E0-270 Machine Learning 02/06/2020

Assignment 1

Instructor: *Submitted by: Neeraj Badal*

1 Valid Kernels

1a

Given $k(x, y) = x^T Ay$, a function from R^d x $R^d \rightarrow R$ and $A \in R^{d \text{ x } d}$, $x, y \in R^d$. We need to derive the necessary and sufficient conditions on A in order to ensure k is a valid kernel.

Necessary part proof

Let $k(x, y)$ be the valid kernel. Therefore,

$$\begin{aligned} k(x, y) &= x^T Ay \\ &= (x^T Ay)^T \text{ since } x^T Ay \text{ is a scalar, so } x^T Ay = (x^T Ay)^T \\ &= y^T A^T x \end{aligned}$$

As k is a valid kernel,

$$\begin{aligned} k(x, y) &= k(y, x) = y^T Ax \\ \therefore y^T A^T x &= y^T Ax \implies A^T = A \end{aligned}$$

Thus A is a symmetric matrix.
Let K be the Gram matrix represented as,

$$K = XAX^T, \text{ where } X \in R^{nxd} \text{ is the data matrix having n input vectors}$$

Since K is built using kernel function k, therefore it is symmetric and positive semi-definite and hence it follows,

$$\begin{aligned} u^T Ku &\geq 0 \ \forall u \in R^n \\ u^T XAX^T u &\geq 0 \text{ since } K = XAX^T \\ (X^T u)^T AX^T u &\geq 0 \implies z^T Az \geq 0, \text{ where } z = X^T u \end{aligned}$$

Thus matrix A is positive semi-definite. Therefore if k is a valid kernel then A is symmetric and positive semi-definite.

Sufficiency part proof

Let A be symmetric and positive semi-definite. Therefore we can represent A in terms of its eigen decomposition as follows,

$$\begin{aligned} A &= Q\Lambda Q^{-1} = Q\Lambda Q^T, \text{ since A is symmetric } \therefore \text{ Q is an orthogonal matrix} \\ \therefore A &= Q\Lambda Q^T = Q\Lambda^{1/2}\Lambda^{1/2}Q^T = (Q\Lambda^{1/2})(Q\Lambda^{1/2})^T \end{aligned}$$

Rewriting kernel function based on this value of A,

$$\begin{aligned} k(x,y) &= x^T Ay = x^T (Q\Lambda^{1/2})(Q\Lambda^{1/2})^T y \\ &= (P^T x)^T (P^T y) = \Phi(x)^T \Phi(y) \text{ , where } P = Q\Lambda^{1/2} \end{aligned}$$

The above result signifies that $k(x,y)$ does the dot product over the transformed vectors.

$$\begin{aligned} k(x,y) &= x^T Ay = (x^T Ay)^T \\ &= y^T A^T x = y^T Ax = k(y,x) \end{aligned}$$

This implies k is a symmetric function. For psd, again consider Gram matrix $K = XAX^T$,

$$\begin{aligned} \therefore u^T Ku &= u^T XAX^T u \ \forall \ u \in R^n \\ &= (X^T u)^T A(X^T u) \\ \implies &= g^T Ag \geq 0 \text{ as A is a psd} \end{aligned}$$

This implies Gram matrix K is also a psd and so if matrix A is symmetric and psd then $k(x,y) = x^T Ay$ is a valid kernel. Hence, the derivation of necessary and sufficient conditions is done.

1b

Considering $\mathcal{X}$ as the data set in R^n, let K be the gram matrix computed from $K_{ij} = k_l(x_i, x_j)$ $\forall \ x_i, x_j \in \mathcal{X}$. The kernel function k_l is a valid kernel iff $K = K^T$ and $u^T Ku \geq 0 \ \forall u \in R^n$. For checking positive semi-definiteness, after ensuring that K is symmetric it will be sufficient to have all the eigenvalues corresponding to K to be ≥ 0. Below functions were tested for n = 900 samples.

i k_1 is not a valid kernel, as the gram matrix K_1 constructed was symmetric but not positive semi-definite.

ii k_2 is likely to implement a valid kernel, as the gram matrix K_2 constructed was symmetric and positive semi-definite.

iii k_3 is not a valid kernel, as the gram matrix K_3 constructed was neither symmetric and positive semi-definite.

iv k_4 is not a valid kernel, as the gram matrix K_4 constructed was symmetric but not positive semi-definite.

v k_5 is not a valid kernel, as the gram matrix K_5 constructed was symmetric but not positive semi-definite.

