ANLY601 Take Home Assignment 3

Xunge Jiang

Feb 10, 2019

1 Using EM algorithm to derive the update equation for the mean parameter for each Gaussian Mixture Component j.

E-Step:

$$Q(\theta;\theta_t) = \underset{j|x}{E}[\sum_{i=1}^{N} ln[P(x_i|j)p_j]] = \sum_{i=1}^{N} \underset{j|x}{E}[ln[P(x_i|j)p_j]]$$

$$= \sum_{i=1}^{N}\sum_{j=1}^{J} P(j|x_i,\theta_t) ln(P(x_i|j)p_j)$$

Since $p(x)$ has a Gaussian Distribution, then

$$Q(\theta;\theta_t) = \sum_{i=1}^{N}\sum_{j=1}^{J} P(j|x_i,\theta_t)[-\frac{1}{2}ln{\sigma_j}^2 - \frac{1}{2{\sigma_j}^2}||x_i-\mu_j||^2 + lnp_j)]$$

M-step:
To maximize $Q(\theta;\theta_t)$ w.r.t to μ, we set $\frac{\partial Q}{\partial \mu_j} = 0$

$$\frac{\partial}{\partial \mu_j}\sum_{i=1}^{N}\sum_{j=1}^{J} P(j|x_i,\theta_t)[-\frac{1}{2}ln{\sigma_j}^2 - \frac{1}{2{\sigma_j}^2}||x_i-\mu_j||^2 + lnp_j)] = 0$$

Since two terms don't contain μ_j, thus, the above equation can be simplified to

$$\frac{\partial}{\partial \mu_j}\sum_{i=1}^{N} P(j|x_i,\theta_t)(-\frac{1}{2{\sigma_j}^2}||x_i-\mu_j||^2) = 0$$

The coefficient before the L2 norm can be cancelled, also,

$$\frac{\partial}{\partial \mu_j}(-||x_i-\mu_j||^2)) = 2(x_i-\mu_j)$$

we get

$$2\sum_{i=1}^{N} P(j|x_i,\theta_t)(x_i-\mu_j)=0$$

$$\sum_{i=1}^{N} P(j|x_i,\theta_t)x_i=\sum_{i=1}^{N} P(j|x_i,\theta_t)\mu_j$$

Result:

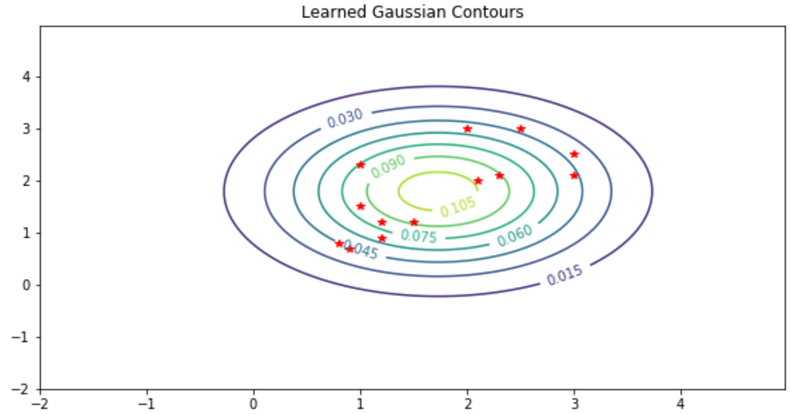
$$\mu_{j_{t+1}}=\frac{\sum_{i=1}^{N} P(j|x_i,\theta_t)x_i}{\sum_{i=1}^{N} P(j|x_i,\theta_t)}$$

2 Assume data set as follow: x[0] = [1, 1.5, 1.2, 1.2, .9, .8, 1, 2.3, 2.1, 2, 3, 2.5, 3] x[1] = [1.5, 1.2, 1.2, .9, .7, .8, 2.3, 2.1, 2, 3, 2.5, 3, 2.1]

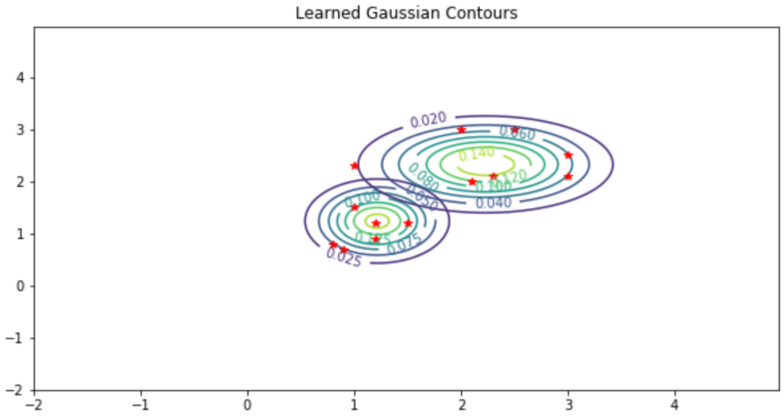
2.1 a) Run your code with 1 component, 2 components, and 3 components. What epsilon did you use? Which component number "fits" the data best? How did you make this determination?

