Parthasarathi Kumar
A59003519

Homework 2

Problem 5

(a) Using the MLE for a multinomial distribution, the prior probabilities turn out to be -

$$P(\text{cheetah})= \frac{250}{250+1053} = 0.1909$$
$$P(\text{grass})=\frac{1053}{250+1053} =0.8081$$

This is same as what I used in homework 1, where I calculated the probability of a certain class as its frequency divided by the total number of samples.

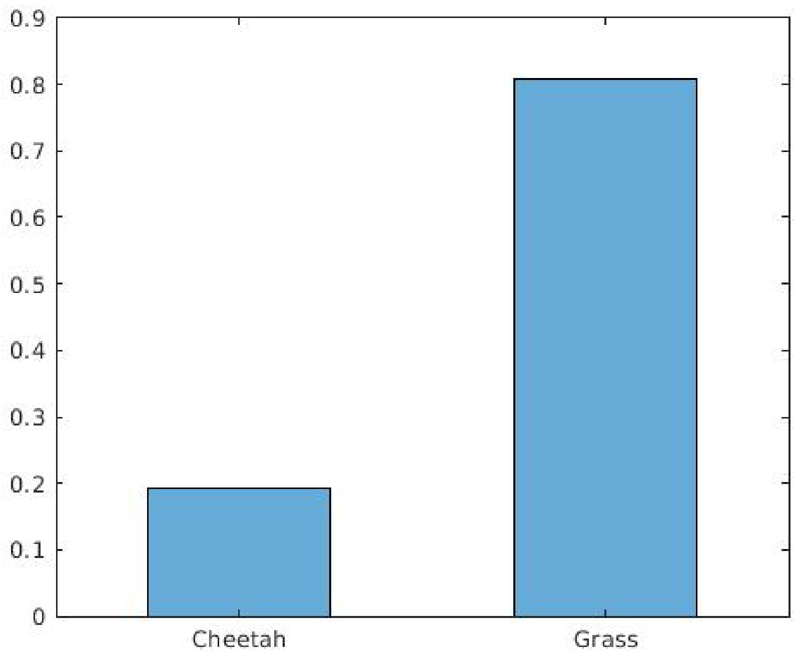

Figure 1: Histogram to calculate prior probabilities.

(b) Using the mean and standard deviation of each feature for the 2 classes, the following 64 marginals densities were obtained. The red dotted line corresponds to grass class and the blue continuous corresponds to the cheetah class.

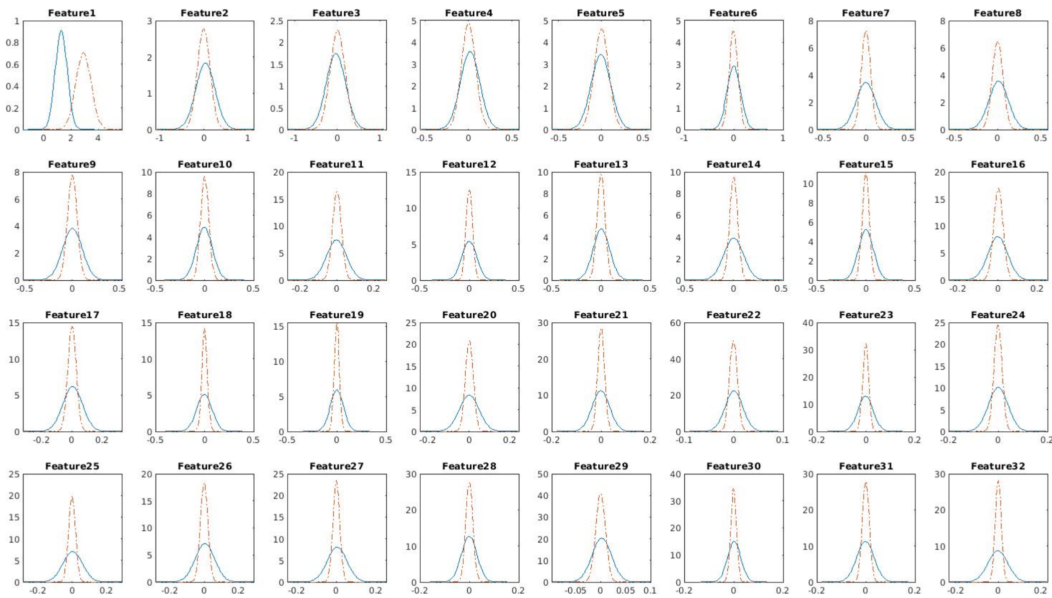

Figure 2: Marginal densities of the first 32 features.

