ECE 271A - Statistical Learning 1 - Homework 1

Shubhan Mital

13th October 2025

Question Subpart 1 - Using the training data in `TrainingSamplesDCT_8.mat`, what are reasonable estimates for the prior probabilities?

From the training data (`TrainingSamplesDCT_8.mat`):

$$P(Y = \text{cheetah}) = \frac{N_{FG}}{N_{FG} + N_{BG}}, \quad P(Y = \text{grass}) = \frac{N_{BG}}{N_{FG} + N_{BG}}$$

where

$$\begin{aligned} N_{FG} &= 250 & N_{BG} &= 1053 \\ P(Y = \text{cheetah}) &\approx 0.192 & P(Y = \text{grass}) &\approx 0.808 \end{aligned}$$

```
--- Prior Probabilities ---
P(Y = Background) = 0.8081
P(Y = Cheetah)    = 0.1919
```

Figure 1: Displayed prior probabilities from the training data: $P(Y = \text{Background}) = 0.8081$, $P(Y = \text{Cheetah}) = 0.1919$, as calculated from `TrainingSamplesDCT_8.mat`.

Question Subpart 2 - Using the training data in `TrainingSamplesDCT_8.mat`, compute and plot the index histograms $P_{X|Y}(x|\textbf{cheetah})$ and $P_{X|Y}(x|\textbf{grass})$

For each class, we compute a histogram of feature indices $(1 - 64)$ and normalize to obtain probabilities:

$$P(X|Y) = \frac{\text{histogram count at } X}{\text{total samples}}$$

Laplace smoothing ($\epsilon = 1$) is applied to avoid zero probabilities:

$$P(X|Y) = \frac{\text{count}(X) + \epsilon}{\sum_X (\text{count}(X) + \epsilon)}$$

We obtain two distributions: $P(X|Y = \text{cheetah})$, $P(X|Y = \text{grass})$.

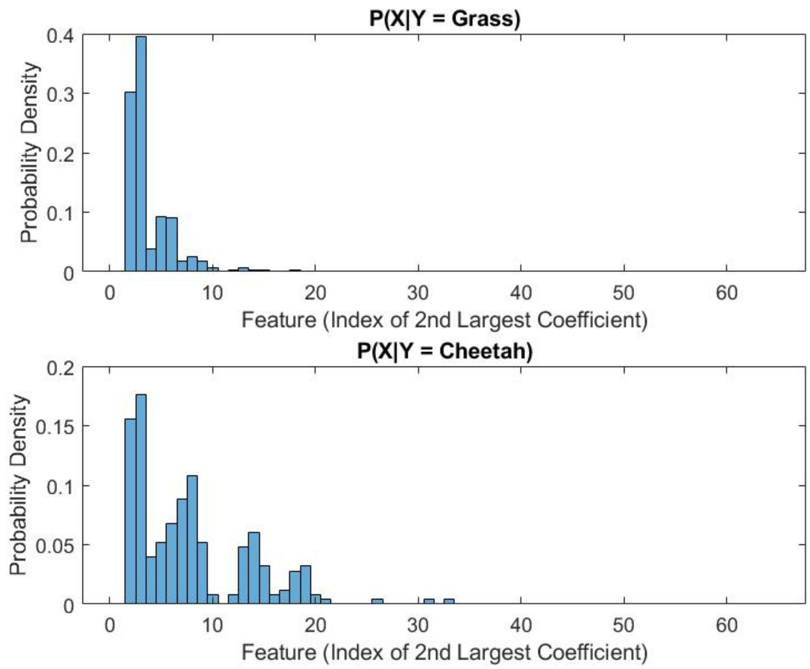

Figure 2: Estimated index histograms showing $P_{X|Y}(x\,|\,\text{cheetah})$ and $P_{X|Y}(x\,|\,\text{grass})$ from normalized training data (`TrainingSamplesDCT_8.mat`). Each histogram illustrates the probability density of the index of the 2nd largest DCT coefficient for cheetah and grass classes.

Question Subpart 3 - For each block in the image `cheetah.bmp`, compute the feature X (index of the DCT coefficient with 2nd greatest energy). Compute the state variable Y using the minimum probability of error rule based on the probabilities obtained in a) and b). Store the state in an array A. Using the commands `imagesc` and `colormap(gray(255))`, create a picture of that array.

For each block in the test image:

$$P(Y = \text{cheetah}|X) = \frac{P(X|Y = \text{cheetah})P(Y = \text{cheetah})}{P(X)}$$

Since $P(X)$ is common, we compare unnormalized posteriors:

$$\text{Decide cheetah if } P(X|Y = \text{cheetah})P(Y = \text{cheetah}) > P(X|Y = \text{grass})P(Y = \text{grass})$$

The resulting binary map A marks cheetah pixels as 1 (white) and grass pixels as 0 (black).

