

### CSE 5522 Homework 3

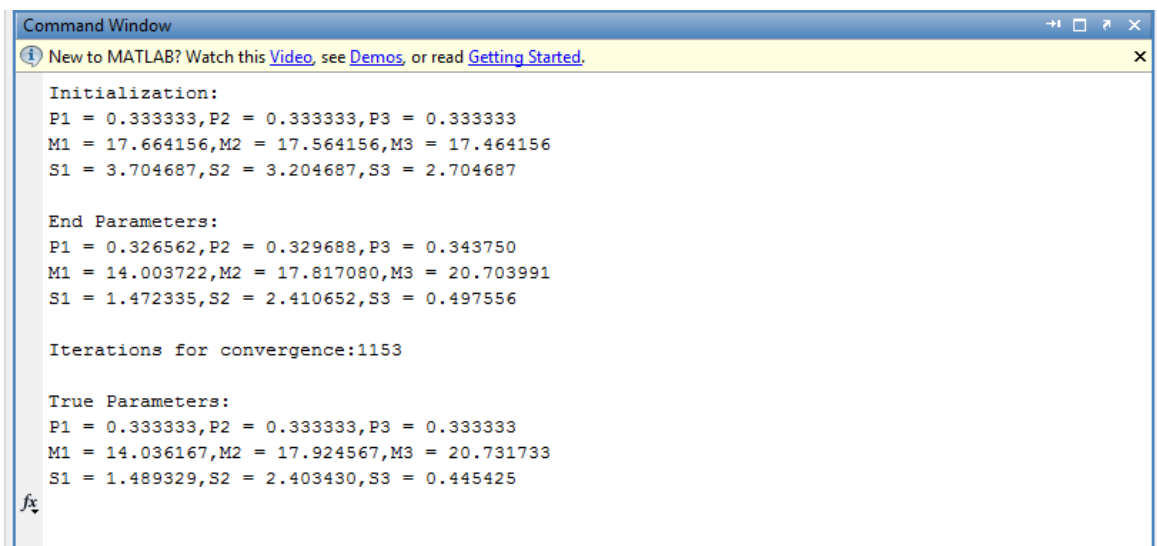
Manjari Akella

11/08/2013

1. Here is a set of one-dimensional data that was generated by 3 Gaussians: 1-dimensional data. You will need to use the code you developed in hw1 for reading in data files, and calculating means and standard deviations.

Create a program to estimate the means, standard deviations, and weights of a mixture of Gaussians via the EM algorithm. You will probably want to have three functions: one that performs the expectation step, one that performs the maximization step, and one that runs the outer loop. For this part, report on the parameters listed above that result from the EM algorithm. Also, turn in your code with an explanation of how to run it; we will evaluate your code on a different data set.

- Figure 1 shows the results of the EM algorithm on the given data for the specified initializations of mean ( $m+0.1, m, m-0.1$ ), standard deviations ( $s+0.5, s, s-0.5$ ) and prior probabilities ( $1/3, 1/3, 1/3$ ) of each gaussian .
- I used  $\epsilon = 1e-9$  for convergence of mean.
- $m$ =global mean,  $s$ =global standard deviation,  $M$ =mean,  $P$ =weights,  $S$ =Standard deviation)

