ECE 271A: Statistical Learning I - Quiz #2

Mazeyu Ji - A59023027

December 1, 2023

Solution a)

Based on the findings from problem 2, the maximum likelihood estimation for prior probabilities is given by:

$$\pi_j^* = \frac{c_j}{n}$$

where c_j represents the count of observations of class j, and n denotes the overall number of observations.
Let:

- n_{FG} represent the number of foreground (cheetah) samples.
- n_{BG} represent the number of background (grass) samples.

The prior probabilities can be estimated as:

$$P_Y(\text{cheetah}) = \frac{n_{FG}}{n_{FG} + n_{BG}} = 0.1919 \tag{1}$$

$$P_Y(\text{grass}) = \frac{n_{BG}}{n_{FG} + n_{BG}} = 0.8081 \tag{2}$$

The following figure shows the histogram of the number of samples and prior probability for each class:

Figure 1: Histograms showing the number of samples and prior probility.

Solution b)

The marginal densities of the two classes are illustrated in Figure 2 for k values ranging from 1 to 64. In the figure, the red lines depict $P_{x_k|Y}(x_k|\text{cheetah})$, while the blue lines showcase $P_{x_k|Y}(x_k|\text{grass})$.

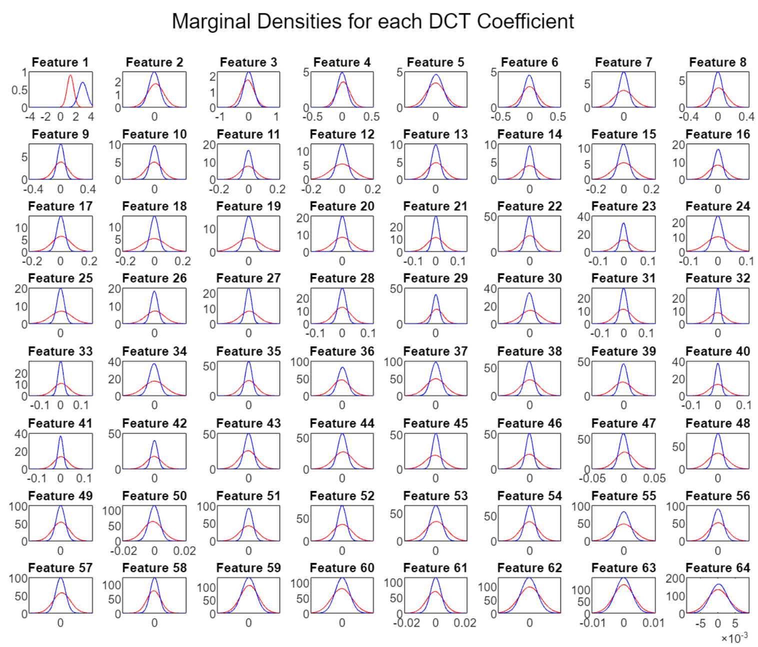

Figure 2: Marginal densities of all 64 features.

By visual inspection, we select [1,14,17,21,32,40,41,45] as the best 8 features, and the marginal densities are shown in Figure 3. These features have significant variances or differences in mean values, which can better assist in classification.

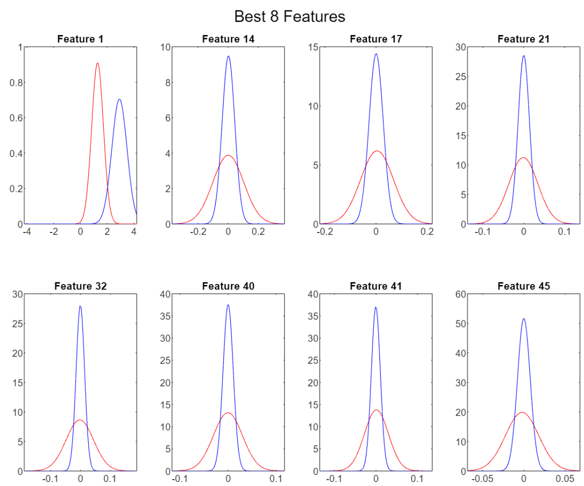

Figure 3: Marginal densities of the best 8 features.

By visual inspection, we select [3,4,5,59,60,62,63,64] as the worst 8 features, and the marginal densities are shown in Figure 4. These features all have similar distributions, making it difficult to distinguish.

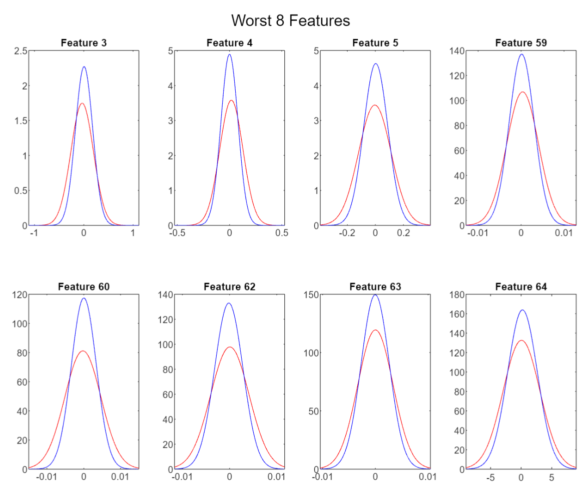

Figure 4: Marginal densities of the worst 8 features.