The cheetah region is captured clearly, although with noise around the edges due to the use of a single scalar feature.

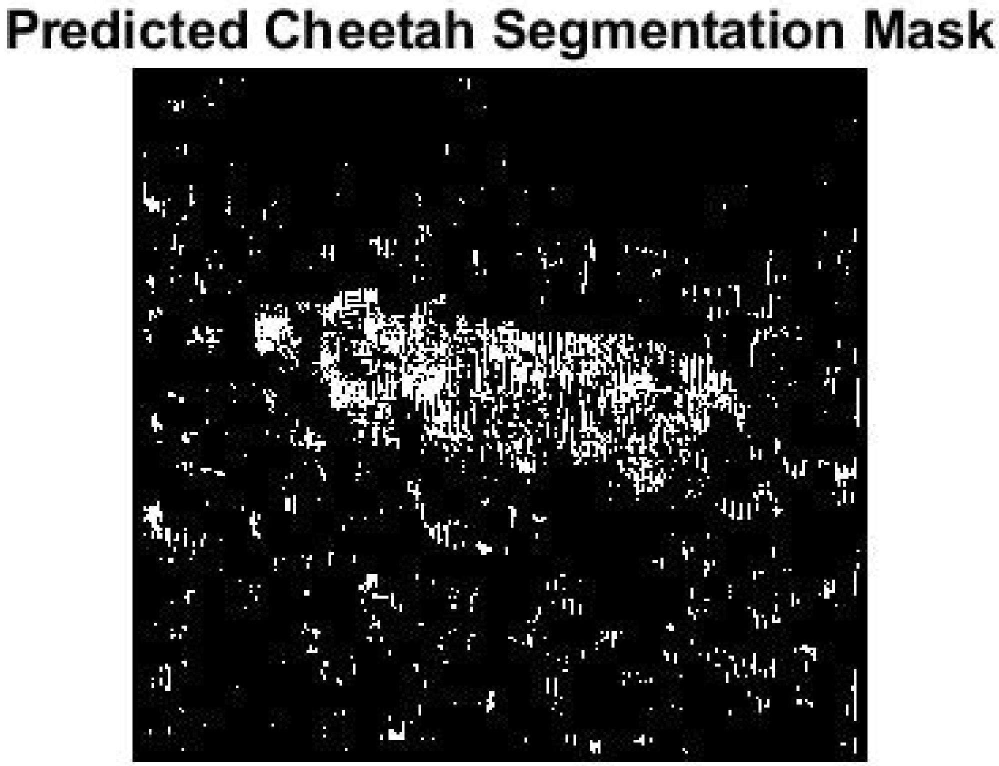

Figure 3: Predicted cheetah segmentation mask using the minimum probability of error rule.

Question Subpart 4 - The array A contains a mask that indicates which blocks contain grass and which contain the cheetah. Compare it with the ground truth provided in image `cheetah_mask.bmp` and compute the probability of error of your algorithm.

To evaluate the performance of the segmentation algorithm, we compare the predicted mask A to the ground truth mask (`cheetah_mask.bmp`). The probability of error, P_e, is calculated as:

$$P_e = \frac{\text{Number of misclassified pixels}}{\text{Total number of pixels}}$$

After aligning and cropping the predicted and ground truth masks, we obtain:

$$P_e = 0.1681$$

This corresponds to an accuracy of approximately 83.19%.

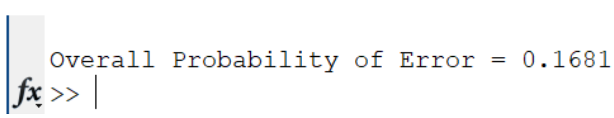

Figure 4: MATLAB output showing the overall probability of error for the segmentation algorithm.

MATLAB Code

1. Load Data and Feature Extraction

```
% --- Load Training Data ---
load('TrainingSamplesDCT_8.mat');
TrainsampleDCT_BG = TrainsampleDCT_BG;
TrainsampleDCT_FG = TrainsampleDCT_FG;

% --- Feature Extraction: index of 2nd largest DCT (ignore DC term) ---
[~, features_BG] = max(TrainsampleDCT_BG(:, 2:64), [], 2);
[~, features_FG] = max(TrainsampleDCT_FG(:, 2:64), [], 2);
features_BG = features_BG + 1; % Matlab 1-indexing
features_FG = features_FG + 1;
```

2. PDFs and Priors

```
% --- Estimate Class-Conditional PDFs ---
count_features_BG = histcounts(features_BG, 1:65);
count_features_FG = histcounts(features_FG, 1:65);
prob_features_BG = (count_features_BG + 1) / sum(count_features_BG + 1);
prob_features_FG = (count_features_FG + 1) / sum(count_features_FG + 1);

% --- Priors ---
total_features_BG = length(features_BG);
total_features_FG = length(features_FG);
total_features = total_features_BG + total_features_FG;
prob_BG = total_features_BG / total_features;
prob_FG = total_features_FG / total_features;
```

