

**INTERFACE 2008**

*May 21-24, 2008 in Durham, NC*

# **Kernel Sliced Inverse Regression With Applications to Classification**

**Han-Ming Wu  
(Hank)**

**Department of Mathematics, Tamkang University  
Taipei, Taiwan**

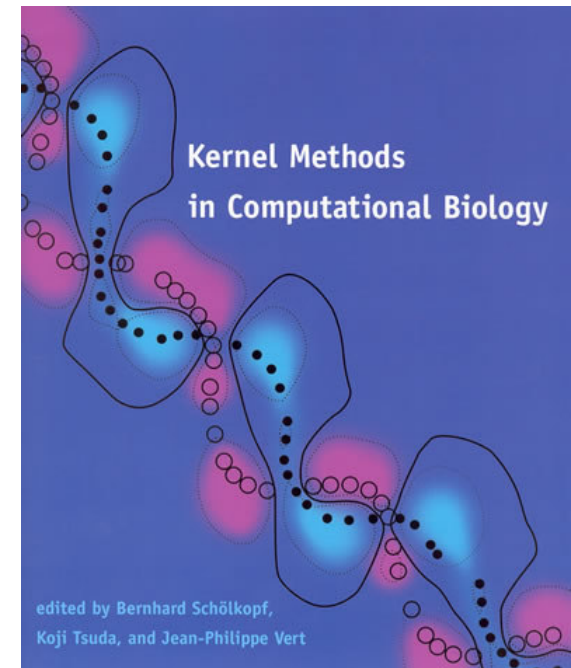
**2008/05/22**

**<http://www.hmwu.idv.tw>**



# Outline

- ❑ **Kernel Methods, Kernel Trick**
- ❑ **Kernel Data and Its Properties**
- ❑ **SIR in the Euclidean Space**
- ❑ **Kernel SIR in a Non-linear Feature Space**
- ❑ **KSIR for Nonlinear Dimension Reduction and Data Visualization**
- ❑ **Experiments on Classification**
- ❑ **Conclusion and Future Direction**



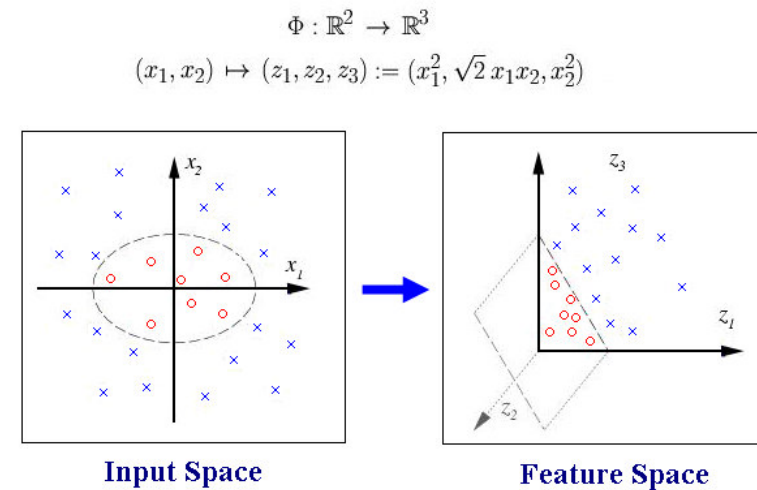
# Kernel Methods (1)

- Aronszajn (1950) and Parzen (1962)
  - ◆ first to employ *kernel methods* in **statistics**.
  
- Aizerman et al. (1964)
  - ◆ used *positive definite kernels* which was closer to “*kernel trick*”,
  - ◆ argued that a positive definite kernel is identical to a *dot product* in the *feature space*.

# Kernel Methods (2)

## □ Boser et al (1992)

- ◆ construct **SVMs**, a generalization of the so-called optimal hyperplane algorithm.



## □ Scholkopf et al (1998)

- ◆ point out that **kernels** can be used to construct generalization of any algorithm that can be carried out **in terms of dot products**.

## □ For last 10 years

- ◆ there have seen a large number of **kernelization** of various algorithms. (e.g., PCA, LDA, CCA, PLS,...)

# Prepare Kernel Data

Raw Data  $\mathbf{X}_{n \times p} = \{\mathbf{x}_i, i = 1, \dots, n\}, \mathbf{x}_i \in R^p$ .

