HW5_271A

Reya Sadhu
A59026753

We assume the likelihood distribution is a mixture of gaussian distributions.

$$P_{x|i}(x|i) = \sum_{c=1}^{C} G(x, \mu_c, \Sigma_c)\pi_c$$

Using the EM algorithm for determining μ_c, Σ_c, π_c
E step:
$Q(\psi; \psi^{(n)}) = h_{ij} \log [E_{X|Z}(x_i|e_j, \psi)\pi_j, where\ h_{ij} = P_{Z|X}(e_j|x_i; \psi^{(n)})$
M step:
$\psi^{(n+1)} = \underset{\psi}{\arg max}\, h_{ij} \log [E_{X|Z}(x_i|e_j, \psi)\pi_j$

When x follows a gaussian distribution and we have N samples of x, we update the parameters using the following equation at each step.

$$h_{ij} = \frac{G\left(x_i, \mu_j^{(n)}, \Sigma_j^{(n)}\right)\pi_j^{(n)}}{\sum_{c=1}^{C} G\left(x_i, \mu_c^{(n)}, \Sigma_c^{(n)}\right)\pi_c^{(n)}}$$

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

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

$$\Sigma_j^{(n+1)} = \frac{\sum_i h_{ij}(x_i - \mu_j^{(n)})^2}{\sum_i h_{ij}}, this\ is\ a\ diagonal\ matrix\ of\ size\ i \times i.$$

For simplification, I have initialized the parameters randomly.

For $\pi_c^{(0)}$, I have taken random numbers from a uniform distribution between 1 to 10. As the sum of the weights should be 1, the numbers are then divided by their sum.

For, $\mu_c^{(0)}$, I have taken random samples from the distribution of x.

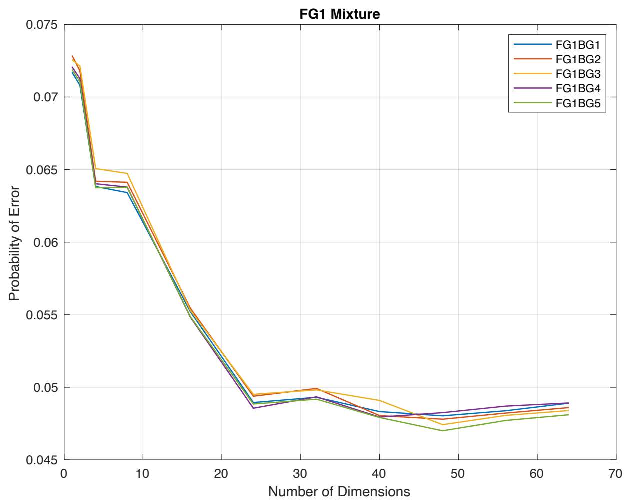
For, $\Sigma_c^{(0)}$, I have taken random numbers from a normal distribution with mean 5 and sigma 0.3 (They can be anything, but we need to make sure that they are not zero) and make a diagonal matrix with them.

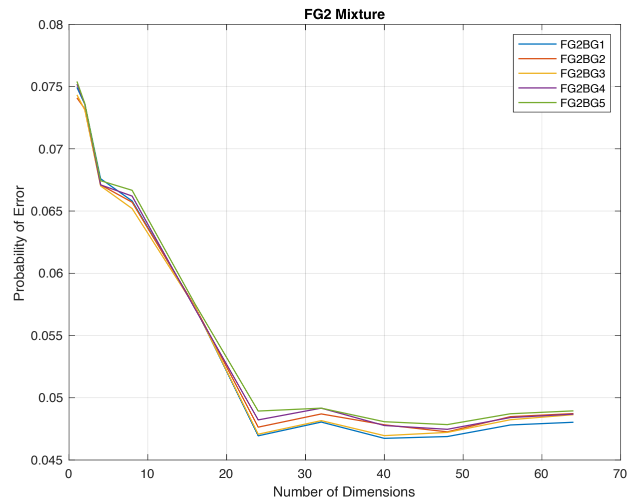
At each update, to make sure the updated covariance matrix is positive semidefinite, I have added a small value 1e-6 to each diagonal value.

