Homework 4 for Deep Learning and Data Mining

Xiaoqian Chen

June 3, 2019

1 generation of 5-group data

In the 2 dimensional real space, we generate 5 group of data, and 500 points for each group.

- Given a square with length L=100, generating 5 pairs of uniformly distributed random numbers $c_i = (x_i, y_i), i = 1, ..., 5$, which is the central point of i-th group.
- Calculate the distance between central points $d(i,j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, i \neq j$, and save the smallest distant with other point $v_i = min\{d(i,j)\}, j \neq i$.
- For i-th group, generate 500 random points with the 2-dim Gaussian distribution $N(c_i, \frac{v_i}{2})$. To be more specific, points $p_i = (x_i, y_i), i = 1, \ldots, 2500$, x-coordinate $x_i \sim N(c_i(1), \frac{v_i}{\sqrt(2)})$ and y-coordinate $y_i \sim N(c_i(2), \frac{v_i}{\sqrt(2)})$.

Until now, we have totally 2500 points.
Our tasks is that we perform on a data set without labels is to find groups of data in our dataset which are similar to one another – what we call clusters.

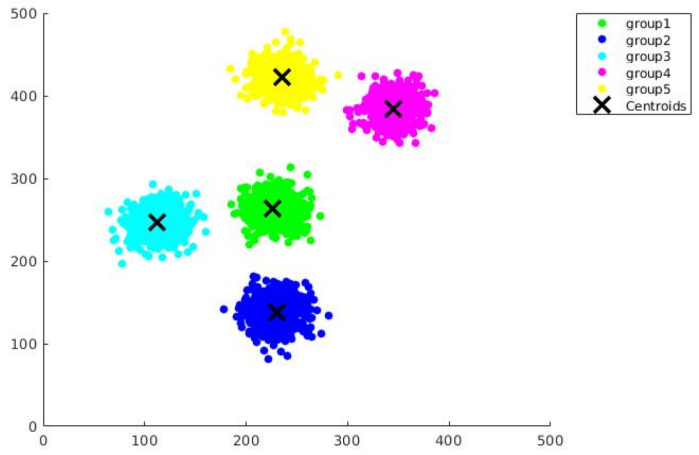


Figure 1: generated 5-group points

2 simulation of 5-group data

Our goal is to partition these points into 5 group with simulation. For classifying 2500 points $(x_i, y_i), i = 1, \ldots, 2500$ into 5 groups, {G1,G2,G3,G4,G5}, we use 2500×5 binary neurons V(i,j) to present the result. The desired result is that there are one of identity vector. $\{e1, e2, e3, e4, e5\}$ for each points. For example, $e1 = [1, 0, 0, 0, 0]$ means that this point is decided into group1, and so on.

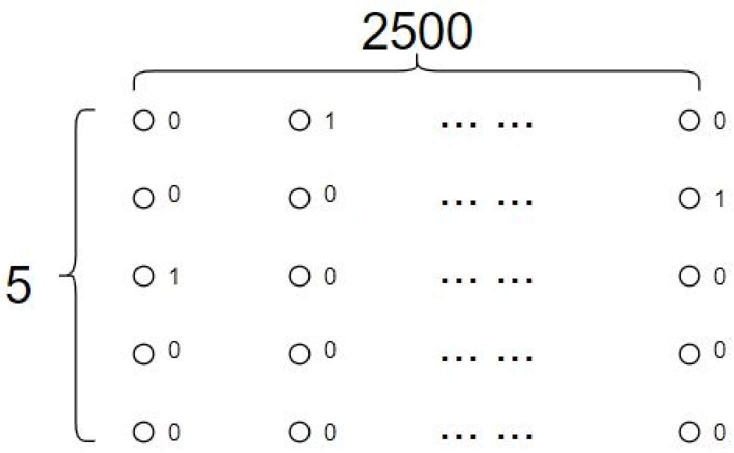


Figure 2: 2500*5 binary neurons

Construction of Cost function is consist of two parts. One part is for clustering and second part is to promise that there is only one active neuron of each 5 neurons.

