E9-253: Neural Networks and Learning Systems-I
HW-3

Submitted by - Ocima Kamboj
Serial Number - 06-02-01-10-51-18-1-15899

Problem-5.10 Clustering with K-means Algorithm

Methodology

Experimental Setup

The programming of the K-means Algorithm was done in MATLAB. K-means algorithm proceeds in two steps -

Step-1 For a given encoder C, the total cluster variance is minimized with respect to the assigned set of cluster means $\{\hat{\mu}_j\}_{j=1}^{K}$; that is, we perform, the following minimization:

$$\min_{\{\hat{\mu}_j\}_{j=1}^{K}} \sum_{j=i}^{K} \sum_{C(i)=j} \|x_i - \hat{\mu}_j\|^2$$

for a given C.

Step-2 Having computed the optimized cluster means $\{\hat{\mu}_j\}_{j=1}^{K}$ in step-1, we next optimizr the encoder as follows -

$$C(i) = arg \min_{1 \leq j \leq K} \|x_i - \hat{\mu}_j\|^2$$

Starting from some initial choice of the encoder C, the algorithm goes back and forth between these two steps until there is no further change in the cluster assignments.

The Pseudocode to implement K-means is -

1. Choose K initial cluster centers. We do this by picking the data points randomly from the distribution that produces the input data set.
2. For a data point, computer the distance to all the cluster centres.
3. Assign each observation to the cluster with the closest mean.
4. Compute the average of the points belong to a cluster-k to get K new cluster centres.
5. Repeat steps-2-4 until the maximum number of iterations has been reached.

Generation of Data Points

$N = 1000$ data points were generated as given in Haykin, with 500 points in each of the half moons. The center radius is $R = 10$, width of the halfmoon is 6, and the distance d between the halfmoons is kept variable, which can be fed to the function.

Number of Clusters

As specified in the question, we fix $K = 6$.

(a) Aim - Experimentally, determine the mean $\hat{\mu}_j$ and variance $\hat{\sigma}_j^2$, $j = 1, 2...6$, for the sequence of eight uniformly spaced vertical separations starting at $d = 1$ and reducing them by one till the final separation $d = -6$ is reached.

The results have been plotted below. The plots also contain the decision boundaries plotted by the K-means, RLS algorithm (details in Question 5.11). *D.B.* refers to the decision boundary. C.A refers to the classification accuracy as achieved on the training data set.