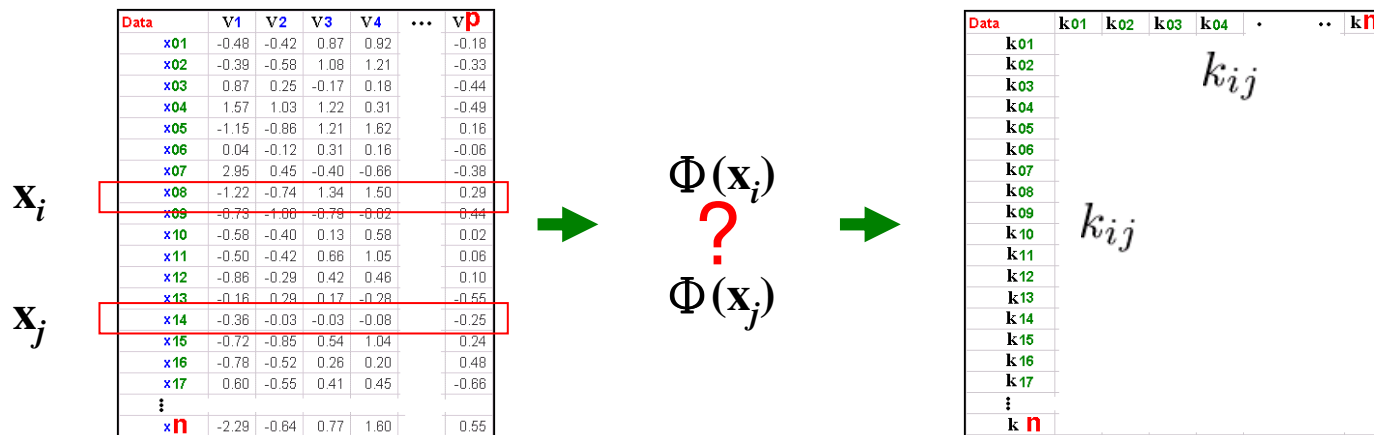
Kernel transformation:  $\mathbf{x}_i \rightarrow \phi(\mathbf{x}_i) := k(\mathbf{x}_i, \cdot)$ .

Kernel Data:  $\{\phi(\mathbf{x}_i), i = 1, \dots, n\}, \phi(\cdot) \in \mathcal{H}_k$ .

Kernel Data  $\mathbf{K}_{n \times n} = \{k_{ij} : k(\mathbf{x}_i, \mathbf{x}_j), i, j = 1, \dots, n\}$ .

theoretically

In fact



- Linear:  $k(x, y) = \langle x, y \rangle$
- Polynomial:  $k(x, y) = (\text{scale} \cdot \langle x, y \rangle + \text{offset})^{\text{degree}}$
- Gaussian Radial Basis Function:  $k(x, y) = \exp\{-\text{scale} \cdot \|x - y\|^2\}$

# Data Representation

- Data are not represented individually anymore, but only through a set of **pairwise comparisons**.
- The size of the matrix used to represent a dataset of  **$n$  objects** is always  **$n$  by  $n$** .

**Definition:** a function  $k : \mathcal{X} \times \mathcal{X} \rightarrow R$  is called a **positive definite kernel** *iff* it is **symmetric**, that is,  $k(\mathbf{x}_i, \mathbf{x}_j) = k(\mathbf{x}_j, \mathbf{x}_i)$  for any two objects  $\mathbf{x}_i, \mathbf{x}_j$  in  $\mathcal{X}$ , and **positive definite**, that is,  $\sum_{i=1}^n \sum_{j=1}^n c_i c_j k(\mathbf{x}_i, \mathbf{x}_j) \geq 0$  for any  $n > 0$ , any choice of  $n$  objects  $\mathbf{x}_1, \dots, \mathbf{x}_n$  in  $\mathcal{X}$ , and any choice of real numbers  $c_1, \dots, c_n$  in  $R$ .

# Kernel as Inner Product

Represent objects  $\mathbf{x} \in \mathcal{X}$  as a vector  $\phi(\mathbf{x}) \in R^p$ ,  $\mathcal{F}$ .  
defining a kernel for any  $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$  by  $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$ .

(Aronszajn, 1950)

**Theorem:** for any kernel  $k$  on a space  $\mathcal{X}$ , there exists a Hilbert space  $\mathcal{F}$  and a mapping  $\phi : \mathcal{X} \rightarrow \mathcal{F}$  such that  $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$ , for any  $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$ , where  $\langle u, v \rangle$  represents the dot product in the Hilbert space between any two points  $u, v \in \mathcal{F}$ .

