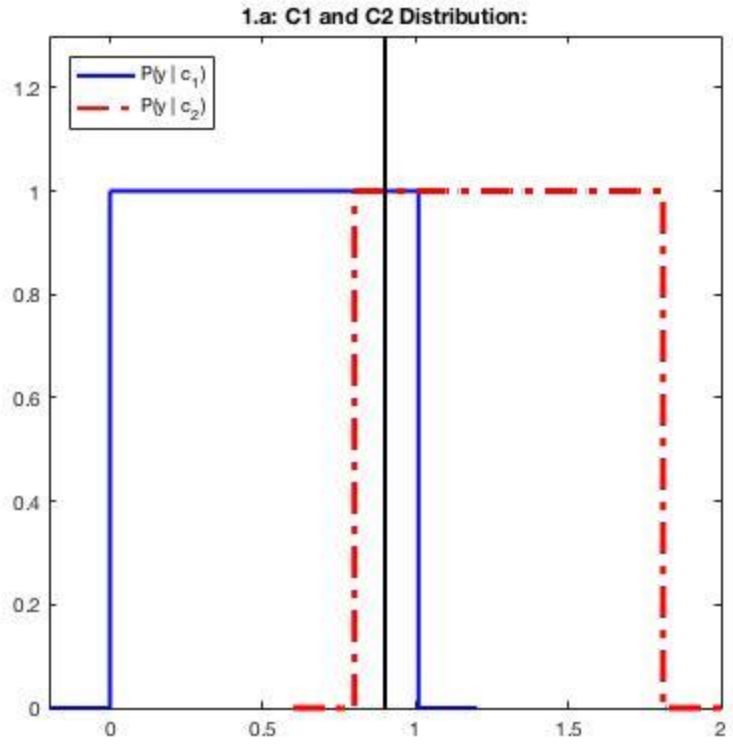
Table of Contents

Q 1

```
% Code:
clear all, close all, clc;
addpath([genpath('./materials') genpath('./functions')])

% A:
pd1 = makedist('Uniform');
pd2 = makedist('Uniform');

A1 = 0;
B1 = 1;
X1 = -0.2:0.01:1.2;
uni1 = unifpdf(X1,A1,B1);
A2 = 0.8;
B2 = 1.8;
X2 = 0.6:0.01:2;
uni2 = unifpdf(X2,A2,B2);

f = figure(1);
ax_plot = axes('Position',[.3 .1 .6 .8]);
stairs(ax_plot, X1, uni1, 'b', 'linewidth', 2), title('1.a: C1 and C2
 Distribution:'), hold on;
stairs(X2, uni2, 'r-.', 'linewidth', 2), xlim([-0.2 2]), ylim([0
 1.3]);

% B:
line([0.9 0.9], get(gca, 'ylim'),'color', 'k', 'linewidth', 2)
legend({'P(y | c_1)','P(y | c_2)' },'Location','northwest')
hold off;

ax_txt = axes('Position',[0 0 1 1],'Visible','off');
axes(ax_txt)
descr = {'1.b:';
    'It is possible that';
    'C2 object will be';
    'falsley classified';
    'between:';
    '0.8 < y < 1'};
text(.025, 0.6, descr, 'FontSize', 15)
% Answers:
```

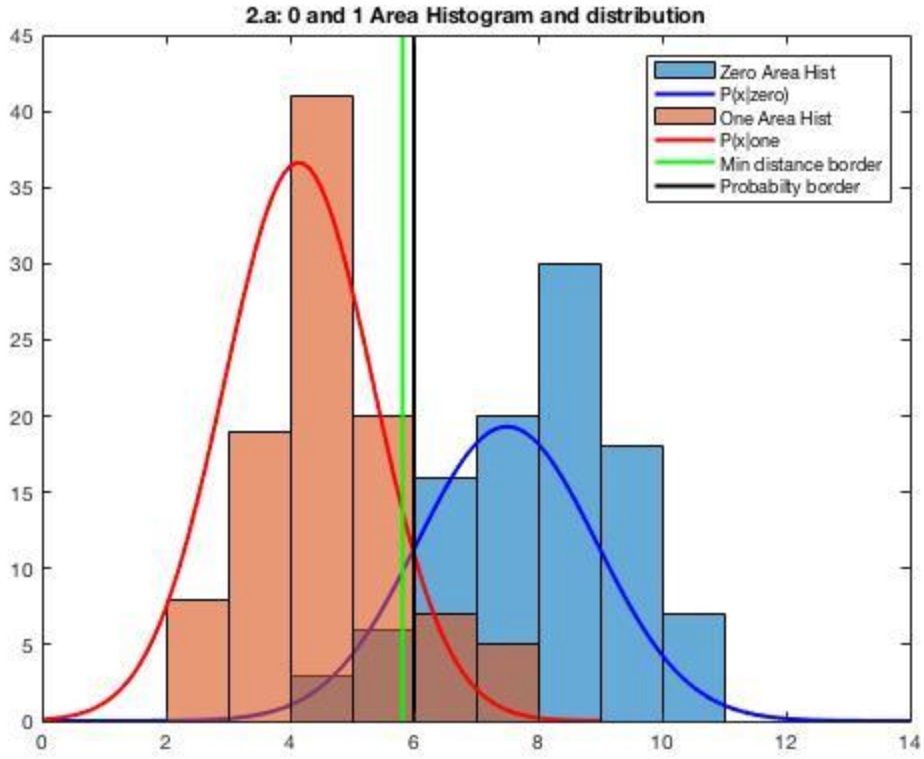
1.b:
It is possible that C2 object will be falsley classified between:
0.8 < y < 1