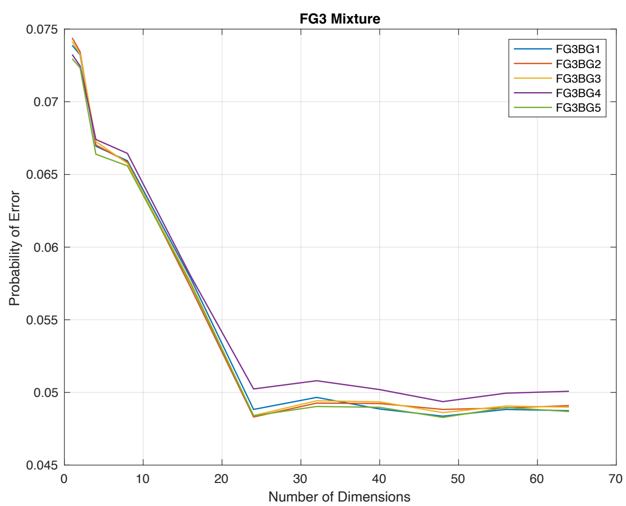
After we get the estimated μ_c, Σ_c, π_c for each class, we can add the each component value to get $P_{X|i}(x|i)$. Then, we apply the BDR,

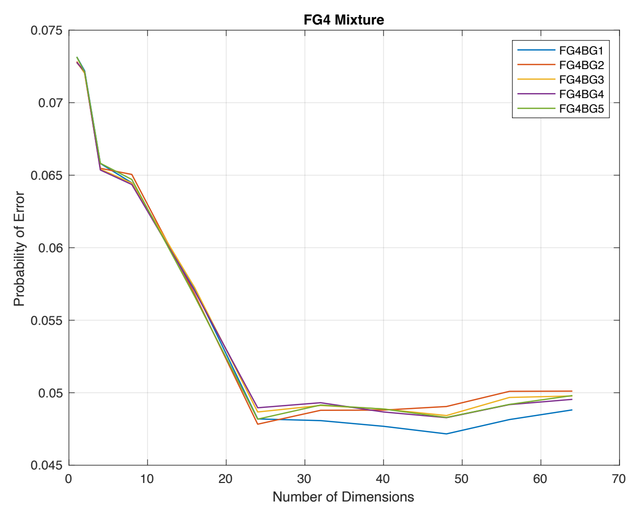
$$i^* = \underset{i}{\arg max}\, P_{i|X}(i|x) = \underset{i}{\arg max}\, P_{X|i}(x|i)P_i(i)$$

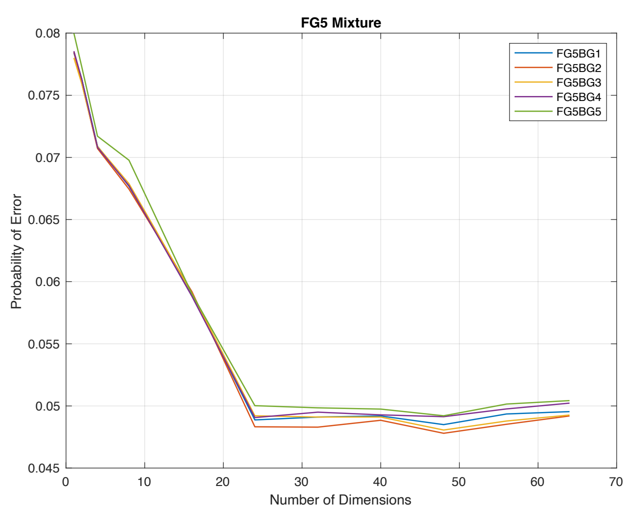
a) We have 5 foreground mixtures, each of then initiated randomly. For each of the 5 mixtures, we have 5 background mixtures of randomly initialized parameters. The graphs shown the probability of error vs dimensions for 25 classifiers.







