Kalvin Goode

a. we calculate the prior probability of two features, cheetah (foreground) and grass (background). The calculation and results are the same as last time.

$$P_Y(cheetah) = \frac{row\ size\ of\ cheetah}{total\ row\ size} \approx 0.19$$

$$P_Y(grass) = 1 - P_Y(cheetah) = \frac{row\ size\ of\ grass}{total\ row\ size} \approx 0.81$$

From problem 2, we know that the maximum likelihood estimate of the parameters of a multinomial distribution is the average of the observations,

$$P_Y(i) = \pi_i = \frac{c_i}{n}, i = 1, \ldots, N$$

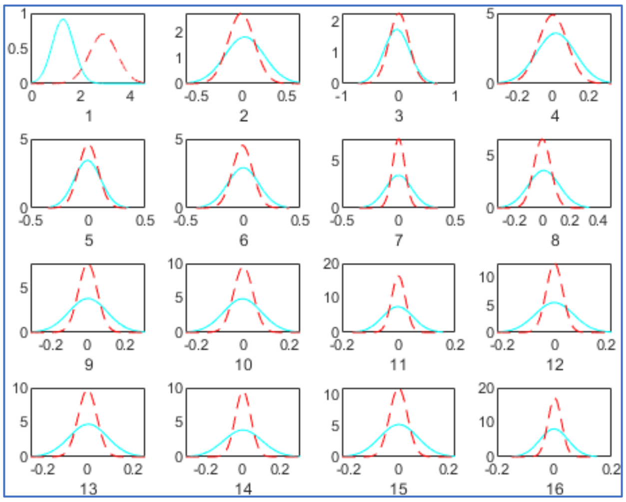
, where c_i is the number of sample has feature *i* and n is the total number of sample from feature 1 to *N*.

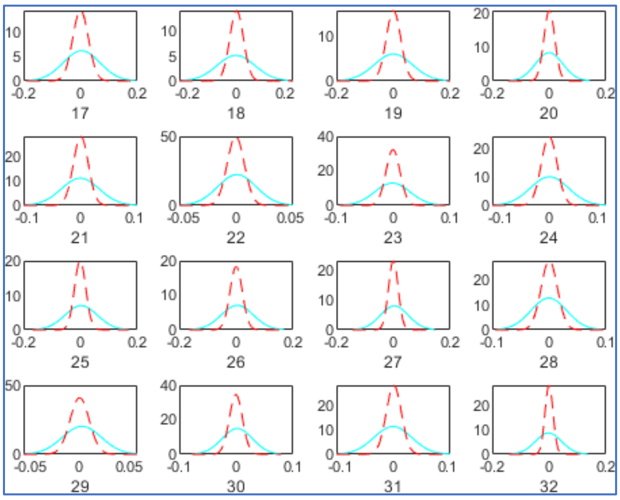
This formula is the same as the calculation obtained last time shows that $P_Y(cheetah)$ and $P_Y(grass)$ are the best estimate according to maximum likelihood estimation.

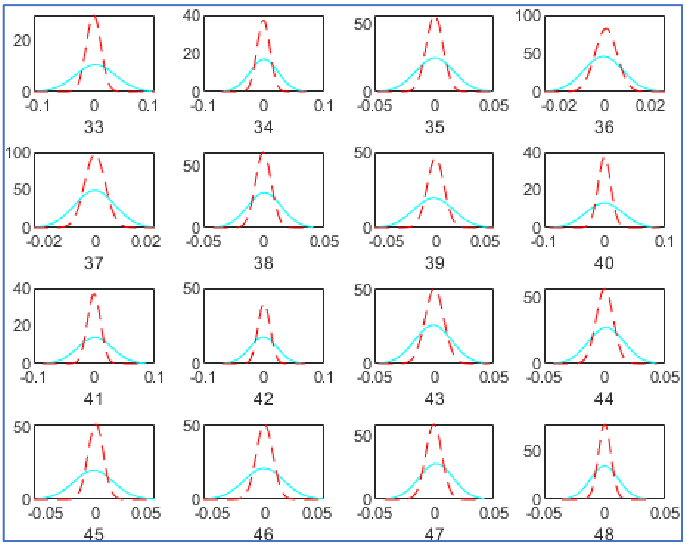
b. To compute maximum likelihood estimates of Gaussian, μ and Σ, we use the formula, $\mu^i = \frac{1}{n}\sum_j x_j^i$ and $\Sigma^{ik} = \frac{1}{n}\sum_{i,k}(x^i - \mu^i)(x^k - \mu^k)^T$

