ECE 271A Statistical Learning HW5

A53268155 Lien-Hsi Lu

Introduction

In this homework, we keep on classifying a picture of cheetah and grass. But this time, we use the mixture model through EM algorithm to get the parameters and plug in to Bayes Decision Rule to solve this problem.

Results & Explanation:

a) **For each class, learn 5 mixtures of C=8 components, using a random initialization (recall that the mixture weights must add up to one). Plot the probability of error vs. dimension for each of the 25 classifiers obtained with all possible mixture pairs. Comment the dependence of the probability of error on the initialization.**

By using the EM algorithm on to the Gaussian Distribution model, we can parameters and plug them into BDR to classify foreground(grass) and background(cheetah). After random initialize the prior, mu and sigma, I start the EM algorithm for 30 iterations. In the E-step, we calculate h_{ij} base on the initialization parameters as below formula.

$$h_{ij} = \frac{G\left(x_i, \mu_j^n, \sigma_j^n\right)\pi_j^n}{\sum_k G(x_i, \mu_k^n, \sigma_k^n)\pi_k^n}$$

In M-step, we update the parameters by using the h_{ij} in the E-step and the formulas are like below.

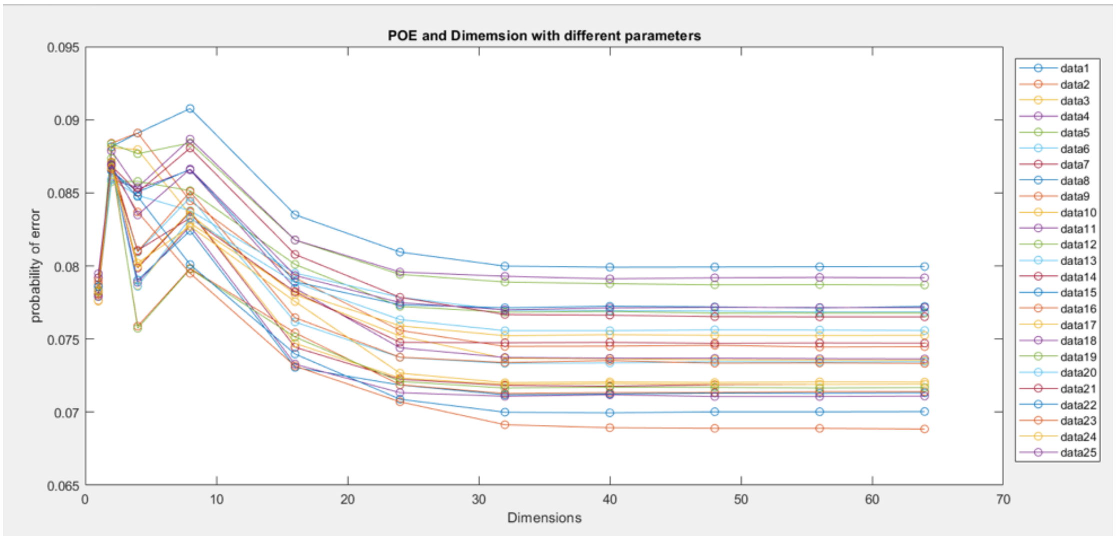
$$\mu_j^{n+1} = \frac{\sum_i h_{ij} x_i}{\sum_i h_{ij}}$$

$${\sigma_j^2}^{n+1} = \frac{\sum_i h_{ij}\left(x_i - \mu_j\right)^2}{\sum_i h_{ij}}$$

$$\pi_j^{n+1} = \frac{1}{n}\sum_i h_{ij}$$

After plugging those parameters into class-conditional probability before applying BDR, we finish our classification and calculate the probability of error by using different dimension of data.

