1. A plot of the sum-squared-error (divided by the sample size n) as a function of iteration number in the K-means algorithm. Only show the clustering with the lowest sum-squared-error among all r tries.

## For dataset1:

I run for r =10 and have the lowest ever SSE I found for that iterations is 2819.6 the final centroids, for 2 clusters:1 and 2 are :

|  |  |  |
| --- | --- | --- |
| Centroid | X1 | X2 |
| C1 | 3.0170 | 0.0456 |
| C2 | -0.1308 | -0.0531 |

And the two clusters are shown as follows:

Convergence occurs on the fifth iteration.

## 

## 

Figure 1 Two Centroids and 2 cluster after convergence, the block boxes are the last updated centroids

## 

Figure 2 1.762 is the lowest SSE/N , N=1600

## For dataset2:

I run for r =10 and have the lowest ever SSE I found for that iterations is 2819.6 the final centroids, for 2 clusters:1 and 2 are :

|  |  |  |
| --- | --- | --- |
| Centroid | X1 | X2 |
| C1 | -0.5218 | 0.0382 |
| C2 | 1.6465 | -0.8508 |
| C3 | 1.7219 | 1.0477 |

And the two clusters are shown as follows:



Figure 3 Three Centroids and 2 cluster after convergence



Figure 4 1028 is the lowest SSE/N , N=1500

1. A sample plot of the log-likelihood as a function of iteration number during EM. Only show the one with the highest log-likelihood.

## For dataset1:

Initialization Strategy 1 (Random):

Figure 5 shows the random initialization where converiences are consider the global converiance multiplied by 1/k, where k equals 2 in this case.



Figure 5 Randomintializationn where means are random and covs are the global cov alike

After EM converges means backed to its right position, and the new converiences take the right elapsed representing new clusters 1 and 2, Figure 6. Highest logliklihood achieved = -5344, and it can be seen clearly that logliklihood is a monotone function, figure 7.



Figure 6



Figure 7: logliklihood is a monotone function as expected

Initialization Strategy 2:

In this part, KMEAN was used to initialize mean, and points from each cluster are used to compute conversances. Figure 8 shows 2 means are in center of ellipsoids, and these ellipsoids are corresponds to conversances. It is intuitive that it takes the shape of cluster data as in figure 1. Results are identical, and logliklihoods are equally alike for the two initialization strategy



Figure 8



Figure 9

Note: EM has a smooth assignment in which each point is a part of cluster with a certain probability. Thus for visualization purpose only, I assign each point to the max probability for being a part of a cluster.



Figure 10: logliklihood is a monotone function as expected

## For dataset2

Initialization Strategy 1:



Figure 11



Figure 12



Figure 13: logliklihood is a monotone function as expected

Initialization Strategy 2:



Figure 14: coveraince ellipsoid clearly reflects the three clusters in figure 3

After running EM figure 15 and 16 shows the new clusters and logliklihood. Figure 16 starts with a higher logliklihood, and that might be for a better selecting of means comparing with random intialization in figure 11.



Figure 15



Figure 16: logliklihood is a monotone function as expected

If you run the code, the exact numeric results for both initializing and last updating means & conveniences will be printed once the algorithm converge. I thought to not put in the report but to visualize it.

3) Add some brief comments on the two different methods for the initialization for Part II.

In general:

Both KMEAN and EM can find local solution but not necessarily global. Resulting logliklihood came back as expected as a monotone function. If I increase r the result would show a higher value of logliklihhood.

For initialization comment:

1. Results tends to very slightly change each time I run the code because initial parameters effects the end results.
2. In initializing EM by evoking KMEAN can firstly start with **a higher value logliklihood** (which is good) in general and decrease number of iteration significantly. This case occurs especially for dataset1 since it converges in only 11 iteration which is better than 98 in the random method.. Meanwhile For the second dataset evoking KMEAN has no really effect since KMEAN has not do a good job in determining a close form clusters if we compare it with ground truth. In general, KMEAN would help EM if there is no high degree of overlap, otherwise KMEAN step in EM is useless as in dataset2.

Ground truth for dataset1 and dataset2 are in figure 17 and 18 respectively:



Figure 17 Ground truth dataset1



Figure 18 ground truth dataset2

References:

[1] Adebisi, A. A., Olusayo, O. E., & Olatunde, O. S. (2012). An Exploratory Study of K-Means and Expectation Maximization Algorithms. *British Journal of Mathematics & Computer Science*, *2*(2).

[2] PLOT\_GAUSSIAN\_ELLIPSOIDS, Gautam Vallabha, File ID: #16543, time access: 4/13/14