2 Support Vector Machines

2a and 2b

Fig. 1a shows the data set $\mathcal{D}$ generated for the given problem. Fig. 1c and d shows the plots of different separating hyperplanes w.r.t values of C tried out over the training and test data set.

The hyper-parameter C was tuned to obtain the best training set accuracy. The best training accuracy achieved was **1.0** for the hyper-parameter **C=1.1**. The test set accuracy for this value of

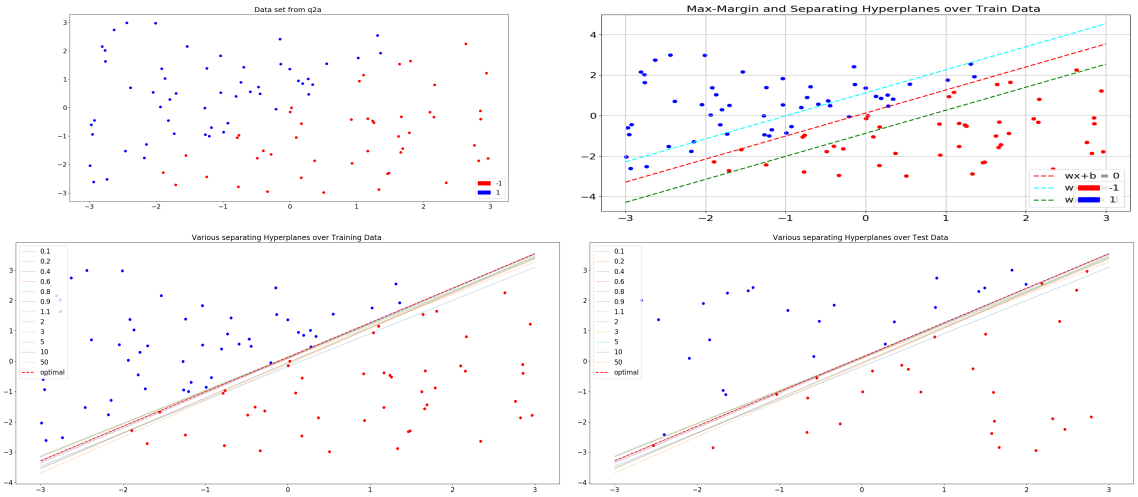


Figure 1: a) Data set, Separating Hyperplanes b) with max-margins over train set c) Training Data set d) Test Set Data set d)

C was **0.98**. Plots shown in the Fig. 2 shows that training set accuracy remains stable for values of $C \geq 1.1$, but the test set accuracy is not found to be stable and in fact it decreases to 0.96 for higher value of $C = 50$. So, $C = 1.1$ was chosen to be optimal, as trade-off between training and test accuracy for this data-set was found to be suitable.

The table 1 reports the optimal paramters obtained by solving primal SVM formulation on this data set.

Hyper-parameter C	w^*	b^*
1.1	$[-2.47262, 2.17398]$	-0.26945

Table 1: Optimal Params

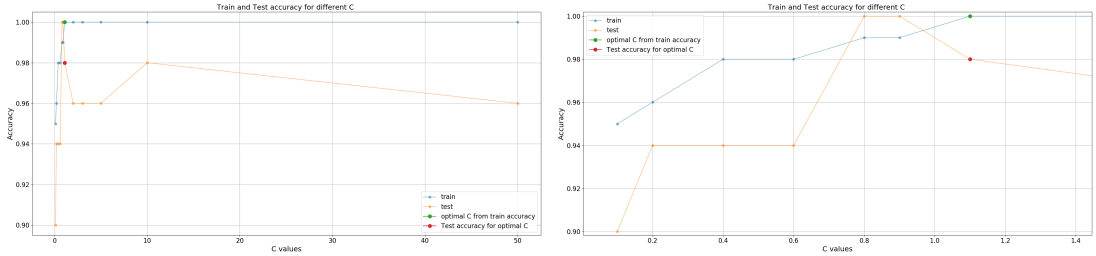


Figure 2: Hyper-parameter C tuning a) Original Scale plot b) Zoomed-Over optimal section

