Report for Assignment 3

Zou Yanyan

March 29, 2016

Task 1

0.1

1. The data is generated from the following spherical Gaussian distribution:
$P(x|\mu,\sigma^2) = \frac{1}{(2\pi\sigma^2)^{\frac{d}{2}}} exp(-\frac{1}{2\sigma^2}||x-\mu||^2)$

We assume the data comes from a mixture of 2 spherical Gaussian distribution. Furthermore, we also assume a priori where the clusters are, i.e., which points are generated from which distribution. Since the clusters may vary by size as well, we also include parameters p_1, p_2 that specify the frequency of points we would expect to see in each cluster. Also, $p_1 + p_2 = 1$.

Now, we need to estimate five parameters, $(1), \sigma_1, p_1, \mu^{(2)}, \sigma_2$. Also, we try to uncover hidden labels, i.e. find the underlying clusters where points are generated from the same spherical Gaussian distribution.

0.2 Hard EM

Before implementation, we show the raw data.

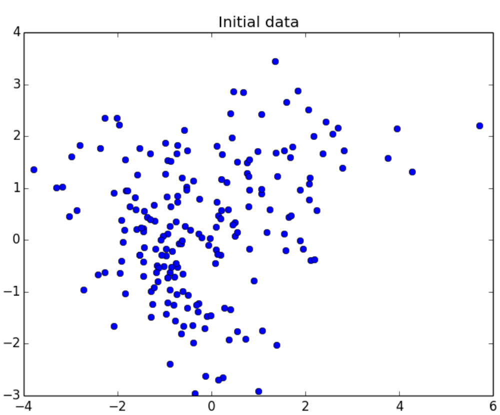

Figure 1. Raw data

To implement this hard EM algorithm, we need to first initialize the mixture parameters which we mentioned in question 1. We initialize the means of each cluster as in the k-means algorithm, randomly setting values. Also, we set variances for each cluster all equal to the overall data variance. This ensures that Gaussian can all "see" all the data points (spread is large enough) that we do not assign points to specific clusters too strongly. Since we have no information about the cluster sizes, we will set $p_i = \frac{1}{k}, i = 1, 2, ..., k$. In this case, k is 2. Moreover, we predefine the error threshold. The algorithm will terminate when the difference between current and last likelihood is smaller than the threshold, where we can consider the algorithm converges. After implementing hard EM, we show some samples of running the hard EM-algorithm.

