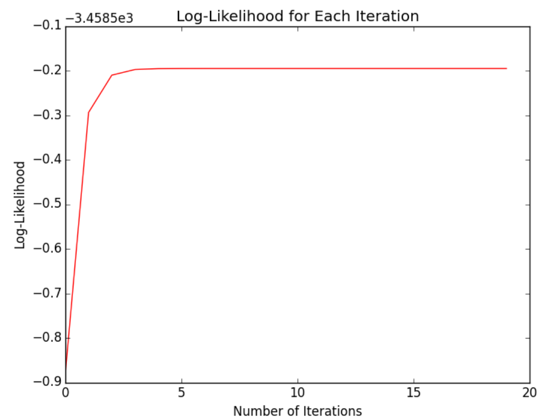
SangHee Kim

EECS 349

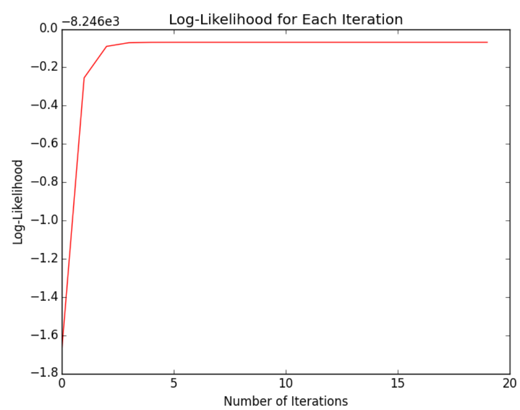
11/13/15

1B) I believe that the program has converged; visually, the graph has flat curve, and numerically, the log-likelihood did not change by more than 0.000001 after few iterations.

Class 1 –
Number of Gaussians = 2
mu = [9.7749, 29.5825]
sigma^2 = [21.9228, 9.7838]
weight = [0.5977, 0.4023]
Log-likelihood converges to -3458.695

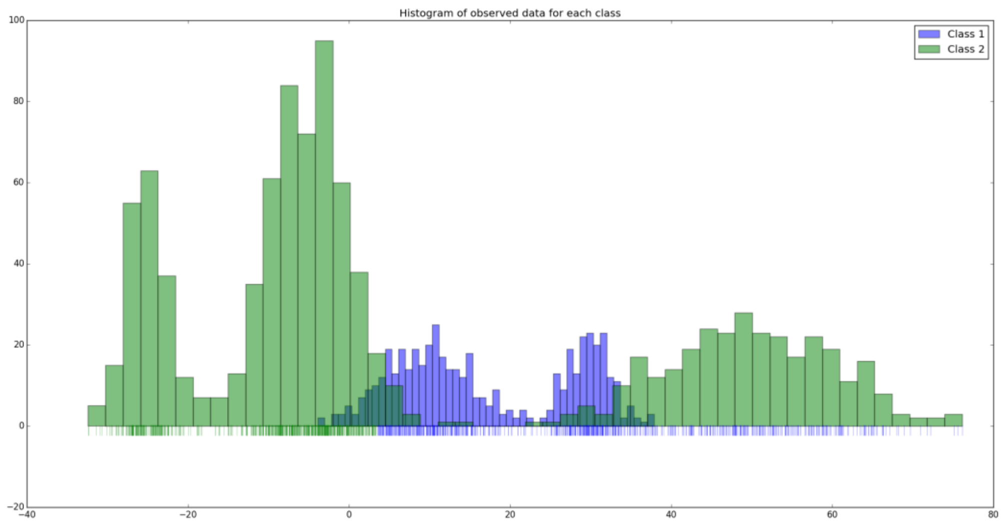


Class 2 –
Number of Gaussians = 3
mu = [-24.8228, -5.0602, 49.6244]
sigma^2 = [7.9473, 23.3227, 100.0243]
weight = [0.2037, 0.4988, 0.2975]
Log-likelihood converges to -8246.069



1D) Accuracy on the gmm_test.csv = 75.33%

For some reason, the estimated parameters from Class 2 threw off the accuracy of the classification. As seen from the graph, the classification is usually correct until after it passes class 1, after which it classifies all class 2 data points to class 1. After looking at the code for extended periods of time, I still was unable to find what was wrong with it, since both class 1 and class 2 went through the same calculation.



2A) It would be possible to use the closed-form solution to find the parameters; we would simply treat the Gaussian we know as the single Gaussian available and disregard the others during the calculation. Since we know that the Gaussian generated the data point x, it means that we also know that the Gaussian is the one with the maximized probability of observing the data. With this in mind, we can use the equation

$$\Theta^* = \underset{\operatorname{argmax}\Theta}{\sum_{i=1}^{n}\left(\frac{-(x_i-\mu)^2}{2\sigma^2} - \log\sigma\right)}$$

and the resulting partial derivatives

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^2$$

to find the best parameters.

2B) You would have to use the EM algorithm in this case; not knowing what Gaussian generated each data point means that one would have to find the best estimated parameters to find the 'correct' Gaussian. Having multiple Gaussian also means that there will be no closed-form solutions, and since there's no one Gaussian we can be sure to be the answer, we will have to go through and compare the probability of each of the data points being observed in each Gaussian one at a time, which is what the EM algorithm is meant for in the first place.