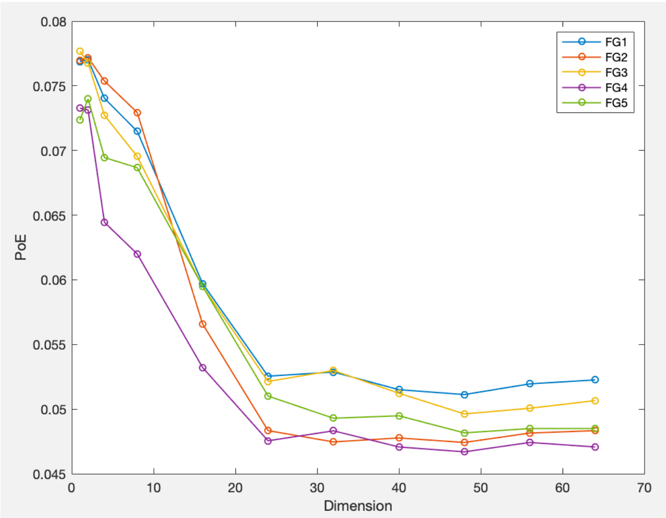
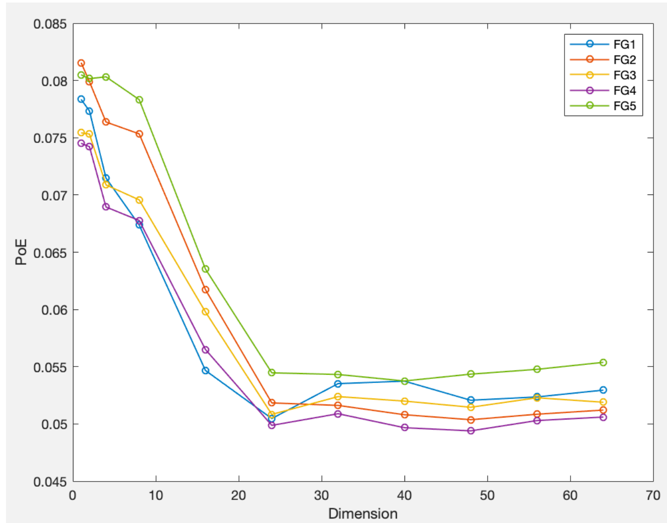
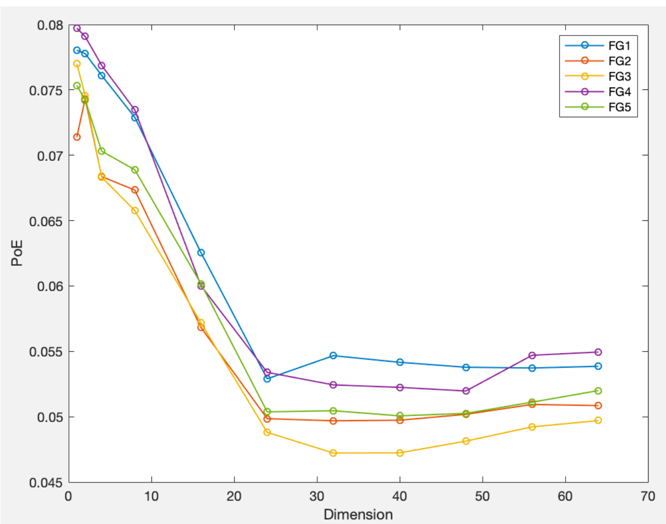
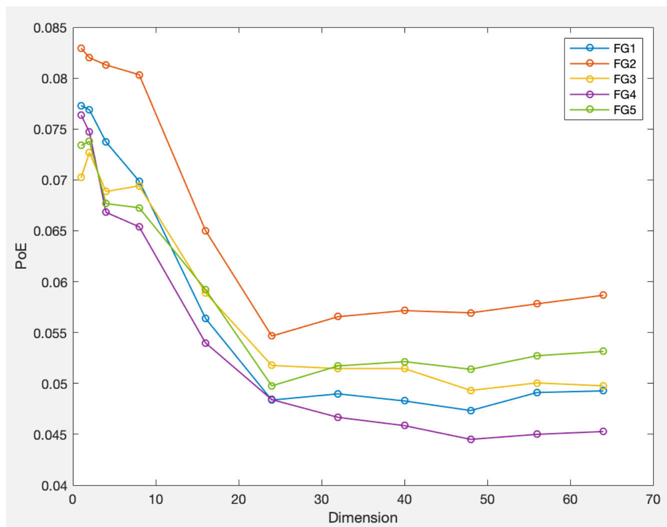
Bolin He A53316428