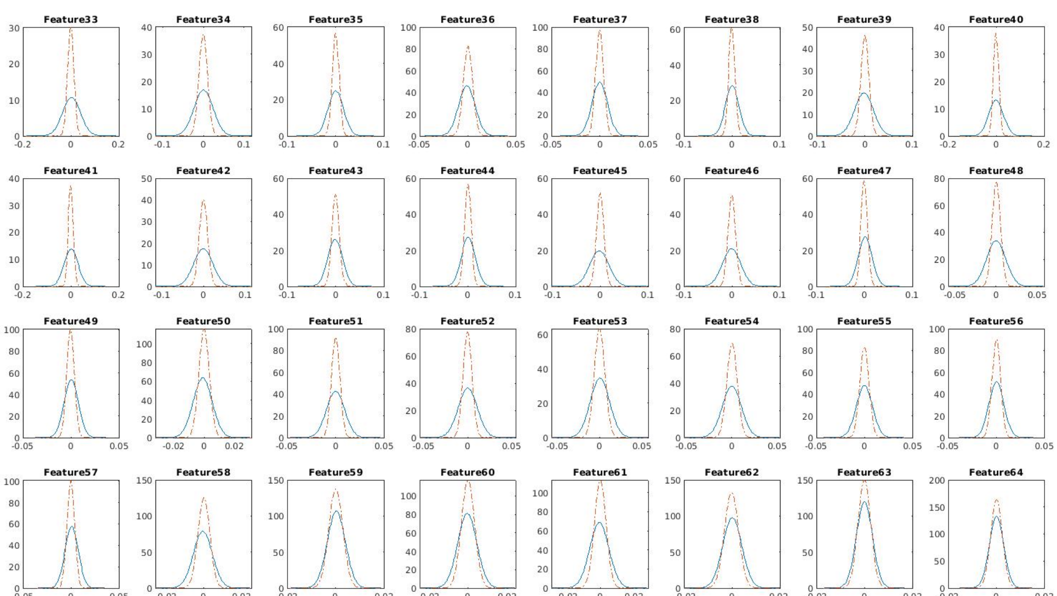

Figure 3: Marginal densities of the last 32 features.

The extent of overlap of the 2 distributions across the 2 class was observed. Lower the overlap, better the feature. Based on this criterion, following are the marginal densities of the best and worst 8 features.

The marginals of worst 8 and best 8 features are plotted below.

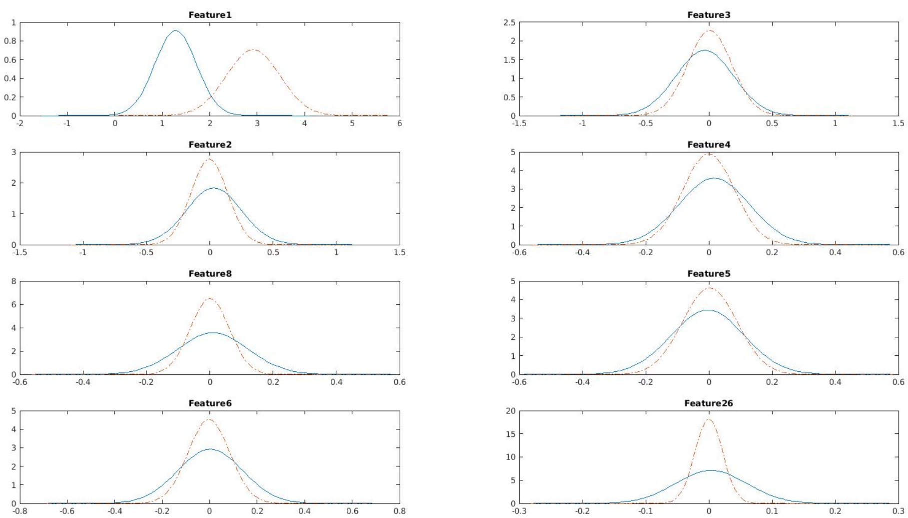

Figure 4: Plots of marginals for the best 8 features.

