

## Kernel PCA

Steps:

### 1. Compute Kernel

Compute square distances between every 2 points for linear kernel

Compute  $\gamma \times \text{square distances}$  for every 2 points for RBF kernel

### 2. Center Kernel

### 3. Compute Eigen Values and Eigen Vectors OF the Kernel

### 4. Sort EVals and Evecs based on Evecs in descending order

### 5. Take first "redu\_dim" sorted vectors

### 6. This gives you kernel PCA

Best results: For linear svm and linear kernel

```
bhaktipriya@Karyakshetra:~/COURSES/sem5/SMIA/Assignment3$ python q2.py
(100, 10000)
(100,)
=====For 10 components=====
(100, 100)
RBF svm, RBF pca score: 0.45
Linear svm, RBF pca score : 0.45
RBF svm, linear pca score: 0.45
Linear svm, linear pca score : 0.425
=====For 99 components=====
(100, 100)
RBF svm, RBF pca score: 0.45
Linear svm, RBF pca score : 0.45
RBF svm, linear pca score: 0.45
Linear svm, linear pca score : 0.775
```

## 2. Kernel lda:

Steps:

### 1. Compute Kernel

Compute square distances between every 2 points for linear kernel

Compute  $\gamma \times \text{square distances}$  for every 2 points for RBF kernel

### 2. Compute M1 and M2 for classes 1 and 2

### 3. Compute $M = \text{np.dot}((M2 - M1), (M2 - M1).T)$

### 4. Compute N1 and N2

### 5. Compute $N = N1 + N2 + \text{eps}$

### 6. Get alphas by solving for $N(\alpha) = M$

```
bhaktipriya@Karyakshetra:~/COURSES/sem5/SMAI/Assignment3$ python kllda.py
(100, 10000)
(100,)
Class 1 samples 44
Class 2 samples 56
(100, 100) (100, 100) (100, 100)
alpha inv (100,)
alpha solv (100,)
shape (100, 100) (100,)
(100,) result
rbfsvm 1.0
linear svm 1.0
```