2c

i Fig. 3a shows the data set $\mathcal{D}$ generated as per the question.

ii Fig. 3b and d, shows the classification result over the train and test data set. The SVM linear classifier from the previous question 2b could not separate the two classes. For different values of C, no change in train and test set accuracy was observed. The train accuracy observed was **0.68**, while test accuracy was 0.74. The high value for test accuracy can be attributed to large population of a class where every test data set got classified which can be seen from Fig. 3c and d.

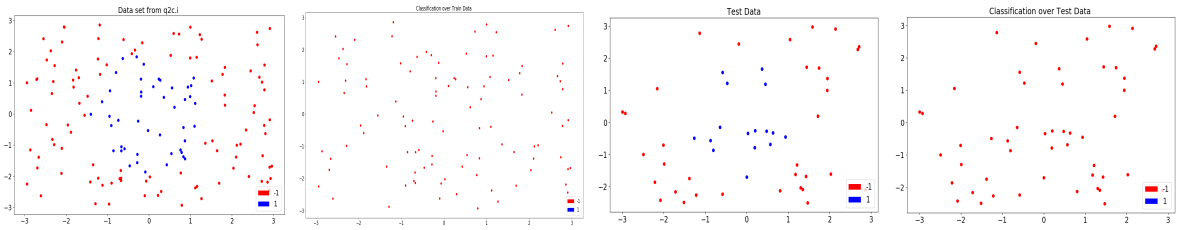


Figure 3: a) Train Data set b) Classification result c) Test data set d) Classification result (Test set)

The table 2 reports the optimal paramters obtained by solving primal SVM formulation on this data set.

Hyper-parameter C	w^*	b^*
0.1	$[-3.79e-08, -3.99e-08]$	-1.00

Table 2: Optimal Params

2d

i Fig. 4a shows the data set $\mathcal{D}$ generated as per the new transformation function.

ii Fig. 4b and c shows the classification result over the train and test data set. The value of$\mathbf{C = 0.4}$ achieves the best accuracy of **0.99** for training data. The test accuracy for the same value of C came out to be **0.98**. However, if the value of C is further increased i.e. $C \geq 0.6$, we can see an improvement in test set accuracy whereas the train set accuracy remains the same. This can be verified from Fig. 4d.

The table 3 reports the optimal paramters obtained by solving primal SVM formulation on the train and test data set.

iii The performance in question 2d.ii has shown improvement over the question 2c due to the fact that the transformed data became linearly separable which was not the case in 2c.