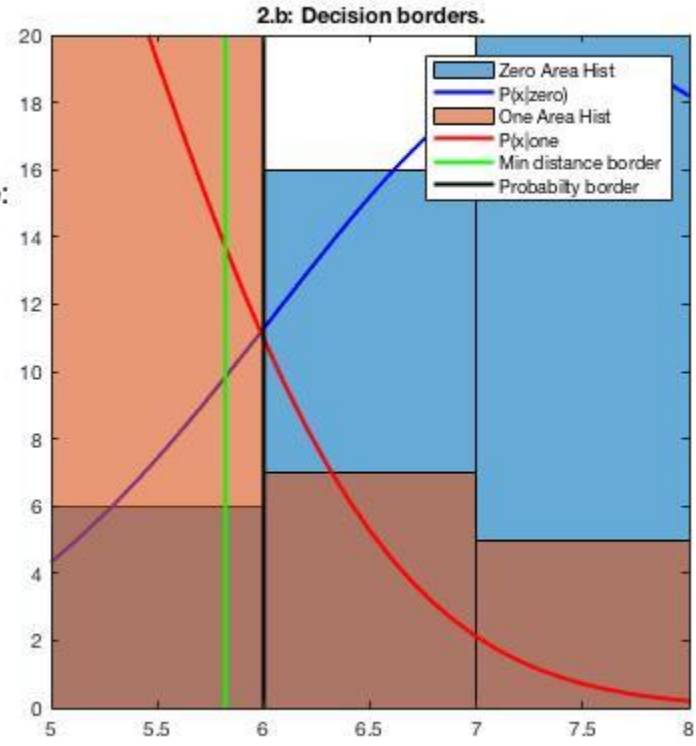


Q 2

```
% Code:
figure(2);
data1 = [repelem(4,3) repelem(5,6) repelem(6,16) repelem(7,20) ...
    repelem(8,30) repelem(9,18) repelem(10,7)];
histogram(data1, 4:11), title('2.a: 0 and 1 Area Histogram and
 distribution'), hold on;
dnorm1  = fitdist(data1(:), 'Normal');
X1 = 0:0.1:14;
Y1 = pdf(dnorm1, X1);
plot(X1, Y1*70,'b', 'linewidth', 2);

data2 = [repelem(2,8) repelem(3,19) repelem(4,41) repelem(5,20) ...
    repelem(6,7) repelem(7,5)];
histogram(data2, 2:8);
dnorm2  = fitdist(data2(:), 'Normal');
X2 = 0:0.1:9;
Y2 = pdf(dnorm2, X2);
plot(X2, Y2*110,'r', 'linewidth', 2);

dist_lm = mean([mean(data1) mean(data2)]);
line([dist_lm dist_lm], get(gca, 'ylim'),'color', 'g', 'linewidth', 2)
line([6 6], get(gca, 'ylim'),'color', 'k', 'linewidth', 2)
```

```
legend({'Zero Area Hist', 'P(x|zero)', 'One Area Hist', 'P(x|
one', 'Min distance border', 'Probabilty border'})

figure(3)
ax_plot = axes('Position',[.37 .1 .57 .8]);
histogram(data1, 4:11),title('2.b: Decision borders.'), hold on;
plot(ax_plot, X1, Y1*70,'b', 'linewidth', 2);


histogram(data2, 2:8);
plot(X2, Y2*110,'r', 'linewidth', 2);

line([dist_lm dist_lm], get(gca, 'ylim'),'color', 'g', 'linewidth', 2)
line([6 6], get(gca, 'ylim'),'color', 'k', 'linewidth', 2)

legend({'Zero Area Hist', 'P(x|zero)', 'One Area Hist', 'P(x|
one', 'Min distance border', 'Probabilty border'})
xlim([5 8]), ylim([0 20])

ax_txt = axes('Position',[0 0 1 1],'Visible','off');
axes(ax_txt)
descr = {'2.b:';
    'Decision by Min distance:';
    sprintf('1 if Area < %g', dist_lm);
    sprintf('0 if Area %s%g', '{\geq}', dist_lm);
    '';
    'Decision by probabilty:';
    '1 if Area <6';
    '0 if Area {\geq}6'};
text(.025,0.6,descr, 'FontSize', 15)
```