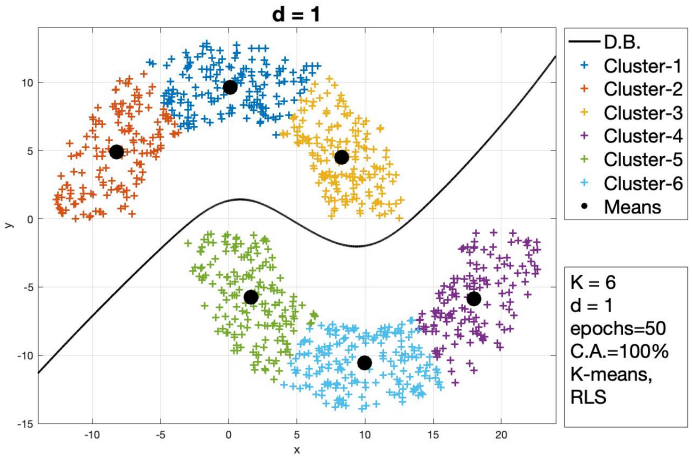

Figure 1

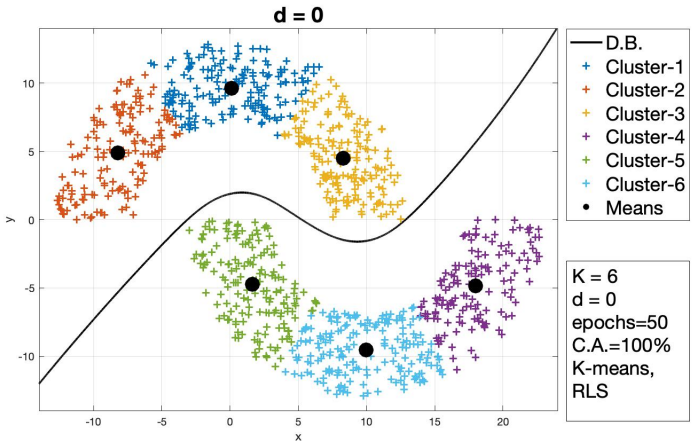

Figure 2

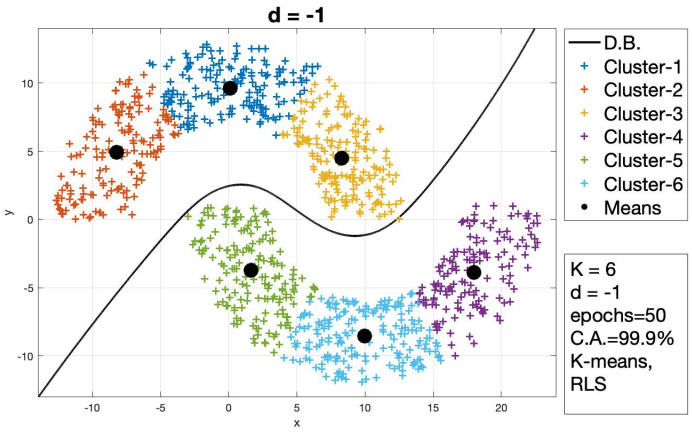

Figure 3

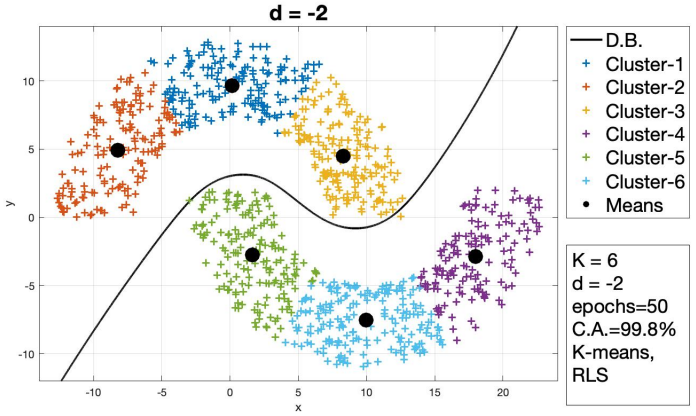

Figure 4

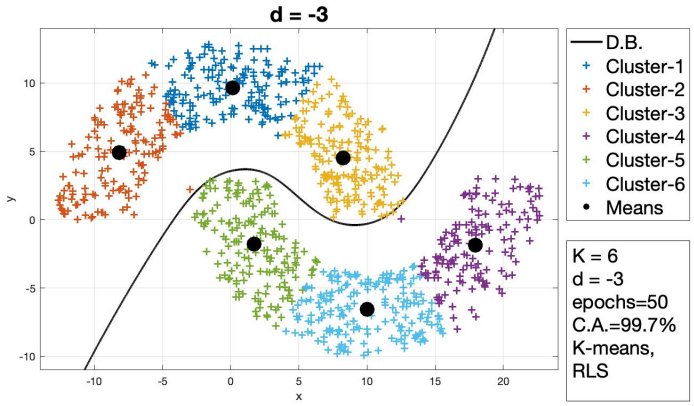

Figure 5

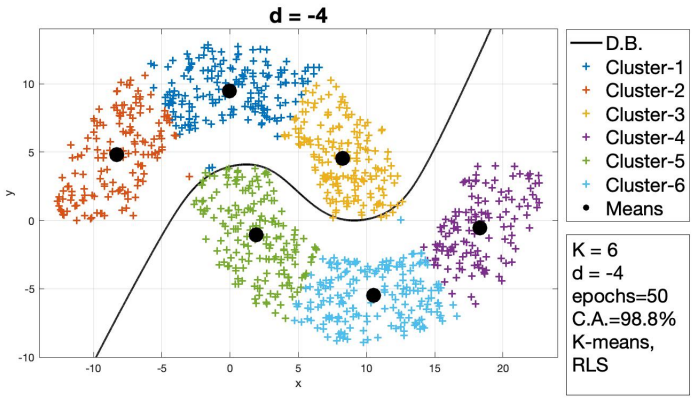

Figure 6

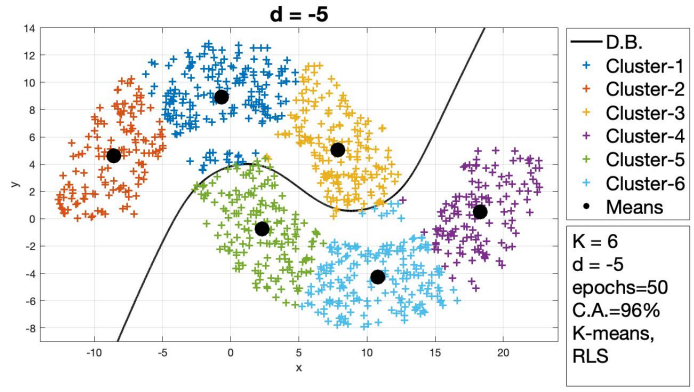

Figure 7

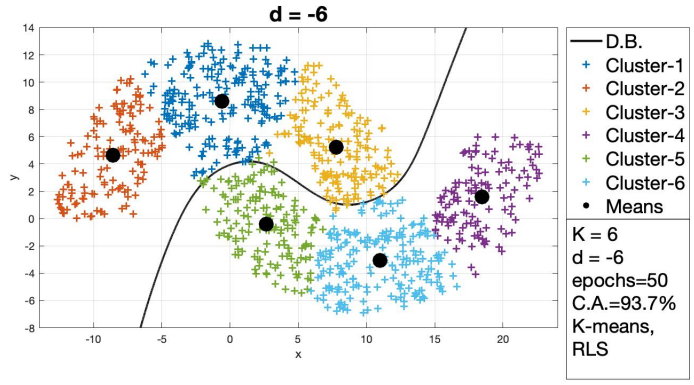

Figure 8

Observations

The classification accuracy decreases as d decreases. This is as expected because the classification problem becomes harder for decreasing d.