Firstly, for our 2500 points$\{p_1, p_2, \ldots, p_{2500}\} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_{2500}, y_{2500})\}$, with the corresponding simulated group $\{g_1, g_2, \ldots, g_{2500}\}$. We calculate 5 cluster centroids,$\{C1, C2, C3, C4, C5\}$ that it uses to define clusters.

$$C_k = (\frac{1}{N_k}\sum_{i=1}^{N_k} x_i, \frac{1}{N_k}\sum_{i=1}^{N_k} y_i), k = 1, 2, 3, 4, 5 \tag{1}$$

Where N_k is the number of the points belonging to group k.

For every cluster k, we compute point-to-cluster-centroid distances of all observations to each centroid.

$$D_k = \frac{1}{N_k}\sum_{i=1}^{N_k} ||p_i - C_k||, k = 1, 2, 3, 4, 5 \tag{2}$$

The cost function is the sum of the within-cluster, sum-of-squares point-to-cluster-centroid distances.

$$Cost = \sum_{k=1}^{5} D_k = \sum_{k=1}^{5}\sum_{i=1}^{N_k} ||p_i - C_k|| \tag{3}$$

Individually assign observations to a different cluster. If the reassignment decreases the cost function, we accept the change. Otherwise, we accept the change with probability

$prob = e^{-\Delta E/T}$. Repeat this process until there is no much change in the cost function or the maximum number of sweep is reached.
The temperature function:

$$T = 0.99^{sweep} \tag{4}$$

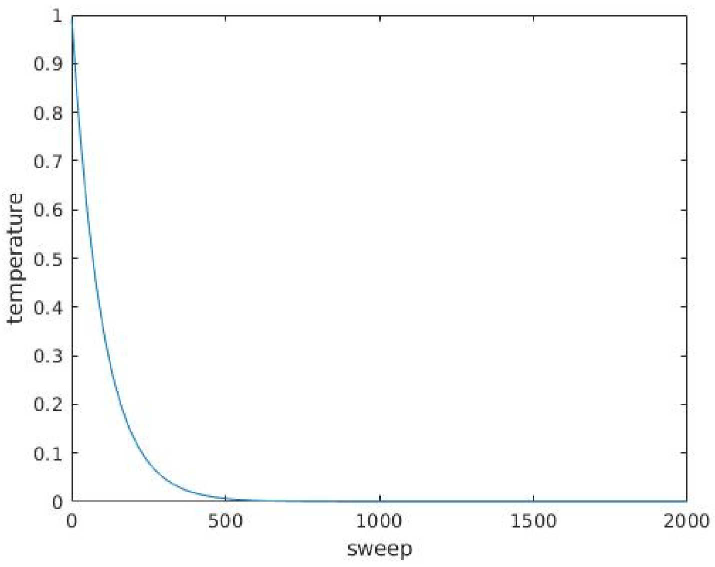


Figure 3: Temperature function

Our task is to partition 2500 points into 5 groups, by minimizing the cost function , which is sum-of-squares point-to-cluster-centroid distances.

$$MinCost = \sum_{k=1}^{5} \sum_{i=1}^{N_k} ||p_i - C_k|| \tag{5}$$

Algorithm:

1. Initialization, create 2500*5 binary neurons, where there is one active neurons for every 5 neurons.

2. Define max-sweep=2 000, sweep = 1: max-sweep

   In each sweep:

3. permuting the data and divide dataset into some batch (eg. 50 neurons in one bacth).

   In each batch:

4. reassigning points to different cluster randomly, where there is one active neurons for every 5 neurons.

5. calculate the new cost E_{new}.

6. calculate $\Delta E = E_{new} - E_{old}$.
7. If $\Delta E < 0$, accept the change. If $\Delta E > 0$, accept the change with probability $prob = e^{-\Delta E/T}$.
8. repeat step 3-7 until there is no much change or the maximum number of epochs is reached.