A screenshot of a MATLAB Command Window. The window has a title bar 'Command Window' and a toolbar with icons for help, run, clear, and close. Below the toolbar is a yellow banner with the text 'New to MATLAB? Watch this Video, see Demos, or read Getting Started.' The main area of the window contains the following text:

```
Initialization:
P1 = 0.333333,P2 = 0.333333,P3 = 0.333333
M1 = 17.664156,M2 = 17.564156,M3 = 17.464156
S1 = 3.704687,S2 = 3.204687,S3 = 2.704687

End Parameters:
P1 = 0.326562,P2 = 0.329688,P3 = 0.343750
M1 = 14.003722,M2 = 17.817080,M3 = 20.703991
S1 = 1.472335,S2 = 2.410652,S3 = 0.497556

Iterations for convergence:1153

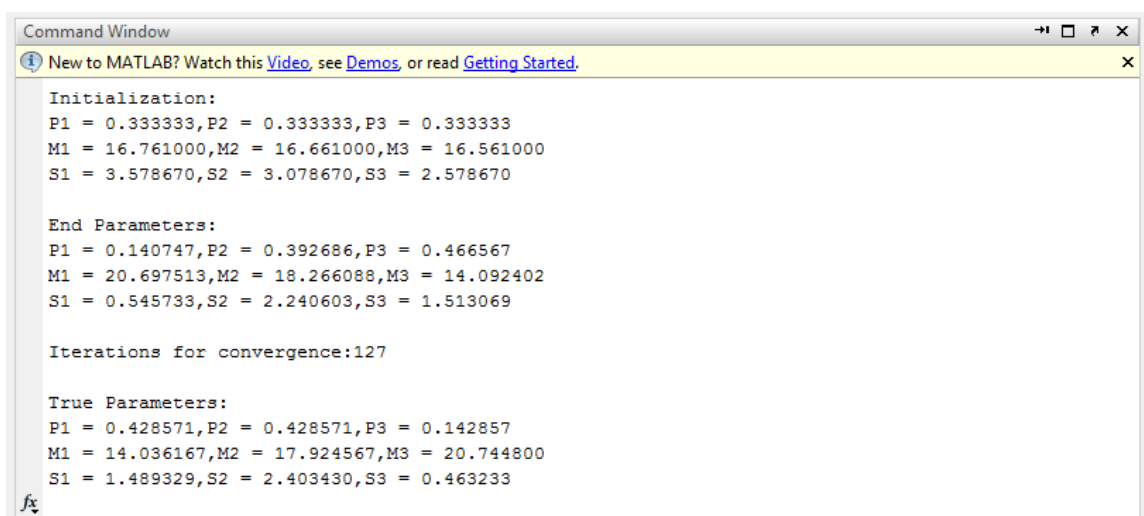
True Parameters:
P1 = 0.333333,P2 = 0.333333,P3 = 0.333333
M1 = 14.036167,M2 = 17.924567,M3 = 20.731733
S1 = 1.489329,S2 = 2.403430,S3 = 0.445425
```

Figure 1

2. Conduct a small experiment where you compare two different scenarios. Make a hypothesis about what outcome you may find before running the experiment, and then evaluate the hypothesis. Choose one of the following:

- Varying the number of training points during training (what happens to parameters)?
- Varying the number of Gaussians used to model the data (more means == better fit)?
- Compare and contrast diagonal versus full covariance matrices in 2-d data. (Here is some 2-d data for download: 2-dimensional data).
- An experiment of your own design.