It is also interesting to note that the decision boundary is correctly distinguishing the clusters that were identified by the K-means algorithm. That is, the decision boundary doesn't cut through a cluster.

The results for $\hat{\mu}_j$ and $\hat{\sigma}_j^2$ have been tabulated below -

		Cluster-1	Cluster-2	Cluster-3	Cluster-4	Cluster-5	Cluster-6
$d = 1$	x	0.12	-8.23	8.27	18.00	1.64	9.96
	y	9.63	4.91	4.49	-5.87	-5.74	-10.54
$d = 0$	x	0.12	-8.23	8.27	18.00	1.64	9.96
	y	9.63	4.91	4.49	-4.87	-4.74	-9.54
$d = -1$	x	0.12	-8.23	8.27	18.00	1.64	9.96
	y	9.63	4.91	4.49	-3.87	-3.74	-8.54
$d = -2$	x	0.14	-8.21	8.27	18.00	1.64	9.96
	y	9.64	4.93	4.49	-2.87	-2.74	-7.54
$d = -3$	x	0.14	-8.18	8.25	17.97	1.71	10.00
	y	9.64	4.91	4.52	-1.85	-1.80	-6.56
$d = -4$	x	-0.04	-8.30	8.23	18.31	1.90	10.53
	y	9.49	4.82	4.54	-0.53	-1.04	-5.46
$d = -5$	x	-0.68	-8.59	7.85	18.30	2.31	10.78
	y	8.91	4.63	5.05	0.50	-0.75	-4.27
$d = -6$	x	-0.59	-8.59	7.79	18.44	2.66	10.99
	y	8.60	4.63	5.22	1.60	-0.39	-3.08

Table 1: $\hat{\mu}_j$ for j = 1,2, .. 6

	Cluster-1	Cluster-2	Cluster-3	Cluster-4	Cluster-5	Cluster-6
$d = 1$	11.79	13.21	10.09	11.80	12.27	12.01
$d = 0$	11.79	13.21	10.09	11.80	12.27	12.01
$d = -1$	11.79	13.21	10.09	11.80	12.27	12.01
$d = -2$	11.71	13.29	10.09	11.80	12.27	12.01
$d = -3$	11.71	13.42	9.94	11.95	12.26	11.89
$d = -4$	12.39	13.02	10.06	10.44	12.54	12.37
$d = -5$	12.64	12.05	11.08	10.54	11.40	12.16
$d = -6$	12.84	12.05	10.45	9.96	10.17	11.91

Table 2: $\hat{\sigma}_j^2$ for j = 1,2, .. 6

(b) Aim - In light of the results obtained in part (a), comment on how the mean $\hat{\mu}_j$ of cluster j is affected by reducing the separation d for j = 1, 2, and 3.

Observations -

The means $\hat{\mu}_1$, $\hat{\mu}_2$, and $\hat{\mu}_3$ remain almost the same with the decreasing distance. This is because the K-means algorithm tries to minimize the total cluster variance. Because of the symmetry of our input data set, it seems intuitive that with K clusters, an optimal clustering would be one in which $K/2$ means are in the top half-moon, and $K/2$ means are in the bottom half moon, with the size of each cluster being almost the same. As the top half-moon doesn't change it's position with changing d, the cluster means corresponding to it remain almost fixed in position.

