Assignment 4

Problem 2:

Pseudo Code for Kernelized PCA:

1. Calculate the Kernel matrix for the given data set.
2. Calculate k eigenvectors corresponding to k largest eigenvalues for the kernel Matrix.
3. The product of Kernel matrix with the k-eigenvector matrix gives the required training dataset in k dimensions.
4. Now, for a test data, calculate its kernel matrix with the training dataset and the resultant vector in k dimension is just its product with the eigenvector matrix

Pseudo Code for Kernelized LDA:

1. Determine the Kernel matrix for each class.
2. Then calculate the values of M and N from Problem 1.
3. After that, from N and M, we can easily calculate alpha and thus the required output in one dimension

Accuracy results (from SVM function from sklearn library):

1. Arcene Dataset:
    a. K=10 Kernelized PCA:
        i. with RBF:
            1. 73% (SVM: kernel=Linear)
            2. 70% (SVM: kernel=RBF)
        ii. with Linear function:
            1. Not Diverging (SVM: kernel=Linear)
            2. 56% (SVM: kernel=RBF)
    b. K=100 Kernelized PCA:
        i. with RBF:
            1. 75% (SVM: kernel=Linear)
            2. 56% (SVM: kernel=RBF)
        ii. with Linear function:
            1. Not Diverging (SVM: kernel=Linear)
            2. 56% (SVM: kernel=RBF)
    c. Kernelized LDA:
        i. with RBF:
            1. 60% (SVM: kernel=Linear)
            2. 44% (SVM: kernel=RBF)
        ii. with Linear function:
            1. 44% (SVM: kernel=RBF)
            2. 60% (SVM: kernel=Linear)
2. LSVT Dataset:

a. K=10 Kernelized PCA:
    i. with RBF:
        1. 66.67% (SVM: kernel=Linear)
        2. 66.67% (SVM: kernel=RBF)
    ii. with Linear function:
        1. 66.67% (SVM: kernel=Linear)
        2. 100% (SVM: kernel=RBF)
b. K=100 Kernelized PCA:
    i. with RBF:
        1. 66.67% (SVM: kernel=Linear)
        2. 66.67% (SVM: kernel=RBF)
    ii. with Linear function:
        1. 66.67% (SVM: kernel=Linear)
        2. 100% (SVM: kernel=RBF)
c. Kernelized LDA:
    i. with RBF:
        1. 100% (SVM: kernel=RBF)
        2. Not Diverging (SVM: kernel=Linear)
    ii. with Linear function:
        1. 100% (SVM: kernel=RBF)
        2. Not Diverging (SVM: kernel=Linear)

Assignment - 4
(SMAI)

1 In Fisher's Linear Discriminant Analysis, we project our data in one dimentional space and then separate them.

The ~~kernel trick~~ kernelized LDA is when we first project our data to higher dimensional space and then perform an LDA $\hookrightarrow (x \to \phi(x))$.

The function that needs to be maximized is

$$S_B^\phi = (m_2^\phi - m_1^\phi)(m_2^\phi - m_1^\phi)^T$$

$$S_W^\phi = \sum_{i=1,2} \sum_{n=1}^{l_i} (\phi(x_n^i) - m_i^\phi)(\phi(x_n^i) - m_i^\phi)^T$$

[Assuming 2 classes]

$$\& \quad m_i^\phi = \text{mean} = \frac{1}{l_i} \sum_{j=1}^{l_i} \phi(x_j^i)$$

\# We need to calculate the weight vector w to reduce to 1 dimentional space

$$w = \sum_{i=1}^{l} \alpha_i \phi(x_i)$$

$\hookrightarrow$ kernel function

Let $$(M_i)_j = \frac{1}{l_i} \sum_{k=1}^{l_i} k(x_j, x_k^i)$$

$$\therefore \quad w^T S_B^\phi w = \alpha^T (m_2^\phi - m_1^\phi)(m_2^\phi - m_1^\phi)^T w$$

$$= \alpha^T M \alpha$$

where $M = (M_2 - M_1)(M_2 - M_1)^T$

Similarly, $w^T S_W^\Phi w = \alpha^T N \alpha$

where $N = \sum_{j=1,2} K_j (I - 1_{l_j}) K_j^T$

where 1_{l_j} is the matrix with all elements $1/l_j$

Now,

$$w^T S_W^\Phi w = \left(\sum_{i=1}^{l} \alpha_i \Phi^T(x_i)\right) \left(\sum_{j=1,2} \sum_{n=1}^{l_j} (\Phi(x_n^j) - m_j^\Phi)\right.$$

$$\left.(\Phi(x_n^j) - m_j^\Phi)^T\right) \left(\sum_{k=1}^{l} \alpha_k \Phi(x_k)\right)$$

$$= \sum_{j=1,2} \sum_{i=1}^{l} \sum_{n=1}^{l_j} \sum_{k=1}^{l} \alpha_i \Phi^T(x_i)(\Phi(x_n^j) - m_j^\Phi)(\Phi(x_n^j) - m_j^\Phi)^T \alpha_k \Phi(x_k)$$

$$= \sum_{j=1,2} \alpha^T K_j K_j^T \alpha - \alpha^T K_j 1_{l_j} K_j^T \alpha$$

$$= \alpha^T N \alpha$$

$$J(\alpha) = \frac{\alpha^T M \alpha}{\alpha^T N \alpha}$$

Differentiating, we get

$$(\alpha^T M \alpha) N \alpha = (\alpha^T N \alpha) M \alpha$$

$$\Rightarrow \alpha = N^{-1} (M_2 - M_1)$$

$$\therefore y(x) = w \cdot \Phi(x) = \sum_{i=1}^{l} \alpha_i k(x_i, x)$$