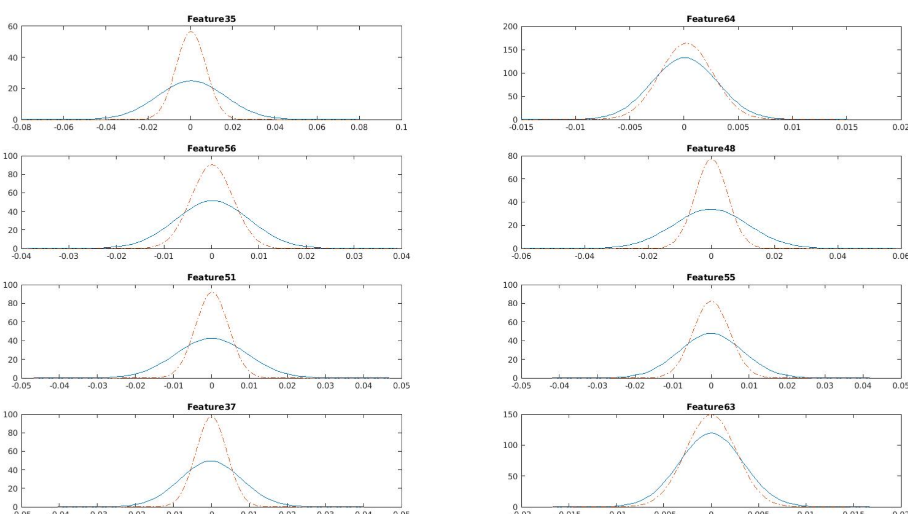

Figure 5: Plots of marginals for the worst 8 features.

(c) Following are the masks calculated with the 8 best features and with all 64 features. The results with 8 best features is better visually than taking all 64 features. The probability of error reflects the same. The possible reason for this is that the additional features are worsening the separability of the 2 classes. By including these, the 2 classes have more overlap leading to wrong classifications.

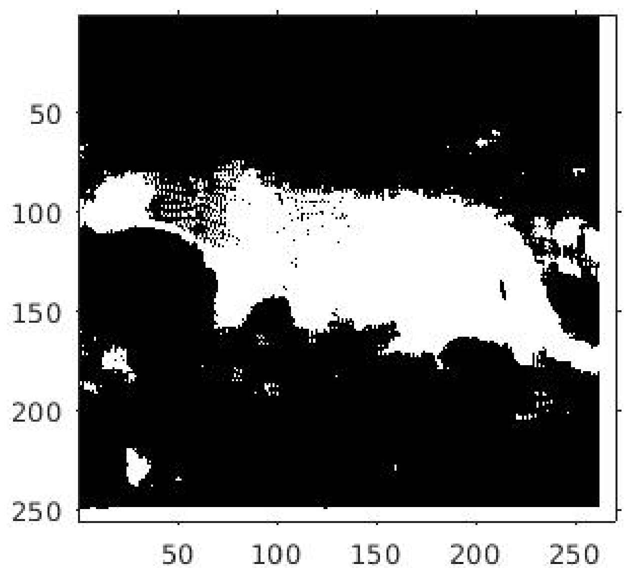

Figure 6: Mask calculated using the best 8 features.

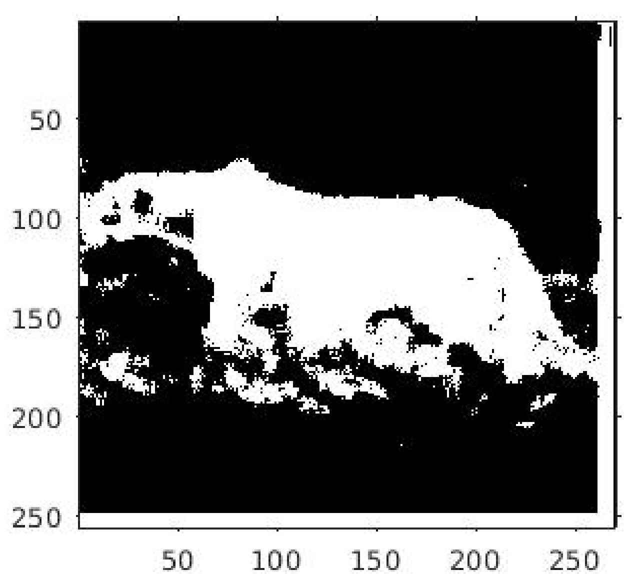

Figure 7: Mask calculated using all 8 features.

