Project 4 : Kernel PCA and Kernel Regression

1. The included code creates a data set X in the plane that consists of points having radius in $[0, 3/4] \cup [3/2, 3]$. We label the data points with radius< 1 as class '2' and label the points with radius> 1 as class '1', and we collect these labels in a vector c.

2. Kernel PCA is a nonlinear version of PCA that is really more closely connected to MDS. The idea is to replace the Gram matrix in MDS with a nonlinear kernel functions evaluated on all pairs of data points, $K_{ij} = k(x_i, x_j)$. Map the above data nonlinearly into $\mathbb{R}^3$ by taking the first three components of Kernel PCA using the following kernels (the first one is already done in the included code):

   (a) Exponential Radial Basis Function (RBF) kernel: $k(x, y) = \exp(-||x - y||^2/\delta)$
   (b) Polynomial kernel: $k(x, y) = (1 + x \cdot y)^\delta$
   (c) Nonlocal kernel: $k(x, y) = \frac{1}{1+||x-y||^2/\delta}$
   (d) Hyperbolic kernel: $k(x, y) = \cosh^{-1}(1 + ||x - y||/\delta)$
   (e) Sigmoid kernel: $k(x, y) = \tanh(\delta + x \cdot y)$

   You can quickly compute the matrix of pairwise distances using `d=pdist2(X',X');` and the matrix of pairwise inner products by `G = X'*X`. Experiment with coloring your scatter plot using the radius `r` and the class labels `c`. You may need to tune the parameter δ. Notice that when well tuned all of these kernels map the data into a space where the two classes can be separated by a plane. This is one method of clustering, in the next part we explore another method.

3. Now consider the kernel regression method of extending the class function $c(x)$ where $c(x_i) = c_i$ from this training data set to other data points. Generate a grid on $[-3, 3]^2$ by

```
t=-3:.1:3;
[x,y]=meshgrid(t,t);
x=x(:)';y=y(:)';
Z=[x;y];
```

   then apply the kernel regression:

   (a) Using the exponential kernel, build the kernel matrix $\mathbf{K}_{ij} = k(x_i, x_j)$ for the data set X.
   (b) Compute the coefficients of the class functions, $\tilde{c} = (\mathbf{K} + \epsilon I_{N\times N})^{-1}\vec{c}$.
   (c) Using `d2 = pdist2(Z',X');` compute the joint kernel matrix $(\mathbf{K_Z})_{ij} = k(z_i, x_j)$. Notice that $\mathbf{K_Z}$ is not square, the dimensions will be the number of data points in Z by the number of data points in X.
   (d) Evaluate $c(z_i) = \sum_j k(z_i, x_j)\tilde{c}_j$ for the new data set by multiplying $\vec{c}_Z = K_Z\tilde{c}$.
   (e) Plot the extended function: `surf(t,t,reshape(cZ,length(t),length(t)));`

   Experiment with different parameters such as $\delta \in \{0.01, 0.1, 1, 5\}$ in the kernel and $\epsilon \in \{10^{-10}, 10^{-5}, 10^{-1}, 10^2, 10^4\}$ in the regression.

   When ϵ is too small we are overfitting and when ϵ is too large we are over-smoothing.