Kernels can all be thought of as dot products in feature space  $\mathcal{F}$ .

The point  $\mathbf{x} \in \mathcal{X}$  are viewed as point  $\phi(\mathbf{x})$  in  $\mathcal{F}$ .

# Kernel Trick

- The *kernel trick* transforms any algorithm that solely depends on the *dot product* between two vectors.
- Whenever a *dot product* is used, it is replaced with the *kernel function*.
- The non-linear algorithm is the linear algorithm operating in the *feature space*.
- *Kernelization*: the operation that transforms a linear algorithm into a more general kernel method.

# SIR in the Euclidean Space

K.C. Li, (1991),

Sliced inverse regression for dimension reduction, *JASA* **86**, 316 – 342.

$y$  is a univariate variable.

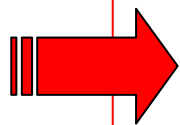
$f$  is an arbitrary function.

$$y = f(\beta'_1 \mathbf{x}, \dots, \beta'_K \mathbf{x}, \epsilon)$$

$\epsilon$  is a random variable independent of  $\mathbf{x}$ .

The  $\beta$ 's are referred to effective dimension reduction (*e.d.r.*) or projection directions.

$\mathbf{x}$  is a random vector with dimension  $p \times 1$ ,  $p \geq K$ .



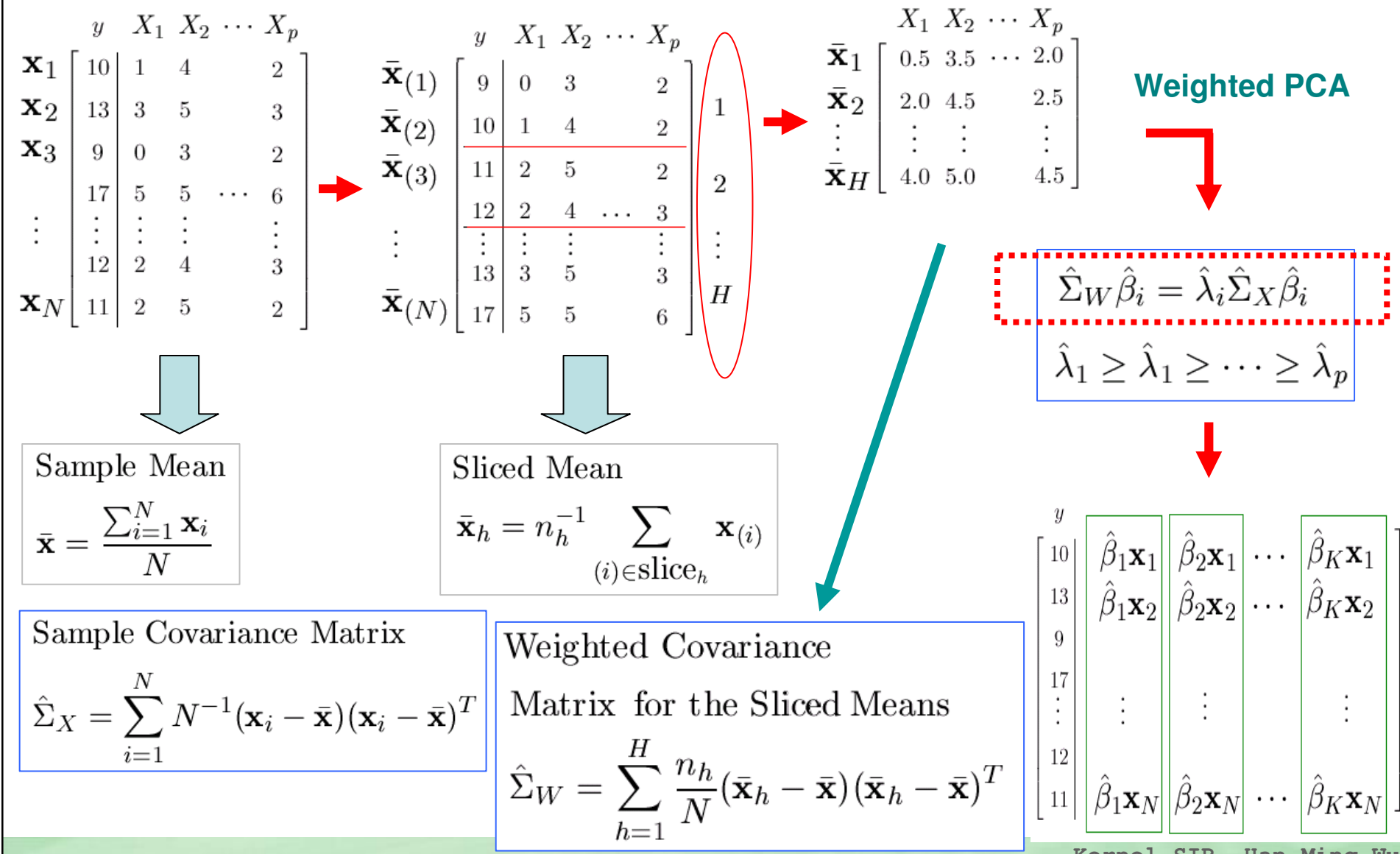
Sliced inverse regression (SIR) is a method for estimating the *e.d.r.* directions based on  $y$  and  $\mathbf{x}$ .

