ECE-271A

Haonan Peng

A14765890

HW2 Report

1. From problem 2, the prior probability can be calculated by , where is the number of samples in the specific class, and is the total number of all samples in this question. Then we have:

The calculated results are the same as the results we obtained last week.

Chart, bar chart

Description automatically generated

1. Here are the 64 marginal densities for the two classes under Gaussian assumption. (The blue plots are features of the class of foreground, and the red plots are the features of the class of background)

Diagram, engineering drawing

Description automatically generatedDiagram, engineering drawing

Description automatically generatedDiagram, engineering drawing

Description automatically generatedDiagram, engineering drawing

Description automatically generatedDiagram, engineering drawing

Description automatically generated

By visual inspection, the best 8 features for classification are: [1,18,25,27,32,33,40,41]. The worst 8 features are: [3,4,5,59,60,62,63,64].

1. The two figures shown below are the prediction image of the “Cheetah” by using 1) all 64-dimensional Gaussian, and 2) the best 8-dimensional Gaussian.

Graphical user interface

Description automatically generated

Comparing to both predictions of the cheetah image, the performance of the 8-dimensional Gaussian with is obviously better than the one of the 64-dimensional Gaussian with . When counting the probabilities of features other than the best 8 features, the performance becomes worse, because the probabilities of two classes of the features are so close that can misguide the Bayesian decision by using

Code

%% a)

samples= load('TrainingSamplesDCT\_8\_new.mat');

FG = samples.TrainsampleDCT\_FG;

BG = samples.TrainsampleDCT\_BG;

PY\_FG = size(FG,1) / (size(FG,1)+size(BG,1));

PY\_BG = size(BG,1) / (size(FG,1)+size(BG,1));

h = [ones(size(FG,1),1);zeros(size(BG,1),1)];

hist = categorical(h,[1 0],{'Foreground','Background'});

figure(1)

histogram(hist);

title('The Histogram of the Estimate');

%% b)

mean\_FG = mean(FG);

mean\_BG = mean(BG);

cov\_FG = cov(FG);

cov\_BG = cov(BG);

MLE\_FG = zeros(1,size(FG,1));

MLE\_BG = zeros(1,size(BG,1));

% generate all 64 gaussian probability densities

for i = 1:4

figure(i+1)

for j = 1:16

sig\_FG = sqrt(cov\_FG((i-1)\*16+j,(i-1)\*16+j));

sig\_BG = sqrt(cov\_BG((i-1)\*16+j,(i-1)\*16+j));

m\_FG = mean\_FG((i-1)\*16+j);

m\_BG = mean\_BG((i-1)\*16+j);

x\_FG = (m\_FG-4\*sig\_FG):8\*sig\_FG/199:(m\_FG+4\*sig\_FG);

x\_BG = (m\_BG-4\*sig\_BG):8\*sig\_BG/199:(m\_BG+4\*sig\_BG);

md\_FG = exp(-(2\*sig\_FG^2)^(-1)\*(x\_FG-m\_FG).^2)/(sig\_FG\*sqrt(2\*pi));

md\_BG = exp(-(2\*sig\_BG^2)^(-1)\*(x\_BG-m\_BG).^2)/(sig\_BG\*sqrt(2\*pi));

subplot(4,4,j);

plot(x\_FG,md\_FG);

hold on

plot(x\_BG,md\_BG);

hold off

title(['Index ' num2str((i-1)\*16+j)]);

end

end

best = [1,18,25,27,32,33,40,41];

worst = [3,4,5,59,60,62,63,64];

% generate 8 best gaussian probability densities and 8 worst ones

figure(6)

for i = 1:8

sig\_FG\_b = sqrt(cov\_FG(best(i),best(i)));

sig\_BG\_b = sqrt(cov\_BG(best(i),best(i)));

sig\_FG\_w = sqrt(cov\_FG(worst(i),worst(i)));

sig\_BG\_w = sqrt(cov\_BG(worst(i),worst(i)));

m\_FG\_b = mean\_FG(best(i));

m\_BG\_b = mean\_BG(best(i));

m\_FG\_w = mean\_FG(worst(i));

m\_BG\_w = mean\_BG(worst(i));

x\_FG\_b = (m\_FG\_b-4\*sig\_FG\_b):8\*sig\_FG\_b/199:(m\_FG\_b+4\*sig\_FG\_b);

x\_BG\_b = (m\_BG\_b-4\*sig\_BG\_b):8\*sig\_BG\_b/199:(m\_BG\_b+4\*sig\_BG\_b);

x\_FG\_w = (m\_FG-4\*sig\_FG\_w):8\*sig\_FG\_w/199:(m\_FG+4\*sig\_FG\_w);

x\_BG\_w = (m\_BG-4\*sig\_BG\_w):8\*sig\_BG\_w/199:(m\_BG+4\*sig\_BG\_w);