(d) The probability of error by the following formula -

$$P(\text{error}) = \frac{\text{No. of mislabeled cheetah}}{\text{No. of cheetah in GT}} \times P(\text{cheetah}) + \frac{\text{No. of mislabeled grass}}{\text{No. of grass in GT}} \times P(grass)$$

The probability of error in case of using 8 features is -

$$P(\text{error}) = \frac{3132}{13209} \times 0.1909 + \frac{1238}{55641} \times 0.0635$$

$$P(\text{error}) = 0.0635$$

The probability of error in case of using all 64 features is -

$$P(\text{error}) = \frac{6721}{13209} \times 0.1909 + \frac{931}{55641} \times 0.1111$$

$$P(\text{error}) = 0.1111$$

MATLAB Code

```
clc;
clear all;
close all;

load('TrainingSamplesDCT_8_new.mat');

%% 5a)
% Prior probabilities
pFG = size(TrainsampleDCT_FG,1)/(size(TrainsampleDCT_BG,1)+size(
    TrainsampleDCT_FG,1));
pBG = 1 - pFG;

disp('The priror probability of cheetah is ')
disp(pFG);

disp('The priror probability of background is ')
disp(pBG);

% Plot histogram
figure();
X = [ones(size(TrainsampleDCT_FG,1),1);  ones(size(
    TrainsampleDCT_BG,1),1)*2];
C = categorical(X,[1 2],{'Cheetah','Grass'});
hBG = histogram(C,'BarWidth',0.5, ...
      Normalization = 'probability');

% Calculate the mean and std of the 64 features
% Mean
meanBG = mean(TrainsampleDCT_BG);
meanFG = mean(TrainsampleDCT_FG);

% Standard deviataion.
stdBG = std(TrainsampleDCT_BG);
stdFG = std(TrainsampleDCT_FG);

%% 5b)
% plot the class conditional densities
% Feature 1 - 32
figure;
for i = 1 : size(meanFG,2)/2
    % Iterate through each feature.

    % Plot histograms
    subplot(4,size(meanFG,2)/8,i);
```

```matlab
    [x_BG, x_FG, y_BG, y_FG] = getXYdata(stdFG(i), stdBG(i), meanFG(i), meanBG(i));
    plot(x_FG,y_FG,'-',x_BG,y_BG,'-.');
    title(strcat('Feature ',int2str(i)));
end

figure;
% Feature 33 - 64
for i = size(meanFG,2)/2+1 : size(meanFG,2)
    % Iterate through each feature.

    % Plot histograms
    subplot(4,size(meanFG,2)/8,i-32);
    [x_BG, x_FG, y_BG, y_FG] = getXYdata(stdFG(i), stdBG(i), meanFG(i), meanBG(i));
    plot(x_FG,y_FG,'-',x_BG,y_BG,'-.');
    title(strcat('Feature ',int2str(i)));
end

figure;
% Best 8 features
idxB = [1,11,20,25,31,40,44,41];
% plot the best 8 features
for i = 1 : 8
    % Plot histograms
    subplot(4,2,i);
    [x_BG, x_FG, y_BG, y_FG] = ...
        getXYdata(stdFG(idxB(i)), stdBG(idxB(i)), meanFG(idxB(i)), meanBG(idxB(i)));
    plot(x_FG,y_FG,'-',x_BG,y_BG,'-.');
    title(strcat('Feature ',int2str(idxB(i))));
end

figure;
% Worst 8 features.
idxW = [2,5,58,59,60,62,63,64];
% Plot the worst 8 features
for i = 1 : 8
    % Iterate through each feature.

    % Plot histograms
    subplot(4,2,i);
    [x_BG, x_FG, y_BG, y_FG] = ...
        getXYdata(stdFG(idxW(i)), stdBG(idxW(i)), meanFG(idxW(i)), meanBG(idxW(i)));
    plot(x_FG,y_FG,'-',x_BG,y_BG,'-.');
```

```
    title(strcat('Feature ',int2str(idxW(i))));
end