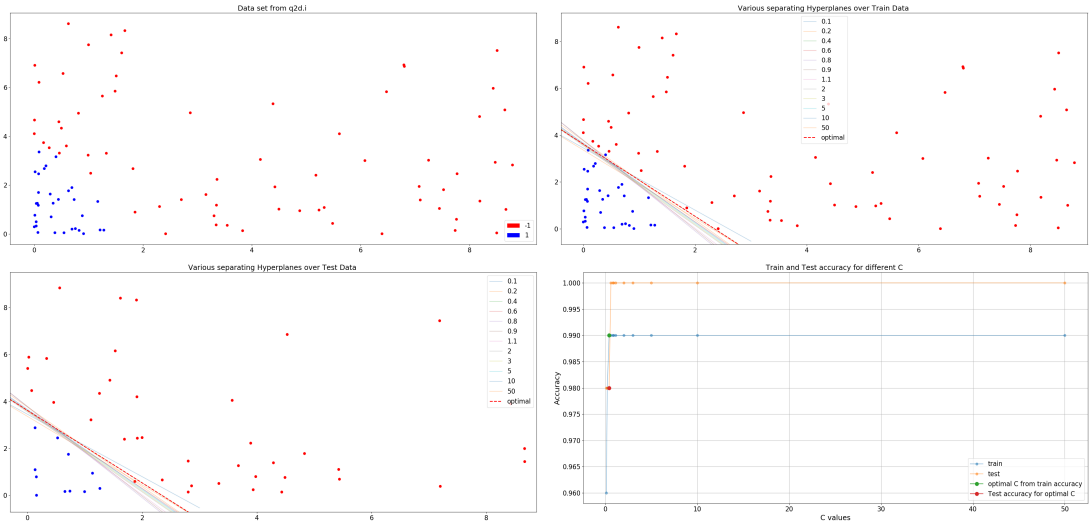


Figure 4: a) Train Data set b) Various Separating Hyperplane for different C over Train data c) Over Test Set d) Classification Accuracy

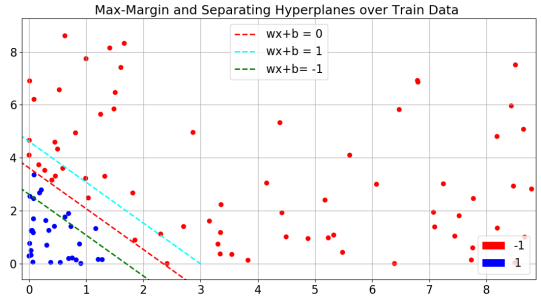


Figure 5: separating hyperplane with max-margins over train set

Hyper-parameter C	w^*	b^*
0.4	$[-1.466, -0.951]$	3.438

Table 3: Optimal Params

2e

For the mentioned primal problem, we can construct the Lagrangian as eq.1,

$$L(w, b, \alpha, \mu) = \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i[y^{(i)}(w^T x^{(i)} + b) - 1 + \xi_i] - \sum_{i=1}^{n}\mu_i\xi_i \tag{1}$$

The dual optimization problem can then be formulated as,

$$\begin{aligned} minimize_{\alpha} \quad & W(\alpha) = \frac{1}{2}\sum_{i,j=1}^{n} y^{(i)}y^{(j)}\alpha_i\alpha_j < x^{(i)}, x^{(j)} > -\sum_{i=1}^{n}\alpha_i \\ \text{subject to} \quad & 0 \leq \alpha_i \leq C, \text{ i=1..n} \\ & \sum_{i=1}^{n}\alpha_i y^{(i)} = 0 \end{aligned} \tag{2}$$

After finding optimum α^*, the optimum values w^*, b^* and the prediction for a new input x, can be found as follows,

$$w^* = \sum_{i \in S}\alpha_i y^{(i)} x^{(i)} \text{ S : set of support vectors} \tag{3}$$

$$b^* = \frac{1}{|S|}\sum_{s \in S}(y^{(s)} - \sum_{m \in S}\alpha_m y^{(m)} x^{(m)}.x^{(s)}) \tag{4}$$

$$\hat{y} = \sum_{i \in S}\alpha_i y^{(i)} < x^{(i)}, \hat{x} > +b^* \tag{5}$$

For chosing a suitable kernel function, polynomial kernel and RBF kernels were tried out. Polynomial kernel with degree $\in \{2, 3, 4, 5\}$ and RBF kernel with sigma $\in \{1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 5.0\}$ were used. Fig. 6 shows the performance of using the mentioned kernels. It can be noted for both the kernels as we increase the value of hyper-parameters degree & sigma each one of them shows fall in train and test accuracy. So, to keep the model simple yet having good accuracy, polynomial kernel with degree 2 was chosen.