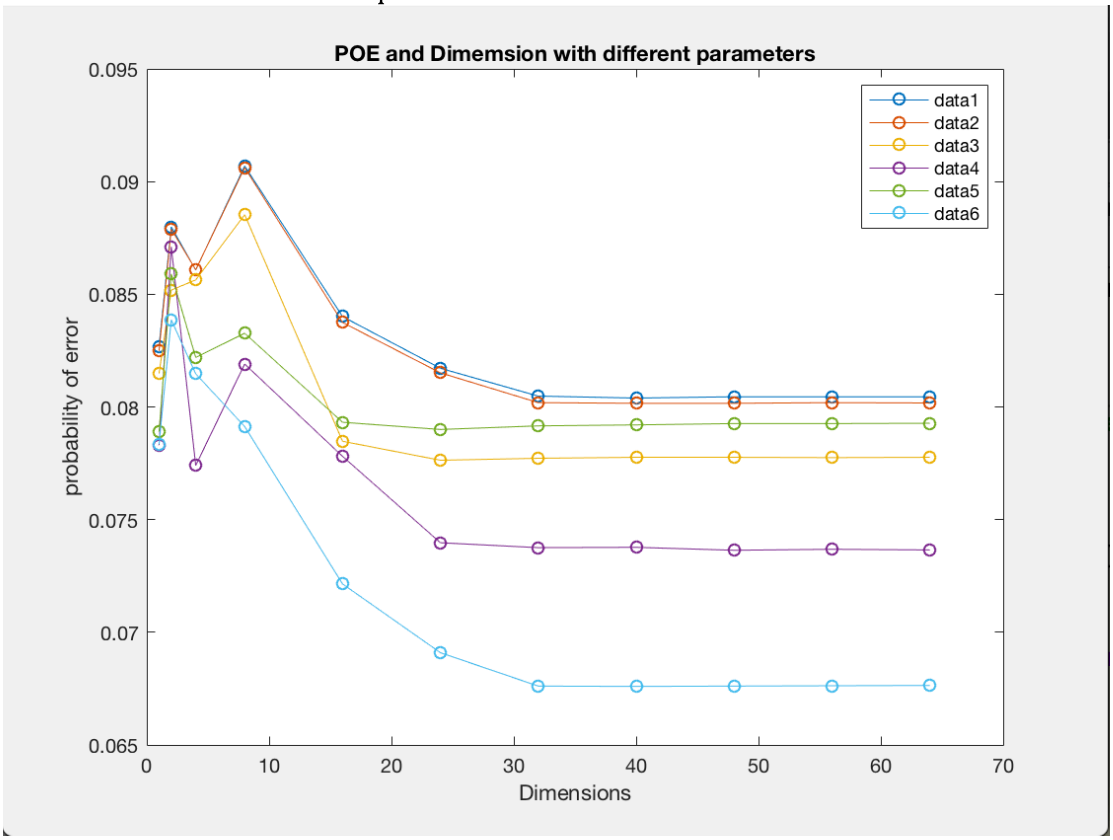
Because of learning 5 sets of parameters for foreground and background, we the number of combination is 25. So the result is a plot that its x axis is the dimensions and the y axis is the probability of error with 25 lines. Below is the plot of the result.



Based on my observation, the probability of error will first go up to about 9% in small dimensions of data. But when the dimension increases, the probability of error start to go down and are nearly the same when the dimensions are big (32, 40, 48……) for each curve. The probability is about 7.5 %~8% at last. We can see that mixture model by using EM algorithm really gets lower POE than the method we used in previous homework. There are some hidden state for the cheetah and the grass. First of all, although the initialization are different and random, the trend of the POE for 25 different classifier are very similar because of the information provided by each dimension. In addition, the POE doesn't change when using higher dimensions such as bigger than 40, the reason of this is because features at those dimensions provided less information than lower ones. At last, the difference between each classifiers get larger when the dimension increases, so the initialization seems very important when the feature gives enough information.

b) For each class, learn mixtures with C ∈ {1, 2, 4, 8, 16, 32}. Plot the probability of error vs dimension for each number of mixture components. What is the effect of the number of mixture components on the probability of error?

In this problem, we do only one set of foreground & background parameters but for the number of classes C ∈ {1, 2, 4, 8, 16, 32}. Again, we need to plot the POE and dimension plot and we can see the result in below.