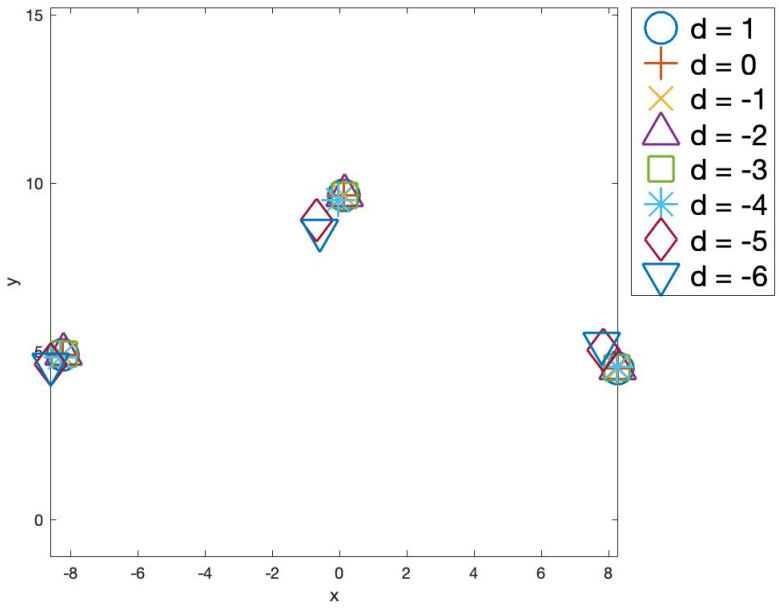

Figure 9: $\hat{\mu}_1$, $\hat{\mu}_2$ and $\hat{\mu}_3$ vs. d

(c) Aim - Plot the variance $\hat{\sigma}_j^2$ versus the separation d for j = 1, 2, ..., 6.