2.b:
Decision by Min distance:
1 if Area < 5.82
0 if Area $\geq$5.82

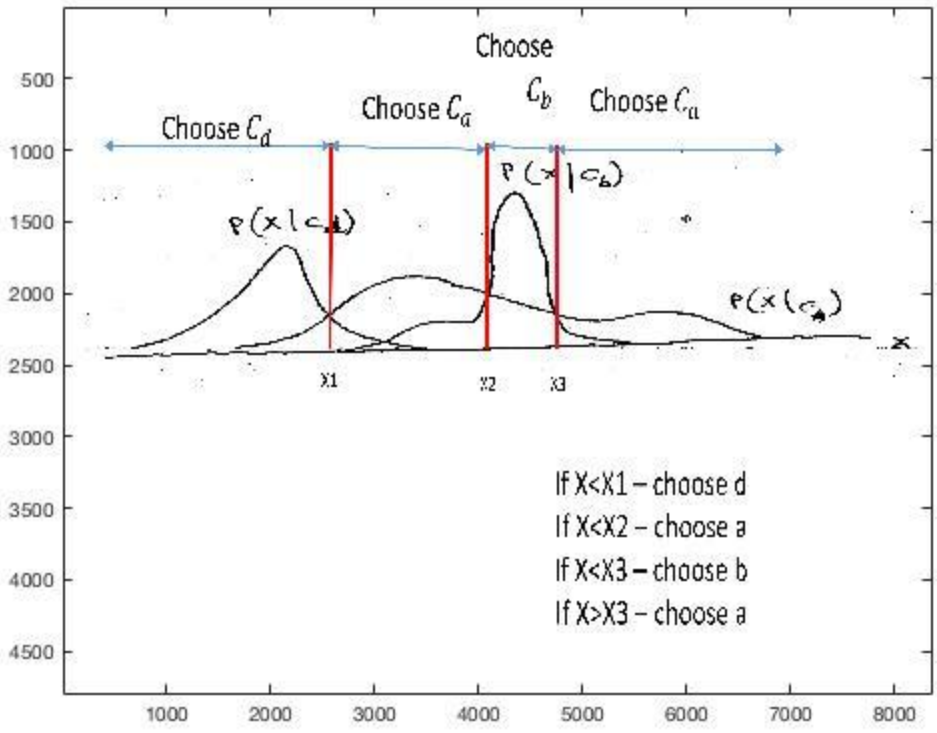
Decision by probabilty:
1 if Area <6
0 if Area $\geq$6

Q 3

```
% A:
letters =
 ['A' 'B' 'C' 'D' 'E' 'F' 'G' 'H' 'I' 'J' 'K' 'L' 'M' 'N' 'O' 'P' 'Q' 'R' 'S' 'T'
num_edges = [2 0 2 0 3 3 2 4 2 2 4 2 2 2 0 1 1 2 2 3 2 2 2 4 3 2];
num_T = [2 1 0 0 1 1 0 2 0 0 2 0 0 0 0 1 1 2 0 1 0 0 0 4 1 0];
num_X = [0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0];

fprintf('3.a:\n')
fprintf('P(letter=P|edges=1) = 0.5 | P and Q has 1 edge\n')
fprintf('P(letter=H|edges=4) = 1/3 | H K and X has 4 edges\n')
fprintf('P(letter=B|edges=0 and T=1) = 1 | only B has 0 edges and 1 T
 junction\n')

% B:
fprintf('3.b:\n')
fprintf('They will not be valid, corupted text can have totaly
 different number of Edges and T junctions.\n')

% Answers:

3.a:
P(letter=P|edges=1) = 0.5 | P and Q has 1 edge
P(letter=H|edges=4) = 1/3 | H K and X has 4 edges
P(letter=B|edges=0 and T=1) = 1 | only B has 0 edges and 1 T junction
3.b:
They will not be valid, corupted text can have totaly different number
 of Edges and T junctions.
```

