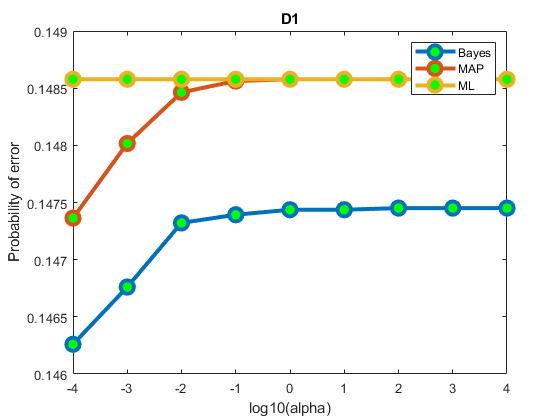
Name: Adrian Mai

SID: A13224086

ECE 271A

Programing assignment 3 report

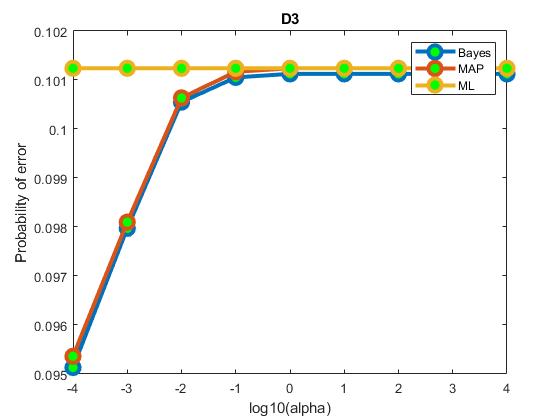
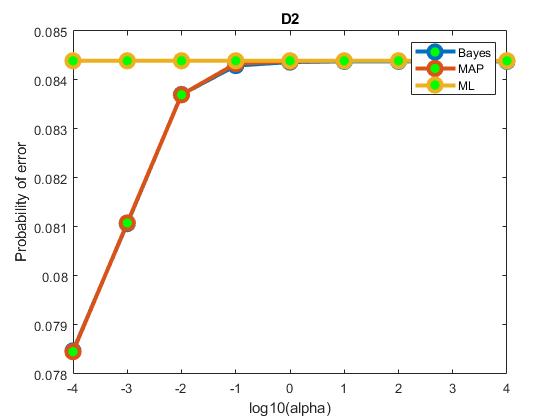
Part a-c:

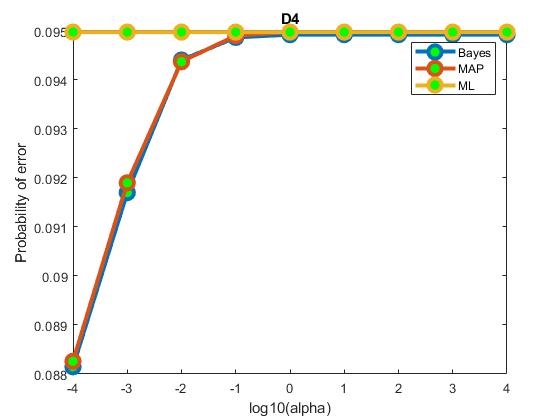


When alpha is small the prior seems to make a big different and therefore the best estimator is Bayesian then follow by MAP and lastly ML. However, when as alpha get bigger the Probability of error seems to get flat out, indicate that the probability of errors now depend more on data (data dominates).

As predicted Bayesian Estimator is the best!

Part d:

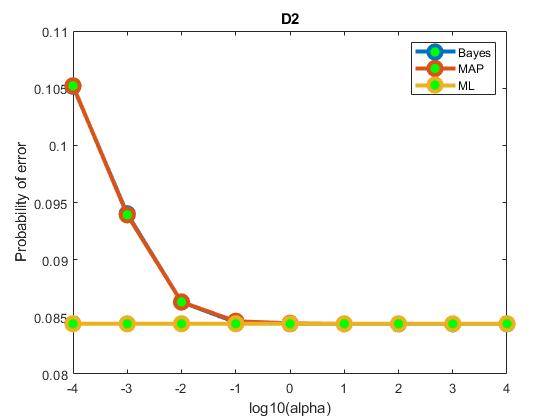
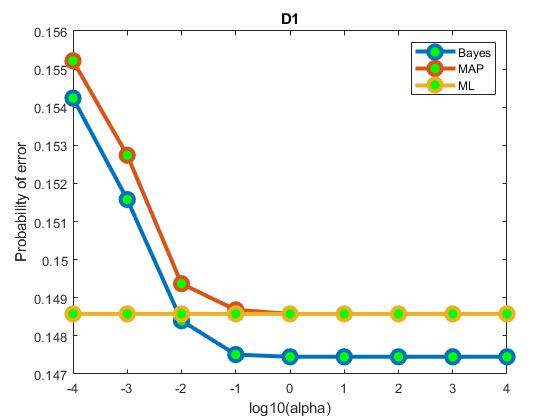


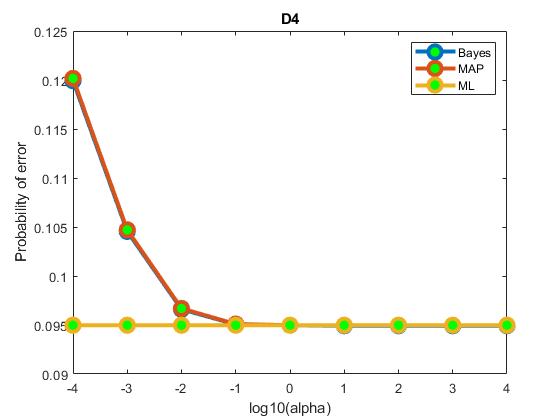
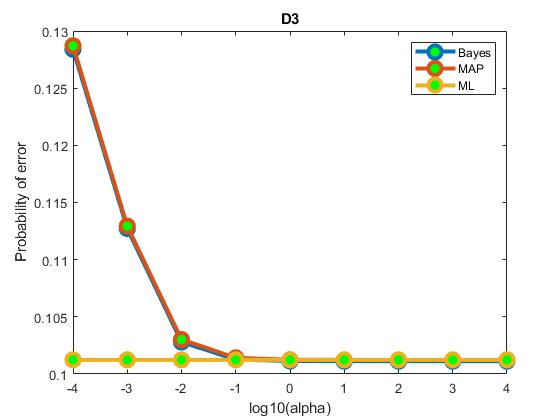


In those 3 data sets, the probability of errors tend to follow similar shape. The MAP and Bayesian Estimator are really close now, maybe because the scaling of the y axis is now bigger.

Similar behavior as Dataset 1, when alpha is small prior seem to dominate and when alpha get bigger prior get less power. There are not significant sign on how the prior dominate from smaller to bigger data set, it seems like the estimator for the variance play a big factor in this.

Part e:





Once again, the prior seems to dominate when alpha is small.

Strangely with this strategy, ML seems to perform the best.

The reason is because we’ve chosen a pretty bad priors(same for both classes). In the consequences, when the prior dominate, Bayesian Estimator will make a prediction of equally likely between both class and therefor lead to a higher error rate than ML and so for in the case of MAP. The error seem to flat out an convert to a point when priors dominate

Code:

S = load('TrainingSamplesDCT\_subsets\_8.mat');

Prior\_1 = load('Prior\_1.mat');

Prior\_2 = load('Prior\_2.mat');

alpha\_m = load('Alpha.mat');

alpha = alpha\_m.alpha;

%Extract data

% D1\_BG = S.D1\_BG;

% D2\_BG = S.D2\_BG;

% D3\_BG = S.D3\_BG;

% D4\_BG = S.D4\_BG;

%

% D1\_CT = S.D1\_FG;

% D2\_CT = S.D2\_FG;

% D3\_CT = S.D3\_FG;