For monitoring the quality of clustering, we have three performance indicators, including cost, 90% qua tile point in the histogram of point-to-cluster-centroid distances and the correct rate.

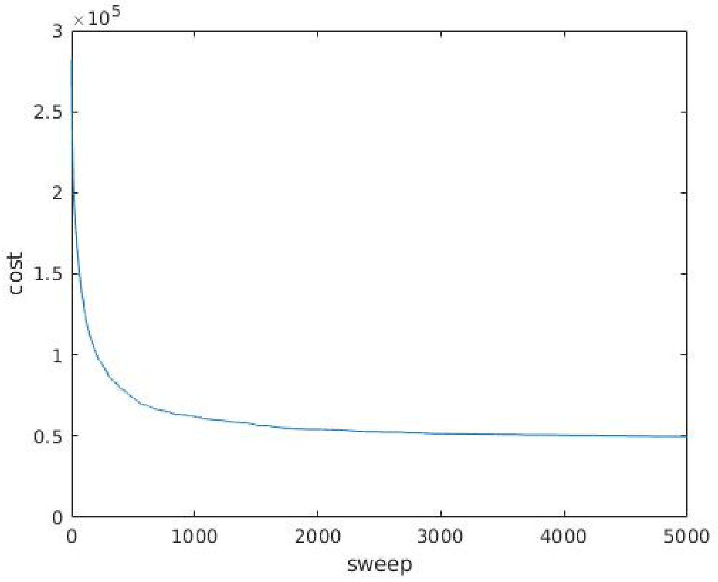


Figure 4: cost function

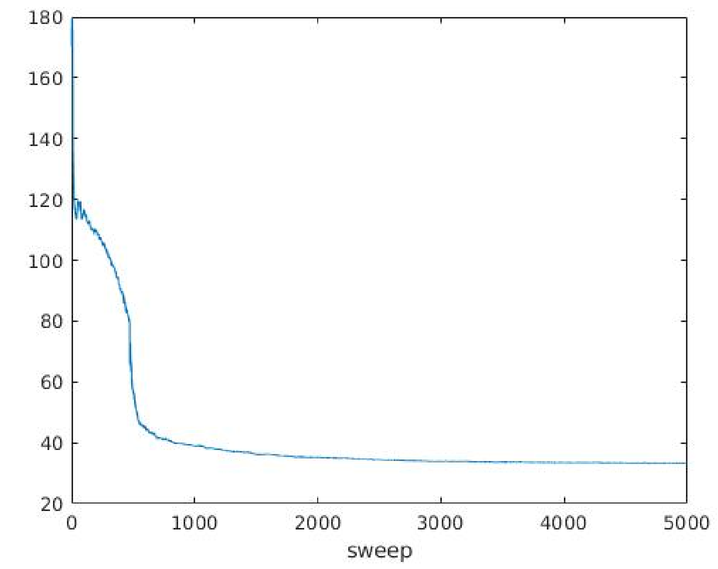


Figure 5: the 90% quantile point of point-to-cluster-centroid distances, for data set 1

Figure (4) shows the cost function with respect to the sweep. The cost function keep decreasing trend ,until does not change much at about 0.5×10^5 after 2000 sweep. And Figure (5) shows the 90% point in the histogram of point-to-cluster-centroid distances. It also decrease until a stable level about 35 after sweep 2000, which means the points are getting more and more close to the cluster centroid.

