In the Name of God
Project-1
Expectation Maximization
Department of Electrical Engineering - Shiraz University

Instructor: Dr. Hamed Masnadi-Shirazi Spring 2016

1 Introduction

In this project we will use the EM algorithm for Gaussian mixture models to segment an image by classifying each pixel in the image as either belonging to the foreground or background class. We proved the convergence of the general EM algorithm and derived the EM algorithm for Gausiian mixtures in class. We can summarized the algorithm below. For a multidimensional Gaussian, the E-step involves computing

$$h_{ij} = \frac{\mathcal{N}(x_i, \mu_j^{(t)}, \Sigma_j^{(t)})\pi_j^{(t)}}{\sum_{k=1}^{C} \mathcal{N}(x_i, \mu_k^{(t)}, \Sigma_k^{(t)})\pi_k^{(t)}},$$

where i is the index for the number of data samples n, and j is the index for the number of mixture C and $\mu_j^{(t)}$, $\Sigma_j^{(t)}$ and $\pi_j^{(t)}$ are the mean vector, covariance matrix and mixture weight of the jth mixture at iteration t respectively. The M-step involves computing in order

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

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

$$\Sigma_j^{(t+1)} = \frac{\sum_i^n h_{ij}(x_i - \mu_j^{(t+1)})(x_i - \mu_j^{(t+1)})^T}{\sum_i^n h_{ij}}.$$

Note that the new $\mu_j^{(t+1)}$, found at iteration $t+1$, is used to find $\Sigma_j^{(t+1)}$. In this project we will assume that $\Sigma_j^{(t+1)}$ is diagonal and so we can find each term on the diagonal from

$$\sigma_{j_k}^{2(t+1)} = \frac{\sum_i^n h_{ij}(x_{i_k} - \mu_{j_k}^{(t+1)})^2}{\sum_i^n h_{ij}}.$$

where $\sigma_{j_k}^{2(t+1)}$ is the kth element on the diagonal of $\Sigma_j^{(t+1)}$ and x_{i_k} and μ_{j_k} are the kth elements of the vectors x_i and μ_j.

2 General Assignment

In this project you will first be given a set of 1-dimensional data sampled from a mixture of $C = 2$ Gaussians. You will implement the EM algorithm and learn the parameters of the two Gaussians from which the data where sampled. This will help you check that your EM algorithm is working correctly. In the second stage, you will check your algorithm on 2-dimensional data sampled from a mixture of $C = 2$ Gaussians. You should use these two steps to write your code such that it works on a mixture of Gaussians of any dimension and any number of mixtures.

In the final step you will be given two sets of data, one set will be data sampled from the foreground of an image and the other will be data sampled from the background of an image. Each set will contain 64-dimensional data corresponding to the DCT features of an 8x8 pixel block of the image as explained in class. You will use the EM algorithm to learn a mixture of $C = 4$ Gaussians for each of the background and foreground classes. Finally, you will use the Bayes decision rule to classify the pixels of an image into the foreground or background classes.

3 Matlab details

STEP 1. Use the matlab code below to load the training data.

```
load('C:\TrainDATAtoyGaussian1D.mat','x1');
```

where x1 is data sampled from two mixtures of Gaussians with the following parameters: $\mu_1 = 4$, $\sigma_1^2 = 0.5$, $\pi_1 = 0.75$ and $\mu_2 = 1$, $\sigma_2^2 = 1$, $\pi_2 = 0.25$. To implement the EM algorithm we need an initial Guess for the parameters. Use the following initial values

```
muS= [0 3];
SIGMAS=[0.5 1];
PIS=[0.5 0.5];
```

We also need a stooping criteria for the EM algorithm. Compute the Q-function using

$$Q^{(t+1)} = \sum_{i,j} h_{ij} \log(\mathcal{N}(x_i, \mu_j^{(t+1)}, \Sigma_j^{(t+1)}) \pi_j^{(t+1)}) \tag{1}$$

and set the stopping criteria to be

```
while Qnew-Qold > epsilon
```

where

```
epsilon=0.01;
```

You can use the mvnpdf.m function in Matlab to make your life easier. Also try to write your code using matrix algebra rather than multiple for-loops, since Matlab is optimized for matrix algebra and slower with for-loops. If you did things correctly, you should get the following results after 19 iterations

```
muS = 1.0274    4.0281
SIGMAS = 1.2836    0.4755
PIS = 0.2530    0.7470
counter = 19
```

As you can see, these values are very close to the true distribution values from which the training data was sampled from. What is the final value of Qnew? Include this in your report.