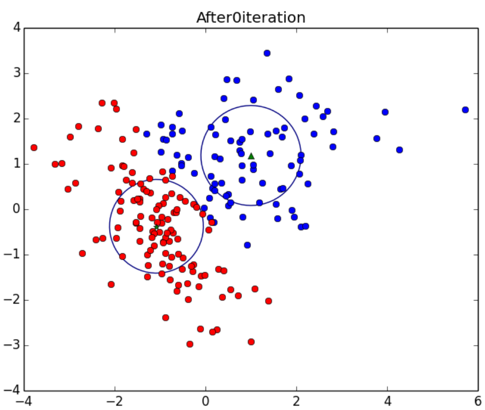

Figure 2. Initial

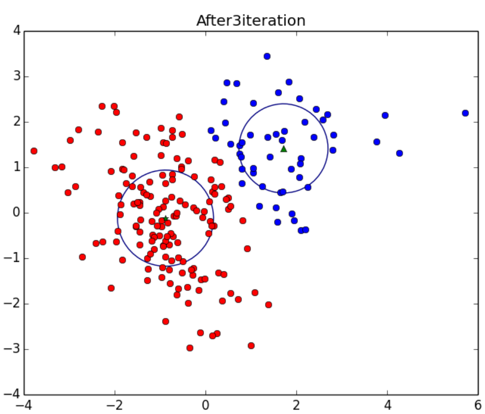

Figure 3. After 3 iterations

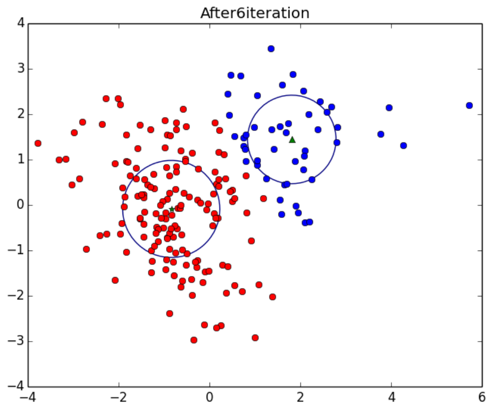

Figure 4. After 6 iterations

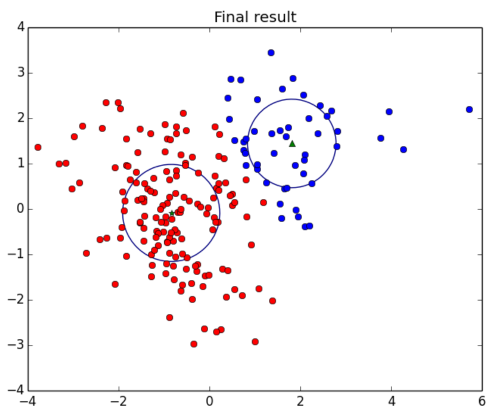

Figure 5. Final result

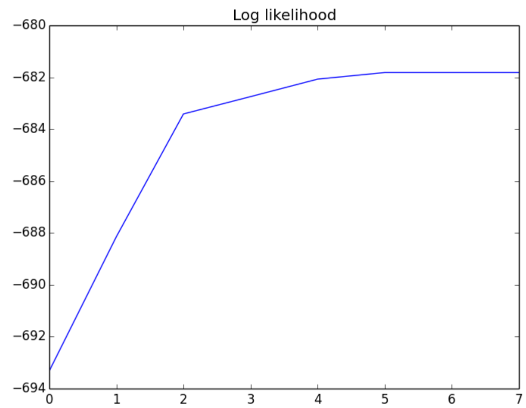

Figure 6. Log joint likelihood

hard em	k =1	k =2
initial_p	0.5	0.5
initial_mean	[-0.33490708, -0.05961256]	[0.05233886, 0.26039476]
initial_variance	2.01377335	2.01377335
initial_likelihood	-702.9680195	
final_p	0.76	0.24
final_mean	[-0.84015567 -0.08720252]	[1.81346957 1.44428993]
final_variance	1.14559217	0.9522242
final_likelihood	-681.8172729	

Figure 7. Results

Based on the above observations, the value of log joint likelihood increases after each iteration before converge. In the first part, it increases fast and then it increases slower.

0.3 Soft EM

Now, we implement the soft EM algorithm to solve this problem. The method to initializing the model and terminating the algorithm is the same in the hard EM.