## Sufficient Dimension Reduction

**NOTE:** For more details, please see Dr. Dennis Cook, School of Statistics, University of Minnesota. (> 50 related articles published!)



# Classical SIR: Algorithm





# Kernel SIR in a Non-linear Feature Space

## Kernel SIR: Kernelize the SIR algorithm

- ▶ first map the data nonlinearity in to a feature space  $\mathcal{F}$  by

$$\phi : R^p \rightarrow \mathcal{F}, \mathbf{x} \mapsto \phi(\mathbf{x})$$

- ▶ We will show that even if  $\mathcal{F}$  has arbitrarily large dimensionality, for certain choices of  $\phi$ , we can still perform SIR in  $\mathcal{F}$ .
- ▶ Assume for the moment that our data mapped into feature space,  $\phi(\mathbf{x}_1), \dots, \phi(\mathbf{x}_n)$ , is centered, i.e.  $\sum_{i=0}^n \phi(\mathbf{x}_i) = 0$ .

# KSIR: Algorithm (1)

We have to find eigenvalues  $\lambda \geq 0$  and eigenvectors  $\beta \in \mathcal{F} \setminus \{0\}$  satisfying  $\Sigma_{\mathbf{wz}}\beta = \lambda\Sigma_{\mathbf{zz}}\beta$ .

$$\Sigma_{\mathbf{zz}} = \frac{1}{n} \sum_{i=1}^n \phi(\mathbf{x}_i)\phi(\mathbf{x}_i)^T.$$

$$p_h = \frac{\sum_{i=1}^n \delta_h(y_i)}{n} = \frac{n_h}{n}, \delta_h(y_i) = 1, \text{ if } y_i \in I_h, \delta_h(y_i) = 0, \text{ o.w.}$$

$$\Sigma_{\mathbf{wz}} = \sum_{h=1}^H p_h \bar{\phi}(\mathbf{m}_h)\bar{\phi}(\mathbf{m}_h)^T.$$

$$\bar{\phi}(\mathbf{m}_h) = \frac{1}{np_h} \sum_{i=1}^n \phi(\mathbf{x}_i)\delta_h(y_i)$$

All solutions  $\beta$  lie in span  $\{\phi(\mathbf{x}_1), \dots, \phi(\mathbf{x}_n)\}$ .

➤ The equivalent system  $\lambda \langle \phi(\mathbf{x}_k), \Sigma_{\mathbf{zz}}\beta \rangle = \langle \phi(\mathbf{x}_k), \Sigma_{\mathbf{wz}}\beta \rangle$ , for all  $k = 1, \dots, n$ .

➤ there exists  $\alpha_1, \dots, \alpha_n$  such that  $\beta = \sum_{i=1}^n \alpha_i \phi(\mathbf{x}_i)$ .

Define  $\mathbf{K} := \{\mathbf{k}_{ij} = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle\}_{n \times n}$ .