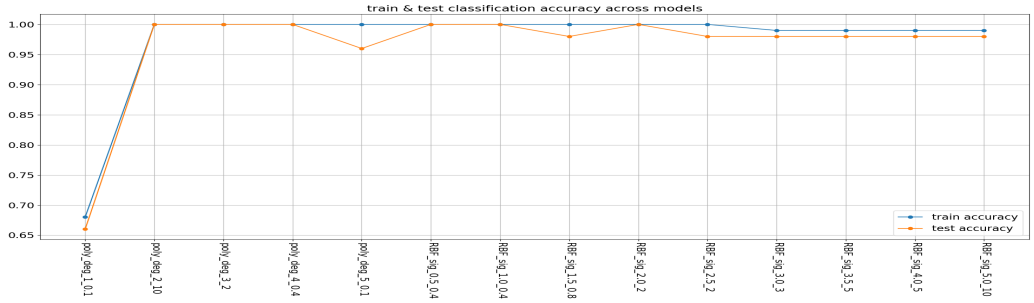


Figure 6: Training and Test set accuracy for Polynomial Kernel and RBF Kernel

The expression for the chosen polynomial kernel function having degree 2 is given as,

$$k(x, y) = \left(\sum_{i=1}^{n} x_i y_i + 1\right)^2$$

The best training accuracy obtained was **1.0**, test accuracy corresponding to it was **1.0**. The tuned hyper-parameters were $\mathbf{C = 10}$ and $\mathbf{degree = 2}$. Fig. 7a and b shows the classification performed over the test set with the chosen kernel and it's corresponding hyper-parameter tuning plot.

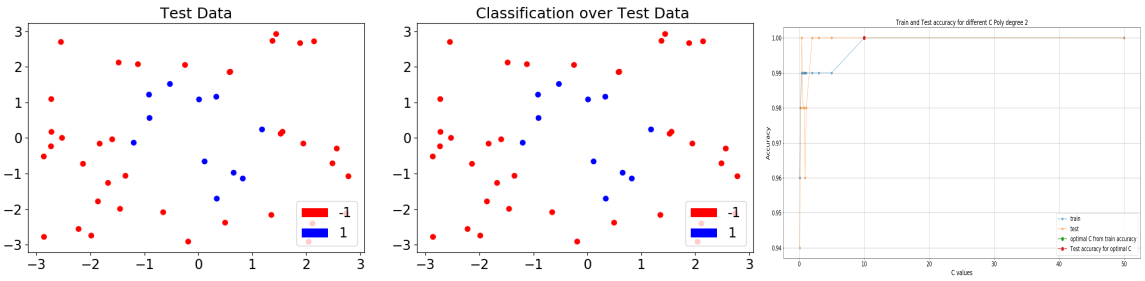
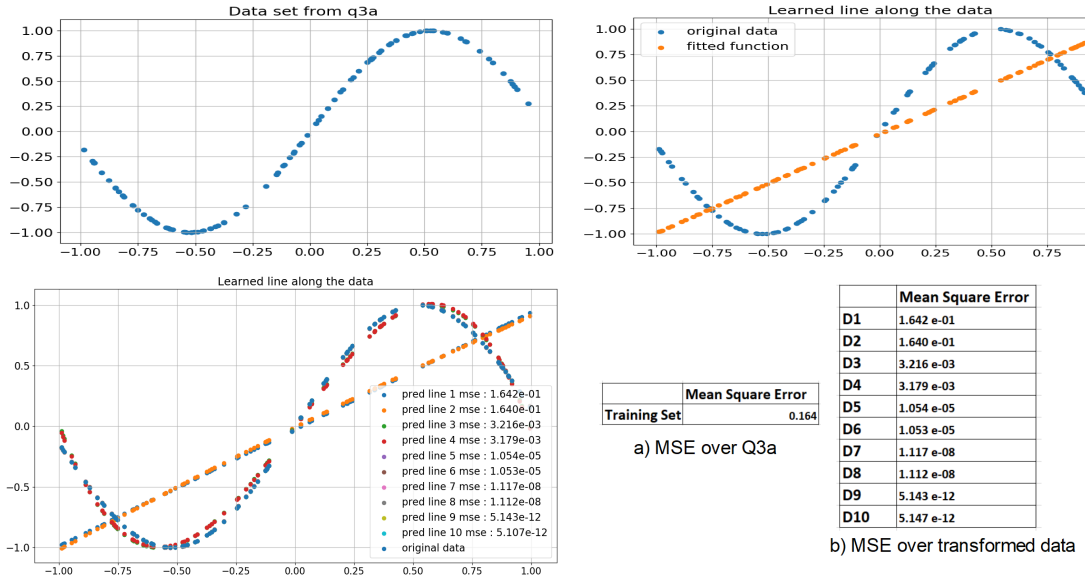