% D4\_CT = S.D4\_FG;

data\_cell = cell(2,4);

data\_cell{1,1} = S.D1\_BG;

data\_cell{1,2} = S.D2\_BG;

data\_cell{1,3} = S.D3\_BG;

data\_cell{1,4} = S.D4\_BG;

data\_cell{2,1} = S.D1\_FG;

data\_cell{2,2} = S.D2\_FG;

data\_cell{2,3} = S.D3\_FG;

data\_cell{2,4} = S.D4\_FG;

Prior\_1\_W0 = Prior\_1.W0;

Prior\_1\_m0\_BG = Prior\_1.mu0\_BG;

Prior\_1\_m0\_CT = Prior\_1.mu0\_FG;

Prior\_2\_W0 = Prior\_2.W0;

Prior\_2\_m0\_BG = Prior\_2.mu0\_BG;

Prior\_2\_m0\_CT = Prior\_2.mu0\_FG;

img = imread('cheetah.bmp');

img = im2double(img);

%Vectorize 8-8 block with zig-zag patter

% Padding to the image

B = padarray(img, [7 7], 'symmetric','pre');

m\_Cheetah = zeros(255, 270);

m\_Background = zeros(255, 270);

% Compute ans sliding window and decision matrix for 64 features

B\_masking\_m\_Cheetah = zeros(255, 270);

B\_masking\_m\_Background = zeros( 255, 270);

ML\_masking\_m\_Cheetah = zeros( 255, 270);

ML\_masking\_m\_Background = zeros( 255, 270);

MAP\_masking\_m\_Cheetah = zeros( 255, 270);

MAP\_masking\_m\_Background = zeros( 255, 270);

B\_e = zeros(9,1);

MAP\_e = zeros(9,1);

ML\_e = zeros(9,1);

masked\_cheetah = imread('cheetah\_mask.bmp');

masked\_cheetah = im2double(masked\_cheetah);

v = find(masked\_cheetah);

v1 = find(~masked\_cheetah);

for d = 1:4

cov\_BG = cov(data\_cell{1,d});

cov\_CT = cov(data\_cell{2,d});

me\_BG = mean(data\_cell{1,d},1);

me\_CT = mean(data\_cell{2,d},1);

Prior\_Cheetah = size(data\_cell{2,d}, 1) / (size(data\_cell{1,d},1) + size(data\_cell{2,d}, 1));

Prior\_Background = size(data\_cell{1,d}, 1) / (size(data\_cell{1,d},1) + size(data\_cell{2,d}, 1));

for al = 1:9

sigma0 = diag(alpha(al) .\* Prior\_1\_W0);

mun\_1\_BG = sigma0 \* ((sigma0 + (1/(size(data\_cell{1,d}, 1))) \* cov\_BG) \ me\_BG') + (1/size(data\_cell{1,d}, 1)) \* cov\_BG \* ((sigma0 + (1/size(data\_cell{1,d}, 1)) \* cov\_BG) \ Prior\_1\_m0\_BG');

mun\_1\_CT = sigma0 \* ((sigma0 + (1/(size(data\_cell{2,d},1))) \* cov\_CT) \ me\_CT') + (1/size(data\_cell{2,d},1)) \* cov\_CT \* ((sigma0 + (1/(size(data\_cell{2,d},1))) \* cov\_CT) \ Prior\_1\_m0\_CT');

cov\_n\_1\_BG = sigma0 \* ((sigma0 + (1/(size(data\_cell{1,d}, 1))) \* cov\_BG) \ ((1/(size(data\_cell{1,d}, 1))) \* cov\_BG));

cov\_n\_1\_CT = sigma0 \* ((sigma0 + (1/(size(data\_cell{2,d}, 1))) \* cov\_CT) \ ((1/(size(data\_cell{2,d}, 1))) \* cov\_CT));

B\_cov\_BG = cov\_n\_1\_BG + cov\_BG;

B\_cov\_CT = cov\_n\_1\_CT + cov\_CT;

for i = 1:size(img, 1)

for j = 1:size(img, 2)

temp\_v = compute\_dct\_vector(B(i:i+7, j:j+7));

temp\_v = temp\_v';