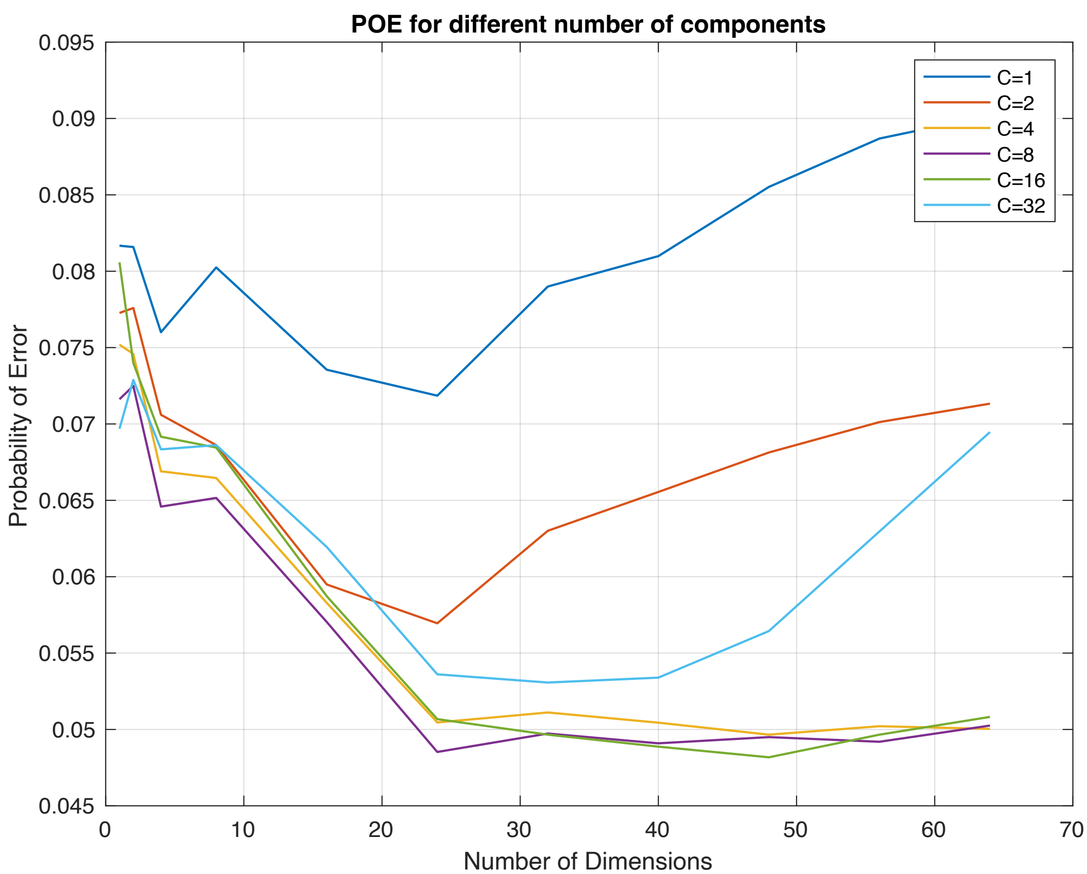




We can see, the combination of different foreground and background mixtures makes different models, as they are randomly initialized. The only trend we can observe is that no matter what initial parameters we use, the error decreases with the increase in dimensions. But also, after a certain value of dimensions it started to increase. This is similar to the HW2, where we have seen best 8 features work better than the all 64 features. More features can introduce confusions to the decision rule, for which the error increases.
Here, we can see the for the dimensions between 24 to 48, the PoE reaches its minimum.

b) In part b we change the number components for the mixture. As we can see if we for c=1, i.e., if we consider the likelihood to be made of only 1 gaussian, that has significantly more error than the other component numbers. So, the likelihood is definitely not made of any 1 component.

   For c=4,8, or 16, the error is the lowest. So, using one of these values can give us the best segmentation result. However, we cannot tell the precise optimal number for the number of components from this single experiment.



