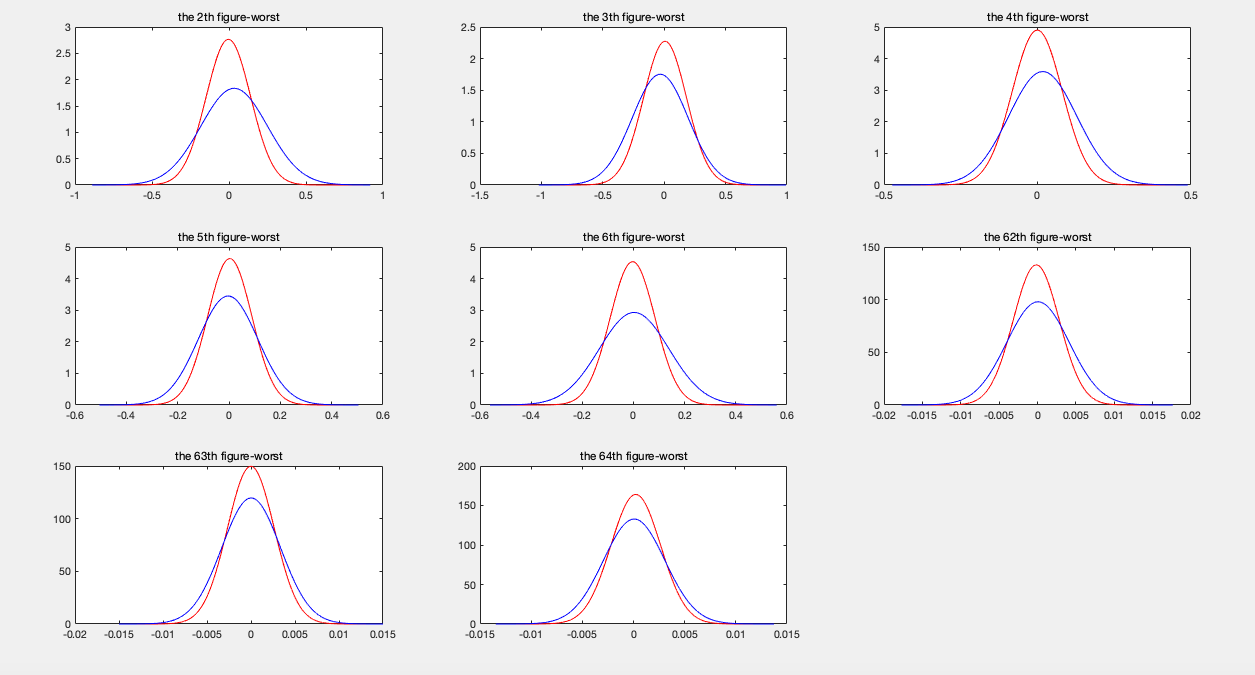
64 plots with the marginal densities:



The best 8 features:[1,17,19,20,25,27,32,33]



The worst 8 features: [2,3,4,5,6,62,63,64]

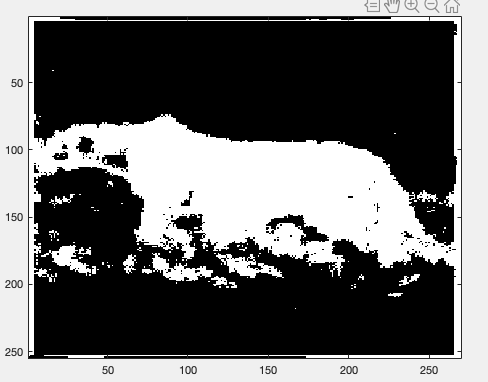


The best features should be the most distinct Gaussians distribution between two classes, in this way, it can be easily classified to be different classes with less error. And the worst features should be the most similar distinct Gaussians distribution between two classes.

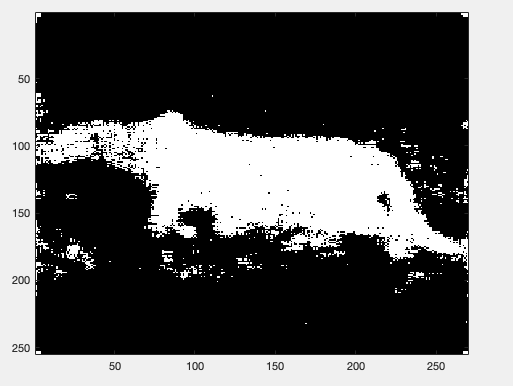
(c)

**Result:**

**The classification mask for 64-dimensional Gaussians:**



**The classification mask for 8-dimensional Gaussians:**



**P\_error:(P\_error\_64 is under 64-dimensional Gaussians, and the other is under 8-dimensional Gaussians)**

/Users/zenghailong/Desktop/ECE 271 A/homework2/homework2结果截图/屏幕快照 2018-10-30 下午8.05.17.png

Using the 8-dimensional Gaussian distribution with the best 8 features has less possibility to be misclassified since the 8 best features are more distinct between two classes. So the error with respect to the 8-dimensional Gaussian distribution should be less.

