==
- Kernel Statistics toolbox
- Last Update:2006/10/29
- For questions or comments, please email
  Yuh-Jye Lee, yuh-jye@mail.ntust.edu.tw or
  Yi-Ren Yeh, yeh@stat.sinica.edu.tw or
  Su-Yun Huang, syhuang@stat.sinica.edu.tw
- Web site: http://dmlab1.csie.ntust.edu.tw/downloads/
==

Table of Contents
=================

Introduction
============
Kernel Statistics toolbox is still in development. Two algorithms are now
available. One is kernel principal component analysis (KPCA). The other
is kernel sliced inverse regression (KSIR).

Key Features
============
* Construct principal components in the input space and feature space.
* Provide a preprocess for preventing ill-posed problem encountered in KSIR.
* Support linear, polynomial and radial basis kernels.
* Can handle large scale problems by using reduced kernel.

Data Format
===========
Kernel Statistics toolbox is implemented in Matlab. Use a data format which
can be loaded into Matlab. The instances are represented by a matrix (rows
for instances and columns for variables) and the labels (1,2,...,k) or
responses are represented by a column vector. Note that you also can

represent the labels of binary classification by 1 or -1.

For classification

```
  instances
 |  10  -5   0.8  | => inst 1
 |  15  -6   0.2  | => inst 2
 |       .        |      .
 |       .        |      .
 |       .        |      .
 |  21   1  -0.1  | => inst n

  labels
 |   1   | => label of inst 1  => class 1
 |   5   | => label of inst 2  => class 5
 |   .   |           .
 |   .   |           .
 |   .   |           .
 |  10   | => label of inst n  => class 10
```

For regression

```
  instances
 |  11  -5   0.2  | => inst 1
 |  14  -7   0.8  | => inst 2
 |       .        |      .
 |       .        |      .
 |       .        |      .
 |  20   2  -0.9  | => inst n

  labels
 |    3.2   | => response of inst 1
 |    1.7   | => response of inst 2
 |     .    |           .
 |     .    |           .
 |     .    |           .
 |   -1.1   | => response of inst n
```

Code Usage with Examples
===

Kernel Statistics toolbox contains two main functions: KDR for constructing PCs and UseKDR for using the results of KDR. Besides, users also can directly use the two core programs, KPCA and KSIR, for a specific kernel matrix.

Description for codes

```
KDR     : the main program for constructing PCs via different methods.
UseKDR  : using the results of KDR to build up the projected data matrix.
KPCA    : finds dimension reduction directions by PCA of a kernel matrix.
KSIR    : finds dimension reduction directions by sliced inverse regression of a kernel
matrix.
```

```
Usage of KDR:

  >>[Info] = KDR(label, inst, 'options')

------------------------------------------------------------------

*Inputs of KDR:
  label  : training data class label or response
  inst   : training data inputs
  options:
  -s  statistic method. 0-PCA, 1-SIR                      (default:0)
  -t  kernel type. 0-linear, 1-polynomial, 2-radial basis (default:2)
  -r  ratio of random subset size to the full data size   (default:1)
  -z  number of slices                                    (default:20)
      If NumOfSlice >= 1, it represents NumOfSlice slices.
      If NumOfSlice = 0,  it extracts slices according to
      class labels.
  -p  number of principal components                      (default:1)
      If NumOfPC= r >= 1, it extracts the first r leading
      eigenvectors.
      If NumOfPC= r < 1, it extracts leading eigenvectors
      whose sum of eigenvalues is greater than 100*r% of
      the total sum of eigenvalues.
  -g  gamma in kernel function                            (default:0.1)
  -d  degree of polynomial kernel                         (default:2)
  -b  constant term of polynomial kernel                  (default:0)
  -m  scalar factor of polynomial kernel                  (default:1)



*Outputs of KDR:
  Info        results of Kernel Statistics method (a structure in Matlab)
    .PC       principal components of data
    .EV       eigenvalues respect to the principal components
    .Ratio
    .RS       reduced set
    .Space    the space of Kernel Statistics method
    .Params   parameters specified by the user in the inputs



Example: Construct ten PCs via KPCA by using Gaussian kernel

  >>[Info_one] = KDR([], inst, '-s 0 -t 2 -g 0.1 -p 10');

Example: Construct five PCs via reduced KPCA (10%) by using Gaussian kernel

  >>[Info_two] = KDR([], inst, '-s 0 -t 2 -p 5 -g 0.2 -r 0.1');

Example: Construct PCs of 90% eigenvalue via KPCA by using reduced polynomial kernel

  >>[Info_three] = KDR([], inst, '-s 0 -t 1 -p 0.9 -r 0.1 -d 2 -m 3 -b 2');

Example: Construct one PCs via KSIR by using Gaussian kernel for 2-class problems


  >>[Info_four] = KDR(label, inst, '-s 1 -t 2 -p 1 -z 0 -g 0.3');
```

Example: Construct 5 PCs via KSIR (30 slices) by using Gaussian kernel for regression problems

```
  >>[Info_five] = KDR(label, inst, '-s 1 -t 2 -p 1 -z 30 -g 0.2');
```

Usage of UseKDR:

```
  >>[ProjInst] = UseKDR(inst, Info)
```

---

*Inputs of UseKDR:
  inst :testing data inputs
  Info :results of Kernel Statistics method