- I varied the number of training points. I tested 2 subsets (1-700 points, 200-800 points (indices of original points)).
- I kept the initializations of means same as in question 1 (mean ( $m+0.1, m, m-0.1$ ), standard deviations ( $s+0.5, s, s-0.5$ ) and prior probabilities ( $1/3, 1/3, 1/3$ )) of each gaussian. However global mean/standard deviation here is the mean/standard deviation of the reduced set of training points.
- ( $m$ =global mean,  $s$ =global standard deviation,  $M$ =mean,  $P$ =weights,  $S$ =Standard deviation)
- I choose epsilon =  $1e-9$  for convergence similar to question 1
- Reducing the number of training data points should I believe affect the end parameters. They will now adapt based on the new training points. I think that the parameters will depend on how many points of each Gaussian (true) are there in the considered subset.
- The results for different number of training points are shown in Figures 2, 3. The end parameters were close to the true parameters in each case except for the weights. In figure 2, I expected to observe a larger value for  $P1$  and a smaller value for  $P3$ . But this wasn't the case.
- This problem wasn't seen in the second scenario I tested. This is probably because the bias in the first subset of points was quite less on the 3<sup>rd</sup> Gaussian.



```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

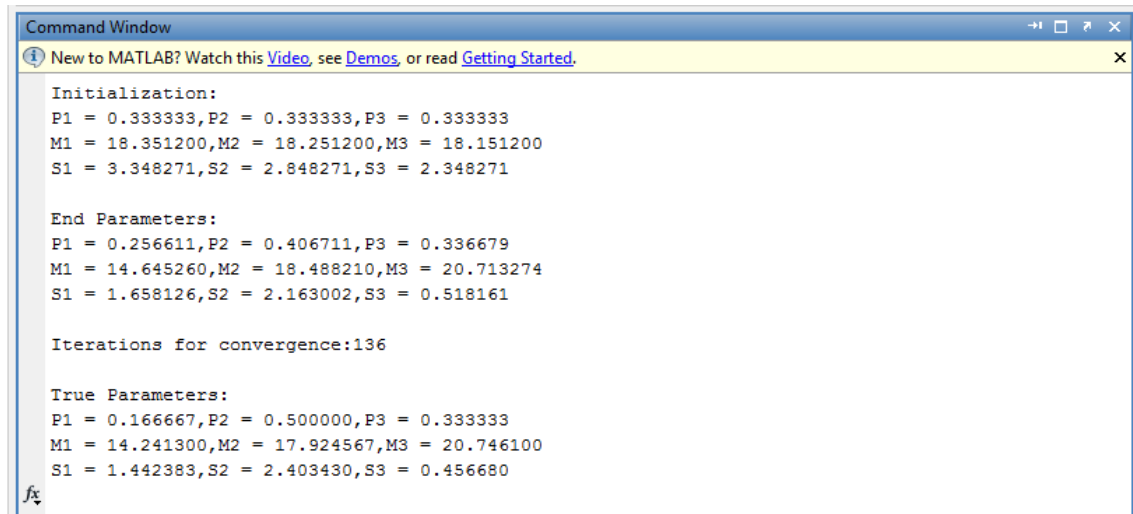
Initialization:
P1 = 0.333333,P2 = 0.333333,P3 = 0.333333
M1 = 16.761000,M2 = 16.661000,M3 = 16.561000
S1 = 3.578670,S2 = 3.078670,S3 = 2.578670

End Parameters:
P1 = 0.140747,P2 = 0.392686,P3 = 0.466567
M1 = 20.697513,M2 = 18.266088,M3 = 14.092402
S1 = 0.545733,S2 = 2.240603,S3 = 1.513069

Iterations for convergence:127

True Parameters:
P1 = 0.428571,P2 = 0.428571,P3 = 0.142857
M1 = 14.036167,M2 = 17.924567,M3 = 20.744800
S1 = 1.489329,S2 = 2.403430,S3 = 0.463233
```

