Bolin He
PID: 53316428
ECE 271A HW02
Oct 20, 2019

a)
All the calculation shows the same result. That
PY(cheetah) = 0.1919
PY(grass) = 0.8081

b) The details of maximum likelihood estimate can be referred to the codes. After plotting all the figures, we choose the best ones as [1, 18, 19, 25, 32, 34, 40, 41] and the worst ones as [4, 5, 6, 59, 60, 62, 63, 64].

The result are:

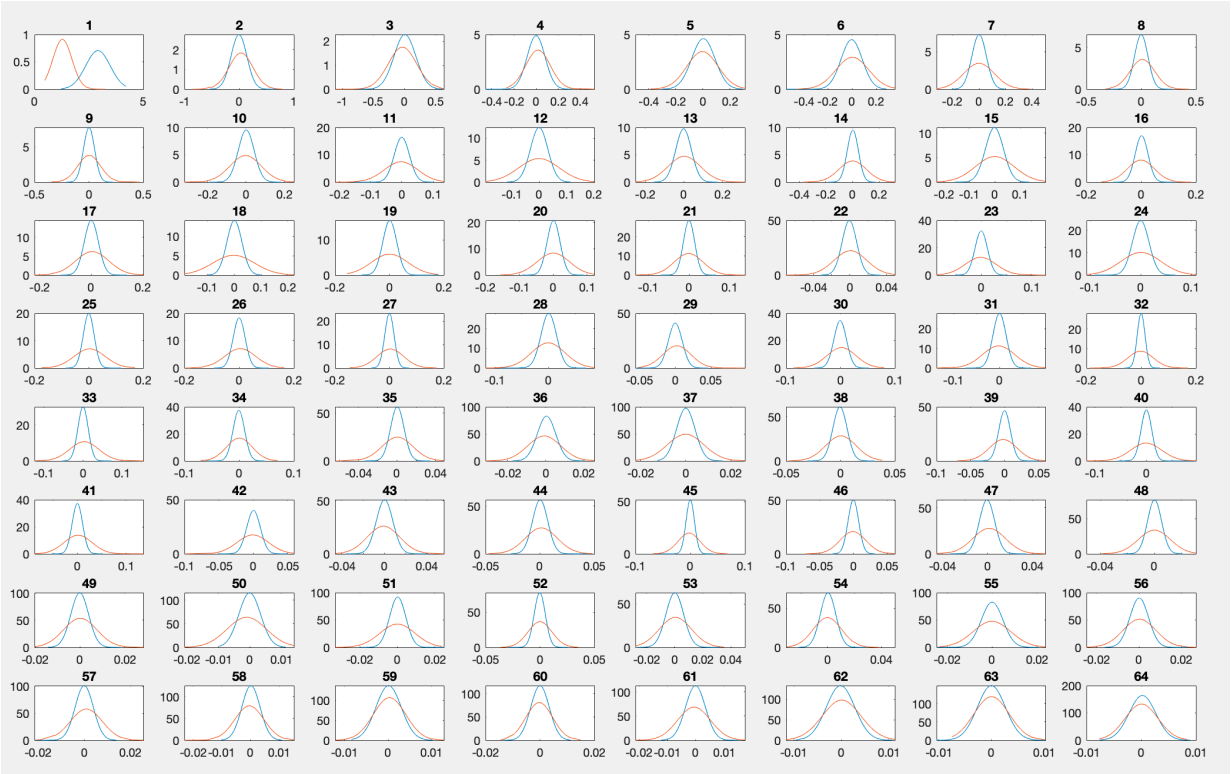

Fig.1 Whole set of 64 plots

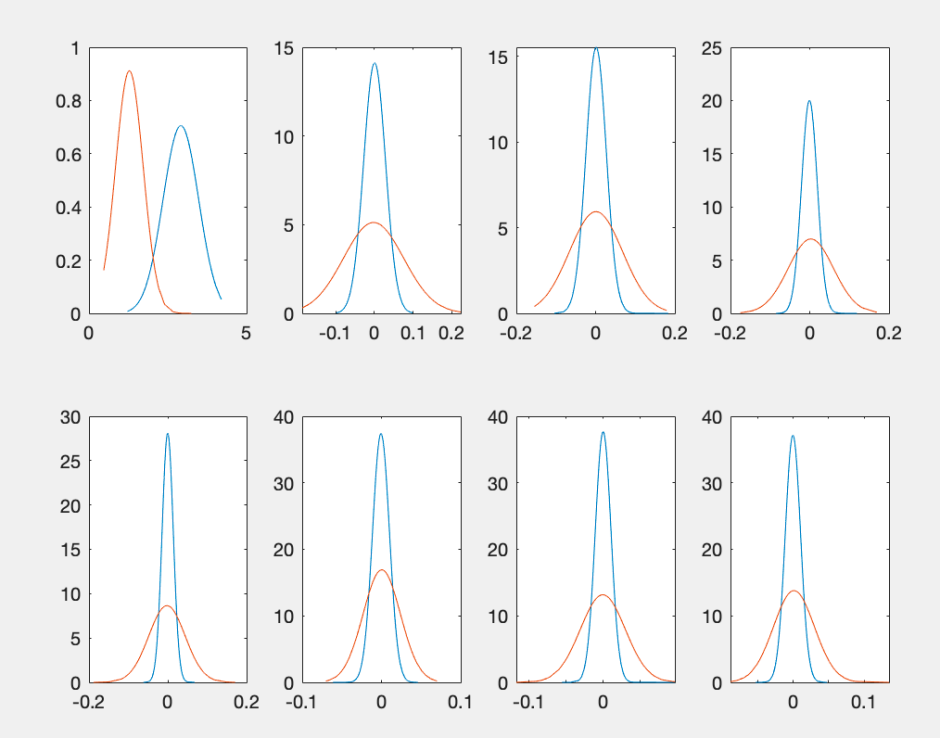

Fig.2 The best 8 plots

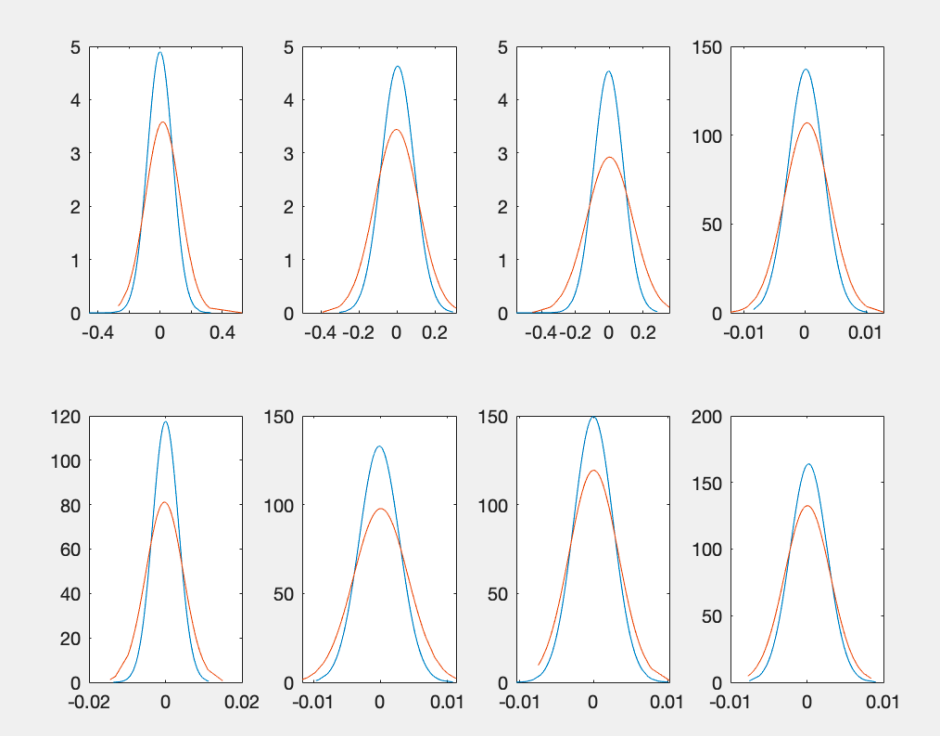

Fig.3 The worst 8 plots

c)