Code:

```
load('/Users/reyasadhu/Desktop/Masters/ECE 271A Statistical
Learning/HW4/TrainingSamplesDCT_8_new.mat');
num_FG = size(TrainsampleDCT_FG, 1);
num_BG = size(TrainsampleDCT_BG, 1);
p_FG=num_FG/(num_FG+num_BG);
p_BG=num_FG/(num_FG+num_BG);

I = imread('/Users/reyasadhu/Downloads/homework1/cheetah.bmp');
I=im2double(I);
[row,col]=size(I);
A=zeros((row-7)*(col-7),64);
index=1;

for i = 1:row - 7
    for j = 1:col - 7
        block = I(i:i+7, j:j+7);
        block_dct = dct2(block);
        dct_flat=zigzag(block_dct);
        A(index,:)=dct_flat;
        index=index+1;
    end
end


% Part a
C=8;
dim= [1,2,4,8,16,24,32,40,48,56,64];
error_mat=zeros(25,11);
for i=1:1
    [pi_FG,mu_FG,cov_FG]=EM(8,TrainsampleDCT_FG,1000);
    for j=1:1
        [pi_BG,mu_BG,cov_BG]=EM(8,TrainsampleDCT_BG,1000);
        for d=1:1
            mask=zeros(size(A,1),1);
            for k=1:size(A,1)
                prob_bg=0;
                prob_fg=0;
                for l=1:C

prob_bg=prob_bg+mvnpdf(A(k,1:dim(d)),mu_BG(l,1:dim(d)),cov_BG(1:dim(d),1:dim(d),l))*pi_BG(l);

prob_fg=prob_fg+mvnpdf(A(k,1:dim(d)),mu_FG(l,1:dim(d)),cov_FG(1:dim(d),1:dim(d),l))*pi_FG(l);
                end
                if(prob_fg*p_FG>prob_bg*p_BG)
                    mask(k)=1;
                end
            end
            mask=reshape(mask,263,248)';
            mask_resized=zeros(255,270);
            mask_resized(1:248,1:263)=mask;
            error_mat(5*(i-1)+j,d)=error(mask_resized);
        end
    end
end

figure;
for j=1:5
    subplot(3,2,j);
    for i = 5*(j-1)+1:5*(j-1)+5
        plot(dim,error_mat(i, :), 'LineWidth', 1);
        hold on;
    end
    xlabel('Number of Dimensions');
    ylabel('Probability of Error');
    title(sprintf('FG%d Mixture',j));
```

```
legend(sprintf('FG%dBG1',j),sprintf('FG%dBG2',j),sprintf('FG%dBG3',j),sprintf('FG%dBG4',j),sprintf(
'FG%dBG5',j));
    grid on;
    hold off;
end


% Part b
mixtures=[1,2,4,8,16,32];
error_mat_b=zeros(length(mixtures),length(dim));
for c=1:length(mixtures)
    [pi_FG,mu_FG,cov_FG]=EM(mixtures(c),TrainsampleDCT_FG,1000);
    [pi_BG,mu_BG,cov_BG]=EM(mixtures(c),TrainsampleDCT_BG,1000);
    for d=1:length(dim)
        disp(c);
        disp(d);
        mask=zeros(size(A,1),1);
        for i=1:size(A,1)
            prob_bg=0;
            prob_fg=0;
            for j=1:mixtures(c)

prob_bg=prob_bg+mvnpdf(A(i,1:dim(d)),mu_BG(j,1:dim(d)),cov_BG(1:dim(d),1:dim(d),j))*pi_BG(j);

prob_fg=prob_fg+mvnpdf(A(i,1:dim(d)),mu_FG(j,1:dim(d)),cov_FG(1:dim(d),1:dim(d),j))*pi_FG(j);
            end
            if(prob_fg*p_FG>prob_bg*p_BG)
                mask(i)=1;
            end
        end
        mask=reshape(mask,263,248)';
        mask_resized=zeros(255,270);
        mask_resized(1:248,1:263)=mask;
        error_mat_b(c,d)=error(mask_resized);
    end
end

figure;
for i = 1:6
    plot(dim,error_mat_b(i, :), 'LineWidth', 1);
    hold on;
end
xlabel('Number of Dimensions');
ylabel('Probability of Error');
title('POE for different number of components');
legend('C=1', 'C=2', 'C=4', 'C=8', 'C=16', 'C=32');
grid on;
hold off;

% Execute EM algorithm for given numstep
function [pi_c,mu,cov]=EM(C,sample,numstep)
    % Initialization
    n=size(sample,1);
    pi_c= randi([1,10], C,1);
    pi_c= pi_c / sum(pi_c);
    mu=sample(randi([1 200],1,C),:);
    cov=zeros(64,64,C);
    for i=1:C
        cov_temp = normrnd(5, 0.3, 1, 64);
        cov(:,:,i) = diag(cov_temp);
    end


    hij=zeros(n,C);
    for step=1:numstep
        %E step
        for i=1:C
```

```
            hij(:,i)=mvnpdf(sample,mu(i,:),cov(:,:,i))*pi_c(i);
        end
        hij=hij./sum(hij,2);

        %M step
        pi_c=sum(hij)/n;
        mu=hij'*sample./sum(hij)';

        for c=1:C
            cov(:,:,c)=diag(diag(((sample-mu(c,:))'.*hij(:,c)'*(sample-
mu(c,:))./sum(hij(:,c),1))+1e-6));
        end

    end
end


function output= zigzaged(input)
    zigzag=importdata('/Users/reyasadhu/Downloads/homework1/Zig-Zag Pattern.txt') ;
    zigzag=zigzag+1;
    output=zeros(1,64);
    for i=1:8
        for j=1:8
            output(zigzag(i,j))=input(i,j);
        end
    end
end

function poe=error(image)
    im_test = imread('/Users/reyasadhu/Downloads/homework1/cheetah_mask.bmp');
    im_test=im2double(im_test);
    err=abs(im_test-image);
    poe=sum(err,"all")/(255*270);
end
```