Figure 7: a) Test set , b) Classification over it c) Hyper-parameter C tuning for poly-kernel degree 2

3 Kernelized-Regression

3a, 3b and 3c

Fig. 8 a and b shows the train data along with the learned linear function over it. Fig. 8c shows different function learned based on the mapping Φ_k $k \in [1, 10]$. The respective mean square error for the learned functions are reported in Fig.8d.



	Mean Square Error
Training Set	0.164

a) MSE over Q3a

	Mean Square Error
D1	1.642 e-01
D2	1.640 e-01
D3	3.216 e-03
D4	3.179 e-03
D5	1.054 e-05
D6	1.053 e-05
D7	1.117 e-08
D8	1.112 e-08
D9	5.143 e-12
D10	5.147 e-12

b) MSE over transformed data

Figure 8: a) Train Data , b)Learned Linear Function c) Learned function over various D_k $k \in [1, 10]$ d) Reported MSE

3d

For choosing a suitable kernel function, polynomial kernel and RBF kernels were tried out along with tuning over their hyper-parameters degree and sigma values respectively. Amongst them, RBF kernel with $\sigma = 0.5$ had minimum MSE as seen in Fig. 9 a and b.

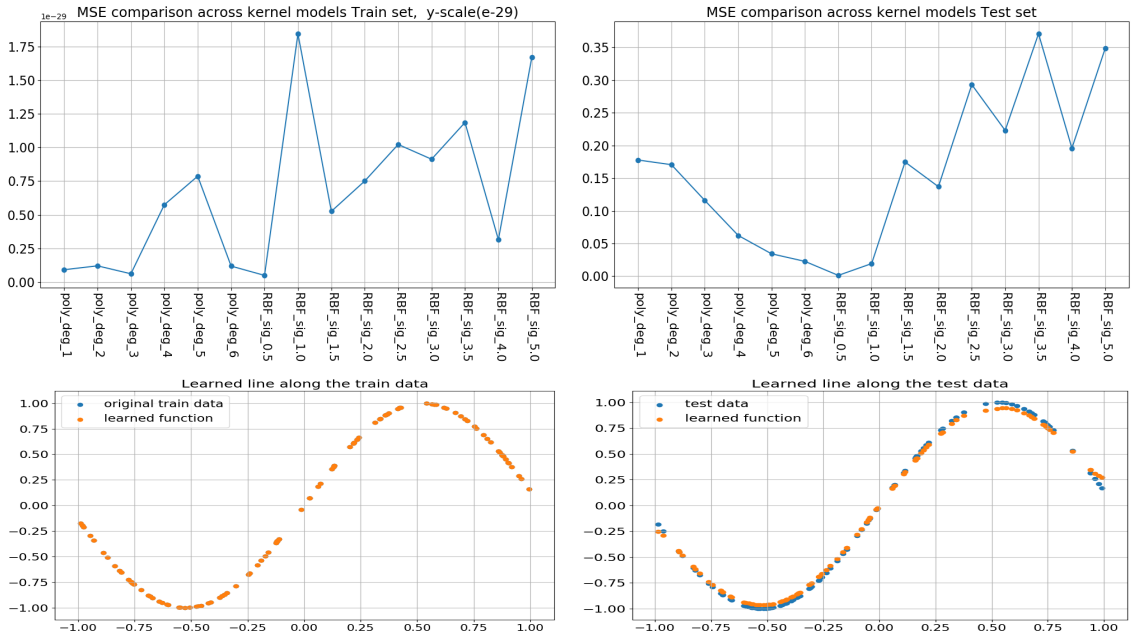


Figure 9: a) MSE over Train Data b)MSE over test data. c)Learned Function RBF($\sigma = 0.5$) over Train data d) Learned Function RBF($\sigma = 0.5$) Test data

Fig. 9 c and d, shows the learned function plotted over the train and test data. The expression for the chosen RBF kernel function for $\sigma = 0.5$ used here is given as,

$$k(x, y) = exp\left(- \frac{||x - y||^2}{2\sigma^2} \right)$$

The training set mean square error was around $4.844e - 31$ and test set mse was around 0.000973.