The error can be calculated by the prior probability and the misclassified probability.

error = error(Grass) + error(Cheetah)

error(Grass) = Grass misclassified as Cheetah / Grass Pixels * Prior probability of Grass

error(Cheetah) = Cheetah misclassified as Grass / Cheetah Pixels * Prior probability of Cheetah

Finally we find the error for 64-dimension model is 0.0896, while the error for 8-dimension model is 0.0538

The result shows that the 8-dimension model works better than the 64-dimension model. Because the 8- dimension model contains more distinct features. Those distinct features are the keys for us to classify objects. Even though the 64-dimension model has more feature, much of them are useless and probably confuse us. This reminds us that we can reduce dimensionality by eliminating the useless futures, which might add burden on processing.

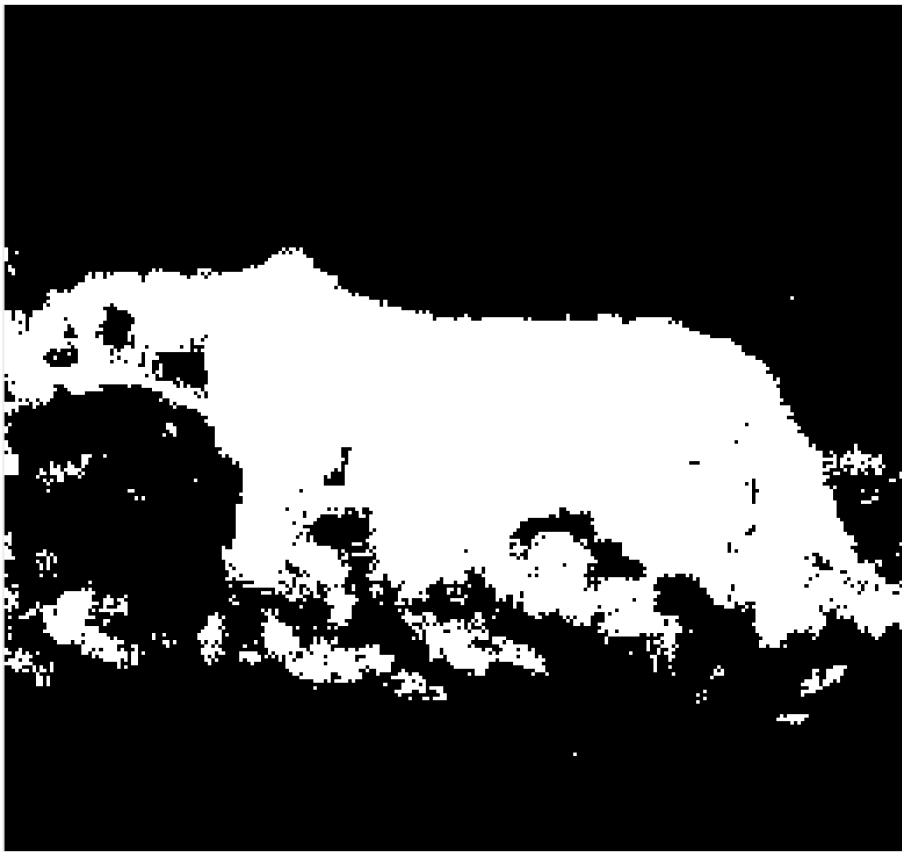

Fig.4 64-dimension

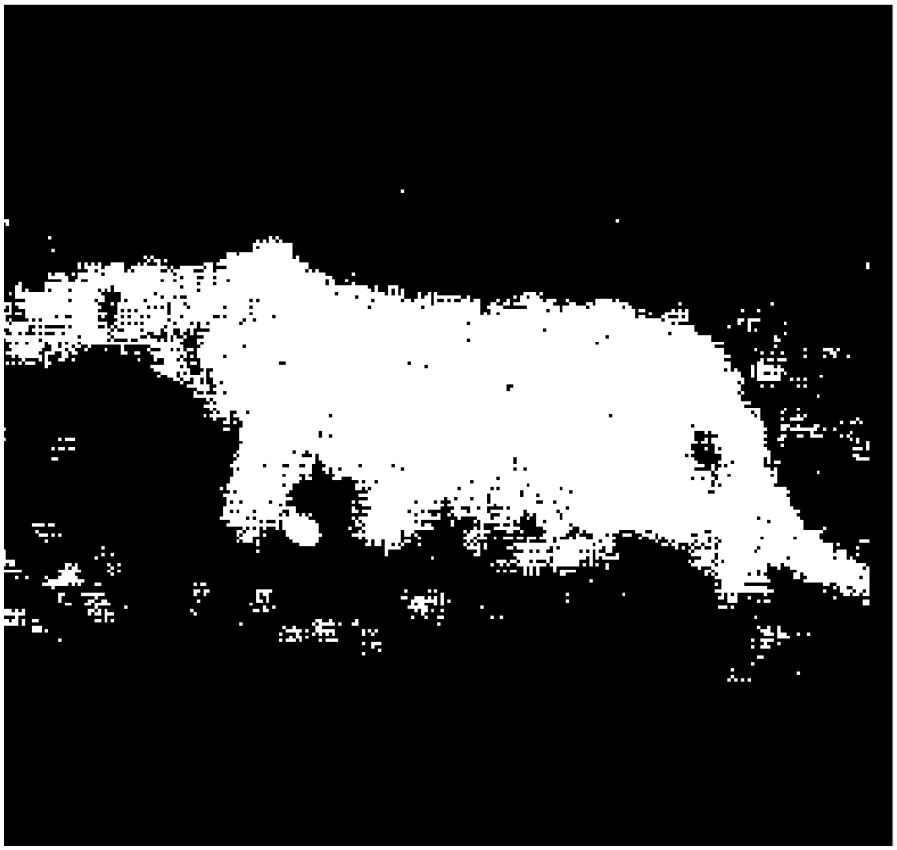

Fig.5 8-dimension

Bolin He, PID: A53316428, Hw02

Table of Contents

Oct 18,2019

Question a

```
load('TrainingSamplesDCT_8_new.mat');
TB = TrainsampleDCT_BG;
TF = TrainsampleDCT_FG;

[xtb,ytb] = size(TB);
[xtf,ytf] = size(TF);

PY = xtf/(xtf+xtb); % cheetah
PX = xtb/(xtf+xtb); % grass
```

Question b

maximum likelihood estimates

```
mTB = mean(TB);
cTB = cov(TB);
sTB = std(TB);
vTB = var(TB);
mTF = mean(TF);
cTF = cov(TF);
sTF = std(TF);
vTF = var(TF);

% foreground
mleTF = zeros(xtf,1);
for i=1:xtf
    tmp = 0;
    for j=1:64
