CS 470 Machine Learning

Assignment-3

Gaussian Mixture Models

Instructor: Hassan Aqeel Khan

Due Date: Friday March 29, 2019

Background: In this assignment you will learn to generate and visualize Gaussian Mixture Data. You will also learn to fit a Gaussian Mixture Model to data. We will be using some opensource data available online [1]. The associated scripts are available in the 'GMMCode' folder. The pdf of a GMM is given by the expression below:

$$f_X(\mathbf{x}) = \sum_{j=1}^{K} w_j \mathcal{N}(\mathbf{x}|\, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$$

Here, $\mathbf{x} = [x_1, x_2]^T \in \mathbb{R}^2$ represents the 2-dimensional data vector; $\boldsymbol{\mu}_j = [\mu_1, \mu_2]^T$ is the mean vector of the j^{th} Gaussian in the mixture; $\boldsymbol{\Sigma}_j$ is the (2×2) covariance matrix of the j^{th} mixture component. The vector $\mathbf{w} = [w_1, \ldots., w_K]^T$ is the vector containing the weights assigned to the different components of the mixture pdf.

Task-1 Write Matlab code to generate 1000 samples from a bivariate (or 2-dimensional) Gaussian Mixture Model with the following specifications:

$$\boldsymbol{\mu}_1 = [0, 0]^T;\ \boldsymbol{\mu}_2 = [3, 3]^T;\ \boldsymbol{\mu}_3 = [0, 4]^T;$$

$$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 1 & 0.4 \\ 0.4 & 1 \end{bmatrix};\ \boldsymbol{\Sigma}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix};\ \boldsymbol{\Sigma}_3 = \begin{bmatrix} 0.4 & 0 \\ 0 & 0.1 \end{bmatrix};$$

$$\mathbf{w} = \left[\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right]^T$$

Plot your data samples on a graph and briefly describe how you generated them and whether agree with what you expected.

Hint: There are a number of ways of doing this. You can either generate samples using the Matlab function mvnrnd( ) which allows you to sample a single multi-variate Gaussian pdf of a specified mean and covariance. To create a mixture of three gaussians you can generate 33.33% of the total samples from each of the three multivariate Gaussians specified above and plot the result. The second option can be employ the functions gmdistrbution( ) and random( ) to sample from GMM directly. See the following link for details: https://www.mathworks.com/help/stats/gmdistribution.random.html. An example output obtained by sampling the GMM specified above is plotted in Figure 1. The samples from each of the mixture components are shown in different colors to emphasize the difference.

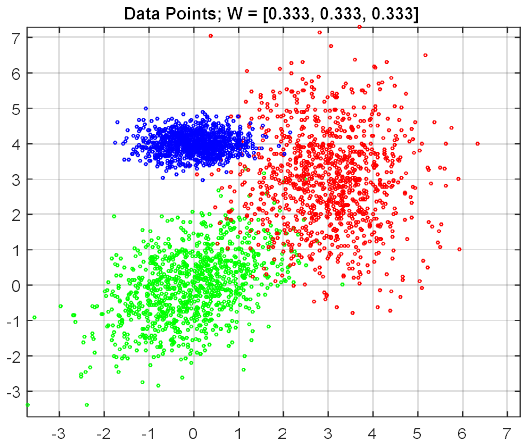

Figure 1 Plot of 1000 samples obtained by sampling the desired GMM.

Task-2 Assume for a second that your knowledge of the underlying pdf is limited; all you only know is that your data samples come from a GMM containing only 3 components however, you do not know the parameters $(w_j, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$ of the underlying GMM pdf. Your objective in this task is to estimate the values of these parameters from data. Given below is a basic guideline on how this can be accomplished:

1. These parameters can be estimated using the Expectation Maximization (or EM) algorithm. This can be performed by calling the function gmdistribution.fit( ). By default, the EM algorithm will start with random values of the parameters and then try to refine them based on the observed data samples. However, the performance of the EM algorithm can vary significantly depending on initial values and therefore; it is generally preferable to obtain initial estimates using the *k-means* clustering algorithm (I encourage you to Google the k-means in your own time). Use the given function EM_init_kmeans( ) to obtain initial estimates for the parameters $(w_j, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$.
2. After you obtain initial estimates for the parameters use the function gmdistribution.fit to refine your estimates. You can read the Matlab help to understand how this can be done. Below is some sample code that you can use however; please do not blindly copy it and make sure you understand what it means.

```
s = struct('mu',MuInit,'Sigma',SigInit,'PComponents',Priors);
options = statset('Display','final');
e = 1e-5;
GMModel =
gmdistribution.fit(X,k,'CovType','full','Options',options,'Start'
,s,'Regularize',e);
muModel = GMModel.mu;
SigModel = GMModel.Sigma;
```

Here; MuInit, SigInit & Priors are the respective (initial) estimates of the parameters $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ and $\mathbf{w}$, obtained from the function EM_init_kmean( ). 'X' is a (1000×2) matrix containing the data samples.