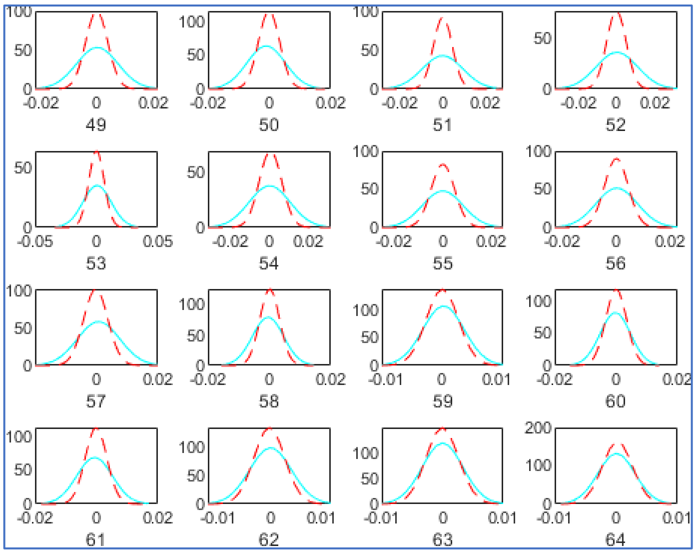
, where x_j^i is the value of *j*th observation of *i*th feature, n is the number of total observations, *T* is the transpose of matrix, μ^i and Σ^{ik} are sample mean and sample covariance, respectively.

After calculating these estimates, we can find the distribution of for each feature *i,* below are probability density function for each feature. Blue solid line is $P_{X|Y}(x|cheetah)$ and red dash line is $P_{X|Y}(x|grass)$

