**Dataset 1**

A close up of a map

Description automatically generatedA close up of a map

Description automatically generated

*Figure 1: Dataset 1/strategy 1 percent Figure 2: Dataset 1/strategy 2 percent*

*error vs log(alpha) for predictive equation error vs log(alpha) for predictive equation*

A close up of text on a white background

Description automatically generatedA close up of a map

Description automatically generated

*Figure 3: Dataset 1/strategy 1 percent Figure 4: Dataset 1/strategy 2 percent error vs log(alpha) for MAP approach error vs log(alpha) for MAP approach*

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

*Figure 5: Dataset 1/strategy 1 percent Figure 6: Dataset 1/strategy 2 percent error vs log(alpha) for ML approach error vs log(alpha) for ML approach*

**Dataset 2**

A close up of a map

Description automatically generatedA close up of a map

Description automatically generated

*Figure 7: Dataset 2/strategy 1 percent Figure 8: Dataset 2/strategy 2 percent*

*error vs log(alpha) for predictive equation error vs log(alpha) for predictive equation*

A close up of a map

Description automatically generatedA close up of a map

Description automatically generated

*Figure 9: Dataset 2/strategy 1 percent Figure 10: Dataset 2/strategy 2 percent error vs log(alpha) for MAP approach error vs log(alpha) for MAP approach*

A screenshot of a social media post

Description automatically generatedA screenshot of a social media post

Description automatically generated

*Figure 11: Dataset 2/strategy 1 percent Figure 12: Dataset 2/strategy 2 percent error vs log(alpha) for ML approach error vs log(alpha) for ML approach*

**Dataset 3**

**A close up of a map

Description automatically generated**A close up of a map

Description automatically generated

*Figure 13: Dataset 3/strategy 1 percent Figure 14: Dataset 3/strategy 2 percent*

*error vs log(alpha) for predictive equation error vs log(alpha) for predictive equation*

A close up of a map

Description automatically generatedA screenshot of a cell phone

Description automatically generated

*Figure 15: Dataset 3/strategy 1 percent Figure 16: Dataset 3/strategy 2 percent error vs log(alpha) for MAP approach error vs log(alpha) for MAP approach*

A screenshot of a social media post

Description automatically generatedA screenshot of a social media post

Description automatically generated

*Figure 17: Dataset 3/strategy 1 percent Figure 18: Dataset 3/strategy 2 percent*

*error vs log(alpha) for ML approach error vs log(alpha) for ML approach*

**Dataset 4**

**A close up of a map

Description automatically generatedA close up of a map

Description automatically generated**

*Figure 19: Dataset 4/strategy 1 percent Figure 20: Dataset 4/strategy 2 percent*

*error vs log(alpha) for predictive equation error vs log(alpha) for predictive equation*

A close up of a map

Description automatically generatedA close up of a map

Description automatically generated

*Figure 21: Dataset 4/strategy 1 percent Figure 22: Dataset 4/strategy 2 percent error vs log(alpha) for MAP approach error vs log(alpha) for MAP approach*

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

*Figure 23: Dataset 4/strategy 1 percent Figure 24: Dataset 4/strategy 2 percent*

*error vs log(alpha) for ML approach error vs log(alpha) for ML approach*

**Analysis**

All the curves for the predictive equation and MAP approaches are asymptotic, and the “curves” for the ML approach are linear (as the function for ML is not dependent on alpha). For strategy 1, as alpha increases the error percentage increases toward an upper limit. For strategy 2, as alpha increases the error percentage decreases toward a lower limit. These upper and lower limits, however, are equivalent (i.e. strategy 1 increases toward the limit, whereas strategy 2 decreases toward that same limit value). This limit value, spanning all datasets (with one exception) and strategy types is the error percentage calculated by the ML approach. The one exception to this trend is observed in dataset 1. In dataset 1 (for both strategies) the MAP approach tends toward an asymptote of approximately 15.6, whereas the predictive equation approach tends toward an asymptote of approximately 15.5. The ML approach has an error percentage of approximately 15.5. From this observation, it seems as if the MAP approach will result in a larger percentage of error relative to the predictive equation approach when the dataset is small**\***. As the dataset size increases (dataset 2, 3, and 4 are all larger than dataset 1), however, both the MAP and predictive equation approaches approach the same asymptote value – this asymptote value is the value of the ML percentage of error.