# KSIR: Algorithm (2)

The equivalent system  $\lambda \langle \phi(\mathbf{x}_k), \Sigma_{\mathbf{z}\mathbf{z}} \boldsymbol{\beta} \rangle = \langle \phi(\mathbf{x}_k), \Sigma_{\mathbf{w}\mathbf{z}} \boldsymbol{\beta} \rangle$ , for all  $k = 1, \dots, n$ .

$$\begin{aligned}
 \lambda \langle \phi(\mathbf{x}_k), \Sigma_{\mathbf{z}\mathbf{z}} \boldsymbol{\beta} \rangle &= \lambda \langle \phi(\mathbf{x}_k), \left\{ \frac{1}{n} \sum_{j=1}^n \phi(\mathbf{x}_j) \phi(\mathbf{x}_j)^T \right\} \left\{ \sum_{i=1}^n \alpha_i \phi(\mathbf{x}_i) \right\} \rangle \\
 &= \lambda \frac{1}{n} \sum_{i=1}^n \alpha_i \langle \phi(\mathbf{x}_k), \sum_{j=1}^n \phi(\mathbf{x}_j) \rangle \langle \phi(\mathbf{x}_j), \phi(\mathbf{x}_i) \rangle \\
 &= \lambda \frac{1}{n} \sum_{i=1}^n \alpha_i \sum_{j=1}^n K_{kj} K_{ji}, \quad \forall k = 1, \dots, n \\
 &\Rightarrow \lambda \frac{1}{n} \mathbf{K} \mathbf{K}^T \boldsymbol{\alpha}
 \end{aligned}$$

# KSIR: Algorithm (3)

$$\langle \phi(\mathbf{x}_k), \Sigma_{\mathbf{wz}} \beta \rangle$$

$$= \langle \phi(\mathbf{x}_k), \left\{ \sum_{h=1}^H p_h \bar{\phi}(\mathbf{m}_h) \bar{\phi}(\mathbf{m}_h)^T \right\} \left\{ \sum_{i=1}^n \alpha_i \phi(\mathbf{x}_i) \right\} \rangle$$

$$= \sum_{i=1}^n \alpha_i \langle \phi(\mathbf{x}_k), \sum_{h=1}^H p_h \bar{\phi}(\mathbf{m}_h) \rangle \langle \bar{\phi}(\mathbf{m}_h), \phi(\mathbf{x}_i) \rangle$$

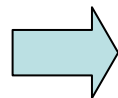
$$= \sum_{i=1}^n \alpha_i \sum_{h=1}^H \frac{\sum_{j=1}^n \mathbf{K}_{kj} \delta_h(y_j)}{n} \frac{\sum_{j=1}^n \mathbf{K}_{ji} \delta_h(y_j)}{\sum_{j=1}^n \delta_h(y_j)}$$

$$= \frac{1}{n} \sum_{i=1}^n \alpha_i \sum_{h=1}^H \frac{\sum_{j=1}^n \mathbf{K}_{kj} \delta_h(y_j)}{\sqrt{\sum_{j=1}^n \delta_h(y_j)}} \frac{\sum_{j=1}^n \mathbf{K}_{ji} \delta_h(y_j)}{\sqrt{\sum_{j=1}^n \delta_h(y_j)}}, \quad \forall k = 1, \dots, n$$

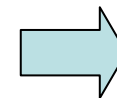
$$\Rightarrow \frac{1}{n} \mathbf{K} \mathbf{E}_H \mathbf{K} \alpha$$

$$\mathbf{E}_H = \sum_{h=1}^H \frac{\mathbf{1}_h \mathbf{1}_h^t}{n_h}, \quad \mathbf{1}_h = [\delta_h(y_1) \cdots \delta_h(y_n)]^t.$$

$$\Sigma_{\mathbf{wz}} \beta = \lambda \Sigma_{\mathbf{zz}} \beta$$



$$\lambda \mathbf{K} \mathbf{K} \alpha = \mathbf{K} \mathbf{E}_H \mathbf{K} \alpha$$



$$\lambda \mathbf{K} \alpha = \mathbf{E}_H \mathbf{K} \alpha$$

$$\begin{aligned} \langle \phi(\mathbf{x}_k), \sum_{h=1}^H p_h \bar{\phi}(\mathbf{m}_h) \rangle &= \sum_{h=1}^H p_h \langle \phi(\mathbf{x}_k), \bar{\phi}(\mathbf{m}_h) \rangle \\ &= \sum_{h=1}^H p_h \langle \phi(\mathbf{x}_k), \frac{\sum_{j=1}^n \phi(\mathbf{x}_j) \delta_h(y_j)}{\sum_{j=1}^n \delta_h(y_j)} \rangle \\ &= \sum_{h=1}^H \frac{\sum_{j=1}^n \mathbf{K}_{kj} \delta_h(y_j)}{n} \end{aligned}$$