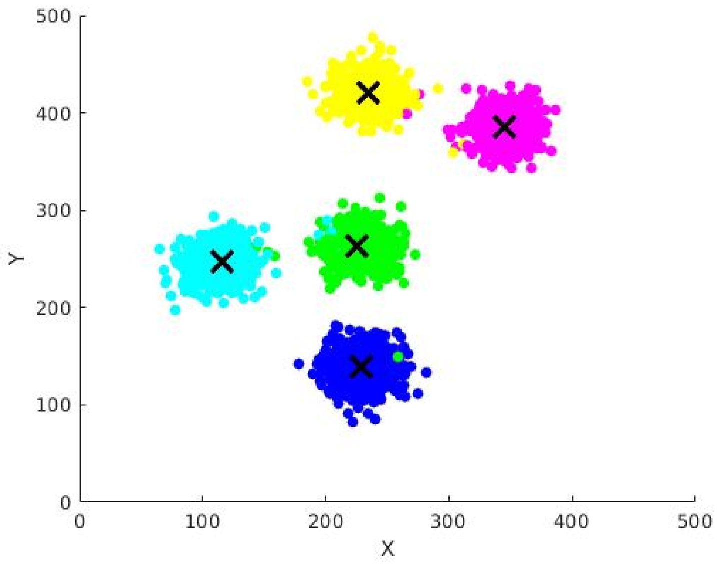


Figure 6: clustering result, for data set 1

Figure (6) is the simulated result of partition, we mark points in one group with same color. Let us compare the simulated partition with the true partition, which is decided in the generation of data. The black points have the same group label as the true group label

,which is right clustering points. Otherwise, we remark incorrect clustering points with red color.

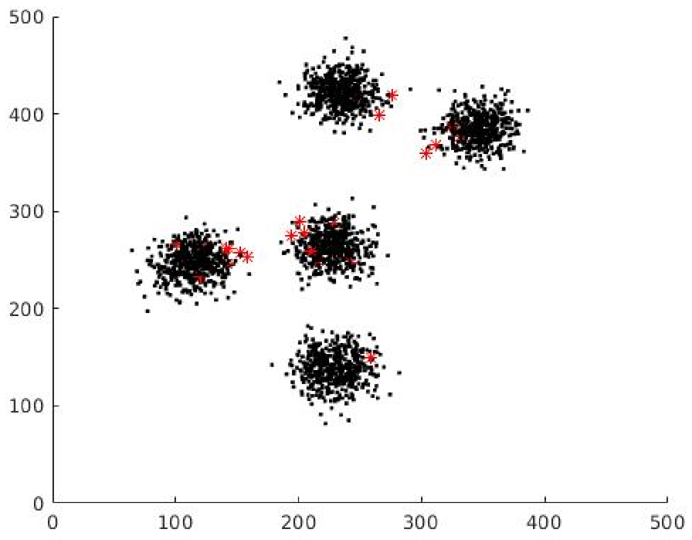


Figure 7: correct and incorrect clustering, for data set 1

We have accuracy rate

$$correctrate = \frac{N_{correctclusteringpoints}}{N_{totalpoints}} = \frac{2480}{2500} = 99.2\% \quad (6)$$

3 Implement of more difficult data set

The above generated data is very ideal data. Now, we enhance the variance in the data generation ($\times\sqrt{2}$). So that the generated cluster will be close to each other and there are more points in the boundary . There are more difficult to clustering.

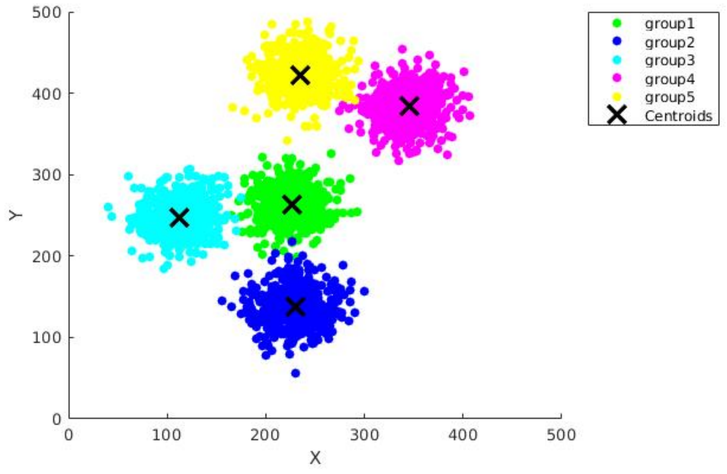


Figure 8: generated data set 2

For monitoring the quality of clustering, we have three performance indicators, including cost, 90% quan tile point in the histogram of point-to-cluster-centroid distances and the correct rate.