**\***Small in this case is 300 DCT coefficient arrays for the background and 75 DCT coefficient arrays for the foreground.

**Code**

%Joseph Bell

%ECE271A HW3 and 4

clc;

clear;

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

load('Alpha.mat');

%%%%% Starting off with Data Set 1 and strategy 1%%%%%

% Strategy 1 = ?0 is smaller for the (darker) cheetah class (?0 = 1)

% and larger for the (lighter) grass class (?0 = 3)

%%%%% Part a %%%%%

%load('Prior\_1.mat'); %priors for strategy 1 - consists of weights, mu0\_FG,

%and mu0\_BG

load('Prior\_2.mat'); %priors for strategy 2 - consists of weights, mu0\_FG,

%and mu0\_BG

% Reading in cheetah image and mask

cheetah\_img = imread('cheetah.bmp');

cheetah\_img = im2double(cheetah\_img); %converting to double values since training data is of type double

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

cheetah\_mask = im2double(cheetah\_mask);

[cheetah\_rows, cheetah\_cols] = size(cheetah\_img);

cheetah\_img = cheetah\_img(1:8\*floor(cheetah\_rows/8),1:8\*floor(cheetah\_cols/8)); %modifying image so it can be split into 8x8 blocks

[cheetah\_rows, cheetah\_cols] = size(cheetah\_img); %overwriting for modified dimensions

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%% STARTING LOOP %%%%%%%%%% Loops 4 times due to 4 data sets

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for d=1:4

alpha\_values = []; %array of alpha values used for plotting

error\_percentages = []; %array of error percentages per alpha used for plotting

dataset\_FG = []; %stores selected foreground data set

dataset\_BG = []; %stores selected background data set

%%%% IF/ELSEIF for selecting data set to operate on

if d == 1

dataset\_FG = D1\_FG;

dataset\_BG = D1\_BG;

elseif d == 2

dataset\_FG = D2\_FG;

dataset\_BG = D2\_BG;

elseif d == 3

dataset\_FG = D3\_FG;

dataset\_BG = D3\_BG;

elseif d == 4

dataset\_FG = D4\_FG;

dataset\_BG = D4\_BG;

end

%Taking rows and cols used for calculations

[data\_rows\_FG, data\_cols\_FG] = size(dataset\_FG);

[data\_rows\_BG, data\_cols\_BG] = size(dataset\_BG);

%zig zag scanning of data to order coefficents from 1-64

%1 row per DC coefficient (First row should have largest values)

cheetah\_zigzag = zeros(64, data\_rows\_FG);

grass\_zigzag = zeros(64, data\_rows\_BG);

%Credit to Alexey Sokolov from https://www.mathworks.com/matlabcentral/fileexchange/15317-zigzag-scan

%for the zig zag code

%Reading into zig zag and transposing to create n\*64 matrix

for row=1:data\_rows\_FG

cheetah\_zigzag(:,row) = zigzag(dataset\_FG(row,:));

end

for row=1:data\_rows\_BG

grass\_zigzag(:,row) = zigzag(dataset\_BG(row,:));

end

means\_variance\_cheetah = zeros(64, 1, 2); %(:,:,1) = mean

means\_variance\_grass = zeros(64, 1, 2); %(:,:,2) = variance

% Calculating mean and variance of each coefficient

% CODE COPIED FROM MY ASSIGNMENT 2

%cheetah loop

for row=1:data\_cols\_FG

N = data\_rows\_FG;

sample\_mean = 0;

sample\_variance = 0;

sample = cheetah\_zigzag(row,:); %grab each coefficient row

for i=1:N

sample\_mean = sample\_mean + sample(1,i);

end

sample\_mean = sample\_mean/N;

for i=1:N

sample\_variance = sample\_variance + (sample(1,i) - sample\_mean)^2;

end

sample\_variance = sample\_variance/N;

means\_variance\_cheetah(row,1,1) = sample\_mean;