4 Kernel K-Means

4a

The distance between $\Phi(x)$ and $\Phi(y)$ is computed as,

$$\begin{aligned} dist(\Phi(x), \Phi(y)) &= ||\Phi(x) - \Phi(x)||^2 \\ &= \Phi(x)^T\Phi(x) + \Phi(y)^T\Phi(y) - 2\Phi(x)^T\Phi(y) \\ &= k(x,x) + k(y,y) - 2k(x,y) \end{aligned}$$

The sum of all entries of $D_{ij} = dist(\Phi(e_i), \Phi(e_j))$ was obtained to be $\mathbf{0}$.

4b

The distance between $\Phi(x)$ and μ_c is computed as,

$$\begin{aligned} dist(\Phi(x), \mu_c) &= ||\Phi(x) - \mu_c||^2 \\ &= \Phi(x)^T\Phi(x) - 2\frac{\sum_{i\in c}\Phi(x)^T\Phi(i)}{|c|} + \frac{\sum_{i\in c}\sum_{j\in c}\Phi(i)^T\Phi(j)}{|c|^2} \\ &= k(x,x) - 2\frac{\sum_{i\in c}k(x,i)}{|c|} + \frac{\sum_{i\in c}\sum_{j\in c}k(i,j)}{|c|^2} \end{aligned}$$

The sum of distances of each input e_i towards the mean μ was obtained to be $\mathbf{0}$.

4c

Fig.10a and b, shows the result of kernel k-Means over a dataset started with 2 randomly assigned clusters.

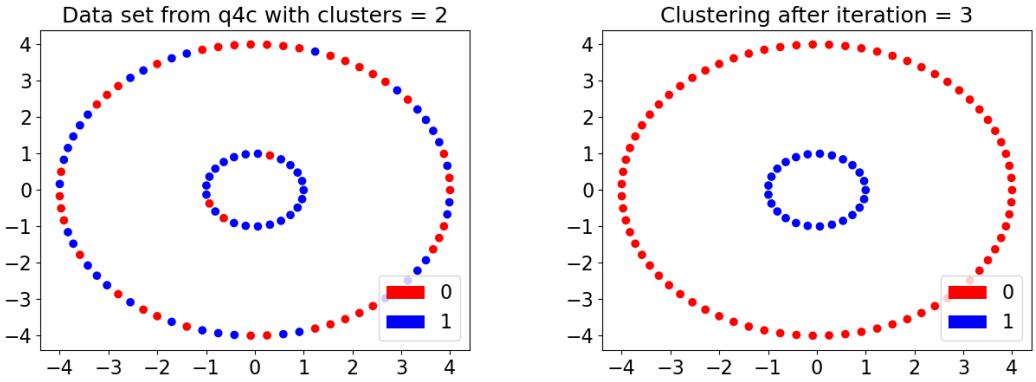


Figure 10: a) Train Data with initial k = 2, b) Final Clustering Result

5 Kernel Fisher's Discriminant Analysis

5a, 5b and 5c

Fig. 11a and b shows the train data and the given transformation applied over it respectively. Fig. 11c shows the projection line corresponding to learned direction w using Fisher's Discriminant analysis.