md\_FG\_b = exp(-(2\*sig\_FG\_b^2)^(-1)\*(x\_FG\_b-m\_FG\_b).^2)/(sig\_FG\_b\*sqrt(2\*pi));

md\_BG\_b = exp(-(2\*sig\_BG\_b^2)^(-1)\*(x\_BG\_b-m\_BG\_b).^2)/(sig\_BG\_b\*sqrt(2\*pi));

md\_FG\_w = exp(-(2\*sig\_FG^2)^(-1)\*(x\_FG-m\_FG).^2)/(sig\_FG\*sqrt(2\*pi));

md\_BG\_w = exp(-(2\*sig\_BG^2)^(-1)\*(x\_BG-m\_BG).^2)/(sig\_BG\*sqrt(2\*pi));

subplot(4,4,i);

plot(x\_FG\_b,md\_FG\_b);

hold on

plot(x\_BG\_b,md\_BG\_b);

hold off

title(['Best: Index ' num2str(best(i))]);

subplot(4,4,i+8);

plot(x\_FG\_w,md\_FG\_w);

hold on

plot(x\_BG\_w,md\_BG\_w);

hold off

title(['Worst: Index ' num2str(worst(i))]);

end

%% c.

img= im2double(imread('cheetah.bmp'));

[row, colm] = size(img);

A\_64 = zeros(row,colm);

A\_8 = zeros(row,colm);

%read Zig-Zag Pattern.txt file

ZigZag = fopen('Zig-Zag Pattern.txt','r');

zzPat = fscanf(ZigZag,'%d',[8,8]);

fclose(ZigZag);

cov\_FG8 = cov(FG(:,best));

cov\_BG8 = cov(BG(:,best));

mean\_FG8 = mean\_FG(best);

mean\_BG8 = mean\_BG(best);

% using all 64 features

for i = 1:row-8

for j = 1:colm-8

dctImg = dct2(img(i:i+7,j:j+7));

zzScan= zeros([1, 64]);

for x = 1:8

for y = 1:8

zzScan(zzPat(x,y)+1) = dctImg(x,y);

end

end

PX\_FG = sqrt((2\*pi)^64\*det(cov\_FG))^(-1)\*exp(-(zzScan-mean\_FG)/cov\_FG\*(zzScan-mean\_FG)'/2);

PX\_BG = sqrt((2\*pi)^64\*det(cov\_BG))^(-1)\*exp(-(zzScan-mean\_BG)/cov\_BG\*(zzScan-mean\_BG)'/2);

PX\_FG8 = sqrt((2\*pi)^8\*det(cov\_FG8))^(-1)\*exp(-(zzScan(best)-mean\_FG8)/cov\_FG8\*(zzScan(best)-mean\_FG8)'/2);

PX\_BG8 = sqrt((2\*pi)^8\*det(cov\_BG8))^(-1)\*exp(-(zzScan(best)-mean\_BG8)/cov\_BG8\*(zzScan(best)-mean\_BG8)'/2);

if PX\_FG\*PY\_FG > PX\_BG\*PY\_BG

A\_64(i,j) = 1;

end

if PX\_FG8\*PY\_FG > PX\_BG8\*PY\_BG

A\_8(i,j) = 1;

end

end

end

ground\_truth = im2double(imread('cheetah\_mask.bmp'));

pred = padarray(A\_64, [4,4], 0);

pred8 = padarray(A\_8, [4,4], 0);

missFG = 0;

missBG = 0;

missFG8 = 0;

missBG8 = 0;

gtFG = 0;

gtBG = 0;

for i = 1:size(ground\_truth,1)

for j = 1:size(ground\_truth,2)

if ground\_truth(i,j) == 1

gtFG = gtFG + 1;

if pred(i,j) ~= ground\_truth(i,j)

missFG = missFG + 1;

end

if pred8(i,j) ~= ground\_truth(i,j)

missFG8 = missFG8 + 1;

end

else

gtBG = gtBG + 1;

if pred(i,j) ~= ground\_truth(i,j)

missBG = missBG + 1;

end

if pred8(i,j) ~= ground\_truth(i,j)

missBG8 = missBG8 + 1;

end

end

end

end

% Calculate error

errFG = missFG / gtFG \* PY\_FG;

errBG = missBG / gtBG \* PY\_BG;

errFG8 = missFG8 / gtFG \* PY\_FG;

errBG8 = missBG8 / gtBG \* PY\_BG;

err = errFG + errBG;

err8 = errFG8 + errBG8;

figure

subplot(1,2,1)

imagesc(A\_64);

colormap(gray(255));

title({['The Predicted Image with 64-Dimensional Gaussian '] ['Error = ' num2str(err)]});

subplot(1,2,2)

imagesc(A\_8);

colormap(gray(255));

title({['The Predicted Image with 8-Dimensional Gaussian '] ['Error = ' num2str(err8)]});