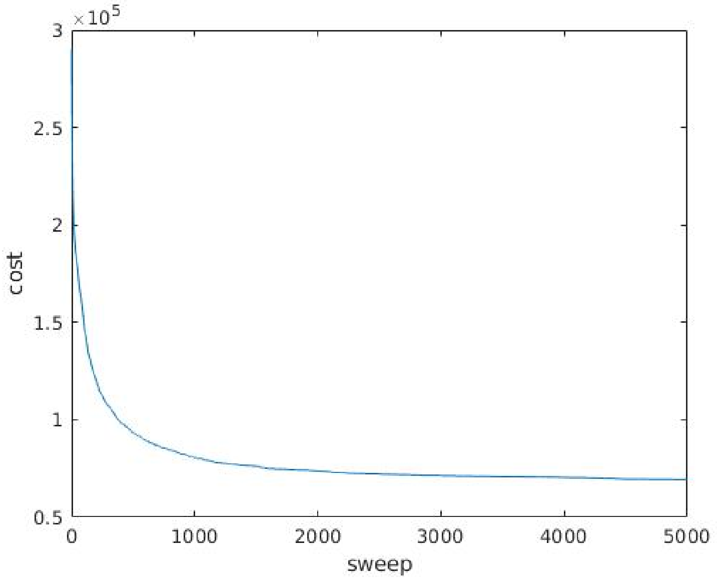


Figure 9: cost function, for data set 2

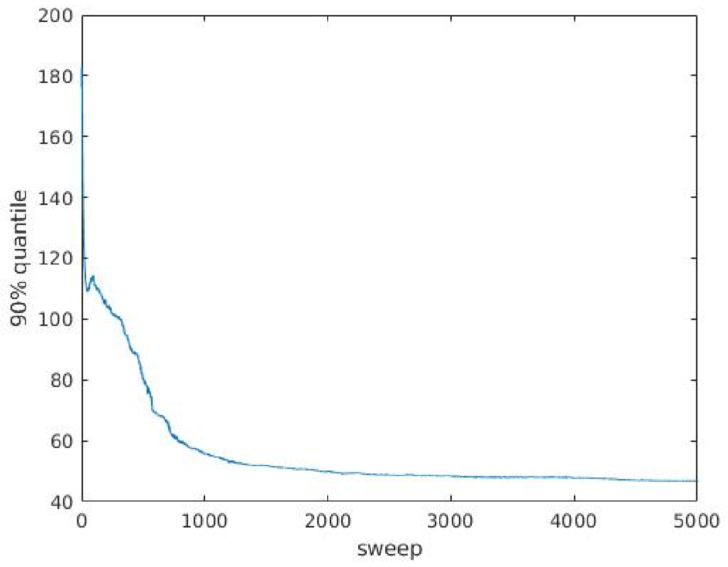


Figure 10: the 90% quantile point of point-to-cluster-centroid distances, for data set 2

In Figure (9), the cost function keep decreasing trend ,until does not change much at about 0.75×10^5 after 2000 sweep. $0.75 \times 10^5 > 0.75 \times 10^5$. The minimum value of cost function is larger than the the first example. Because the second data set is generated with large variance, the clustering level is weaker. And Figure (10) shows the 90% point in the histogram of point-to-cluster-centroid distances decrease until a stable level about 50 after sweep 2000, which means the points are getting more and more close to the cluster centroid.