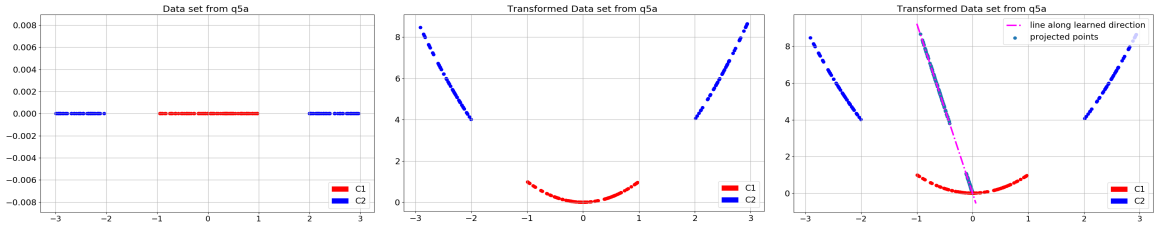


Figure 11: a) Train Data $\mathcal{D}$ b)Transformation over $\mathcal{D}$ c) Line corresponding to learned direction $w \in R^2$

5d

For choosing a suitable kernel function, polynomial kernel and RBF kernels were tried out along with tuning over their hyper-parameters degree and sigma values respectively. The kernel function that had maximum classification accuracy and maximum value for the Fisher's objective function signifying maximum separation between class means & minimum variance inside each class was chosen as the appropriate kernel. Fig.**??**a and b shows the classification accuracy and objective function values for the kernel functions tried over different hyper-parameters. Amongst them, RBF kernel with $\sigma = 0.5$ had shown the maximum performance and so was chosen to be used for this problem.

The expression for the chosen RBF kernel function for $\sigma = 0.5$ used here is given as,

$$k(x, y) = exp\bigg(- \frac{||x - y||^2}{2\sigma^2} \bigg)$$

The classification accuracy obtained was **1**. As bias term b was used for computing the projection data, the threshold θ was obtained around $-2.82e-14 \approx 0$. So, the projected data which lies below θ will be assigned to C_1 and to C_2 o.w. The separation of class means and smaller variance of the projected data can be seen in Fig.12c.

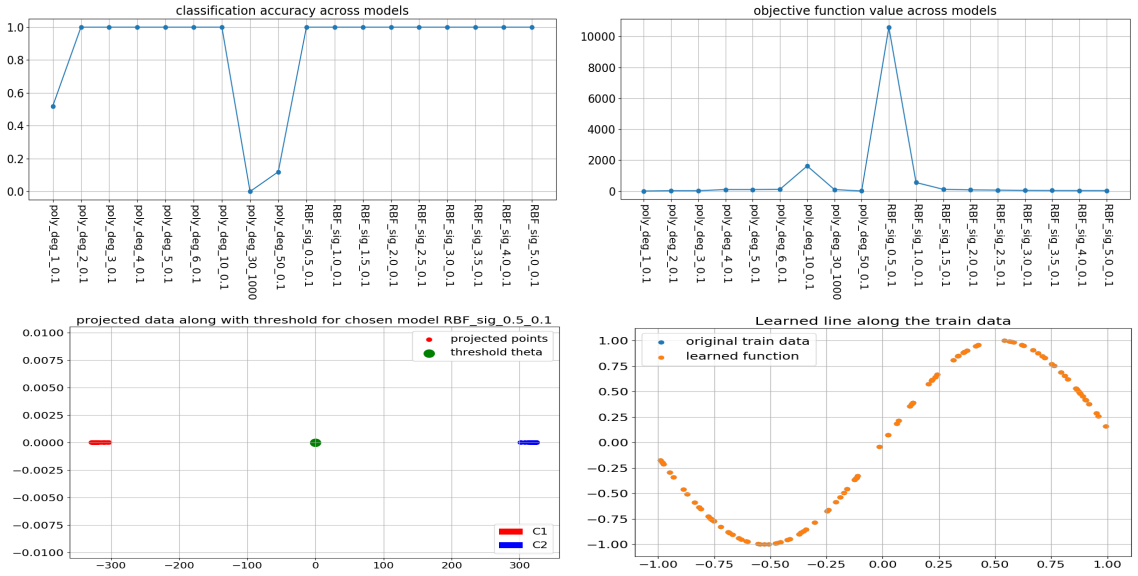


Figure 12: a) Classification accuracy b) Objective function variation with different kernels c) Projection of data based on learned alphas d) learned α_x for $x \in C_1 \cup C_2$.