**Coding:**

%%(a)

clear;

clc;

%Load TrainingSamples\_DCT\_8\_new.mat

load('TrainingSamplesDCT\_8\_new.mat');

[m1,n1]=size(TrainsampleDCT\_BG); %m1 is the row number of DCT\_BG matrix,and n1 is the column number

[m2,n2]=size(TrainsampleDCT\_FG); %m1 is the row number of DCT\_BG matrix,and n1 is the column number

%n independent observations from Y

n=m1\*n1+m2\*n2;

C0=m1\*n1; %O represent grass

C1=m2\*n2; %1 represent cheetah

%According to homework2-2

P\_Y\_0=C0/n; %1 represent cheetah

P\_Y\_1=C1/n; %O represent grass

%print the prior probabilities

fprintf('%s=%.4f\n','P\_Y(grass)',P\_Y\_0);

fprintf('%s=%.4f\n','P\_Y(cheetah)',P\_Y\_1);

%%(b)

X\_0=TrainsampleDCT\_BG'; %Denote TrainsampleDCT\_BG' as X\_0

X\_1=TrainsampleDCT\_FG'; %Denote TrainsampleDCT\_FG' as X\_1

mu\_0=sum(X\_0,2)/size(X\_0,2); %Obtain the mu for X\_0

mu\_1=sum(X\_1,2)/size(X\_1,2); %Obtain the mu for X\_1

%To obtain the sigma for X\_0

for j=1:size(X\_0,2)

s\_0(:,j)=X\_0(:,j)-mu\_0;

end

sigma\_0=(s\_0\*s\_0')/size(X\_0,2); %Covariance matrix for X\_0(grass)

%Obtain the sigma for X\_1

for j=1:size(X\_1,2)

s\_1(:,j)=X\_1(:,j)-mu\_1;

end

sigma\_1=(s\_1\*s\_1')/size(X\_1,2); %Covariance matrix for X\_1(cheetah)

for i=1:64

xmin(i)=(mu\_0(i)+mu\_1(i)-5\*sigma\_0(i,i)^0.5-5\*sigma\_1(i,i)^0.5)/2;

xmax(i)=(mu\_0(i)+mu\_1(i)+5\*sigma\_0(i,i)^0.5+5\*sigma\_1(i,i)^0.5)/2;

x\_step(i)=0.1\*(sigma\_0(i,i)+sigma\_1(i,i))/2;

end

for i=1:64

x=xmin(i):x\_step(i):xmax(i);

y\_0=1/(2\*pi\*sigma\_0(i,i)).^0.5\*exp(-(x-mu\_0(i)).^2/(2\*sigma\_0(i,i)));

y\_1=1/(2\*pi\*sigma\_1(i,i)).^0.5\*exp(-(x-mu\_1(i)).^2/(2\*sigma\_1(i,i)));

subplot(8,8,i);

set(gcf,'Position',[1600 1600 1600 1600]);

plot(x,y\_0,'r'); %plots of the marginal densities of grass

hold on;

plot(x,y\_1,'b'); %plots of the marginal densities of cheetah

title(sprintf('the %dth figure',i));

end

k=[1,17,19,20,25,27,32,33]; %Choose the best 8 features

figure;

for i=1:8

x=xmin(k(i)):x\_step(k(i)):xmax(k(i));

y\_0=1/(2\*pi\*sigma\_0(k(i),k(i))).^0.5\*exp(-(x-mu\_0(k(i))).^2/(2\*sigma\_0(k(i),k(i))));

y\_1=1/(2\*pi\*sigma\_1(k(i),k(i))).^0.5\*exp(-(x-mu\_1(k(i))).^2/(2\*sigma\_1(k(i),k(i))));

subplot(3,3,i);

set(gcf,'Position',[1600 1600 1600 1600]);

plot(x,y\_0,'r'); %plots of the marginal densities of grass

hold on;

plot(x,y\_1,'b'); %plots of the marginal densities of cheetah

title(sprintf('the %dth figure-best',k(i)));

end

W=[2,3,4,5,6,62,63,64]; %Choose the worst 8 features

figure;

for i=1:8

x=xmin(W(i)):x\_step(W(i)):xmax(W(i));

y\_0=1/(2\*pi\*sigma\_0(W(i),W(i))).^0.5\*exp(-(x-mu\_0(W(i))).^2/(2\*sigma\_0(W(i),W(i))));

y\_1=1/(2\*pi\*sigma\_1(W(i),W(i))).^0.5\*exp(-(x-mu\_1(W(i))).^2/(2\*sigma\_1(W(i),W(i))));

subplot(3,3,i);

set(gcf,'Position',[1600 1600 1600 1600]);

plot(x,y\_0,'r'); %plots of the marginal densities of grass

hold on;

plot(x,y\_1,'b'); %plots of the marginal densities of cheetah

title(sprintf('the %dth figure-worst',W(i)));

end

%%(c)