STEP 2. In this step we will make sure that our code works for multidimensional Gaussians. Specifically we will check the algorithm on 2-dimensional data sampled from a mixture of $C = 2$ Gaussians. Use the matlab code below to load the training data.

```
load('C:\TrainDATAtoyGaussian2D.mat','x1');
```

To implement the EM algorithm we need an initial Guess for the parameters. Use the following initial values

```
muS= [0 3 ; 0 0];  % a matrix where each column is a mean vector
SIGMAS=[0.4 0.8; 0.6 0.7];  % a matrix where each column is the diagonal terms of the variance matrix
PIS=[0.5 0.5];
```

and set the stopping criteria parameter to be

```
epsilon=0.01;
```

Report the final values of the means, covariances and weights that your algorithm converges to. What is the final Qnew and after how many iterations does your algorithm stop?

STEP 3. In this step we will find the Gaussian mixture model that represents the background image. The data sampled from the background is available by writing

```
load('C:\TrainingSamplesDCT_8_new');
x1=TrainsampleDCT_BG;
[r c]=size(x1);
n=r; % number of data points
d=c; % number of dimensions of data
C=4; % choose number of mixtures
```

Note that if you have written your code correctly, you only need to load the new x1 data matrix as above and change the number of mixtures to C=4. Keep the stopping criteria as

```
epsilon=0.01;
```

You should also use the code below to load the initial values for the means, variances and weights. These values are usually set at random in practice, but we require that everyone use the same initial values for grading purposes.

```
load('C:\INITValues_BackGround.mat','muinit','varinit','PIS');
```

Report the final values of the means, covariances and weights that your algorithm converges to. Save these values. What is the final Qnew and after how many iterations does your algorithm stop?

STEP 4. In this step we will find the Gaussian mixture model that represents the foreground image. The data sampled from the foreground is available by writing

```
load('C:\TrainingSamplesDCT_8_new');
x1=TrainsampleDCT_FG;
[r c]=size(x1);
n=r; % number of data points
d=c; % number of dimensions of data
C=4; % choose number of mixtures
```

Note that if you have written your code correctly, you only need to load the new x1 data matrix as above and change the number of mixtures to C=4. Keep the stopping criteria as

```
epsilon=0.01;
```

You should also use the code below to load the initial values for the means, variances and weights. These values are usually set at random in practice, but we require that everyone use the same initial values for grading purposes.

```
load('C:\INITValues_ForeGround.mat','muinit','varinit','PIS');
```

Report the final values of the means, covariances and weights that your algorithm converges to. Save these values. What is the final Qnew and after how many iterations does your algorithm stop?

STEP 5. In this step we will load an image and classifiy each pixel as either belonging to the background or the foreground. Use the following code to load the image and its mask that defines the foreground and background pixels.

```
I=im2double(imread('C:\cheetah.bmp'));
Imask=im2double(imread('C:\cheetah_mask.bmp'));
imshow(I)
imshow(Imask)
```

include these images in your report. If the EM algorithm works perfectly it should classify each pixel exactly like the mask. To see how well the EM algorithm does, load the EM results from STEP3 and STEP4. Also, load the DCT coefficient numbers as explained in class using the code below

```
zigzag=load('C:\zig-zag pattern.txt');
[val, plc]=sort(zigzag(:));
```

What does the above code do exactly?

Now, for each **overlapping** 8x8 block of the image, you should compute the DCT coefficients using the code below as explained in class. (**Note**: each 8x8 block represents the center pixel of the block within the image. Pixels on the border of the image should be assumed to belong to the Background class.)

```
dctblock = dct2(block8by8);
dctblock = dctblock(plc);
```

You should next compute the conditional probability of dctblock given the Background ($P(dctblock|Background)$) and the Foreground ($P(dctblock|Foreground)$), using the mixture of Gaussian models you learned from the EM algorithms. To compute the probability of $P(Background|dctblock)$ and $P(Foreground|dctblock)$ using the Bayes rule, we need to estimate a value for the priors. We will find the priors using the number of data points for each class in the training data as explained in class. You should now write the Bayes rule and assign this pixel to either the foreground or background class.

Finally, make a binary image that has all pixels of the foreground class set to 1 and all pixels of the background class set to 0. Include this image in your report. How close is this image to the mask? Report what percentage of the Foreground class pixels have been classified correctly. Report what percentage of the Background class pixels have been classified incorrectly.

4 What to turn in

You should turn in a CD which includes (1) A report on your project (2) and a Matlab .m file of your program. Your report should include your results for each STEP. You should clearly divide your report into STEPs. Failing to do so will get you a zero grade! You Matlab program should have a comment for each line and each segment of code explaining what that line or segment of code does. If you fail to comment your code, you will get a zero grade! I should be able to run your code directly from the CD and see the plots and results. If your code has errors when I run it or does not work when I run it from the CD, you will get a zero grade!

5 CHEATING :-(

Copying a portion of someones report or Matlab code is considered cheating. If even a portion of your report or Matlab code is similar to someone else you will both get a zero on the project, be introduced to the university disciplinary committee (Komite Enzebaty) and possibly fail the course. SO DO NOT EVEN SHOW YOUR CODE OR REPORT TO ANYONE ELSE!