Q 4

```
figure(4)
answer = imread('q4.jpg');
image(answer)
```



Q 5

```
% Code:
% A:
[mu1 sigma1] = group_density('cell1.bmp');
[mu2 sigma2] = group_density('cell2.bmp');

group_prob = @(xi, mu, sigma) exp(-((xi-mu)^2 / (2*sigma))) /
 (2*pi*sigma) ^ (1/2);

C = objects_features('test.bmp');

cell1 = 0;
cell2 = 0;

% B: Most Likelihood
for c = C
    cell1_prob = group_prob(c, mu1, sigma1);
    cell2_prob = group_prob(c, mu2, sigma2);

    if cell1_prob > cell2_prob
        cell1 = cell1 + 1;
    else
        cell2 = cell2 + 1;
    end
```

```
end

fprintf('5.b:\nAccording to %s: we have %g cell1 and %g
 cell2\n', 'most likelihood', cell1, cell2);

cell1 = 0;
cell2 = 0;

for c = C
    cell1_dist = (c - mu1) ^ 2;
    cell2_dist = (c - mu2) ^ 2;

    if cell1_dist < cell2_dist
        cell1 = cell1 + 1;
    else
        cell2 = cell2 + 1;
    end

end

fprintf('According to %s: we have %g cell1 and %g cell2\n', 'Smallest
 distance', cell1, cell2);

% Answers:

5.b:
According to most likelihood: we have 7 cell1 and 8 cell2
According to Smallest distance: we have 7 cell1 and 8 cell2
```

Used Functions:

```
function [ mu, sigma ] = group_density( file_name )

            C = objects_features(file_name);
            n = length(C);
            mu = sum(C) / n;
            sigma = sum((C-mu).^2) / n;

end

function [ C ] = objects_features( file_name )
%OBJECTS_FEATURES Summary of this function goes here
%   Detailed explanation goes here
img = imread(file_name);

[seg_im, vals] = segmentation(img);
C = [];
for val=vals(1 : end-1)
    c = circularity(seg_im, val);
    C = [C c];
end

end
```

```
function [ c ] = circularity( image, val )
% circularity Calculates circularity for segmented shape.
% c = circularity calculation.
% image = segmented image.
% val = segmented shape value.

area = length(find(image == val));
perim = myPerim(image, val);
c = (4*pi*area)/perim^2;
end

function [ perim ] = laplace_perim( image, val )

image(image ~= val) = 255;
dseg = double(image);
del_seg = del2(dseg);
perim = length(find(del_seg<0));
end

function [ perim ] = myPerim( image, val )
perim = 0;
[nrows, ncols] = size(image);
for x = 1:ncols
    for y = 1:nrows
        if image(y, x) == val
            k = 0;
            for i = -1:1
                for j = -1:1
                    if x+i < 1 || y+j < 1
                        continue
                    end
                    if image(y+j, x+i) == 255
                        k = k + 1;
                        break;
                    end
                end
                if k ~= 0
                    break
                end
            end
            if k ~= 0
                perim = perim + 1;
                k = 0;
            end
        end
    end
end
end

function [ seg_pic, vals ] = segmentation( image )

gray_pic = rgb2gray(image);
seg_pic = gray_pic;
[nrows, ncols] = size(gray_pic(:,:,1));
vals(1) = 1;
```

```
ind = 1;
for x = 2:ncols-1
    for y = 2:nrows-1
        if seg_pic(x,y) == 0
            seg_pic = region_grow(seg_pic, vals(ind), x, y);
            ind = ind + 1;
            vals(ind) = vals(ind-1) + 10;
        end
    end
end

function [gray_pic] = region_grow(gray_pic, val, x, y)
    gray_pic(x,y) = val;
    [nrows, ncols] = size(gray_pic);
    for i = -1:1
       for j= -1:1
            if x+i < 1 || y+j < 1
                continue
            end
            if x+i > nrows || y+j > ncols
                break
            end
            if gray_pic(x+i, y+j)==0
            gray_pic = region_grow(gray_pic, val, x+i, y+j);
            end
        end
    end
end
end
```

Published with MATLAB® R2017b