$$\begin{aligned} \langle \bar{\phi}(\mathbf{m}_h), \phi(\mathbf{x}_i) \rangle &= \left\langle \frac{\sum_{j=1}^n \phi(\mathbf{x}_j) \delta_h(y_j)}{\sum_{j=1}^n \delta_h(y_j)}, \phi(\mathbf{x}_i) \right\rangle \\ &= \frac{\sum_{j=1}^n \mathbf{K}_{ji} \delta_h(y_j)}{\sum_{j=1}^n \delta_h(y_j)} \end{aligned}$$

# Normalization and Projection

Let  $\lambda_1 \geq \dots \geq \lambda_n$  denote the eigenvalues, and  $\alpha_1, \dots, \alpha_n$  the corresponding complete set of eigenvectors, with  $\lambda_t$  being the first nonzero eigenvalues.

We normalize  $\alpha_1, \dots, \alpha_n$  by requiring that the corresponding vectors in  $\mathcal{F}$  be normalized:  $\langle \beta_k, \beta_k \rangle = 1$  for all  $k = 1, \dots, t$ .

## Normalization Condition:

$$1 = \sum_{i=1}^n \sum_{j=1}^n \alpha_i^k \alpha_j^k \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle = \langle \alpha^k, \mathbf{K} \alpha^k \rangle = \lambda_k \langle \alpha^k, \alpha^k \rangle$$

## Projections on the eigenvectors $\beta_k$ in $\mathcal{F}$ , $k = 1, \dots, t$ :

Let  $\mathbf{x}$  be a test point, with an image  $\phi(\mathbf{x})$  in  $\mathcal{F}$ , then

$$\langle \beta_k, \phi(\mathbf{x}) \rangle = \sum_{i=1}^n \alpha_i^k \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}) \rangle = \sum_{i=1}^n \alpha_i^k \mathbf{K}(\mathbf{x}_i, \mathbf{x})$$

# Reduced Features

**Data Transform Dialog**

Data Set: SuYun-sinData.txt [200 x 5]

New Data: tSuYun-sinData.txt [200 x 200]

Transform Methods: Kernel Transform

Settings:

- Standardize Data by Columns
- Kernel Type with Parameters: Gaussian RBF, degree: 2, scale: 0.05, offset: 0
- Sampling Columns
  - Random 200
  - Leading SVD% 95 %
  - Leading SVD# 50

OK Cancel

***For Theoretical details:***

Lee, Y.J. and Huang, S.Y. (2006), [Reduced support vector machines: a statistical theory](#), *IEEE Transactions on Neural Networks*, accepted.

<http://dmlab1.csie.ntust.edu.tw/downloads>

# KSIR for Nonlinear Dimensional Reduction and Data Visualization

SIR  $\Rightarrow$   $y = f(\beta_1^t \mathbf{x}, \dots, \beta_B^t \mathbf{x}, \epsilon)$

PCA performed on the random vector  $E(\mathbf{x}|y)$  instead of  $\mathbf{x}$ .

KSIR  $\Rightarrow$   $y = f(\beta_1^t \Phi(\mathbf{x}), \dots, \beta_B^t \Phi(\mathbf{x}), \epsilon)$ , where  $\beta_b \in \mathbb{R}^d$ ,  $d \leq \infty$ .

PCA performed on the random vector  $E(\phi(\mathbf{x})|y)$  instead of  $\phi(\mathbf{x})$ .