Figure 2: 1-700 points

A screenshot of the MATLAB Command Window. At the top, there is a yellow banner with the text "New to MATLAB? Watch this Video, see Demos, or read Getting Started." Below this, the command window displays the output of an EM algorithm. It starts with "Initialization:" followed by three rows of parameter values for P, M, and S. Then it shows "End Parameters:" with another set of three rows for P, M, and S. Next, it says "Iterations for convergence:136". Finally, it displays "True Parameters:" with three rows of parameter values for P, M, and S. The window has a standard MATLAB interface with a title bar and window controls.

```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

Initialization:
P1 = 0.333333,P2 = 0.333333,P3 = 0.333333
M1 = 18.351200,M2 = 18.251200,M3 = 18.151200
S1 = 3.348271,S2 = 2.848271,S3 = 2.348271

End Parameters:
P1 = 0.256611,P2 = 0.406711,P3 = 0.336679
M1 = 14.645260,M2 = 18.488210,M3 = 20.713274
S1 = 1.658126,S2 = 2.163002,S3 = 0.518161

Iterations for convergence:136

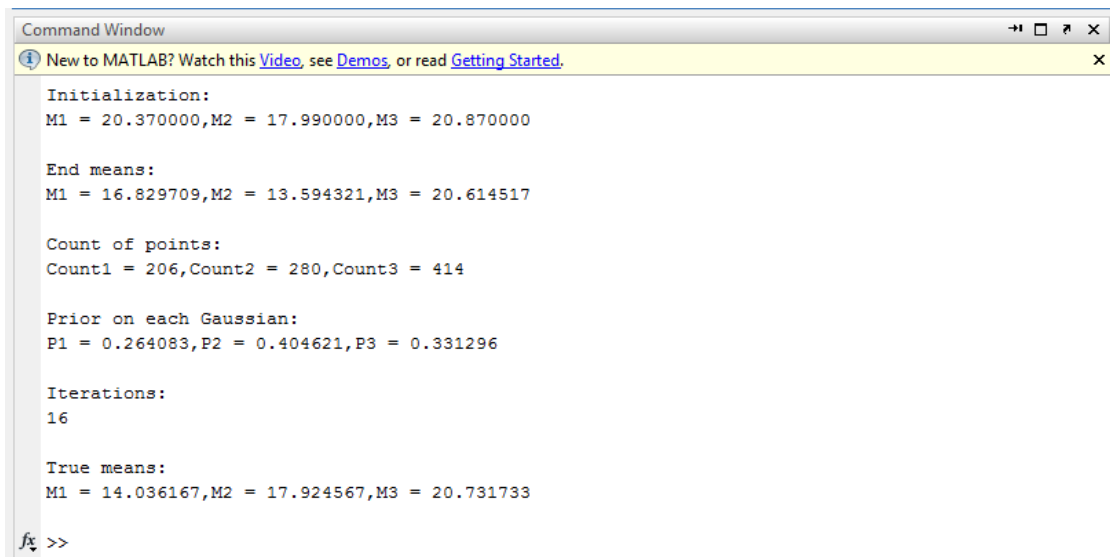
True Parameters:
P1 = 0.166667,P2 = 0.500000,P3 = 0.333333
M1 = 14.241300,M2 = 17.924567,M3 = 20.746100
S1 = 1.442383,S2 = 2.403430,S3 = 0.456680
```

Figure 3: 200-800 points

### 3. Extra Credit

Implement k-means clustering on the 1-dimensional data. Comment on how the results differ from the EM results. You may need to compute the probability of each data point corresponding to a class to do a meaningful comparison. Oh, by the way: the first 300 points were generated from class 1, the second 300 from class 2, and the third 300 from class 3. Are either of these better for classifying the data?

- Figure 4 shows the results of the k-means algorithm with random initialization of means where  $k=3$ .
- To get the probability of a point belonging to a class, I normalized over the 3 Euclidian distances, i.e. – added the 3 euclidian distances (from each of the means) and divided each by the sum. Further to get the prior of each class I normalized the soft counts of each gaussian.
- The means from both EM and  $k(3)$ -means is almost similar. So, at least for the given data, I don't think either of them is better at classifying. However, k-means converges really quickly. Also, the soft counts don't quite match up to the EM results (which gives results more closer to the true value).



The image shows a MATLAB Command Window with a yellow header bar containing a message: "New to MATLAB? Watch this [Video](#), see [Demos](#), or read [Getting Started](#)." Below the header, the following text is displayed in a monospaced font:

```
Initialization:
M1 = 20.370000,M2 = 17.990000,M3 = 20.870000

End means:
M1 = 16.829709,M2 = 13.594321,M3 = 20.614517

Count of points:
Count1 = 206,Count2 = 280,Count3 = 414

Prior on each Gaussian:
P1 = 0.264083,P2 = 0.404621,P3 = 0.331296

Iterations:
16

True means:
M1 = 14.036167,M2 = 17.924567,M3 = 20.731733

fx >>
```

The Command Window interface includes standard window controls (minimize, maximize, close) in the top right corner and a cursor icon at the bottom left.

Figure 4: