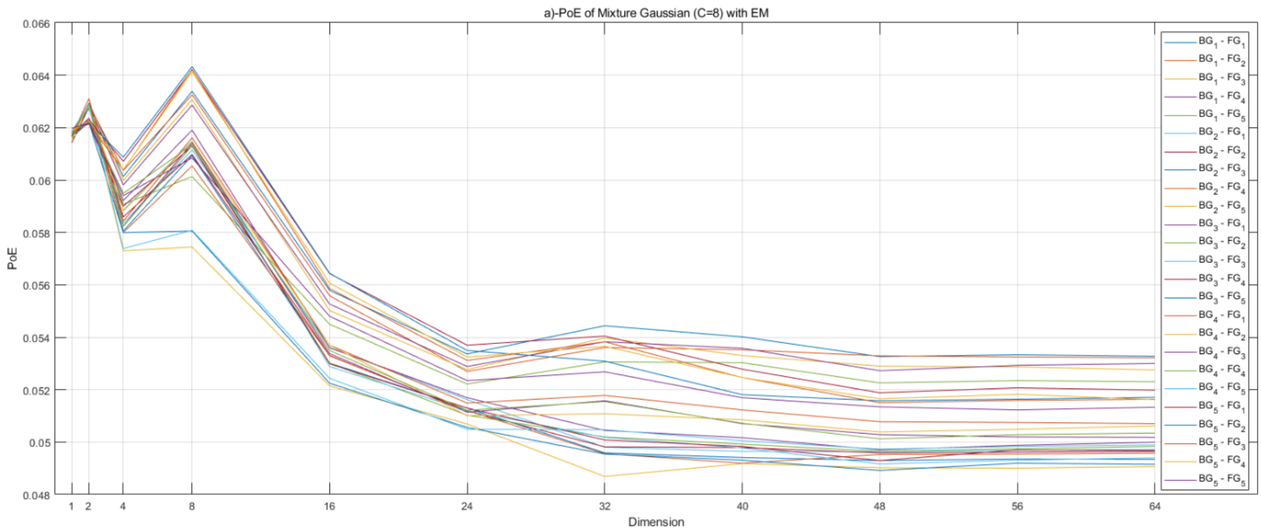
[ECE 271A] Homework5 Report

Yang Yue | A53301503

Learning 5 Gaussian mixtures with C = 8 components

i) Plot of PoE vs dimension



ii) Comments to the result

The plot of the error rate of multi-gaussian classifier with various random initializations vs. the dimension of test data is shown above. The plot implies a generally decreasing trend of error rate with the increasing dimensions.

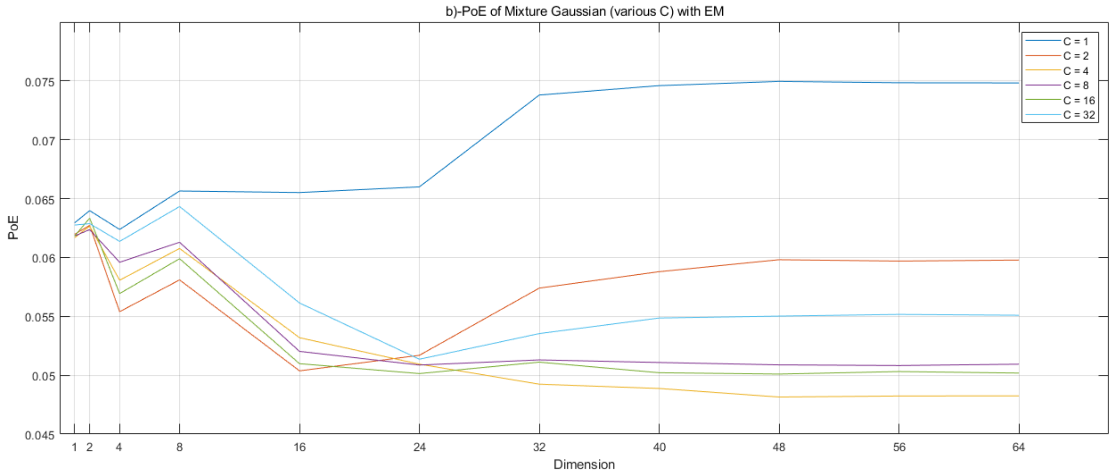
As 1 dimension is used, the PoEs with different initials are close, which implies that the random-initialized parameters of 1-D multi-gaussian of training set is converged to the same values.