       tmp = tmp + ((TF(i,j)-mTF(j))/sqrt(varFG(j)))^2;
    end
    mleTF(i,1) = exp(-32*log(2*pi)-64*log(sqrt(vTF(j)))-0.5*tmp);
end

% background
mleTB = zeros(xtb,1);
for i=1:xtb
    tmp = 0;
    for j=1:64
```

```
       tmp = tmp + ((TB(i,j)-mTB(j))/sqrt(vTB(j)))^2;
    end
    mleTB(i,1) = exp(-32*log(2*pi)-64*log(sqrt(vTB(j)))-0.5*tmp);
end


% 64 dimension
for i = 1:64
    subplot(8,8,i)
    plot(sort(TB(1:1053,i)),normpdf(sort(TB(1:1053,i)),mTB(i),sTB(i)))
    hold on;
    plot(sort(TF(1:250,i)),normpdf(sort(TF(1:250,i)),mTF(i),sTF(i)))
    hold off;
    title(i)
end


% By obervation, we choose the best figures [1, 18, 19, 25, 32, 34,
 40, 41]
% We choose the worst figures [4, 5, 6, 59, 60, 62, 63, 64].

% The best 8 and the worst 8 plots
best = [1, 18, 19, 25, 32, 34, 40, 41];
worst = [4, 5, 6, 59, 60, 62, 63, 64];
figure;
for i = 1:64
    for j = 1:8
        if i == best(j)
        subplot(2,4,j)

 plot(sort(TB(1:1053,i)),normpdf(sort(TB(1:1053,i)),mTB(i),sTB(i)))
        hold on;

 plot(sort(TF(1:250,i)),normpdf(sort(TF(1:250,i)),mTF(i),sTF(i)))
        hold off;
        end
    end
end


figure;
for i = 1:64
    for j = 1:8
        if i == worst(j)
        subplot(2,4,j)

 plot(sort(TB(1:1053,i)),normpdf(sort(TB(1:1053,i)),mTB(i),sTB(i)))
        hold on;

 plot(sort(TF(1:250,i)),normpdf(sort(TF(1:250,i)),mTF(i),sTF(i)))
        hold off;
        end
    end
end
```

```
Index in position 1 exceeds array bounds (must not exceed 1).

Error in ECE_271_hw2 (line 63)
figure;
```