Solution c)

To construct the mask, the top-left pixel of an 8×8 block is marked as '1' if the block is identified as containing the cheetah, otherwise, it is set to '0'. Implementing a sliding window mechanism that shifts by one pixel at each iteration, the final array A is produced. For a given block, the state can be identified as 'cheetah', expressed as:

$$\frac{P_{X|Y}(x|\text{cheetah})}{P_{X|Y}(x|\text{grass})} > \frac{P_Y(\text{grass})}{P_Y(\text{cheetah})} = T \tag{3}$$

Where:

- $P_{X|Y}(x|\text{cheetah})$ and $P_{X|Y}(x|\text{grass})$ are class conditional probabilities estimated from the training data.
- $P_Y(\text{cheetah})$ and $P_Y(\text{grass})$ represent the prior probabilities estimated from the training data.
- T denotes the decision threshold.

The conditional probabilities $P_{X|Y}(x|\text{cheetah})$ and $P_{X|Y}(x|\text{grass})$ are determined by the equation:

$$P_{X|Y}(x^i) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_i|}} \exp\left\{-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i)\right\}$$

The generated mask is shown in Figure 5, The accuracy using the best 8 features is 5.51%, while using all 64 features, the accuracy is 8.96%.
The probability of error of all the 64 features is computed with:

$$P_E = E_Y[P_{X|Y}(g(x) \neq Y|Y)] = \sum_i P_{X|Y}(g(x) \neq i|i)P_Y(i)$$

$$= P_{X|Y}(g(x) = cheetah|grass)P_Y(grass) + P_{X|Y}(g(x) = grass|cheetah)P_Y(cheetah)$$

$$= 0.0935 \times 0.8081 + (1 - 0.9269) \times 0.1919$$

$$\approx 0.0896$$

The probability of error of the best 8 features is computed with:

$$P_E = E_Y[P_{X|Y}(g(x) \neq Y|Y)] = \sum_i P_{X|Y}(g(x) \neq i|i)P_Y(i)$$

$$= P_{X|Y}(g(x) = cheetah|grass)P_Y(grass) + P_{X|Y}(g(x) = grass|cheetah)P_Y(cheetah)$$

$$= 0.0417 \times 0.8081 + (1 - 0.8887) \times 0.1919$$

$$\approx 0.0551$$

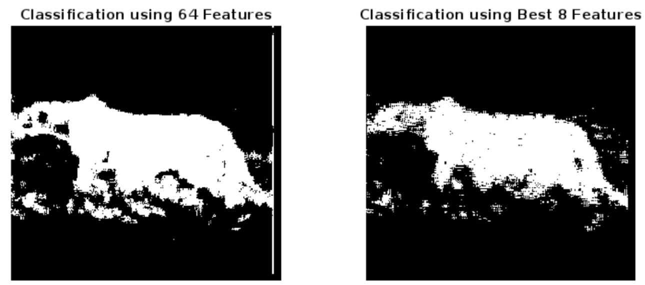

Figure 5: The left image is the mask generated using all 64 features, while the right image is the mask generated using the best 8 features.

The improved accuracy with the best 8 features compared to using all 64 features can be attributed to the following reasons:

- **Feature Relevance**: The best 8 features were selected based on their variance and mean difference, which likely means they are highly discriminative. Such features can more effectively distinguish between the two classes than features with lesser variance or mean difference.
- **Noise Reduction**: Incorporating all 64 features might introduce irrelevant or noisy data into the model, potentially reducing its accuracy. By selecting only the top 8 features, the model is focusing on the most pertinent information and avoiding unnecessary noise.

Appendix

```
%% (a)Estimate the prior probabilities
load('TrainingSamplesDCT_8_new.mat')
[rowFG, columnFG] = size(TrainsampleDCT_FG);
[rowBG, columnBG] = size(TrainsampleDCT_BG);
priorFG = rowFG / (rowBG + rowFG);
priorBG = rowBG / (rowBG + rowFG);
% Plot the number of samples for each class
figure;
bar([rowFG, rowBG]);
set(gca, 'XTickLabel', {'Cheetah (Foreground)', 'Grass (
    Background)'});
ylabel('Number of Samples');
title('Number of Samples for Each Class');
grid on;
% Plot the prior probability for each class
figure;
bar([priorFG, priorBG]);
set(gca, 'XTickLabel', {'Cheetah (Foreground)', 'Grass (
    Background)'});
ylabel('Prior Probability');
title('Prior Probability for Each Class');
ylim([0 1]);
grid on;


%% (b)Plot the marginal densities for the two classes
% Compute the MLE for parameters (mean and covariance) for
    both classes
meanFG = mean(TrainsampleDCT_FG);
meanBG = mean(TrainsampleDCT_BG);
covFG = cov(TrainsampleDCT_FG);
covBG = cov(TrainsampleDCT_BG);

% Create 64 plots for the marginal densities
figure;

for k = 1:64
    subplot(8,8,k);

    % Determine the range for x centered at 0
    absMax = max(abs([TrainsampleDCT_FG(:, k);
        TrainsampleDCT_BG(:, k)]));
    % Gaussian PDF for the k-th DCT coefficient for Cheetah
        and Grass
    x = linspace(-absMax, absMax, 100);
    yFG = my_normpdf(x, meanFG(k), sqrt(covFG(k,k)));
    yBG = my_normpdf(x, meanBG(k), sqrt(covBG(k,k)));

