Expectation Maximization Algorithm - Report

Before explaining the results of Expectation-Maximization Algorithm on the given data, let talk about what Expectation-Maximization Algorithm is and how it does do. It includes two steps as we understand by its name. In expectation step, expected value of each data point is calculated by using initial estimated parameters and given data. This step enables to get the values of the missing data. Then, in maximization step, the maximum likelihood parameters are determined by using the parameters in expectation step. After expectation-maximization, new parameters are obtained by using initialized parameters and data. In Pattern recognition and machine learning book(2006), how to calculate the parameters for expectation-maximization.

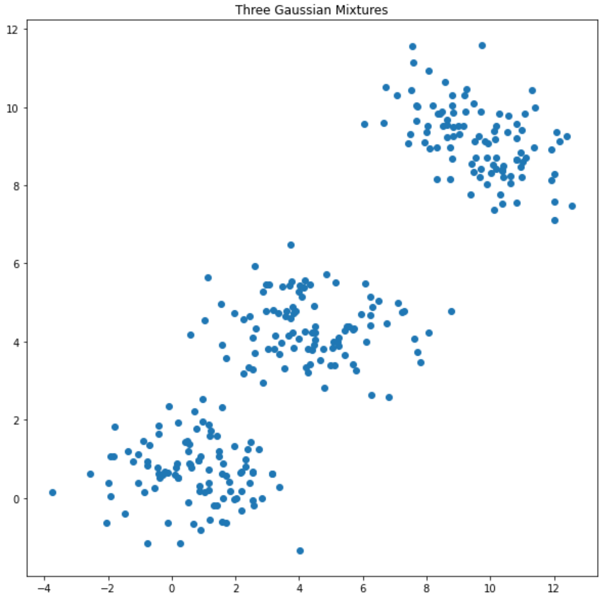

Figure 1: Visualize given data

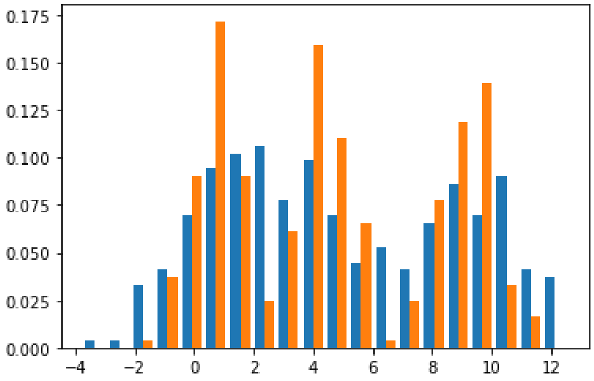

Figure 2: Histogram of given data

Before not applying any algorithm for data, our data includes three groups as we clearly see in the Figure 1. Also, Figure 2 shows histogram of data; that is, it says the distribution of data. Now, let talk about our initial parameters. Initial mean (μ_k) values are selected randomly from the data points. For our purpose, we need to 3 points to initialize mean. Also, a sigma value is defined to draw contours for classification of the data.

$$sigma = \begin{bmatrix} 0.000003 & 0 \\ 0 & 0.000003 \end{bmatrix}$$

Also, we have 3 covariance matrices 2 by 2 and they are defined as equal. Initial covariance matrices can be selected arbitrarily. Initial π_k's are also selected as equal. Since we have 3 Gaussians, each π_k is $\frac{1}{3}$

$$\Sigma_k = \begin{bmatrix} 2.000003 & 0 \\ 0 & 2.000003 \end{bmatrix}$$

$$\pi_k = [0.333333330.333333330.33333333]$$

Therefore, our initial likelihood value is calculated as

$$initial\ loglikelihood = [0.9205115469167344, 0.9205115469167344, 0.9205115469167344]$$

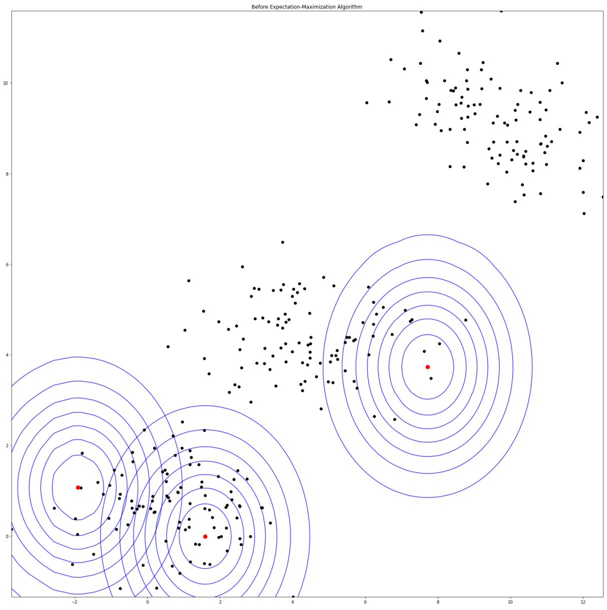

Figure 3: Before Expectation-Maximixation Algorithm

As we see in Figure 3, before Expectation-Maximixation Algorithm, we are not arrived true classification of data. These contours are very random. We want to create true classification of data using 3 Multivariate Gaussians.