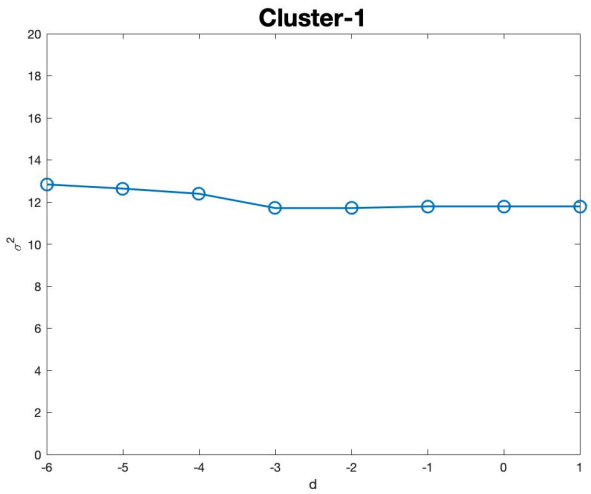

Figure 10: $\hat{\sigma}_1^2$ vs. d

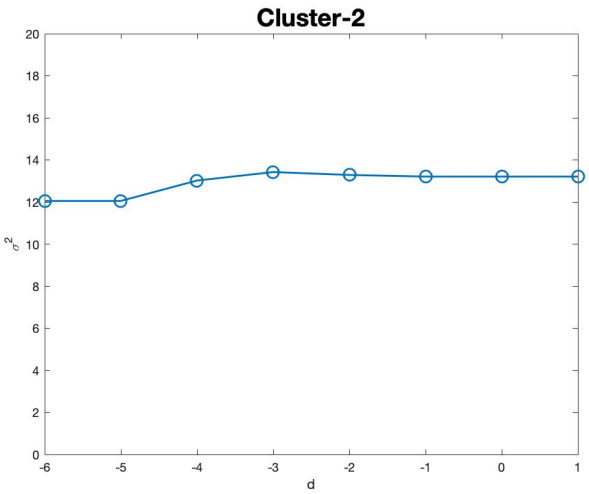

Figure 11: $\hat{\sigma}_2^2$ vs. d

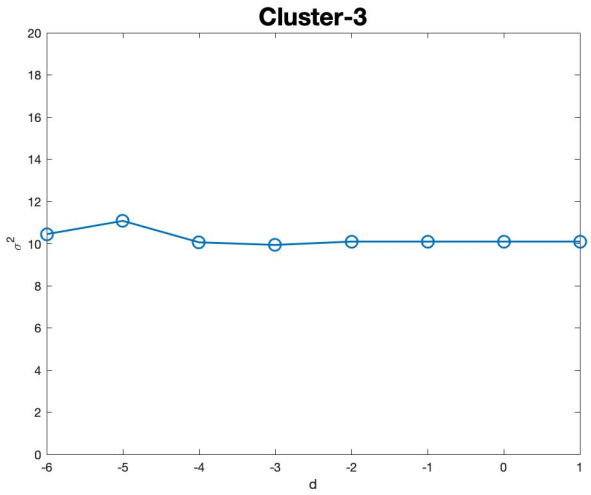

Figure 12: $\hat{\sigma}_3^2$ vs. d

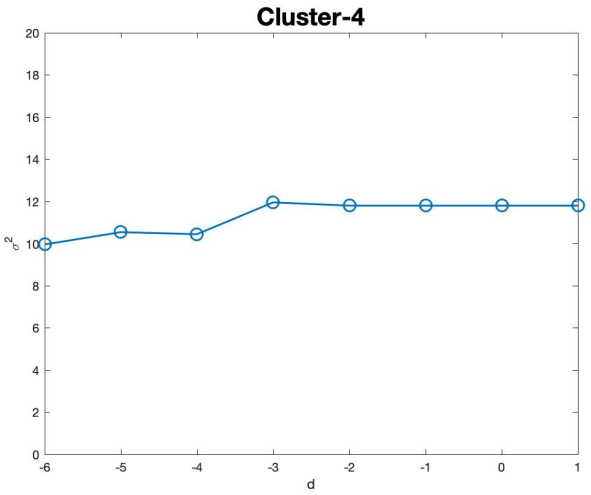

Figure 13: $\hat{\sigma}_4^2$ vs. d

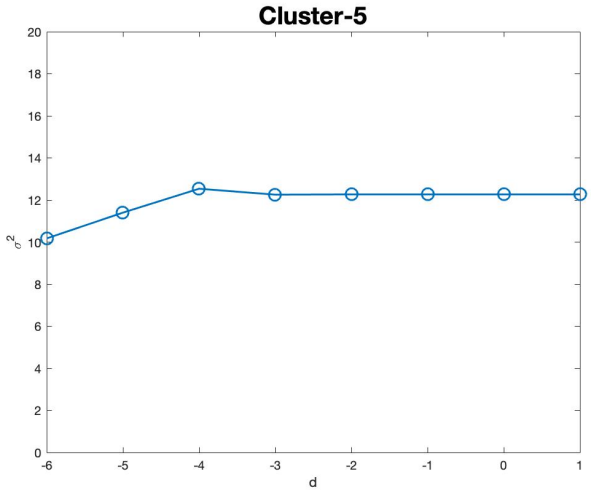

Figure 14: $\hat{\sigma}_5^2$ vs. d

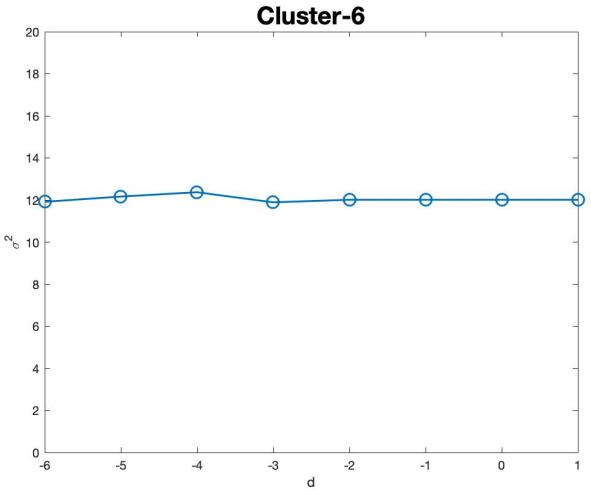

Figure 15: $\hat{\sigma}_6^2$ vs. d

Observations

All the clusters remain almost the same in size as d decreases. σ^2 is a measure of within-the-cluster distance variance, and this distance remains almost the same if the size of the cluster doesn't change. Therefore, we observe that variances remain almost the same for all the clusters.

(d) Aim - Compare the common σ^2 computed in accordance with the empirical formula of Eq. (5.49) with the trends exhibited in the plots obtained in part (c).

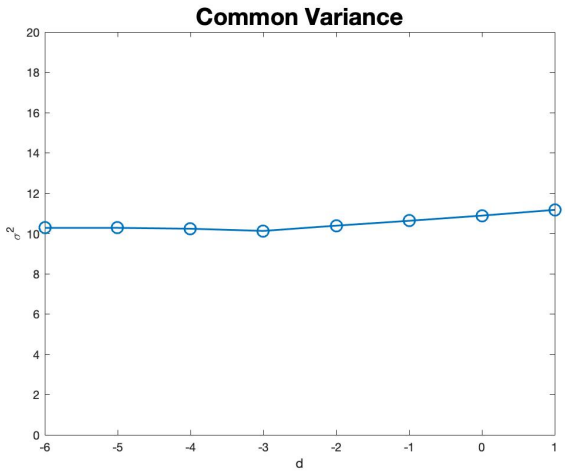

Figure 16: σ^2

Observations

In part(c) we saw that variances for all the clusters remain almost the same for decreasing d. The common variance that is calculated follows the same trend.

Problem-5.11 Comparision of K-means,RLS and K-means,LMS Algorithm

Methodology

Experimental Setup
The programming for the Radial Basis Function Network was done on MATLAB. The network consists of one input layer with 2 neurons, one hidden layer with K neurons, and one output neuron. The Gaussian Function is used as the radial basis function for the hidden layer.

Both the algorithms consist of two steps. The first step performs an unsupervised learning task to map the input space $\{x_i\}$ to the feature space $\{\varphi_j(x_i)\}_{j=1}^{K}$. This is done with the K-means algorithm that was discussed in Problem-5.10. The number of clusters K for K-means equal the number of hidden neurons, and this K acts as a hyperparameter. At the end of this step, we get the centroids and the variances, and this for any input $\{x_i\}$, the corresponding point in feature space becomes ϕ_i, where

$$\phi(x_i) = [\varphi(x_i, \mu_1), \varphi(x_i, \mu_2), ..., \varphi(x_i, \mu_K)]^T$$

and

$$\varphi(x_i, \mu_j) = e^{\frac{-\|x_i - \mu_j\|^2}{2\sigma_j^2}}$$

The second step performs a supervised training on the labelled set of training data $\{\phi(i), d(i)\}_{i=1}^{N}$, where N is the total number of examples in each data set. This is done differently for RLS and LMS algorithm.