3. Write down values of the estimated parameters can compare them with the true values listed on page one. Discuss your results.

Task-4 Plot the contours of the estimated mixture components and discuss them. Sample code on how you can do this given below.

```
figure;
hold on; grid on;
Wmodel = GMModel.ComponentProportion;
plotGMM(muModel', SigModel, [.8 0 0], 1);
title(sprintf('Contours of GMM, Wmodel = [%0.3f, %0.3f,
%0.3f]',sort(Wmodel,'descend')));
```

Here, muModel, SigModel and Wmodel are the final estimates of the mean vectors, covariance matrices and weight vector obtained after the application of the gmdistribution.fit function to the data.

Task-5 Up until now, we have assumed that the number of mixtures,K, was known to us. This is somewhat of an unrealistic assumption since in most practical situations we do not have knowledge of K. Therefore, (in addition to the parameters $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ & $\mathbf{w}$) we also need to estimate K from the data samples. This is generally, accomplished by employing a metric called the "Bayesian Information Criterion" (BIC) [2].

$$BIC = \ln(n)\, K - 2\ln(\hat{L})$$

Where,

n = The number of data samples or observations.

$\hat{L}$ = The maximized value of the likelihood function obtained after application of the EM-algorithm.

Load the data contained in the file Lab-07.mat and plot the 2-D data vector using the following command: plot(X(:,1),X(:,2),'go','MarkerSize',2);. You should get a plot similar to Figure 2. Try to estimate the number of mixtures in the GMM pdf that generated these data points. Estimate K and then give rough estimates for the corresponding mean vectors and covariance matrices.

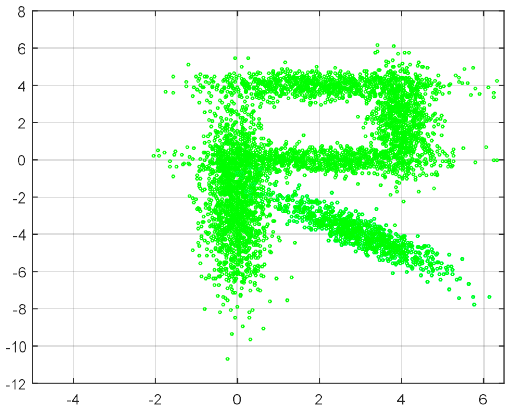

Figure 2 Plot of given data samples.

Task-6 Fit GMM with two mixtures $(K = 2)$ to this data using the code you developed for task-1 to 4. Quick reminder of the distribution fitting procedure: (1) Initialize using k-means then; (2) Use the EM algorithm to get final estimates of the parameters; and (3) Plot the contours of the resulting GMModel. Matlab also returns the BIC (GMModdel.BIC) with the model which you should save for plotting later.

Task-7 Repeat Task-2 for the following values of $\mathrm{K} = [3, 4, 5, 6, 7, 8, 9]$. Save the corresponding BIC and plot the resulting contour. Which value of K gives you the best fit to the data and does it agree with the number of mixtures you estimated in Task-5?

Task-8 Plot the BIC as a function of different values of K. Based on the BIC you should employ the value of K which gives you the minimum value of the BIC.

Side Note: For the data given in this Lab one may argue why go to the trouble of evaluating (and understanding) the BIC when we can easily evaluate the number of mixtures via visual inspection. Indeed, this is a valid criticism for the 2-dimensional toy example given here. However, visualization of data becomes challenging when dealing with 10's or 100's of variables; which happens very frequently in real life data analysis. Imagine trying to visually estimate the number of mixtures (or clusters) in 100-dimensional space. In such a situation you will have no option but to employ the BIC for estimation of the number of mixture components. For practical applications of GMMs please refer to CMU's opensource speech recognition toolkit available at https://cmusphinx.github.io/ .

References

[1] Billard A, Calinon S, Dillmann R, Schaal S. Robot programming by demonstration. In Springer handbook of robotics 2008 (pp. 1371-1394). Springer Berlin Heidelberg.

[2] Schwarz G. Estimating the dimension of a model. The annals of statistics. 1978;6(2):461-4.

[3] Mathworks and Matlab help documentation.