Based on my observation, when the number of class = 1 obtains the largest POE. When the number of class = 8 and 32 obtains the lowest POE.
The reason for this is when class = 1 is the EM algorithm will actually be like a maximum likelihood problem. So, the result from this will be very similar to the homework we did before in ML. In addition, the result also tell us when the hidden state is 2 or more will obtain optimized result. So, when the number of C is more than the actual number of hidden classes, the result will be better.

```
clear all;
%% Training Part
load('./TrainingSamplesDCT_8_new.mat');
[sample_FG, dim] = size(TrainsampleDCT_FG);
[sample_BG, dim] = size(TrainsampleDCT_BG);

priorFG_cell = cell(5, 1);
priorBG_cell = cell(5, 1);
muFG_cell = cell(5, 1);
muBG_cell = cell(5, 1);
sigmaFG_cell = cell(5, 1);
sigmaBG_cell = cell(5, 1);
Class = 8;
Dim = 64;
Iter = 30;

%% Compute 5 sets of FG & BG parameters through EM algorithm (30
iterations)
Mixture = 5;
for M = 1:Mixture
    %E-step initialize
    priorFG = ones(1, Class)./Class;
    priorBG = ones(1, Class)./Class;
    muFG = rand(Class, Dim);
    muBG = rand(Class, Dim);
    diagFG = cell(Class, 1);
    diagBG = cell(Class, 1);
    for j = 1: Class
        temp = rand(1, Dim);
        temp(temp<0.001) = 0.001;
        diagFG{j} = diag(temp);
        temp = rand(1, Dim);
        temp(temp<0.001) = 0.001;
        diagBG{j} = diag(temp);
    end

    %E-step
    h_FG = zeros(sample_FG, Class);
```

```
    h_BG = zeros(sample_BG, Class);
    muFGprev = muFG;
    muBGprev = muBG;
    priorFGprev = priorFG;
    priorBGprev = priorBG;
    diagFGprev = diagFG;
    diagBGprev = diagBG;
    for i = 1: Iter
        i
        for j = 1: sample_FG
            for z = 1: Class
            %compute hij
                pdf = mvnpdf(TrainsampleDCT_FG(j, 1:Dim),
muFGprev(z, :), diagFGprev{z});
                h_FG(j, z) = pdf*priorFGprev(1, z);
            end
            h_FG(j, :) = h_FG(j, :)/sum(h_FG(j, :));
        end

        for z = 1: Class
           muFG(z, :) = h_FG(:, z)'*TrainsampleDCT_FG(:,
1:Dim)./sum(h_FG(:, z));
           priorFG(1, z) = sum(h_FG(:, z))/sample_FG;
        end

        for z = 1: Class
            sig_tmp = zeros(1, dim);
            sum_1 = 0;
            sum_2 = 0;
            for j = 1: sample_FG
                sum_1 = sum_1 + h_FG(j, z)* (TrainsampleDCT_FG(j,
1:Dim)-muFG(z, :)).^2;
                sum_2 = sum_2 + h_FG(j, z);

            end
            diag_tmp = sum_1/sum_2;
            diag_tmp(diag_tmp<0.002) = 0.002;
            diagFG{z} = diag(diag_tmp);
```

```
        end
        if max(max(abs(muFG - muFGprev)./abs(muFGprev))) < 0.01
          break;
        end

        muFGprev = muFG;
        priorFGprev = priorFG;
        diagFGprev = diagFG;
    end

    for i = 1: Iter
          i
          for j = 1: sample_BG
              for z = 1: Class
              %compute hij
                  pdf = mvnpdf(TrainsampleDCT_BG(j, 1:Dim),
muBGprev(z, :), diagBGprev{z});
                  h_BG(j, z) = pdf*priorBGprev(1, z);
              end
              h_BG(j, :) = h_BG(j, :)/sum(h_BG(j, :));
          end
          %BG

          for z = 1: Class
             muBG(z, :) = h_BG(:, z)'*TrainsampleDCT_BG(:,
1:Dim)./sum(h_BG(:, z));
             priorBG(1, z) = sum(h_BG(:, z))/sample_BG;
          end

          for z = 1: Class
              sum_1 = 0;
              sum_2 = 0;
              for j = 1: sample_BG
                  sum_1 = sum_1 + h_BG(j, z)* (TrainsampleDCT_BG(j,
1:Dim)-muBG(z, :)).^2;
                  sum_2 = sum_2 + h_BG(j, z);
              end
              diag_tmp = sum_1/sum_2;
```

```
            diag_tmp(diag_tmp<0.002) = 0.002;
            diagBG{z} = diag(diag_tmp);
        end
        if max(max(abs(muBG - muBGprev)./abs(muBGprev))) < 0.01
          break;
        end

        muBGprev = muBG;