With more dimensions of training data involved in parameter estimation, it is intuitively correct that the PoE curve will decline as the dimension increases. While the result shows that there exist some disturbances of the decreasing trend of the curve. This uniform disturbance occurs when dimension equals 2 and 8. And since the parameters are trained by all 64 dimensions of training data, it may be caused by the information from the 2^{nd} and the 5^{th} to 8^{th} dimensions are not "helpful" for us to distinguish where the test data belong to. Then the disturbance effect is weakened because the status of each dimension is no more outstanding, which causes the decreasing more stable.

Also, it can be seen that the distance between each PoE curve is getting large as the dimension increases. This may be due to the random initialized parameters falls to different local optimized areas. The mix-gaussian model is getting complicated will the data dimension increase, so that the local optimized parameter values for every initialized parameter vary even though there exist only tiny difference between each parameter.

Learning Gaussian mixtures with various Cs

i) Plot of PoE vs dimensions



ii) Explanation of the effect of number of mixture components

The plot of PoE of mixture gaussian vs different numbers of mixture components is shown above. It is noticed that the trend of curve of number of mixture component $C = 1$ is obviously different from others. Since when number of C is too small, it may lead to bias of estimation. Thus, it provides a poor performance of estimating the classification.

We can also observe that the PoE raises again when $C = 32$, this may due to the overfit of training data.

For other cases, the classifier should perform better, and the probability distribution can be fitted more accurately. However, the parameters are randomly initialized at first, the classification result may be disturbed by that fact. It may be concluded that the best number of components may be in the range of $[2,16]$.

Appendix

Matlab Source Code

randomInit.m

```
function [pi_Init, mu_Init, cov_Init] = randomInit(dim, C)
    pi_Init = rand(C,1);
    pi_Init = pi_Init./sum(pi_Init);
    mu_Init = rand(C,dim)-1;
    cov_Init = rand(C,dim)+1;
%     cov_Init(cov_Init<0.0001) = 0.0001;
end
```

EMcalc.m

```
function [pi_fin,mu_fin,cov_fin] = EMcalc(dim,C,xi,pi_init,mu_init,cov_init,num)
% initialize for the first round iteration
pi = pi_init;
mu = mu_init;
cov = cov_init;
h = zeros(size(xi,1),C);

for i_iter = 1:1000
    for j = 1:C
        h(:,j) = mvnpdf(xi,mu(j,:),diag(cov(j,:)))*pi(j);
    end
    h = h./repmat(sum(h,2),1,C);
%     save(['save_h_data/',num2str(num),'_iter_',num2str(i_iter),'.mat'],'h');
    sum_h = sum(h,1);
    pi_next = sum_h'/size(xi,1);
    for j = 1:C
        mu_next(j,:) = sum(repmat(h(:,j),1,dim).*xi,1)/sum_h(j);
        cov_next(j,:) = sum(repmat(h(:,j),1,dim).*(xi-
repmat(mu(j,:),size(xi,1),1)).^2,1)/sum_h(j);
        cov_next(cov_next<0.0001) = 0.0001;
    end
    pi = pi_next;
    mu = mu_next;
    cov = cov_next;
end
pi_fin = pi;
mu_fin = mu;
cov_fin = cov;
end
```

cheetah_BDR.m

```
function cheetah_vec = cheetah_BDR(blocks,C,dim,mu,cov,pi,pri)
    cheetah_vec = 0;
    for i = 1:C
        cheetah_vec = cheetah_vec +
pi(i)*mvnpdf(blocks,repmat(mu(i,1:dim),size(blocks,1),1),diag(cov(i,1:dim)));
    end
    cheetah_vec = cheetah_vec*pri;
end
```

plotPoE.m

```
%% a)
figure('Name','PoE of Mixture Gaussian (C=8) with EM')
str_legend = [];
for i = 1:5
    for j = 1:5
        plot(dim,poe_1{i,j});
        xlabel('Dimension')
        xticks(dim)
        ylabel('PoE')
        str_legend = [str_legend,'BG',num2str(i),'-FG',num2str(j)];
        hold on
    end
end
xticks(dim)
title('a)-PoE of Mixture Gaussian (C=8) with EM')
legend('BG_1 - FG_1','BG_1 - FG_2','BG_1 - FG_3','BG_1 - FG_4','BG_1 - FG_5',...
       'BG_2 - FG_1','BG_2 - FG_2','BG_2 - FG_3','BG_2 - FG_4','BG_2 - FG_5',...
       'BG_3 - FG_1','BG_3 - FG_2','BG_3 - FG_3','BG_3 - FG_4','BG_3 - FG_5',...
       'BG_4 - FG_1','BG_4 - FG_2','BG_4 - FG_3','BG_4 - FG_4','BG_4 - FG_5',...
       'BG_5 - FG_1','BG_5 - FG_2','BG_5 - FG_3','BG_5 - FG_4','BG_5 - FG_5');
grid()
%% b)
figure('Name','PoE of Mixture Gaussian (various C) with EM')
for i = 1:6
    plot(dim,poe_2(i,:))
    xlabel('Dimension')
    ylabel('PoE')
    hold on
end
legend('C = 1','C = 2','C = 4','C = 8','C = 16','C = 32');
xticks(dim)
grid()
title('b)-PoE of Mixture Gaussian (various C) with EM')
```