B\_masking\_m\_Cheetah(i, j) = (temp\_v - mun\_1\_CT)' \* ((B\_cov\_CT) \ (temp\_v - mun\_1\_CT)) + (log((2 \* pi)^64 \* det(B\_cov\_CT)) + Prior\_Cheetah);

B\_masking\_m\_Background(i, j) = (temp\_v - mun\_1\_BG)' \* ((B\_cov\_BG) \ (temp\_v - mun\_1\_BG)) + (log((2 \* pi)^64 \* det(B\_cov\_BG)) + Prior\_Background);

MAP\_masking\_m\_Cheetah(i, j) = (temp\_v - mun\_1\_CT)' \* ((cov\_CT) \ (temp\_v - mun\_1\_CT)) + (log((2 \* pi)^64 \* det(cov\_CT)) + Prior\_Cheetah);

MAP\_masking\_m\_Background(i, j) = (temp\_v - mun\_1\_BG)' \* ((cov\_BG) \ (temp\_v - mun\_1\_BG)) + (log((2 \* pi)^64 \* det(cov\_BG)) + Prior\_Background);

ML\_masking\_m\_Cheetah(i, j) = (temp\_v - me\_CT')' \* ((cov\_CT) \ (temp\_v - me\_CT')) + (log((2 \* pi)^64 \* det(cov\_CT)) + Prior\_Cheetah);

ML\_masking\_m\_Background(i, j) = (temp\_v - me\_BG')' \* ((cov\_BG) \ (temp\_v - me\_BG')) + (log((2 \* pi)^64 \* det(cov\_BG)) + Prior\_Background);

end

end

n\_img = ~( B\_masking\_m\_Background <= B\_masking\_m\_Cheetah );

%Calcualte error for 64 features

flat\_v\_64 = n\_img(:);

%Calculating error

P\_Cheetah\_g\_Cheetah = nnz(flat\_v\_64(v) == 1) / numel(flat\_v\_64(v));

P\_Cheetah\_g\_Backgound = nnz(flat\_v\_64(v1) == 1) / numel(flat\_v\_64(v1));

B\_e(al,1) = P\_Cheetah\_g\_Backgound \* Prior\_Background + (1 - P\_Cheetah\_g\_Cheetah) \* Prior\_Cheetah;

m\_img = ~( MAP\_masking\_m\_Background <= MAP\_masking\_m\_Cheetah );

%Calcualte error for 64 features

flat\_v\_64 = m\_img(:);

%Calculating error

P\_Cheetah\_g\_Cheetah = nnz(flat\_v\_64(v) == 1) / numel(flat\_v\_64(v));

P\_Cheetah\_g\_Backgound = nnz(flat\_v\_64(v1) == 1) / numel(flat\_v\_64(v1));

MAP\_e(al,1) = P\_Cheetah\_g\_Backgound \* Prior\_Background + (1 - P\_Cheetah\_g\_Cheetah) \* Prior\_Cheetah;

d\_img = ~( ML\_masking\_m\_Background <= ML\_masking\_m\_Cheetah);

%Calcualte error for 64 features

flat\_v\_64 = d\_img(:);

%Calculating error

P\_Cheetah\_g\_Cheetah = nnz(flat\_v\_64(v) == 1) / numel(flat\_v\_64(v));

P\_Cheetah\_g\_Backgound = nnz(flat\_v\_64(v1) == 1) / numel(flat\_v\_64(v1));

ML\_e(al,1) = P\_Cheetah\_g\_Backgound \* Prior\_Background + (1 - P\_Cheetah\_g\_Cheetah) \* Prior\_Cheetah;

end

figure;

plot(log10(alpha), B\_e','o-','linewidth',3,'markersize',10,'markerfacecolor','g')

hold on;

plot(log10(alpha), MAP\_e','o-','linewidth',3,'markersize',10,'markerfacecolor','g')

plot(log10(alpha), ML\_e','o-','linewidth',3,'markersize',10,'markerfacecolor','g')

legend('Bayes','MAP','ML');

xlabel('log10(alpha)') ;

ylabel('Probability of error') ;

title(['D',num2str(d)]);

hold off;

end