RLS Algorithm - Given the training sample $\{\phi(i), d(i)\}_{i=1}^{N}$, do the following -

Initialise : w(0) = 0, and P(0) = $\lambda^{-1}I$, where λ is a small positive constant.
For $n = 1, 2, ...$

1. $$P(n) = P(n-1) - \frac{P(n-1)\phi(n)\phi^T(n)P(n-1)}{1 + \phi^T(n)P(n-1) * \phi(n)}$$

2. $$g(n) = P(n)\phi(n)$$

3. $$\alpha(n) = d(n) - w^T(n-1)\phi(n)$$

4. $$w(n) = w(n-1) + g(n)\alpha(n)$$

LMS Algorithm - Given the training sample $\{\phi(i), d(i)\}_{i=1}^{N}$, do the following -

Initialise : w(0) = 0
For $n = 1, 2, ...$

1. $$e(n) = d(n) - w^T(n)\phi(n)$$

2. $$w(n+1) = w(n) + \eta x(n)e(n)$$

Generation of Data Points

Same as the previous question.

Parameters Chosen for Experiment

The parameters were fixed as prescribed in Haykin. That is -
Number of hidden Gaussian units: 20
Number of training samples: 1,000 data points
Number of testing samples: 2,000 data points
Learning-rate parameter η of the LMS algorithm was annealed linearly from 0.6 down to 0.01.

Prediction

If the output from the RBF Network corresponding to an input data point was greater than zero, then the data point was assigned to class-1, otherwise it was assigned to the other class.

(a) Aim - Construct the decision boundary computed for the "K-means, LMS" algorithm for the vertical separation between the two moons set at d = - 5.

The results for both K-means, RLS and K-means, LMS have been shown below. The terms 'C.A.' and 'T.A.' in the plots refer to the classification accuracy (the accuracy achieved on the training data), and the testing accuracy (the accuracy achieved on the testing data). The 'Time' denotes the time taken to train the model.