HW5.mat

```
%% Step1:initialization, calculate DCT and do the zigzag transformation
clc;clear;
% Once again we use the decomposition
% into 8 × 8 image blocks, compute the DCT of each block, and zig-zag scan.
```

```
zigzag = load('Zig-Zag Pattern.txt');
zigzag = reshape(zigzag, 1, []) + 1;

cheetah_img = imread('cheetah.bmp');
cheetah_dw = im2double(cheetah_img);

cheetah_mask = imread('cheetah_mask.bmp');
cheetah_maskdw = im2double(cheetah_mask);

%set a blank padding
% cheetah_pad = [cheetah_dw, zeros([size(cheetah_dw,1),7]);
zeros([7,size(cheetah_dw,2)+7])];
[img_row, img_col] = size(cheetah_dw);
% cheetah_blocks = zeros(img_row*img_col,64);
cheetah_blocks = zeros((img_row-7)*(img_col-7),64);
cnt = 1;

for col = 1:img_col-7
    for row = 1:img_row-7
%         window = cheetah_pad(row:row+7,col:col+7);
        window = cheetah_dw(row:row+7,col:col+7);
        cheetah_blocks(cnt,:) = reshape(dct2(window),[],64);
        cnt = cnt+1;
    end
end
cheetah_blocks = dozigzag(cheetah_blocks,zigzag);
clear cheetah_img cheetah_mask cnt window cheetah_pad zigzag row col
%% 1 - initialization
load('TrainingSamplesDCT_8_new.mat');
priBG =
size(TrainsampleDCT_BG,1)/(size(TrainsampleDCT_BG,1)+size(TrainsampleDCT_FG,1));
priFG = 1-priBG;
dim = [1,2,4,8,16,24,32,40,48,56,64];
times_1 = 5;
%% Step-1 BG
C1 = 8;
muBG = cell(times_1,size(C1,2));
covBG = cell(times_1,size(C1,2));
piBG = cell(times_1,size(C1,2));
count = 1;
for dim_i = 64
    for time_i = 1:times_1
        % Initialize the BG's pre-requisites
        xi_BG = TrainsampleDCT_BG(:,1:dim_i);
        [pi_init_BG,mu_init_BG,cov_init_BG] = randomInit(dim_i,C1);
        [pi,mu,cov] =
EMcalc(dim_i,C1,xi_BG,pi_init_BG,mu_init_BG,cov_init_BG,1);

        muBG{time_i,count} = mu;
        covBG{time_i,count} = cov;
        piBG{time_i,count} = pi;
    end
    count = count + 1;
end
clear mu pi cov count dim_i xi_BG pi_init_BG mu_init_BG cov_init_BG time_i
%% Step-1 FG
muFG = cell(times_1,size(C1,2));
covFG = cell(times_1,size(C1,2));
```

```
piFG = cell(times_1,size(C1,2));
count = 1;
for dim_i = 64
    for time_i = 1:times_1
        % Initialize the FG's pre-requisites
        xi_FG = TrainsampleDCT_FG(:,1:dim_i);
        [pi_init_FG,mu_init_FG,cov_init_FG] = randomInit(dim_i,C1);
        [pi,mu,cov] =
EMcalc(dim_i,C1,xi_FG,pi_init_FG,mu_init_FG,cov_init_FG,2);

        muFG{time_i,count} = mu;
        covFG{time_i,count} = cov;
        piFG{time_i,count} = pi;
    end
    count = count + 1;
end
clear mu pi cov count dim_i xi_FG pi_init_FG mu_init_FG cov_init_FG time_i
%%
poe_1 = cell(5);
for dim_i = 1:size(dim,2)
    test_set = cheetah_blocks(:,1:dim(dim_i));
    for i = 1:times_1
        Px_BG =
cheetah_BDR(test_set,C1,dim(dim_i),muBG{i},covBG{i},piBG{i},priBG);
        for j = 1:times_1
            Px_FG =
cheetah_BDR(test_set,C1,dim(dim_i),muFG{j},covFG{j},piFG{j},priFG);
            cheetah_vec = Px_FG./Px_BG;
            cheetah_vec(cheetah_vec>1) = 1;
            cheetah_vec(cheetah_vec~=1) = 0;