```

```
    plot(x, yFG, 'r-', x, yBG, 'b-');
    title(['Feature ' num2str(k)]);
    % legend('Cheetah','Grass');
    xlim([-absMax, absMax]);
end

sgtitle('Marginal Densities for each DCT Coefficient');

% Seletecting the features
bestFeatures = [1,14,17,21,32,40,41,45];
worstFeatures = [3,4,5,59,60,62,63,64];

% Plotting best features
figure;
for i = 1:8
    k = bestFeatures(i);
    subplot(2,4,i);

    % Determine the range for x centered at 0
    absMax = max(abs([TrainsampleDCT_FG(:, k);
        TrainsampleDCT_BG(:, k)]));
    x = linspace(-absMax, absMax, 100);

    yFG = my_normpdf(x, meanFG(k), sqrt(covFG(k,k)));
    yBG = my_normpdf(x, meanBG(k), sqrt(covBG(k,k)));

    plot(x, yFG, 'r-', x, yBG, 'b-');
    title(['Feature ' num2str(k)]);
    % legend('Cheetah','Grass');
    xlim([-absMax, absMax]);
end
sgtitle('Best 8 Features');

% Plotting worst features
figure;
for i = 1:8
    k = worstFeatures(i);
    subplot(2,4,i);

    % Determine the range for x centered at 0
    absMax = max(abs([TrainsampleDCT_FG(:, k);
        TrainsampleDCT_BG(:, k)]));
    x = linspace(-absMax, absMax, 100);

    yFG = my_normpdf(x, meanFG(k), sqrt(covFG(k,k)));
    yBG = my_normpdf(x, meanBG(k), sqrt(covBG(k,k)));

    plot(x, yFG, 'r-', x, yBG, 'b-');
    title(['Feature ' num2str(k)]);
    % legend('Cheetah','Grass');
```

```
    xlim([-absMax, absMax]);
end
sgtitle('Worst 8 Features');


%% (c)Separate the foreground and background with 64 and 8 features
% Read and preprocess the image
img = imread('cheetah.bmp');
imgDouble = im2double(img);
[height, width] = size(imgDouble);

% Get pattern index
pattern = readmatrix('Zig-Zag Pattern.txt') + 1;

% Calculate the threshold
thresStar = priorBG / priorFG;

% Compute P(x|cheetah) and P(x|grass) and make a decision
mask64 = zeros(height, width);
mask8 = zeros(height, width);

% Loop over the image
for i = 1:height-7
    for j = 1:width-7
        block = imgDouble(i:i+7, j:j+7); % 8x8 block
        dctBlock = dct2(block);
        zigzag = zeros(1,64);
        for m = 1:8
            for n = 1:8
                zigzag(pattern(m,n)) = dctBlock(m,n);
            end
        end

        % 64-dimensional classification
        Px_yFG64 = my_mvnpdf(zigzag, meanFG, covFG);
        Px_yBG64 = my_mvnpdf(zigzag, meanBG, covBG);
        if Px_yFG64 / Px_yBG64 > thresStar
            mask64(i, j) = 1;
        end

        % 8-dimensional classification
        Px_yFG8 = my_mvnpdf(zigzag(bestFeatures), meanFG(bestFeatures), covFG(bestFeatures,bestFeatures));
        Px_yBG8 = my_mvnpdf(zigzag(bestFeatures), meanBG(bestFeatures), covBG(bestFeatures,bestFeatures));
        if Px_yFG8 / Px_yBG8 > thresStar
            mask8(i, j) = 1;
        end
    end
```

```
end

% Display the masks
figure;
subplot(1,2,1);
imshow(mask64);
title('Classification using 64 Features');

subplot(1,2,2);
imshow(mask8);
title('Classification using Best 8 Features');

% Compute error
maskGT = imread('cheetah_mask.bmp');
maskGT = im2double(maskGT);
error64 = sum(sum(mask64 ~= maskGT)) / (height * width);
error8 = sum(sum(mask8 ~= maskGT)) / (height * width);

%% Define functions to calculate the PDF for Normal Distribution
function single_pdf = my_normpdf(x, mu, sigma)
    single_pdf = 1 / (sqrt(2 * pi) * sigma) * exp(-(x - mu).^2 / (2 * sigma^2));
end

function multi_pdf = my_mvnpdf(x, mu, Sigma)
    k = length(mu);
    multi_pdf = 1 / ((2 * pi)^(k/2) * sqrt(det(Sigma))) * exp(-0.5 * (x - mu) * inv(Sigma) * (x - mu)');
end
```