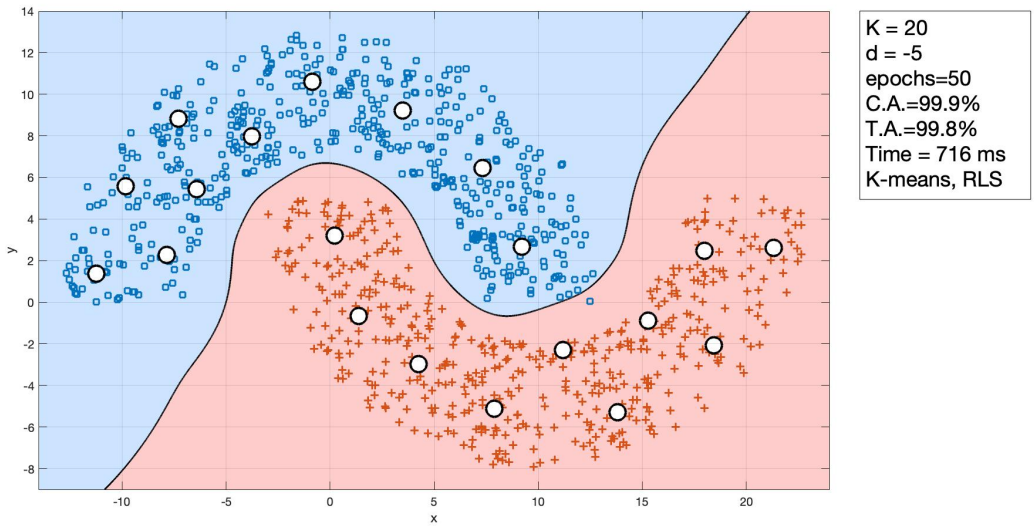

Figure 17: K-means,RLS

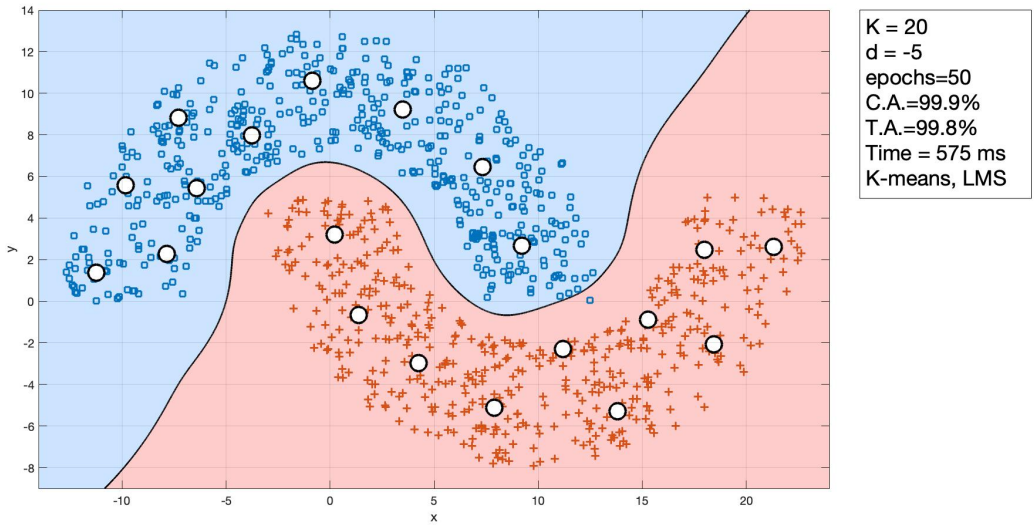

Figure 18: K-means,LMS

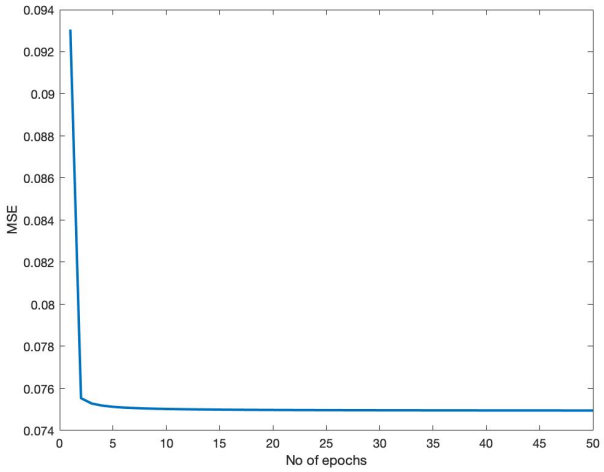

Figure 19: K-means,RLS - Learning Curve

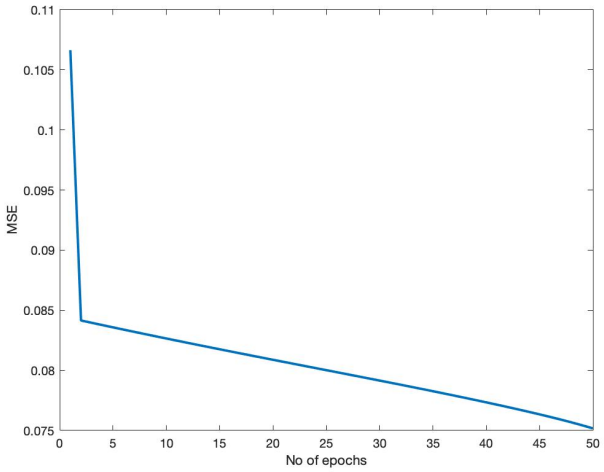

Figure 20: K-means,LMS - Learning Curve

Observations (Covered in (c))

(b) Aim - Repeat the experiment for d = -6.