3. Blockwise Classification

```
A = im2double(imread('cheetah.bmp'));
mask_gt = im2double(imread('cheetah_mask.bmp'));
if size(A,3) == 3, A = A(:,:,1); end
if size(mask_gt,3) == 3, mask_gt = mask_gt(:,:,1); end
[img_row, img_col] = size(A);

test_op_img = zeros(img_row, img_col);
for row = 1:img_row-7
    for col = 1:img_col-7
        block = A(row:row+7, col:col+7);
        dct_block = abs(dct2(block));
        B = zigzag(dct_block);
        [~, sorted_idx] = sort(B);
        feature_idx = sorted_idx(end-1);

        center_r = row + 4;
        center_c = col + 4;
        if center_r <= img_row && center_c <= img_col
            if (prob_features_BG(feature_idx)*prob_BG >= prob_features_FG(
                feature_idx)*prob_FG)
                test_op_img(center_r, center_c) = 0;
            else
                test_op_img(center_r, center_c) = 1;
            end
        end
    end
end
```

4. Error Probability and Output

```
count_gx0_given_y1 = sum(sum((mask_gt == 1) & (test_op_img == 0)));
count_gx1_given_y1 = sum(sum((mask_gt == 1) & (test_op_img == 1)));
count_gx1_given_y0 = sum(sum((mask_gt == 0) & (test_op_img == 1)));
count_gx0_given_y0 = sum(sum((mask_gt == 0) & (test_op_img == 0)));

prob_gx0_given_y1 = (count_gx0_given_y1 + 1) / (count_gx0_given_y1 +
    count_gx1_given_y1 + 2);
prob_gx1_given_y0 = (count_gx1_given_y0 + 1) / (count_gx1_given_y0 +
    count_gx0_given_y0 + 2);
prob_gx0_given_y0 = (count_gx0_given_y0 + 1) / (count_gx0_given_y0 +
    count_gx1_given_y0 + 2);
prob_gx1_given_y1 = (count_gx1_given_y1 + 1) / (count_gx0_given_y1 +
    count_gx1_given_y1 + 2);
prob_error = prob_gx0_given_y1 * prob_FG + prob_gx1_given_y0 * prob_BG;
fprintf('---␣Prior␣Probabilities␣---\n');
fprintf('P(Y␣=␣Background)␣=␣%.4f\n', prob_BG);
fprintf('P(Y␣=␣Cheetah)␣␣␣␣=␣%.4f\n\n', prob_FG);
```

5. Zigzag Function

```
function zz = zigzag(block)
    zz = zeros(64,1);
    index = 1;
    for s = 1:8
        if mod(s,2) == 1
            for i = s:-1:1
                j = s + 1 - i;
                zz(index) = block(i,j);
                index = index + 1;
            end
        else
            for j = s:-1:1
                i = s + 1 - j;
                zz(index) = block(i,j);
                index = index + 1;
            end
        end
    end
    for s = 2:8
        if mod(s,2) == 1
            for i = s:8
                j = 8 - (i - s);
                zz(index) = block(i,j);
                index = index + 1;
            end
        else
            for j = s:8
                i = 8 - (j - s);
                zz(index) = block(i,j);
                index = index + 1;
            end
        end
    end
end
```

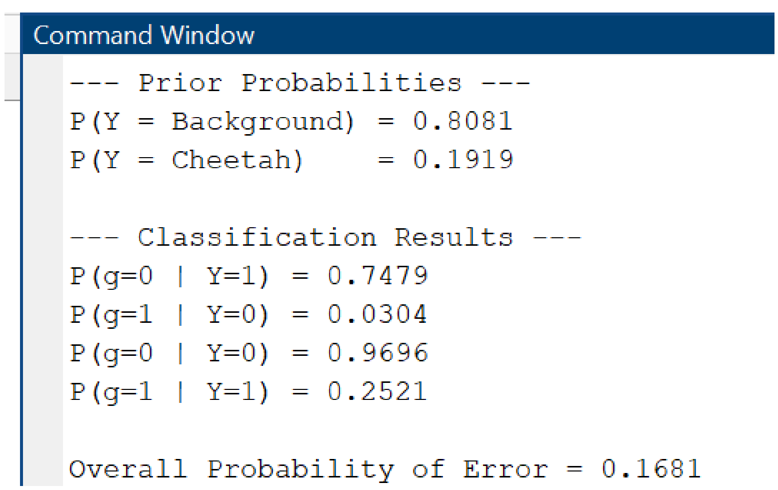

Figure 5: Output in the MATLAB command window showing the prior probabilities, classification results, and overall probability of error produced by the above code.