I choose epsilon = 0.001. And following three graphs are final results of EM algorithm with m = 1, 2, 3 components.

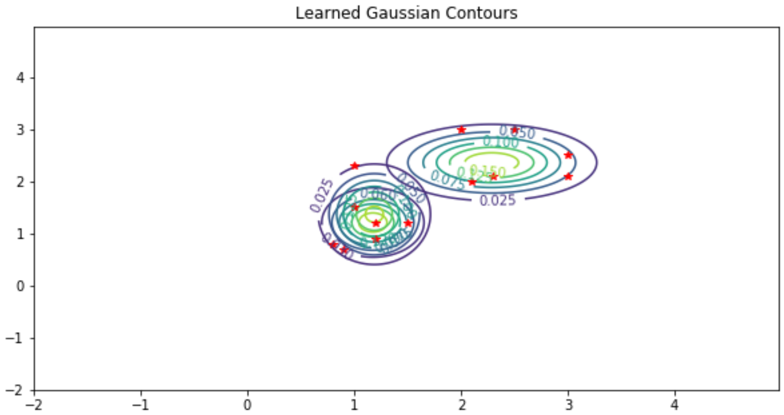
Result of using 1 component:



Result of using 2 components:



Result of using 3 components:

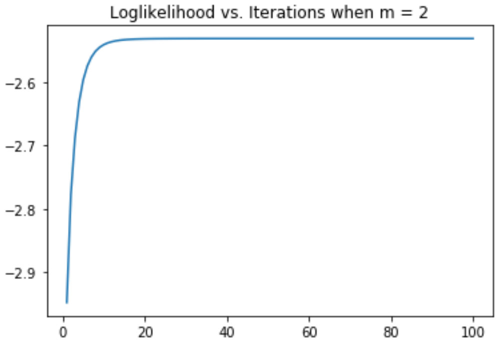


For component number 2, the contour plot fits the data best. Since first, by looking at the raw data (red dots), the data looks like it has two clusters so two clusters would be a better fit. Then by looking at the final result contour plots, when m = 3, two of the three contour plots are overlapping which indicates the additional one is unnecessary. When m = 1, it fits ok but the contour is loose and doesn't look good enough when compares to the plots when m = 2. When m = 2, we can clearly see two contours are covering all points and the boundaries are clear.

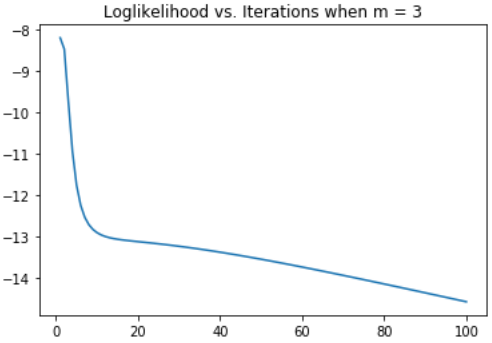
2.2 b) To your existing code, add a function that plots (in a different figure) the log likelihood as a function of iteration. Can you use this information as a stopping criterion rather than epsilon? Can you use this information to determine which number of components is best?

The following plots show the log likelihood function vs. iteration when m = 2 and 3.

Log plot of using 2 components:



Log plot of using 3 components:



I think this plot can be used as a stopping criterion as well. For stopping the algorithm, the criterion on log likelihood function is that it is converging to a number. And when iterations go larger, the y value of the plot will not change by a lot. This can be clearly seen by the plot when m = 2 that after around 20 iterations, log likelihood function becomes stable.

Also, I think it can be useful to determine which number of components is better but not as clearly visualized as contour plot and it can be misleading as well. For the plots above, we can see that when m = 2, it is showing an almost vertical straight line when iterations goes larger, but when m = 3, the line doesn't seem to converge at that point. Thus, comparing by these two cases, we can pick case when m = 2 to be the better one. However, we cannot visually see if two clusters have covered all points and we don't know if we are overfitting or underfitting. Especially, in case m = 3, it looks like it is converging as the line becomes smoother but actually it is overfitting to the dataset.

2.3 c) Given your analysis in Part b), what was the learned parameters of the mixture model that best fit the data

Case m = 2 components,
result of μ_0: [1.22, 1.24]
result of μ_1: [2.23, 2.33]
result of σ_0: [0.34, 0.40]
result of σ_1: [0.59, 0.46]
result of prior0: 0.493
result of prior1: 0.507