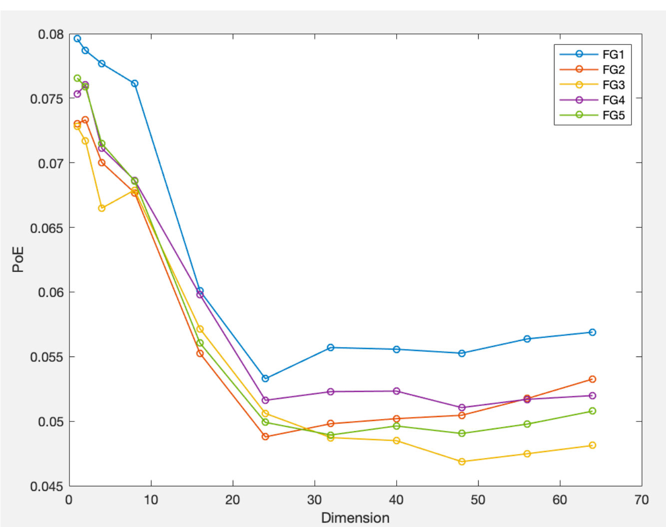
HW 5
Q 1









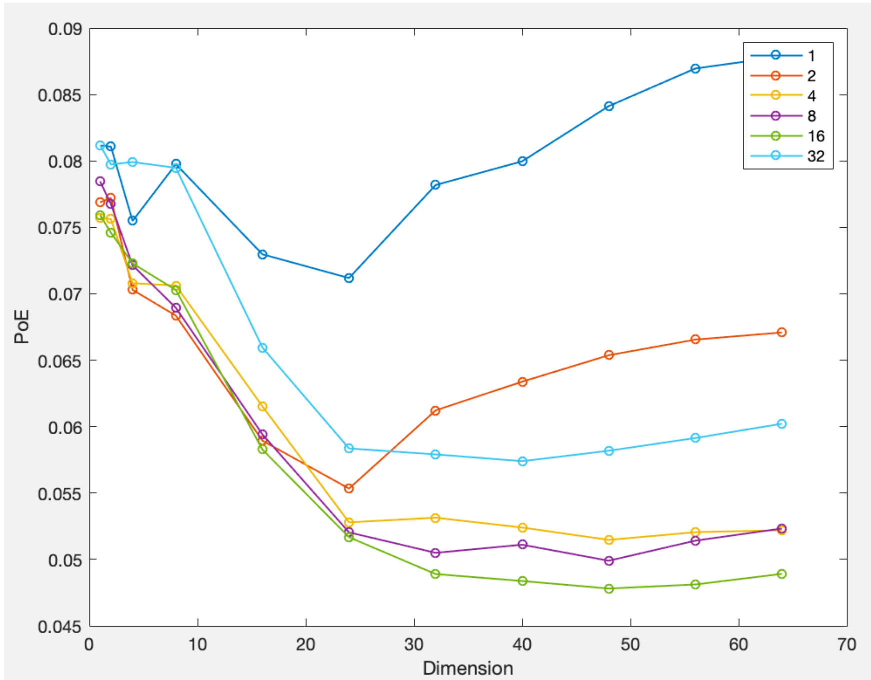


Obverse the curves above, one curve will have less PoE than the other in most dimensions we plot. In other words, if one curve has better performance in one dimension, it would have greater probability to perform better in other dimensions. This phenomenon indicates that the initialization will play an important role in the probability of error.

In fact, the EM algorithm is sensitive to the initial value, and the clustering results fluctuate greatly with different initial values. In general, EM algorithm convergence depends largely on its initial parameters.

EM algorithm can guarantee the stability of convergence to a point, but it does not guarantee convergence to the global maximum. Obviously, if our optimization goal $L(\theta,\theta_L)$ is convex, the EM algorithm can guarantee the convergence to the global maximum, this is the same as other iterative algorithms like gradient descent method. But in the cases of non-convex problem, the result will probably converge to a local maximum. And different initialization will affect not only the rate of convergence, but also the way it converges. In other words, the PoE largely depends on the parameters initialization.

Q 2



The figure above shows that when $C_1 = \{4, 8, 16\}$, the result has less PoE than $C_2 = \{1, 2, 32\}$. With C increasing, the PoE first decreases and then increases.

Obviously, proper selection of C would have great impact on the final result. In our case, whether C is too small or too large, we will have worse prediction. It indicates that C_1 is close to the C_{ground_true}. In other words, if the number of mixture components are close to the ground true condition, we will have less PoE, and vice versa.