By using steps for Expectation-Maximization with 3 Gaussian mixtures in Bishop's Pattern recognition and machine learning (2006) book, we get new parameters.

$$\mu_{new} = \begin{bmatrix} 0.796489 & 0.791044 \\ 4.41856 & 4.39509 \\ 9.60423 & 9.13031 \end{bmatrix}$$

$$\Sigma_{new} = \begin{bmatrix} 2.29495 & 0.157785 \\ 0.157785 & 0.986119 \end{bmatrix}$$

$$\pi_{new} = [0.3482520.3164230.335325]$$

By using these values, the final likelihood values after applying expectation-maximization algorithm are obtained as

$$final\ loglikelihood = [0.71327361.362840611.80766798]$$

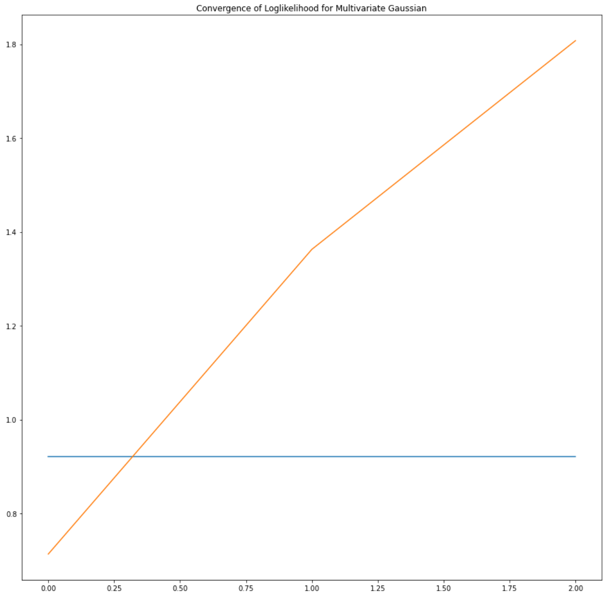

Figure 4: Convergence of Likelihood

In Figure 4, blue line shows final likelihood values after applying expectation-maximization and orange one indicates the likelihood value by using initial values. Here, we can say likelihood function is converged but it would draw better. However, by using multivariate normal distribution from scratch, I get some errors while drawing according to using iterations.

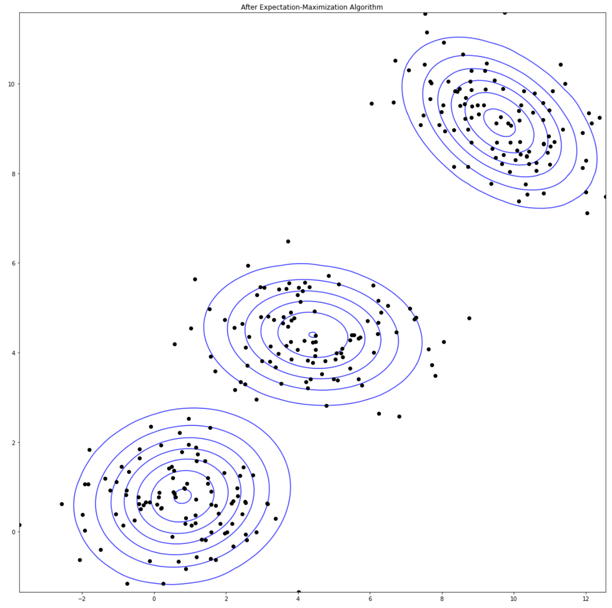

Figure 5: After Expectation-Maximixation Algorithm

In Figure 5, we see better classification of data. However, it has still some outliers or anomalies. I tried some different initial values for the parameters but the problem were continued.

REFERENCE

- Bishop, C. M. (2006). Pattern recognition and machine learning. springer.

- Borman, S. (2004). The expectation maximization algorithm-a short tutorial. *Submitted for publication*, 41.

- Moon, T. K. (1996). The expectation-maximization algorithm. *IEEE Signal processing magazine*, 13(6), 47-60.

- Vlassis, N., & Likas, A. (2002). A greedy EM algorithm for Gaussian mixture learning. *Neural processing letters, 15*(1), 77-87.

- Bernd Klein, B. (2010)
  https://www.python-course.eu/expectation$_{m}aximization_{a}nd_{g}aussian_{m}ixture_{m}odels.php$
  $Sicotte, X.B.(2018)$
  $https : //xavierbourretsicotte.github.io/MLE_{M}ultivariate_{G}aussian.html$