Question c

```
ZZ = load('Zig-Zag Pattern.txt');
ZZ = ZZ+1;
I = imread('cheetah.bmp');
I = im2double(I);

[x,y] = size(I);
NewI64 = zeros(x-7,y-7);

% 64-demensional Gaussians
figure;
    for i=1:x-7
      for j=1:y-7
                SW = I(i:i+7,j:j+7);
                T = dct2(SW);
                Rearrange(ZZ) = T;
                TB64 = mvnpdf(Rearrange,mTB,cTB)*PX;
                TF64 = mvnpdf(Rearrange,mTF,cTF)*PY;

         if TB64 <= TF64
              NewI64(i,j) = uint8(1);
         end
      end

end
NewI64 = padarray(NewI64,[7,7],'post');
imshow(NewI64);

% 8-demensional Gaussians
cTB_best = cov(TB(:,best));
cTF_best = cov(TF(:,best));
mTB_best = mean(TB(:,best));
mTF_best = mean(TF(:,best));

NewI8 = zeros(x-7,y-7);
figure;
for i=1:x-7
      for j=1:y-7
                SW = I(i:i+7,j:j+7);
                T = dct2(SW);
                Rearrange(ZZ) = T;
                TB8 = mvnpdf(Rearrange(best),mTB_best,cTB_best)*PX;
                TF8 = mvnpdf(Rearrange(best),mTF_best,cTF_best)*PY;

         if TB8 <= TF8
              NewI8(i,j) = uint8(1);
         end
```

```
      end

end
NewI8 = padarray(NewI8,[7,7],'post');
imshow(NewI8);

% The probability of error
Imask = imread('cheetah_mask.bmp');
Imask = im2double(Imask);

% 64 dimensions
count = 0;
count2 = 0;
CheetahP = 0;
for i = 1:x
    for j =1:y
        if Imask(i,j) == 1,
            CheetahP = CheetahP + 1;
        end

        if NewI64(i,j) < Imask(i,j) % misclassify cheetah as grass
            count = count+1;
        elseif NewI64(i,j) > Imask(i,j) % misclassify grass as cheetah
            count2 = count2+1;
        end
    end
end

error64 = count/CheetahP*PY + count2/(x*y-CheetahP)*PX;

% 8 dimensions
count3 = 0;
count4 = 0;
for i = 1:x
    for j =1:y
        if NewI8(i,j) < Imask(i,j) % misclassify cheetah as grass
            count3 = count3+1;
        elseif NewI8(i,j) > Imask(i,j) % misclassify grass as cheetah
            count4 = count4+1;
        end
    end
end

error8 = count3/CheetahP*PY + count4/(x*y-CheetahP)*PX;
```

Published with MATLAB® R2018a