*Output of UseKDR:
  ProjInst: the projected instances

Example: Get the projected inst form Info_one

```
  >>[ProjInst_one] = UseKDR(inst, Info_one);
```

Example: Get the projected inst form Info_four

```
  >>[ProjInst_four] = UseKDR(inst, Info_four);
```

Usage of KPCA:

```
  >>[EigenVectors, EigenValues, ratio] = KPCA(K, NumOfPC);
```

---

*Inputs of KPCA
  K   : kernel matrix (reduced or full)
  NumOfPC: If NumOfPC= r >= 1, it extracts the first r leading eigenvectors.
           If NumOfPC= r < 1, it extracts leading eigenvectors whose sum
           of eigenvalues is greater than 100*r% of the total sum of eigenvalues.

*Outputs of KPCA
  EigenValues : leading eigenvalues
  EigenVectors: leading eigenvectors
  ratio   : sum of leading eigenvalues over total sum of all eigenvalues.

Example: Construct ten PCs via KPCA by using a specific kernel matrix.

```
  >>[EigenVectors, EigenValues, ratio] = KPCA(K, 10);

Example: Construct PCs whose ratio to the total sum is 95%.

  >>[EigenVectors, EigenValues, ratio] = KPCA(K, 0.95);
```

Usage of KSIR:

```
  >>[EigenVectors, EigenValues, ratio] = KSIR(K, y, NumOfSlice, NumOfPC)

------------------------------------------------------------------------
*Inputs of KSIR
  K         : kernel matrix (reduced or full)
  y         : class labels or responses
  NumOfSlice: If numerical, it represents the number of slices.
              If a string 'Class', the number of slices is equal to number of
              distinct classes in y.
  NumOfPC   : If NumOfPC >= 1, it extracts the leading NumOfPC eigenvectors.
              If NumOfPC <  1, it extracts leading eigenvectors whose sum
              of eigenvalues is greater than 100*r% of the total sum of eigenvalues.


*Outputs of KISR
  EigenValues : leading eigenvalues
  EigenVectors: leading eigenvectors
  ratio       : sum of leading eigenvalues over total sum of all eigenvalues.


Example: Construct PCs via KSIR for classification by using a specific kernel matrix.

  >>[EigenVectors, EigenValues, ratio] = KSIR(K, y, 'CLASS');

Example: Construct two PCs via KSIR for 3-class problem.

  >>[EigenVectors, EigenValues, ratio] = KSIR(K, y, 'CLASS', 2);

Example: Construct PCs via KSIR for regression (20 slices).

  >>[EigenVectors, EigenValues, ratio] = KSIR(K, y, 20);

Example: Construct five PCs via KSIR for regression .

  >>[EigenVectors, EigenValues, ratio] = KSIR(K, y, 20, 5);

Example: Construct PCs whose ratio to the total sum is 95% via KSIR for regression.

  >>[EigenVectors, EigenValues, ratio] = KSIR(K, y, 20, 0.95);
```

Dimension Reduction Using KPCA or KSIR with Examples
==========

KPCA procedure
---

*Change your current directory to Kernel Statistics toolbox folder

*Load dataset Ionosphere_dataset.mat (can be found in Kernel Statistics toolbox)
```
>>load Ionosphere_dataset.mat
```

*Construct PCs via KPCA
```
>>[Info] = KDR([], inst, '-s 0 -t 2 -g 0.1 -p 10');
```

*Read the contents of Info (PCs, eigenvalues, parameters, ...etc)
```
>>Info
>>Info.PC
```

*Get the projected inst form the PCs

```
>>[ProjInst] = UseKDR(inst, Info);
```

KSIR procedure for classification
---

*Change your current directory to Kernel Statistics toolbox folder

*Load dataset Ionosphere_dataset.mat (can be found in Kernel Statistics toolbox)
```
>>load Ionosphere_dataset.mat
```

*Construct PCs via KSIR (note that we extracts m-1 PCs for m-class problems)
```
>>[Info] = KDR(label, inst, '-s 1 -t 2 -g 0.165 -z 0 -p 1');
```

*Read the contents of Info (PCs, eigenvalues, parameters, ...etc)
```
>>Info
>>Info.PC
```

*Get the projected inst form the PCs

```
>>[ProjInst] = UseKDR(inst, Info);
```

KSIR procedure for regression
---

*Change your current directory to Kernel Statistics toolbox folder

*Load dataset Housing_dataset.txt (can be found in Kernel Statistics toolbox)
```
>>load Housing_dataset.txt
```

*Split the Housing data into inst and label
```
>>inst =Housing_dataset(:,1:13);
>>label = Housing_dataset(:,14);
```

*Construct PCs via KSIR
(note that we usually use 10~30 slices and extracts 3~5 PCs for regression problems)
```
>>[Info] = KDR(label, inst, '-s 1 -t 2 -g 0.0037 -z 20 -p 5');
```

*Read the contents of Info (PCs, eigenvalues, parameters, ...etc)
  >>Info
  >>Info.PC

*Get the projected inst form the PCs

  >>[ProjInst] = UseKDR(inst, Info);

License
=======

This software is available for non-commercial use only. The authors are not responsible for implications from the use of this software.