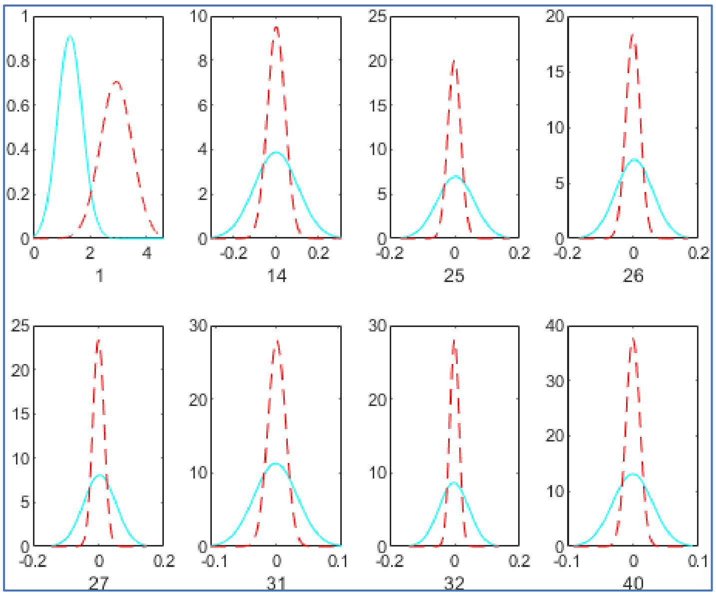




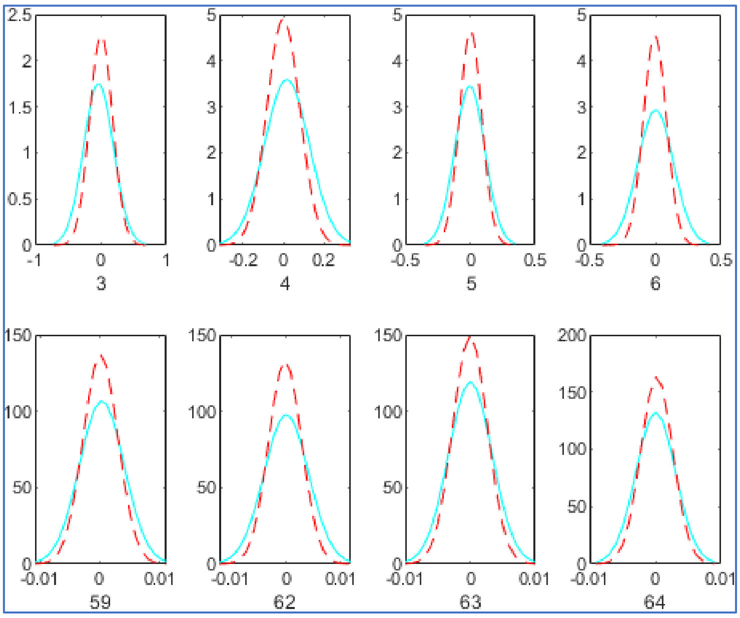




Below are the best 8 plot I chose, which I believe those have the smallest overlapping area between two distribution.



Below are the worst 8 plot I chose, which I believe those have the largest overlapping area between two distribution.

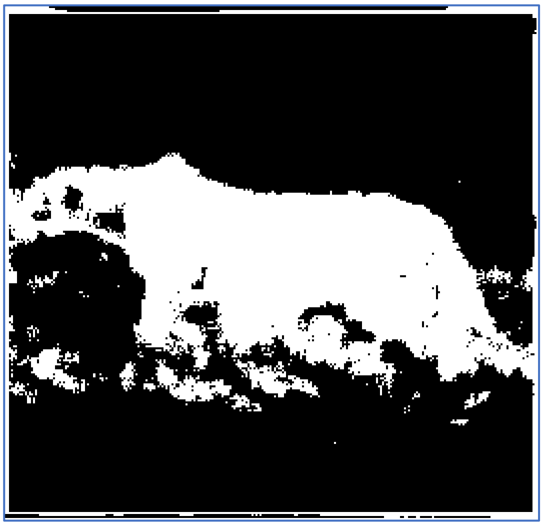


C. From lecture, we know the MAP rule has formula,

$$g^*(x) = \arg\max_i P_{Y|X}(i \mid x) = \arg\max_i P_{X|Y}(x \mid i) P_Y(i)$$

, where g*(x) is the maximum a-posteriori probability rule.

Using all 64-dimension Gaussians distribution to classify cheetah and grass, we obtain the result below,

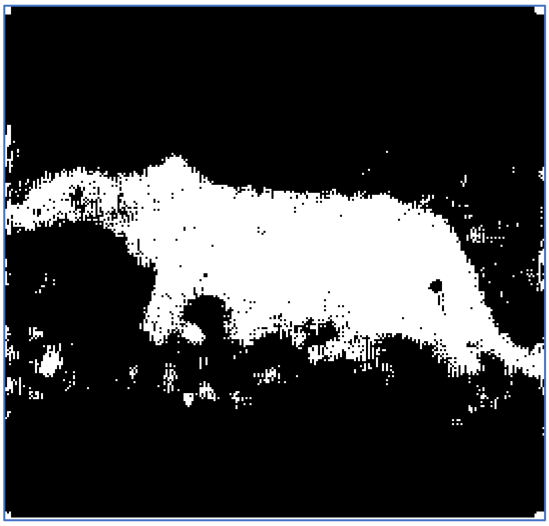


$$Error = \sum_{i=0}^{1} P_Y(i) \frac{amount\ \ of\ \ incorrect\ \ pixel\ \ of\ \ i}{size\ \ of\ \ i\ \ of\ \ truth} = 0.122$$