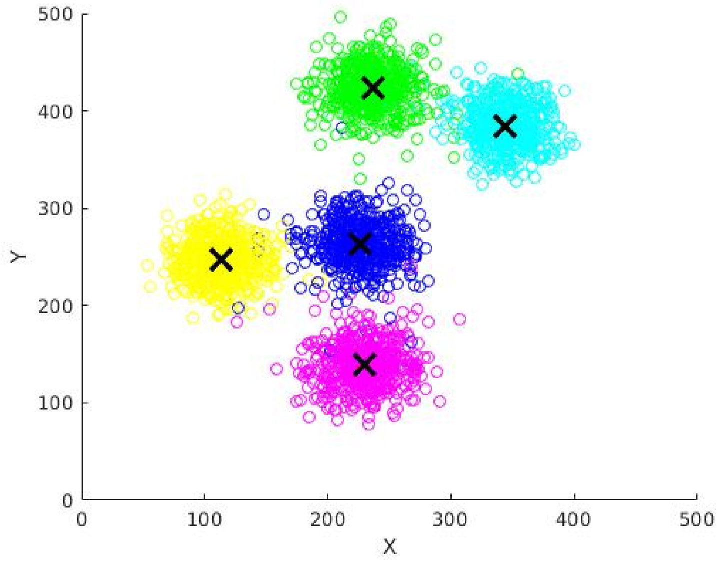


Figure 11: clustering result, for data set 2

Figure (11) is the simulated result of partition, we mark points in one group with same color.

Also, we remark incorrect clustering points with red color and correct clustering points with black color in the figure(12).

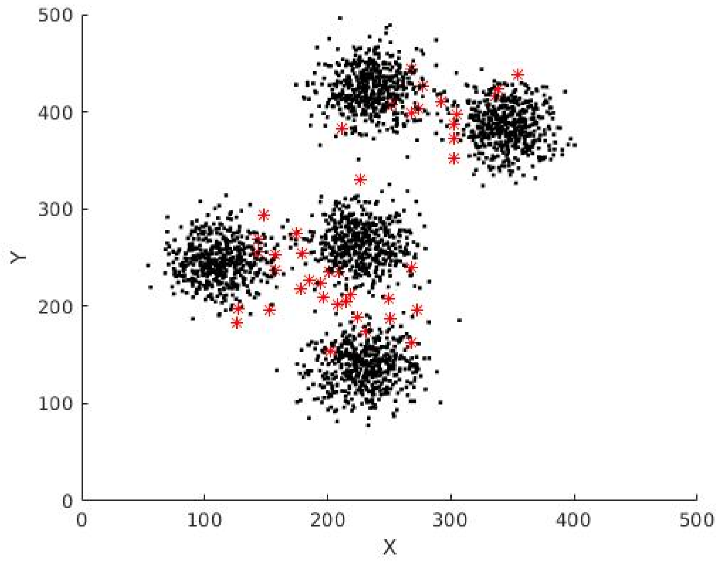


Figure 12: correct and incorrect clustering, for data set 2

We have accuracy rate

$$correctrate = \frac{N_{correctclusteringpoints}}{N_{totalpoints}} = \frac{2458}{2500} = 98.32\% \tag{7}$$

In general, we have good clustering predication. The most incorrect cases happens on the boundary. In the data generation, some clusters are overlap, which cause the extreme points may have shorter distance with neighboring centroid. But still some points should be assign to the closest centroid. If it only reassign one points each time, after many sweep, it should have big chance to have right partition. I may make some mistake in our meeting note??

4 Correctness for the algorithm discussed in the meeting

Rather than change a small batch of neurons randomly, we only change the one-points corresponding neurons , which is equal to set batch size to a set of one-hot neurons. It would be slow due to the number of iteration need per sweep but more accurate. In the 4th step of the algorithm, reassign only one point to different cluster randomly, and the other step are same.