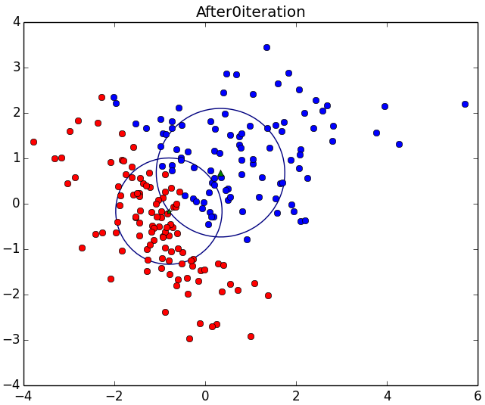

Figure 8. Initial

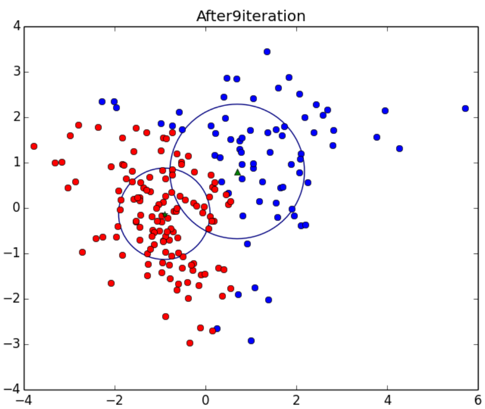

Figure 9. After 9 iterations

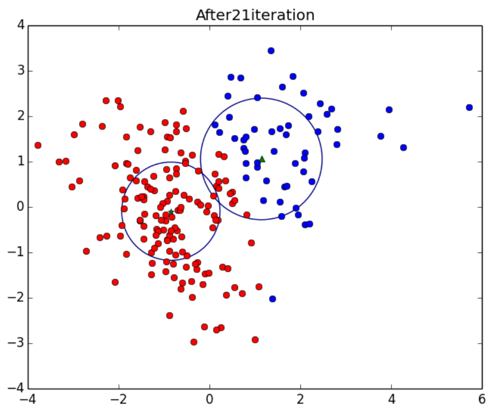

Figure 10. After 21 iterations

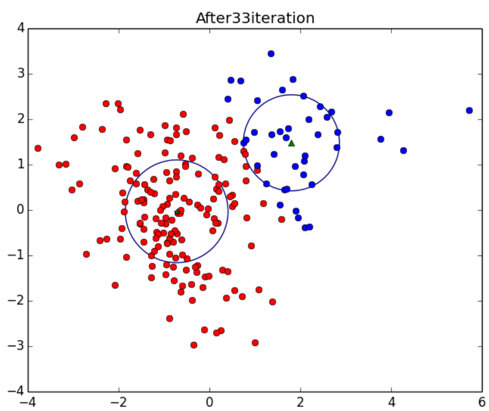

Figure 11. After 33 iterations

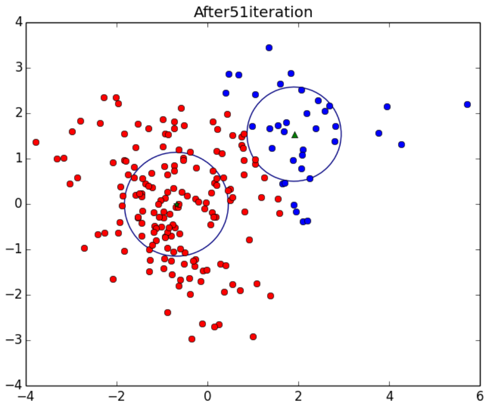

Figure 12. After 51 iterations

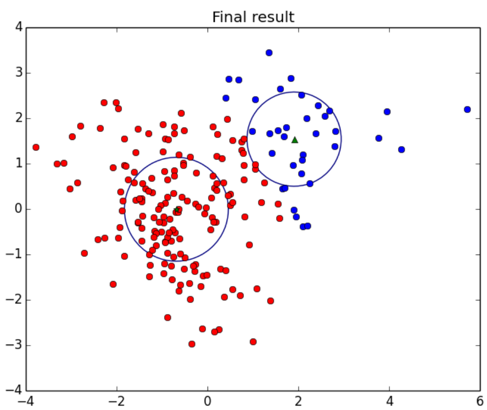

Figure 13. Final results

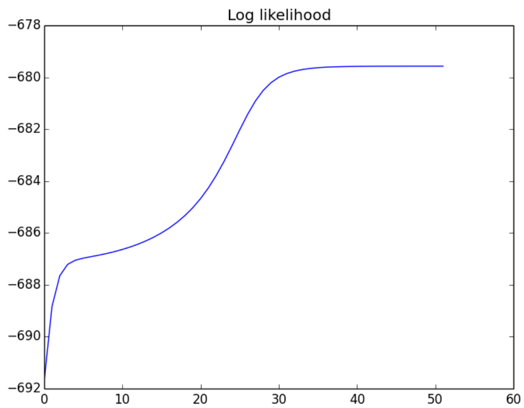

Figure 14. Log joint likelihood

soft em	k = 1	k =2
initial_p	0.5	0.5
initial_mean	[-0.51273099, -0.95468794]	[0.67226552, 0.04067815]
initial_variance	2.01377335	2.01377335
initial_likelihood	-731.0455279	
final_p	0.814858	0.18514194
final_mean	[-0.68327142 -0.00573124]	[1.90925743 1.53949897]
final_variance	1.31134144	1.07826156
final_likelihood	-679.5713974	

Figure 15. Results

Similarly, the value of log joint likelihood increases after each iteration before convergence. Compared to the performance in hard EM, it takes longer time to converge.

EM algorithm is guaranteed to monotonically increase the log-likelihood of the data under the mixture models. However, it may only find a locally optimal solution. Both of soft and hard EM algorithm are sensitive to the initialization of models. That is the reason why we may get different results after running the algorithm. Here, I only present the better result I got. A good initialization could lead algorithm to converge fast.

Task 2

Induction