With best 8 features for classification I chose,

X= $\{X^1, X^{14}, X^{25}, X^{26}, X^{27}, X^{31}, X^{32}, X^{40}\}$

Using all 8-dimension Gaussians distribution to classify cheetah and grass, we obtain the result below,



$$Error = \sum_{i=0}^{1} P_Y(i) \frac{amount\ \ of\ \ incorrect\ \ pixel\ \ of\ \ i}{size\ \ of\ \ i\ \ of\ \ truth} = 0.049$$

We see that even though 64-dim classification has better result than the method (DCT coefficient with 2nd greatest energy) we used from last report, best 8-dim classification has outperformed these two classifications. I believe the reason is 8 features have much more distinction to separate two class and has less ambiguity when *P(cheetah|x)* and *P(grass|x)* are very close. It is also better in terms of computation complexity; when we have more training samples, it is clear that 64-dim classification has more compution time than 8-dim classification. Therefore, this exercise indicates that more features we have does not necessarily correlates with accuracy in testing

Appendix

```
%this training set is the new version of training set
load('TrainingSamplesDCT_8.mat');

%A
pri_f=size(TrainsampleDCT_FG,1)...
    /(size(TrainsampleDCT_BG,1)+size(TrainsampleDCT_FG,1));
pri_b=1-pri_f;

%B
u_f=take_mean(TrainsampleDCT_FG);
u_b=take_mean(TrainsampleDCT_BG);
sig2_f=take_cov(TrainsampleDCT_FG);
sig2_b=take_cov(TrainsampleDCT_BG);

for j=0:16:48
    figure();
    for i=j+1:j+16
        subplot(4,4,i-j)
        low=min(u_f(i)-3*sqrt(sig2_f(i,i)),...
            u_f(i)-3*sqrt(sig2_f(i,i)));
        up=max(u_b(i)+3*sqrt(sig2_f(i,i)),...
            u_b(i)+3*sqrt(sig2_b(i,i)));
        x = [low:.001:up];
        y_f= take_normpdf(x,u_f(i),sqrt(sig2_f(i,i)));
        %x= [-5:.1:5];
        y_b= take_normpdf(x,u_b(i),sqrt(sig2_b(i,i)));
        plot(x,y_f,'-c',x,y_b,'--r')
        xlabel(i)
    end
end
figure();
```

```
best=[1,14,25,26,27,31,32,40];
worst=[3,4,5,6,59,62,63,64];

%take best and worst of 8
j=1;
for i=best
    subplot(2,4,j)
    j=j+1;
    low=min(u_f(i)-3*sqrt(sig2_f(i,i)),...
        u_f(i)-3*sqrt(sig2_f(i,i)));
    up=max(u_b(i)+3*sqrt(sig2_f(i,i)),...
        u_b(i)+3*sqrt(sig2_b(i,i)));
    x = [low:.001:up];
    y_f= take_normpdf(x,u_f(i),sqrt(sig2_f(i,i)));
    %x= [-5:.1:5];
    y_b= take_normpdf(x,u_b(i),sqrt(sig2_b(i,i)));
    plot(x,y_f,'-c',x,y_b,'--r')
    xlabel(i)
end
figure();

j=1;
for i=worst
    subplot(2,4,j)
    j=j+1;
    low=min(u_f(i)-3*sqrt(sig2_f(i,i)),...
        u_f(i)-3*sqrt(sig2_f(i,i)));
    up=max(u_b(i)+3*sqrt(sig2_f(i,i)),...
        u_b(i)+3*sqrt(sig2_b(i,i)));
    x = [low:.001:up];
    y_f= take_normpdf(x,u_f(i),sqrt(sig2_f(i,i)));
    %x= [-5:.1:5];
    y_b= take_normpdf(x,u_b(i),sqrt(sig2_b(i,i)));
    plot(x,y_f,'-c',x,y_b,'--r')