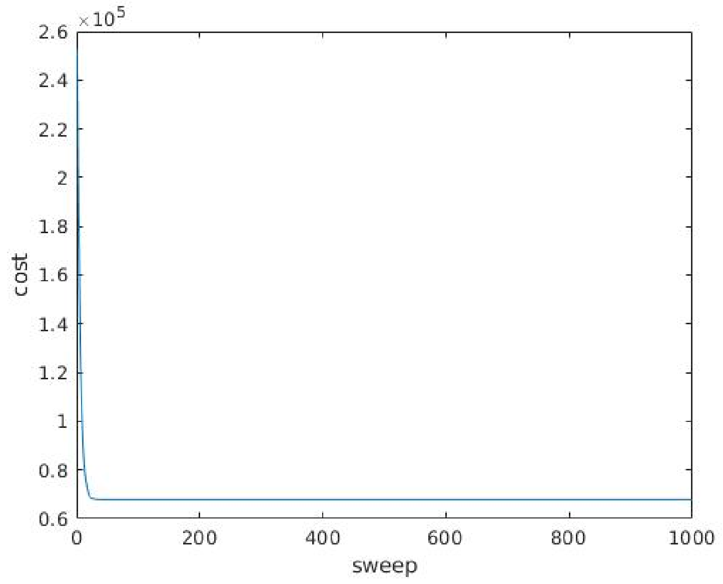


Figure 13: cost function, for data set 2

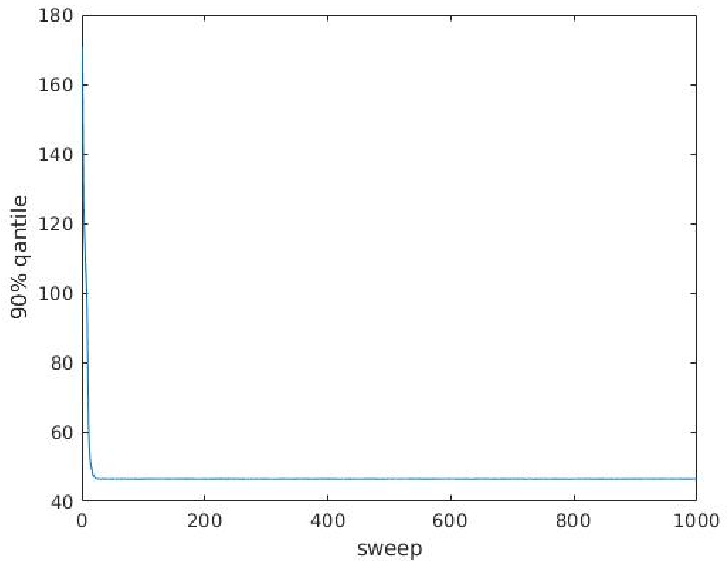


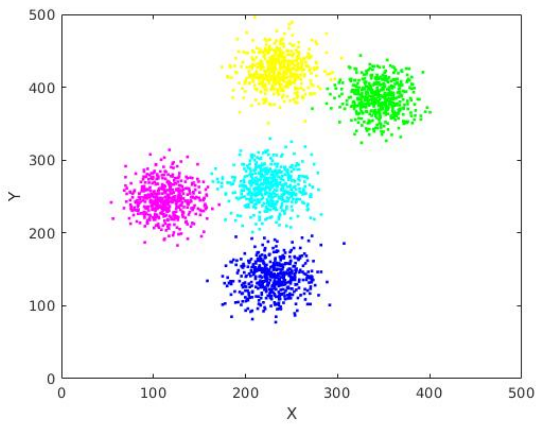
Figure 14: the 90% quantile point of point-to-cluster-centroid distances, for data set 2

YES! Corrected! From the figure(13)(14), we almost have the minimum cost and 90% quantile of histogram after 13 sweep. AND we have better correct rate for clustering. So the 4th step of algorithm should be corrected as reassign one point into random group.