        priorBGprev = priorBG;
        diagBGprev = diagBG;

    end
    priorFG_cell{M} = priorFG;
    priorBG_cell{M} = priorBG;
    muFG_cell{M} = muFG;
    muBG_cell{M} = muBG;
    sigmaFG_cell{M} = diagFG;
    sigmaBG_cell{M} = diagBG;
end

%% prior probability
 P_BG = size(sample_BG,1)/(size(sample_BG,1) + size(sample_FG,1));
 P_FG = 1-P_BG;

%% Testing part
%input bmp file & ZigZag pattern
%store matrix_dct into zigzag which is from 0~63 so plus one to each
number
A = im2double(imread('cheetah.bmp'));
Z = load('Zig-Zag Pattern.txt');
Z = Z+1;
B = padarray(A,[7,7],'symmetric','post');% original size 255*270 --->
need to use padarray() to fill 255+7, 270+7
[q,l] = size(B);
matrix_zigzag=[];
mask = zeros(q,l);
dimension = [1,2,4,8,16,24,32,40,48,56,64];
feature_map = zeros(255*270, 64);
```

```
% sliding window
count = 1;
 for i=1:q-7
    for j=1:l-7
        matrix_dct2 = dct2(B(i:i+7,j:j+7));
        matrix_zigzag(Z)= matrix_dct2;
        matrix_zigzag=reshape(matrix_zigzag,[1,64]);
        feature_map(count,:) = matrix_zigzag;
        count=count+1;
    end
 end

% count # of FG & BG from the original mask
groundtruth_mask = im2double(imread('cheetah_mask.bmp'));
count_grass = 0;
count_cheeta = 0;
for o=1:(q-7)
    for p=1:(l-7)
        if groundtruth_mask(o,p) == 1
            count_cheeta = count_cheeta + 1;
        else
            count_grass = count_grass + 1;
        end
     end
end

%plug in the parameters of 5 FG & BG mixture models to mvnpdf --->
BDR
error_vector=[];
for n1 = 1:2
    n1
    for n2 = 1:2
        n2
        poe_error=[];
        for dim = dimension
            dim
            count = 1;
```

```
          PI_BG = priorBG_cell{n1};
          MU_BG = muBG_cell{n1};
          SIGMA_BG = sigmaBG_cell{n1};
          PI_FG = priorFG_cell{n2};
          MU_FG = muFG_cell{n2};
          SIGMA_FG = sigmaFG_cell{n2};
          for i=1:255
              for j=1:270
                  Pxy_xgrass = 0;
                  Pxy_xcheeta = 0;
                      for c = 1:Class
                          SIGMABG=SIGMA_BG{c};
                          G_grass = mvnpdf((feature_map(count,1:dim)),
MU_BG(c,1:dim), SIGMABG(1:dim,1:dim));
                          Pxy_xgrass = Pxy_xgrass + PI_BG(1,c)*G_grass;
                          SIGMAFG=SIGMA_FG{c};
                          G_cheeta = mvnpdf((feature_map(count,1:dim)),
MU_FG(c,1:dim), SIGMAFG(1:dim,1:dim));
                          Pxy_xcheeta= Pxy_xcheeta +
PI_FG(1,c)*G_cheeta;
                      end

                      if  (Pxy_xcheeta*P_FG) >  (Pxy_xgrass*P_BG)
                          mask(i,j) = 1;
                      else% Pxy_xcheeta*p_foreground <
Pxy_xgrass*p_background
                          mask(i,j) = 0;
                      end
                  count=count+1;
              end
          end
%Error Rate
error = [];

diff_grass = 0;
diff_cheeta = 0;
for o=1:(q-7)
    for p=1:(l-7)
```

```
            if mask(o,p) == 0
                if groundtruth_mask(o,p) == 1
                    diff_cheeta = diff_cheeta + 1;
                end
            elseif mask(o,p) == 1
                if groundtruth_mask(o,p) == 0
                    diff_grass = diff_grass + 1;
                end
            end
         end
    end
                error_cheeta = (diff_cheeta / count_cheeta)*P_FG ;
                error_grass = (diff_grass / count_grass)*P_BG;
                error = error_cheeta + error_grass;

                poe_error = [poe_error error];
            end
            error_vector = [error_vector; poe_error];
        end
    end

%% plot POE and Dimemsion with different parameters
[row,col]=size(error_vector);
figure;
for r = 1:row
    plot(dimension,error_vector(r,:),'-o');
    hold on;
end
title('POE and Dimemsion with different parameters')
xlabel('Dimensions');
ylabel('probability of error');
legend;
hold off;
```