Table of Contents

Bolin He, PID: A53316428, Hw05

Dec 5,2019

```
clear all;
clc;
```

Initialization

```
load('TrainingSamplesDCT_8_new.mat')
I = imread('cheetah.bmp');
I = im2double(I);
I_mask = imread('cheetah_mask.bmp');
I_mask = im2double(I_mask);
[xb,yb] = size(TrainsampleDCT_BG);
[xf,yf] = size(TrainsampleDCT_FG);
x = 255-7;
y = 270-7;
Limit = 1000; % Iteration limits
C = 8; % Components
Dimension = [1,2,4,8,16,24,32,40,48,56,64];

% ZigZag Processing
ZZ = zeros(x*y,64);
for i = 1:x
    for j = 1:y
        Block = I(i:i+7,j:j+7);
        DCT = dct2(Block);
        idx = reshape(1:numel(DCT), size(DCT));
        idx = fliplr(spdiags(fliplr(idx)));
        idx(:,1:2:end) = flipud( idx(:,1:2:end) );
        idx(idx==0) = [];
        ZZ((i-1)*(y)+j,:) = DCT(idx);
    end
end
```

BG EM

Initialize

```
BG_pi = rand(1, C);
BG_pi = BG_pi./sum(BG_pi);
```

```
BG_mu = TrainsampleDCT_BG(randi([1 xb],1,C),:);
BG_cov = zeros(yb,yb,C);
for i =1:C
    BG_cov(:,:,i) = (rand(1,yb)).*eye(yb);
end
Distribution = zeros(xb,C);

for i = 1:Limit
% E-step
    for j = 1:C
        Distribution(:,j) =
 mvnpdf(TrainsampleDCT_BG,BG_mu(j,:),BG_cov(:,:,j))*BG_pi(j);
    end
    h = Distribution./sum(Distribution,2);
    BGLikelihood(i) = sum(log(sum(Distribution,2)));

% M-step
    BG_pi = sum(h)/xb;
    for j = 1:C
        BG_cov(:,:,j) = diag(diag(((TrainsampleDCT_BG-
BG_mu(j,:))'.*h(:,j)'* ...
            (TrainsampleDCT_BG-BG_mu(j,:))./sum(h(:,j),1))+1e-7));
    end
    BG_mu = (h'*TrainsampleDCT_BG)./sum(h)';

% Stop Criterion
    if i > 1
        if abs(BGLikelihood(i) - BGLikelihood(i-1)) < 0.001
            break;
        end
    end
end
```

FG EM

Initialize

```
FG_pi = rand(1, C);
FG_pi = FG_pi / sum(FG_pi);
FG_mu = TrainsampleDCT_FG(randi([1 xf],1,C),:);
FG_cov = zeros(yf,yf,C);
for i =1:C
    FG_cov(:,:,i) = (rand(1,yf)).*eye(yf);
end
Distribution = zeros(xf,C);

for i = 1:Limit
% E-step
    for j = 1:C
        Distribution(:,j) =
 mvnpdf(TrainsampleDCT_FG,FG_mu(j,:),FG_cov(:,:,j))*FG_pi(j);
    end
    h = Distribution./sum(Distribution,2);
```

```
    FGLikelihood(i) = sum(log(sum(Distribution,2)));

% M-step
    FG_pi = sum(h)/xf;
    for j = 1:C
        FG_cov(:,:,j) = diag(diag(((TrainsampleDCT_FG-
FG_mu(j,:))'.*h(:,j)'* ...
            (TrainsampleDCT_FG-FG_mu(j,:))./sum(h(:,j),1))+1e-7));
    end
    FG_mu = (h'*TrainsampleDCT_FG)./sum(h)';

% Stop criterion
    if i > 1
        if abs(FGLikelihood(i) - FGLikelihood(i-1))<0.001
            break;
        end
    end
end
```

PoE

```
    for j = 1:length(Dimension)
        K = Dimension(j);
        mask = zeros(x*y,1);
        for xb = 1:length(ZZ)
            probabilityBG = xb/(xf+xb);
            probabilityFG = xf/(xf+xb);

            for y = 1:size(BG_mu,1)
                probabilityBG = probabilityBG*mvnpdf(ZZ(xb,1:K), ...
                    BG_mu(y,1:K),BG_cov(1:K,1:K,y))*BG_pi(y);
            end

            for y = 1:size(FG_mu,1)
                probabilityFG = probabilityFG*mvnpdf(ZZ(xb,1:K), ...
                    FG_mu(y,1:K),FG_cov(1:K,1:K,y))*FG_pi(y);
            end

            if probabilityBG < probabilityFG
                mask(xb) = 1;
            end
        end

        t_mask = zeros(x,y);
        for xb = 1:x
            t_mask(xb,:) = mask(((xb-1)*(y)+1):xb*(y))';
        end
        mask = t_mask;

        % PoE
        incorrectCount = 0;
        for xb = 1:x
            for y = 1:y
```

```
                if I_mask(xb,y) ~= mask(xb,y)
                    incorrectCount = incorrectCount + 1;
                end
            end
        end
        error(j) = incorrectCount/(255*270);
    end
%         plot(Dimension,error,'o-','markersize',5,'linewidth',2)
%         hold on;

% legend('FG1','FG2','FG3','FG4','FG5')
```

Published with MATLAB® R2018a