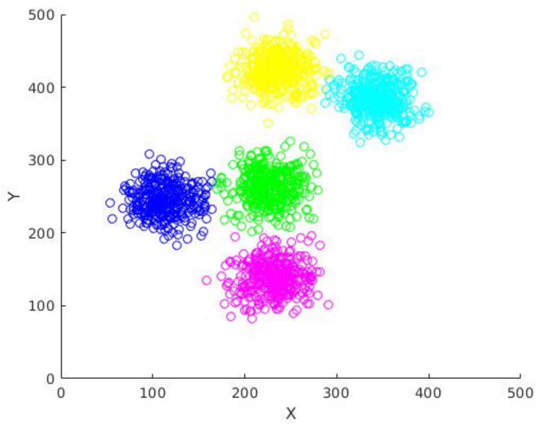
$$correctrate = \frac{N_{correctclusteringpoints}}{N_{totalpoints}} = \frac{2487}{2500} = 99.48\% \tag{8}$$

5 Compartion with K-mean clustering

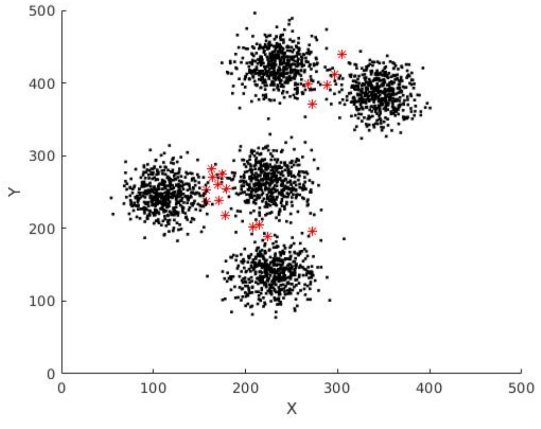
K-mean clustering is a popular method of cluster analysis in data mining. k-means clustering also aims to partition n observations into k clusters. Rather than do the randomly reassign the cluster, the K-mean algorithm decide each observation belongs to the cluster with the nearest mean. It motivate me to test if our clustering algorithm could beat the K-mean method? I use the defaulted "Kmean" function in the Matlab to partition data set 2 into 5 groups.
Figure (15) is the simulated result of partition with K-mean algorithm, we mark points with colors. Also, we remark incorrect clustering points with red color and correct clustering points with black color in the figure(16).



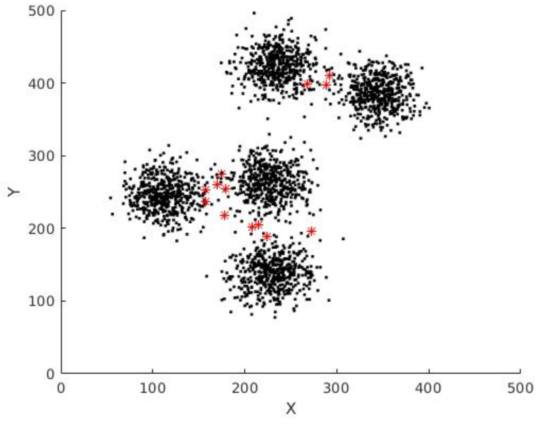
(a) K-mean clustering result, for data set 2



(b) Boltzmann machine clustering result, for data set 2



(a) K-mean clustering, correct and incorrect examples, for data set 2



(b) Boltzmann machine clustering result,correct and incorrect examples, for data set 2

K-mean clustering have accuracy rate 99.28%.

$$correctrate = \frac{N_{correctclusteringpoints}}{N_{totalpoints}} = \frac{2482}{2500} = 99.28\% \tag{9}$$

The Bultmann machine algorithm performs better than K-mean clustering 99.48% > 99.28% in this data set.

6 Conclusion

In this report, we try apply Boltzmann machine algorithm to clustering problem. In the two data set we do practice, Boltzmann machine algorithm have good performance in the partition with correct rate > 99%. Compared with the popular K-mean clustering, we even have better performance in clustering in the correctness rate.