%Read the picture---cheetah.bmp

I=imread('cheetah.bmp');

I=padarray(I,[4 4],0,'both'); %Padding array with 8 row 0s and 8 columns 0s

I=im2double(I); %Represent the image as doubles in [0,1]

Zigzag=importdata('Zig-zag Pattern.txt')+1; %Import the zig-zag Pattern.txt

%Using sliding window to built 255X270's (8X8) cell C

for i=1:(size(I,1)-8)

for j=1:(size(I,2)-8)

Cheetah=I(i:i+7,j:j+7); %Obtain block as sliding windows

Cheetah\_DCT=dct2(Cheetah); %DCT

for b=1:64 %make the blocks zig-zag vector

[m,n]=find(Zigzag==b);

TX(b)=Cheetah\_DCT(m,n);

end

T=TX'; %Transpose and obtain 64\*1 column vector

%64-dimensional Gaussians

P\_X\_grass=1/(((2\*pi)^64\*det(sigma\_0))^0.5)\*exp(-0.5\*(T-mu\_0)'\*inv(sigma\_0)\*(T-mu\_0));

P\_X\_cheetah=1/(((2\*pi)^64\*det(sigma\_1))^0.5)\*exp(-0.5\*(T-mu\_1)'\*inv(sigma\_1)\*(T-mu\_1));

if P\_X\_grass\*P\_Y\_0>=P\_X\_cheetah\*P\_Y\_1 %BRD MAP

A(i,j)=0; %Determine this pixel as grass

else

A(i,j)=1; %Determine this pixel as cheetah

end

%8-dimensional Gaussians

%k=[1,2,3,4,17,18,19,25];

P\_X\_grass\_8=1/((2\*pi)^8\*det(sigma\_0(k,k)))^0.5\*exp(-0.5\*(T(k)-mu\_0(k))'\*inv(sigma\_0(k,k))\*(T(k)-mu\_0(k)));

P\_X\_cheetah\_8=1/((2\*pi)^8\*det(sigma\_1(k,k)))^0.5\*exp(-0.5\*(T(k)-mu\_1(k))'\*inv(sigma\_1(k,k))\*(T(k)-mu\_1(k)));

if P\_X\_grass\_8\*P\_Y\_0>=P\_X\_cheetah\_8\*P\_Y\_1 %BRD MAP

B(i,j)=0; %Determine this pixel as grass

else

B(i,j)=1; %Determine this pixel as cheetah

end

end

end

figure;

imagesc(A);

colormap(gray(255));

figure;

imagesc(B);

colormap(gray(255));

%Compute the error related to 64-dimensional Gaussians and 8-dimensional Gaussians

%Read the cheetah\_mask.bmp

I1=imread('cheetah\_mask.bmp');

I1=im2double(I1); %Represent the image as doubles in [0,1]

%0 in Array A and I1 represents the state grass

%1 in Array A and I1 represents the state cheetah

count1=0; %The number of 0 in array A when the element with the same index in I1 is 0

count2=0; %The number of 0 in array A when the element with the same index in I1 is 1

count3=0; %The number of 1 in array A when the element with the same index in I1 is 1

count4=0; %The number of 1 in array A when the element with the same index in I1 is 0

count1\_8=0; %The number of 0 in array B when the element with the same index in I1 is 0

count2\_8=0; %The number of 0 in array B when the element with the same index in I1 is 1

count3\_8=0; %The number of 1 in array B when the element with the same index in I1 is 1

count4\_8=0; %The number of 1 in array B when the element with the same index in I1 is 0

for i=1:size(A,1)

for j=1:size(A,2)

if I1(i,j)==0

if A(i,j)==I1(i,j)

count1=count1+1;

else

count2=count2+1;

end

if B(i,j)==I1(i,j)

count1\_8=count1\_8+1;

else

count2\_8=count2\_8+1;

end

else

if A(i,j)==I1(i,j)

count3=count3+1;

else

count4=count4+1;

end

if B(i,j)==I1(i,j)

count3\_8=count3\_8+1;

else

count4\_8=count4\_8+1;

end

end

end

end

count=count1+count2+count3+count4;

count\_8=count1\_8+count2\_8+count3\_8+count4\_8;

%P\_error=P\_X|Y(i~=cheetah|cheetah)\*P\_Y(cheetah)+P\_X|Y(i~=grass|grass)\*P\_Y(grass)

%P\_X|Y(i~=grass|grass)\*P\_Y(grass)=count2/(count1+count2)\*(count1+count2)/count=count2/count

%P\_X|Y(i~=cheetah|cheetah)\*P\_Y(cheetah)=count4/(count3+count4)\*(count3+count4)/count=count4/count

fprintf('%s=%.4f\n','P\_error\_64',count2/count+count4/count);

fprintf('%s=%.4f\n','P\_error\_8',count2\_8/count\_8+count4\_8/count\_8);