            cheetah_res = reshape(cheetah_vec,img_row-7,img_col-7);
            cheetah_res_pad = [cheetah_res,zeros([size(cheetah_res,1),7]);
zeros([7,size(cheetah_res,2)+7])];
            poe_1{i,j}(dim_i) = sum(abs(cheetah_maskdw-
cheetah_res_pad),'all')/img_row/img_col;
        end
    end
end
clear i j Px_BG Px_FG cheetah_vec
%%
figure
imagesc(cheetah_res_pad)
colormap('gray')

%% Step-2 BG
times_2 = 1;
C2 = [1,2,4,8,16,32];

%%
muBG = cell(times_2,size(C2,2));
covBG = cell(times_2,size(C2,2));
piBG = cell(times_2,size(C2,2));
count = 1;
for dim_i = 64
    for time_i = 1:times_2
        for c_i = C2
            % Initialize the BG's pre-requisites
            xi_BG = TrainsampleDCT_BG(:,1:dim_i);
```

```matlab
            [pi_init_BG,mu_init_BG,cov_init_BG] = randomInit(dim_i,c_i);
            [pi,mu,cov] =
EMcalc(dim_i,c_i,xi_BG,pi_init_BG,mu_init_BG,cov_init_BG,1);

            muBG{time_i,count} = mu;
            covBG{time_i,count} = cov;
            piBG{time_i,count} = pi;
            count = count + 1;
        end
    end
end
clear mu pi cov count dim_i xi_BG pi_init_BG mu_init_BG cov_init_BG time_i

%% Step-2 FG
muFG = cell(times_2,size(C2,2));
covFG = cell(times_2,size(C2,2));
piFG = cell(times_2,size(C2,2));
count = 1;
for dim_i = 64
    for time_i = 1:times_2
        for c_i = C2
            % Initialize the FG's pre-requisites
            xi_FG = TrainsampleDCT_FG(:,1:dim_i);
            [pi_init_FG,mu_init_FG,cov_init_FG] = randomInit(dim_i,c_i);
            [pi,mu,cov] =
EMcalc(dim_i,c_i,xi_FG,pi_init_FG,mu_init_FG,cov_init_FG,2);

            muFG{time_i,count} = mu;
            covFG{time_i,count} = cov;
            piFG{time_i,count} = pi;
            count = count + 1;
        end
    end
end
clear mu pi cov count dim_i xi_BG pi_init_BG mu_init_BG cov_init_BG time_i

%%
poe_2 = zeros(size(C2,2),size(dim,2));
for dim_i = 1:size(dim,2)
    test_set = cheetah_blocks(:,1:dim(dim_i));
    for c_i = 1:size(C2,2)
        for i = 1:times_2
            Px_BG =
cheetah_BDR(test_set,C2(c_i),dim(dim_i),muBG{c_i},covBG{c_i},piBG{c_i},priBG);
            Px_FG =
cheetah_BDR(test_set,C2(c_i),dim(dim_i),muFG{c_i},covFG{c_i},piFG{c_i},priFG);
            cheetah_vec = Px_FG./Px_BG;
            cheetah_vec(cheetah_vec>1) = 1;
            cheetah_vec(cheetah_vec~=1) = 0;
%                 cheetah_res = reshape(cheetah_vec,img_row,img_col);
            cheetah_res = reshape(cheetah_vec,img_row-7,img_col-7);
            cheetah_res_pad = [cheetah_res,zeros([size(cheetah_res,1),7]);
zeros([7,size(cheetah_res,2)+7])];
            poe_2(c_i,dim_i) = sum(abs(cheetah_maskdw-
cheetah_res_pad),'all')/img_row/img_col;
        end
    end
end
```

```
clear i j Px_BG Px_FG cheetah_vec

%%
figure
imagesc(cheetah_res_pad)
colormap('gray')

%%
figure
plot(dim,poe_2(1,:))

%%
format long
t = datetime('now');
save(['poe_save/poe_1_',datestr(now,30),'.mat'],'poe_1');
save(['poe_save/poe_2_',datestr(now,30),'.mat'],'poe_2');
```