    xlabel(i)
end

zig=load('Zig-Zag Pattern.txt')+1;
cheetah= im2double(imread('cheetah.bmp'));
cheetah_p=padarray(cheetah,[4,3],0,'pre');
```

```
cheetah_p=padarray(cheetah_p,[3,4],0,'post');

n=1;
for i=1:size(cheetah_p,1)-7
    for j=1:size(cheetah_p,2)-7
        temp=dct2(cheetah_p(i:i+7, j:j+7));
        for k=1:8
            for m=1:8
                cheetah_dct(zig(k,m),n)=temp(k,m);
            end
        end
        n=n+1;
    end
end

%MAP with all
like_b=take_mvnpdf(cheetah_dct(:,:)',u_b,sig2_b);
like_f=take_mvnpdf(cheetah_dct(:,:)',u_f,sig2_f);
n=1;
final_a=zeros(size(cheetah,1),size(cheetah,2));
for i=1:size(cheetah,1)
    for j=1:size(cheetah,2)
        if(like_b(n)*pri_b>=like_f(n)*pri_f)
            final_a(i,j)=0;
        else
            final_a(i,j)=1;
        end
        n=n+1;
    end
end
figure();
imshow(final_a)

%MAP with 8 features
like_b=take_mvnpdf(cheetah_dct(best,:)',u_b(best),sig2_b(best,best));
like_f=take_mvnpdf(cheetah_dct(best,:)',u_f(best),sig2_f(best,best));
n=1;
final_b=zeros(size(cheetah,1),size(cheetah,2));
for i=1:size(cheetah,1)
    for j=1:size(cheetah,2)
        if(like_b(n)*pri_b>=like_f(n)*pri_f)
```

```
            final_b(i,j)=0;
        else
            final_b(i,j)=1;
        end
        n=n+1;
    end
end
figure();
imshow(final_b)

%C, calcuate Bayes Error (Risk)
truth=imread('cheetah_mask.bmp');
truth=im2double (truth);
err1=0;
err2=0;
for i=1:size(truth,1)
    for j=1:size(truth,2)
        if (final_a(i,j)~= truth(i,j))
            err1=err1+1;
        end
        if (final_b(i,j)~= truth(i,j))
            err2=err2+1;
        end
    end
end
%err_rate=err/(size(truth,1)*size(truth,2));
disp("E1: " +err1/(size(truth,1)*size(truth,2)))
disp("E2: " +err2/(size(truth,1)*size(truth,2)))

%functions
function u=take_mean(sample)
    u=zeros(1,size(sample,2));
    total=0;
    for i=1:size(sample,2)
        for j=1:size(sample,1)
            total=total+sample(j,i);
        end
        u(1,i)=total/size(sample,1);
        total=0;
    end
    return
```

```
end

function sig=take_cov(sample)
	sig=zeros(size(sample,2));
	for i=1:size(sample,2)
		for j=1:i
			temp=0;
			u_i=take_mean(sample(:,i));
			u_j=take_mean(sample(:,j));
			for k=1:size(sample,1)
				temp=temp+(sample(k,i)-u_i)*(sample(k,j)-u_j);
			end
			sig(i,j)=temp/size(sample,1);
			if(i~=j)
				sig(j,i)=sig(i,j);
			end
		end
	end
end

function y=take_normpdf(x,mu,sig)
	y=zeros(1,size(x,2));
	for i=1:size(x,2)
		y(1,i)=exp(-(x(i)-mu)^2/(2*sig^2))/sqrt(2*pi*sig^2);
	end
end

function y=take_mvnpdf(x,mu,sig)
	y=zeros(size(x,1),1);
	dim=size(x,2);
	de=det(sig);
	invsig=inv(sig);
	for i=1:size(x,1)
		y(i,1)=exp(-1/2*((x(i,:)-mu)*invsig*(x(i,:)-mu)'))/...
			sqrt((2*pi)^dim*de);
	end
end
```