The results have been plotted below -

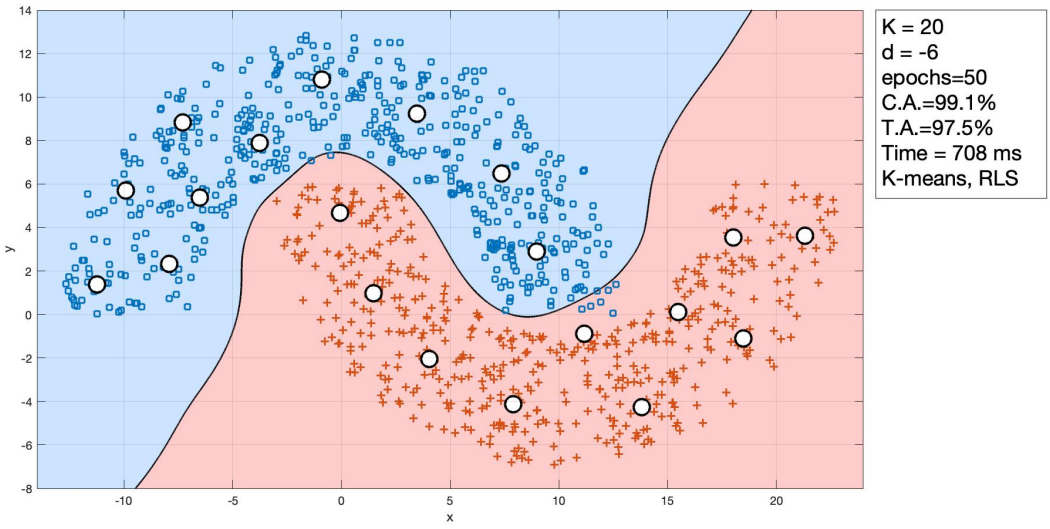

Figure 21: K-means,RLS

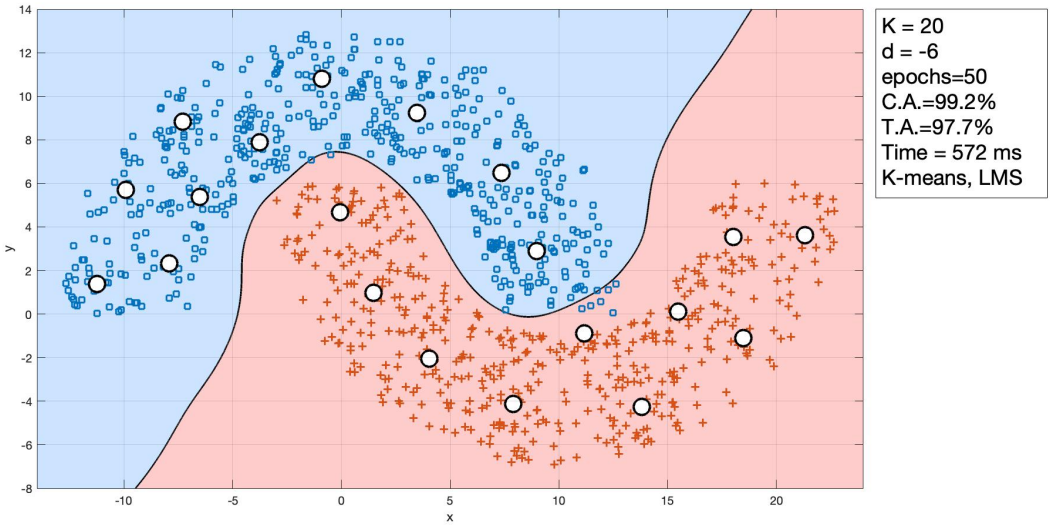

Figure 22: K-means,LMS

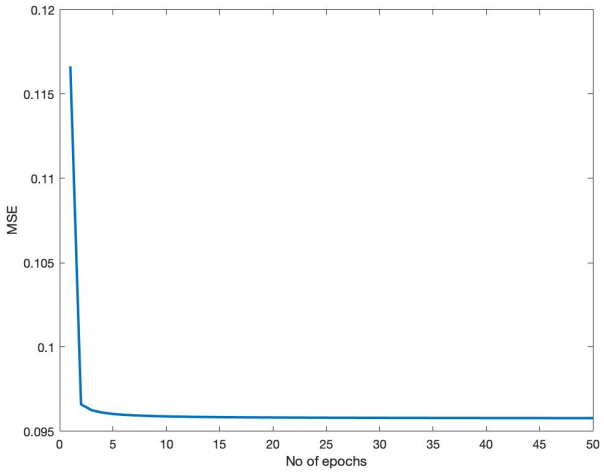

Figure 23: K-means,RLS - Learning Curve

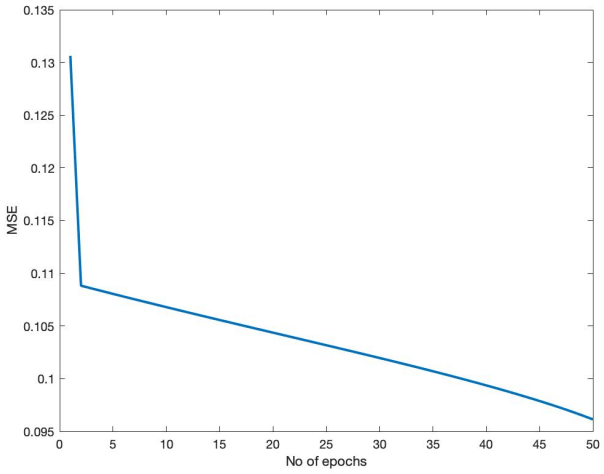

Figure 24: K-means,LMS - Learning Curve

Observations (Covered in (c))

(c) Aim - Compare the classification results obtained using the "K-means, LMS" algorithm with those of the "K-means, RLS" algorithm

Observations

1. Both K-means, LMS and K-means, RLS network are able to achieve almost perfect separability of the two moon-shaped patterns.
2. The time taken to train K-means, RLS is more than K-means, LMS, which is as expected. This is because the weight update in the LMS algorithm is done in less number of steps as compared to the K-means, RLS algorithm.
3. The classification accuracy and the testing accuracy for $d = -5$ case is same for both the algorithms.
4. The classification and testing accuracy for $d = -6$ case is marginally better for K-means, LMS as compared to K-means, RLS algorithm. This result is unexpected, as K-means, RLS algorithm being a second order Optimization routine should perform better.
5. The learning curve for both the cases show that the K-means, RLS routines converges in less number of epochs as compared to K-means, LMS routine. This is expected because K-means, RLS algorithm is a second order method, and K-means, LMS algorithm is a first order method. This also means that, even though K-means, LMS algorithm is giving us better results for $d = -6$, if the computations were expensive, then K-means, RLS would give us better results.