% 8 Features
covFG_8 =  diag(var(TrainsampleDCT_FG(:,idxB(1:8))));
covFGDet_8 = det(covFG_8);
covBG_8 =  diag(var(TrainsampleDCT_BG(:,idxB(1:8))));
covBGDet_8 = det(covBG_8);
meanFG_8 = meanFG(idxB(1:8));
meanBG_8 = meanBG(idxB(1:8));
alphaFG_8 = log((2*pi)^8*covFGDet_8)- 2*log(pFG);
alphaBG_8 = log((2*pi)^8*covBGDet_8)- 2*log(pBG);

% 64 features
covFG_64 =  cov(TrainsampleDCT_FG);
covFGDet_64 = det(covFG_64);
covBG_64 =  cov(TrainsampleDCT_BG);
covBGDet_64 = det(covBG_64);
meanFG_64 = meanFG;
meanBG_64 = meanBG;
alphaFG_64 = log((2*pi)^64*covFGDet_64)- 2*log(pFG);
alphaBG_64 = log((2*pi)^64*covBGDet_64)- 2*log(pBG);

% Predict mask for test image
img = imread('cheetah.bmp');
img = im2double(img);
mask1 = zeros(size(img));
mask2 = zeros(size(img));
img = padarray(img,[7,7],'replicate','post');

% Slide a 8X8 window over the image, calculate its DCT coeffecients. Select
% the index of 2nd largest value as the feature to calculate posterior
% probabilities.
zigZagIdx = readmatrix('Zig-Zag Pattern.txt');
for i = 1 : 255
    for j = 1 : 270
        block = img(i:i+7, j:j+7);
        dctF = dct2(block);
        fIdx(zigZagIdx(:)+1) = dctF(:);
        f = fIdx(idxB(1:8));

        dFG = (f - meanFG_8)*inv(covFG_8)*transpose((f - meanFG_8)) + alphaFG_8;
        dBG = (f - meanBG_8)*inv(covBG_8)*transpose((f - meanBG_8)
```

```
            ) + alphaBG_8;
        if(dFG < dBG)
            mask1(i,j) = 1;
        end

        f = fIdx;
        dFG = (f - meanFG_64)*inv(covFG_64)*transpose((f - meanFG_64)) + alphaFG_64;
        dBG = (f - meanBG_64)*inv(covBG_64)*transpose((f - meanBG_64)) + alphaBG_64;
        if(dFG < dBG)
            mask2(i,j) = 1;
        end
    end
end

figure();
imshow(mask1);

figure();
imshow(mask2);

%% 5c)
gTruth = im2double(imread('cheetah_mask.bmp'));
pError_8 = calculateError(mask1, gTruth, pBG, pFG);
pError_64 = calculateError(mask2, gTruth, pBG, pFG);
disp('Probability of error with best 8 features is ')
disp(pError_8)
disp('Probability of error with all 64 features is ')
disp(pError_64)

%% Helper Functions

% Function to calculate gaussian
function y = gaussian(x, mu, sigma)
%     y = normpdf(x, mu, sigma);
    y =  exp(-power((x-mu)/sigma,2)/2)/ (sigma*sqrt(2*pi));
end

function [x_BG, x_FG, y_BG, y_FG] = getXYdata(stdFG, stdBG, meanFG, meanBG)

    k = 5;
    stdMax = max(stdFG, stdBG);
    x_FG = (meanFG-k*stdMax: stdMax*2*k/100 :meanFG +k*stdMax);
    y_FG = gaussian(x_FG, meanFG, stdFG);
```

```
    x_BG = (meanBG-k*stdMax: stdMax*2*k/100 :meanBG + k*stdMax);
    y_BG = gaussian(x_BG, meanBG, stdBG);
end

function pError = calculateError(mask, gTruth, pB, pF)
    nCheetah = nnz(gTruth);
    nGrass = nnz(1 - gTruth);
    nMislabeledCheetah = nnz((mask-gTruth)>0);
    nMislabeledGrass = nnz((mask-gTruth)<0);
    pError = nMislabeledGrass/nGrass*pB + nMislabeledCheetah/
        nCheetah*pF;

end
```