means\_variance\_cheetah(row,1,2) = sample\_variance;

end

%grass loop

for row=1:data\_cols\_BG

N = data\_rows\_BG;

sample\_mean = 0;

sample\_variance = 0;

sample = grass\_zigzag(row,:); %grab each coefficient row

for i=1:N

sample\_mean = sample\_mean + sample(1,i);

end

sample\_mean = sample\_mean/N;

for i=1:N

sample\_variance = sample\_variance + (sample(1,i) - sample\_mean)^2;

end

sample\_variance = sample\_variance/N;

means\_variance\_grass(row,1,1) = sample\_mean;

means\_variance\_grass(row,1,2) = sample\_variance;

end

% Calculating covariance matrices for foreground (cheetah)

% and background (grass)

% CODE COPIED FROM MY ASSIGNMENT 2

cheetah\_covariances = zeros(64,64);

grass\_covariances = zeros(64,64);

%calculating covariance matrix for cheetah

for i=1:64

p1 = cheetah\_zigzag(i,:); %grab coefficient row

mu\_1 = means\_variance\_cheetah(i,1,1);

for j=1:64

p2 = cheetah\_zigzag(j,:); %grab coefficient row

mu\_2 = means\_variance\_cheetah(j,1,1);

temp = 0;

for k=1:data\_rows\_FG

temp = temp + (p1(1,k) - mu\_1)\*(p2(1,k) - mu\_2);

end

cheetah\_covariances(i,j) = temp/data\_rows\_FG;

end

end

%calculating covariance matrix for grass

for i=1:64

p1 = grass\_zigzag(i,:); %grab coefficient row

mu\_1 = means\_variance\_grass(i,1,1);

for j=1:64

p2 = grass\_zigzag(j,:); %grab coefficient row

mu\_2 = means\_variance\_grass(j,1,1);

temp = 0;

for k=1:data\_rows\_BG

temp = temp + (p1(1,k) - mu\_1)\*(p2(1,k) - mu\_2);

end

grass\_covariances(i,j) = temp/data\_rows\_BG;

end

end

%Used for probability calculations

mu\_n\_hat\_cheetah1 = zeros(1,64);

mu\_n\_hat\_grass1 = zeros(1,64);

mu\_n\_cheetah1 = zeros(1,64);

mu\_n\_grass1 = zeros(1,64);

%Used for MAP part

prior\_cheetah = data\_rows\_FG/(data\_rows\_FG + data\_rows\_BG);

prior\_grass = data\_rows\_BG/(data\_rows\_BG+data\_rows\_FG);

%%%%% Calculating mu\_n\_hat values %%%%

for i=1:64

for j=1:data\_rows\_FG

mu\_n\_hat\_cheetah1(1,i) = mu\_n\_hat\_cheetah1(1,i) + dataset\_FG(j,i);

end

mu\_n\_hat\_cheetah1(1,i) = mu\_n\_hat\_cheetah1(1,i)/data\_rows\_FG;

end

for i=1:64

for j=1:data\_rows\_BG

mu\_n\_hat\_grass1(1,i) = mu\_n\_hat\_grass1(1,i) + dataset\_BG(j,i);

end

mu\_n\_hat\_grass1(1,i) = mu\_n\_hat\_grass1(1,i)/data\_rows\_BG;

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%% STARTING LOOP FOR ALPHA %%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for a=1:9

new\_image = zeros(cheetah\_rows, cheetah\_cols); %Returned image

%Values used for calculations%

sigma\_1 = alpha(1,a)\*diag(W0);

mu\_n\_cheetah1 = sigma\_1 \* inv(sigma\_1 + 1/data\_rows\_FG\*cheetah\_covariances)\*transpose(mu\_n\_hat\_cheetah1) + 1/data\_rows\_FG\*cheetah\_covariances\*inv(sigma\_1+1/data\_rows\_FG\*cheetah\_covariances)\*transpose(mu0\_FG);

mu\_n\_grass1 = sigma\_1 \* inv(sigma\_1 + 1/data\_rows\_BG\*grass\_covariances)\*transpose(mu\_n\_hat\_grass1) + 1/data\_rows\_BG\*grass\_covariances\*inv(sigma\_1+1/data\_rows\_BG\*grass\_covariances)\*transpose(mu0\_BG);

sigma\_n\_cheetah = sigma\_1\*inv(sigma\_1 + 1/data\_rows\_FG\*cheetah\_covariances)\*1/data\_rows\_FG\*cheetah\_covariances;

sigma\_n\_grass = sigma\_1\*inv(sigma\_1 + 1/data\_rows\_BG\*grass\_covariances)\*1/data\_rows\_BG\*grass\_covariances;

sigma\_cheetah\_total = cheetah\_covariances + sigma\_n\_cheetah;

sigma\_grass\_total = grass\_covariances + sigma\_n\_grass;

%Multivariate Gaussian Distribution PDF functions for part a

fun\_cheetah = @(x) 1/sqrt((det(sigma\_cheetah\_total)\*(2\*pi)^64))\*exp(-1/2\*transpose(x-mu\_n\_cheetah1)\*inv(sigma\_cheetah\_total)\*(x-mu\_n\_cheetah1));

fun\_grass= @(x) 1/sqrt((det(sigma\_grass\_total)\*(2\*pi)^64))\*exp(-1/2\*transpose(x-mu\_n\_grass1)\*inv(sigma\_grass\_total)\*(x-mu\_n\_grass1));

%%%%%%%%%% Classifying %%%%%%%%%%%%

for i=1:cheetah\_cols-7 %shift scan pointer over a column

for j=1:cheetah\_rows-7

block = cheetah\_img(j:7+j,i:7+i); %grab 8x8 block

block\_dct = dct2(block);

zzblock\_dct = transpose(zigzag(block\_dct));

P\_x\_D\_cheetah = fun\_cheetah(zzblock\_dct);

P\_x\_D\_grass = fun\_grass(zzblock\_dct);

if P\_x\_D\_cheetah \* prior\_cheetah > P\_x\_D\_grass \* prior\_grass

new\_image(j:j,i:i) = 1;

end

end

end

f = figure() % Returning image per alpha

imagesc(new\_image);

colormap(gray(255));

title(['Dataset: ', num2str(d),' - Alpha: ', num2str(alpha(1,a))])

saveas(f,[pwd,'/results/D\_',num2str(d),'\_A\_',num2str(a),'.png']);

%%%%% Calculating percent error for part a Alpha values %%%%%

counter\_correct = 0;

total\_pixels = cheetah\_rows\*cheetah\_cols;

for i=1:cheetah\_rows

for j=1:cheetah\_cols

if cheetah\_mask(i,j) == new\_image(i,j)

counter\_correct = counter\_correct + 1;

end

end

end

percent\_correct = counter\_correct/total\_pixels\*100;

error\_percentage = 100 - percent\_correct;

alpha\_values = [alpha\_values log(alpha(1,a))];

error\_percentages = [error\_percentages error\_percentage];

end% END OF a=1:9 Still in d data set loop

f = figure() %plotting percent error vs alpha

plot(alpha\_values, error\_percentages);

ylim([floor(min(error\_percentages))-0.1 ceil(max(error\_percentages))+0.1])

title(['Dataset: ', num2str(d), ' - Error Vs Log Alpha']);

saveas(f,[pwd,'/results/error\_plot\_D\_',num2str(d),'.png']);

%%%%%%%%%%%%%%%%%%%%%%% Assignment 2 Method %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Starting loop of alphas for assignment 2 approach

%calculating MLE covariance matrix for cheetah

cheetah\_covariances\_MLE = zeros(64,64);

grass\_covariances\_MLE = zeros(64,64);

for i=1:64

p1 = cheetah\_zigzag(i,:); %grab coefficient row

mu\_1 = means\_variance\_cheetah(i,1,1);

for j=1:64

p2 = cheetah\_zigzag(j,:); %grab coefficient row

mu\_2 = means\_variance\_cheetah(j,1,1);

temp = 0;

for k=1:data\_rows\_FG

temp = temp + (p1(1,k) - mu\_1)\*(p2(1,k) - mu\_2);

end

cheetah\_covariances\_MLE(i,j) = temp/(data\_rows\_FG-1);

end

end

%calculating covariance matrix for grass

for i=1:64

p1 = grass\_zigzag(i,:); %grab coefficient row

mu\_1 = means\_variance\_grass(i,1,1);

for j=1:64

p2 = grass\_zigzag(j,:); %grab coefficient row

mu\_2 = means\_variance\_grass(j,1,1);

temp = 0;

for k=1:data\_rows\_BG

temp = temp + (p1(1,k) - mu\_1)\*(p2(1,k) - mu\_2);

end

grass\_covariances\_MLE(i,j) = temp/(data\_rows\_BG-1);

end

end

alpha\_values\_Asn2 = []; %array of alpha values used for plotting

error\_percentages\_Asn2 = []; %array of error percentages per alpha used for plotting

for a=1:9

new\_image2 = zeros(cheetah\_rows, cheetah\_cols);

fun\_cheetah\_Asn2 = @(x) 1/sqrt((det(cheetah\_covariances\_MLE)\*(2\*pi)^64))\*exp(-1/2\*transpose(x-means\_variance\_cheetah(:,:,1))\*inv(cheetah\_covariances\_MLE)\*(x-means\_variance\_cheetah(:,:,1)));

fun\_grass\_Asn2= @(x) 1/sqrt((det(grass\_covariances\_MLE)\*(2\*pi)^64))\*exp(-1/2\*transpose(x-means\_variance\_grass(:,:,1))\*inv(grass\_covariances\_MLE)\*(x-means\_variance\_grass(:,:,1)));

for i=1:cheetah\_cols-7 %shift scan pointer over a column

for j=1:cheetah\_rows-7

block = cheetah\_img(j:7+j,i:7+i); %grab 8x8 block

block\_dct = dct2(block);

zzblock\_dct = transpose(zigzag(block\_dct));

P\_x\_y\_cheetah = fun\_cheetah\_Asn2(zzblock\_dct);

P\_x\_y\_grass = fun\_grass\_Asn2(zzblock\_dct);

if P\_x\_y\_cheetah \* prior\_cheetah > P\_x\_y\_grass \* prior\_grass

new\_image2(j:j,i:i) = 1;

end

end

end

%Calculating percent error of assignment 2 method

counter\_correct\_Asn2 = 0;

total\_pixels = cheetah\_rows\*cheetah\_cols;

for i=1:cheetah\_rows

for j=1:cheetah\_cols

if cheetah\_mask(i,j) == new\_image2(i,j)

counter\_correct\_Asn2 = counter\_correct\_Asn2 + 1;

end

end

end

percent\_correct\_Asn2 = counter\_correct\_Asn2/total\_pixels\*100;

error\_percentage\_Asn2 = 100 - percent\_correct\_Asn2;

alpha\_values\_Asn2 = [alpha\_values\_Asn2 log(alpha(1,a))];

error\_percentages\_Asn2 = [error\_percentages\_Asn2 error\_percentage\_Asn2];

end%END OF a=1:9 for Assignment 2 MLE method

f = figure() %plotting assignment 2 method (1 per data set just plotting last one they're all the same)

imagesc(new\_image2); %as I'm not multiplying covariace by alpha for this part only using MLE

colormap(gray(255)); %percent error chart should be a horizontal line

title(['Dataset: ', num2str(d), ' - Assignment 2 Method'])

saveas(f,[pwd,'/results/Asn2\_D\_',num2str(d),'.png']);

f = figure() %plotting percent error vs alpha (should only be 4 of these)

plot(alpha\_values\_Asn2, error\_percentages\_Asn2);

ylim([floor(min(error\_percentages\_Asn2))-0.1 ceil(max(error\_percentages\_Asn2))+0.1])

title(['Asn2 Approach - Dataset: ', num2str(d), ' - Error Vs Log Alpha']);

saveas(f,[pwd,'/results/error\_plot\_Asn2\_D\_',num2str(d),'.png']);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%% MAP APPROACH %%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

alpha\_values = []; %Stores alpha values for plotting

error\_percentages = []; %Stores error percentage per alpha

%Same stuff except using MAP for mu

for a=1:9

new\_image = zeros(cheetah\_rows, cheetah\_cols);

sigma\_1 = alpha(1,a)\*diag(W0);

mu\_n\_cheetah1 = sigma\_1 \* inv(sigma\_1 + 1/data\_rows\_FG\*cheetah\_covariances)\*transpose(mu\_n\_hat\_cheetah1) + 1/data\_rows\_FG\*cheetah\_covariances\*inv(sigma\_1+1/data\_rows\_FG\*cheetah\_covariances)\*transpose(mu0\_FG);

mu\_n\_grass1 = sigma\_1 \* inv(sigma\_1 + 1/data\_rows\_BG\*grass\_covariances)\*transpose(mu\_n\_hat\_grass1) + 1/data\_rows\_BG\*grass\_covariances\*inv(sigma\_1+1/data\_rows\_BG\*grass\_covariances)\*transpose(mu0\_BG);;

%sigma\_n\_cheetah = sigma\_1\*inv(sigma\_1 + 1/data\_rows\_FG\*cheetah\_covariances)\*1/data\_rows\_FG\*cheetah\_covariances;

%sigma\_n\_grass = sigma\_1\*inv(sigma\_1 + 1/data\_rows\_BG\*grass\_covariances)\*1/data\_rows\_BG\*grass\_covariances;

sigma\_n\_cheetah = 0;

sigma\_n\_grass = 0;

sigma\_cheetah\_total = cheetah\_covariances + sigma\_n\_cheetah;

sigma\_grass\_total = grass\_covariances + sigma\_n\_grass;

fun\_cheetah\_D1c = @(x) 1/sqrt((det(sigma\_cheetah\_total)\*(2\*pi)^64))\*exp(-1/2\*transpose(x-mu\_n\_cheetah1)\*inv(sigma\_cheetah\_total)\*(x-mu\_n\_cheetah1));

fun\_grass\_D1c= @(x) 1/sqrt((det(sigma\_grass\_total)\*(2\*pi)^64))\*exp(-1/2\*transpose(x-mu\_n\_grass1)\*inv(sigma\_grass\_total)\*(x-mu\_n\_grass1));

for i=1:cheetah\_cols-7 %shift scan pointer over a column

for j=1:cheetah\_rows-7

block = cheetah\_img(j:7+j,i:7+i); %grab 8x8 block

block\_dct = dct2(block);

zzblock\_dct = transpose(zigzag(block\_dct));

P\_x\_D\_cheetah = fun\_cheetah\_D1c(zzblock\_dct);

P\_x\_D\_grass = fun\_grass\_D1c(zzblock\_dct);

if P\_x\_D\_cheetah \* prior\_cheetah > P\_x\_D\_grass \* prior\_grass

new\_image(j:j,i:i) = 1;

end

end

end

f = figure()%Plotting per alpha for MAP approach

imagesc(new\_image);

colormap(gray(255));

title(['Dataset: ', num2str(d), ' - MAP Approach - Alpha: ', num2str(alpha(1,a))])

saveas(f,[pwd,'/results/MAP\_D\_',num2str(d),'\_A\_',num2str(a),'.png']);

%Calculating percent error for alpha and MAP approach

counter\_correct = 0;

total\_pixels = cheetah\_rows\*cheetah\_cols;

for i=1:cheetah\_rows

for j=1:cheetah\_cols

if cheetah\_mask(i,j) == new\_image(i,j)

counter\_correct = counter\_correct + 1;

end

end

end

percent\_correct = counter\_correct/total\_pixels\*100;

error\_percentage = 100 - percent\_correct;

alpha\_values = [alpha\_values log(alpha(1,a))];

error\_percentages = [error\_percentages error\_percentage];

end

f = figure()%Potting percent error vs alpha values for MAP

plot(alpha\_values, error\_percentages);

ylim([floor(min(error\_percentages))-0.1 ceil(max(error\_percentages))+0.1])

title(['Dataset: ', num2str(d), '- MAP Approach', ' - Error Vs Log Alpha']);

saveas(f,[pwd,'/results/error\_MAP\_D',num2str(d),'.png']);

end