## Simulation Data

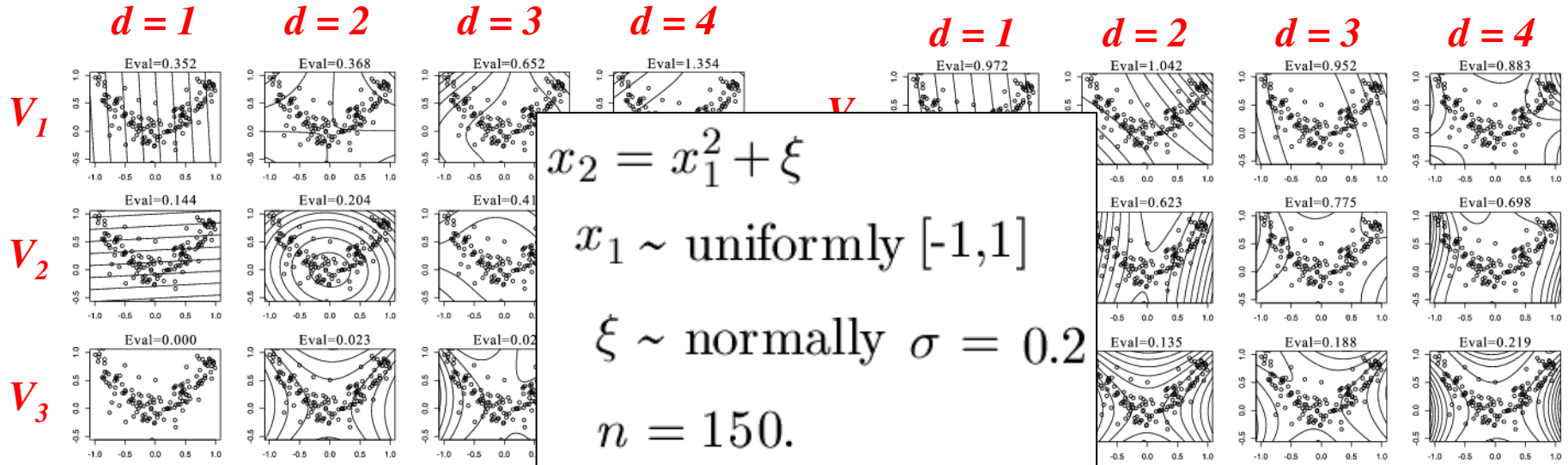
- ❑ Square Data (150x2, na)
- ❑ Three Clusters Data (220x2, no.class=3)
- ❑ Li Data Model (6.3) (400x10, y=conti)

## Real Data

- ❑ Wine Data (178x18, no.class=3)
- ❑ Pendigit Data (7494x16, no.class=10)

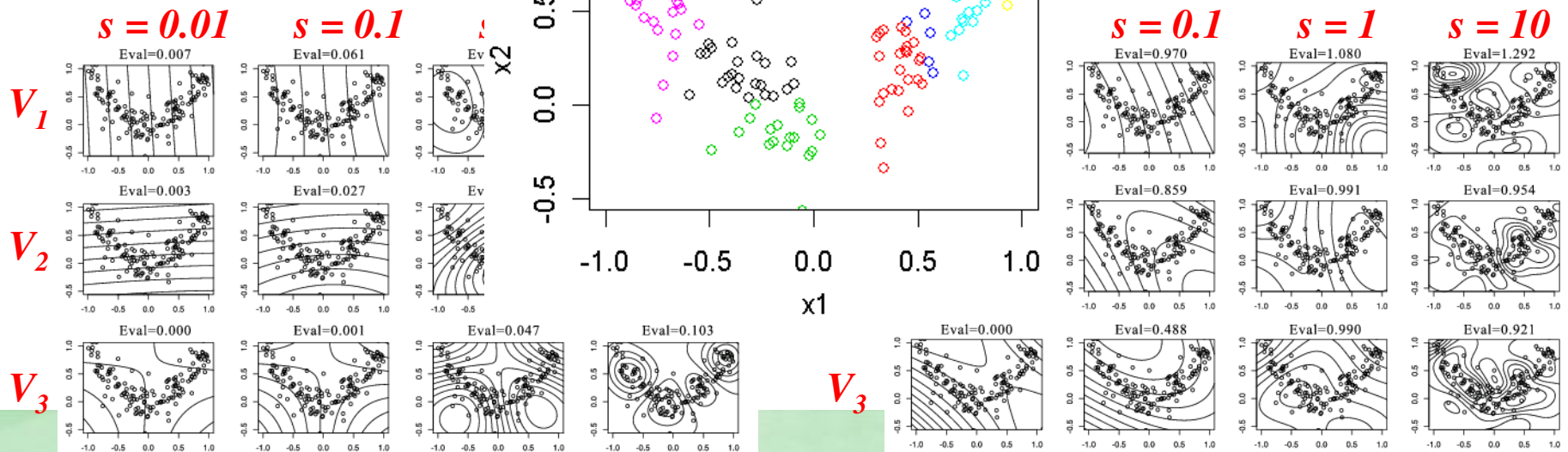
# Visualization (1): Square Data

H=8

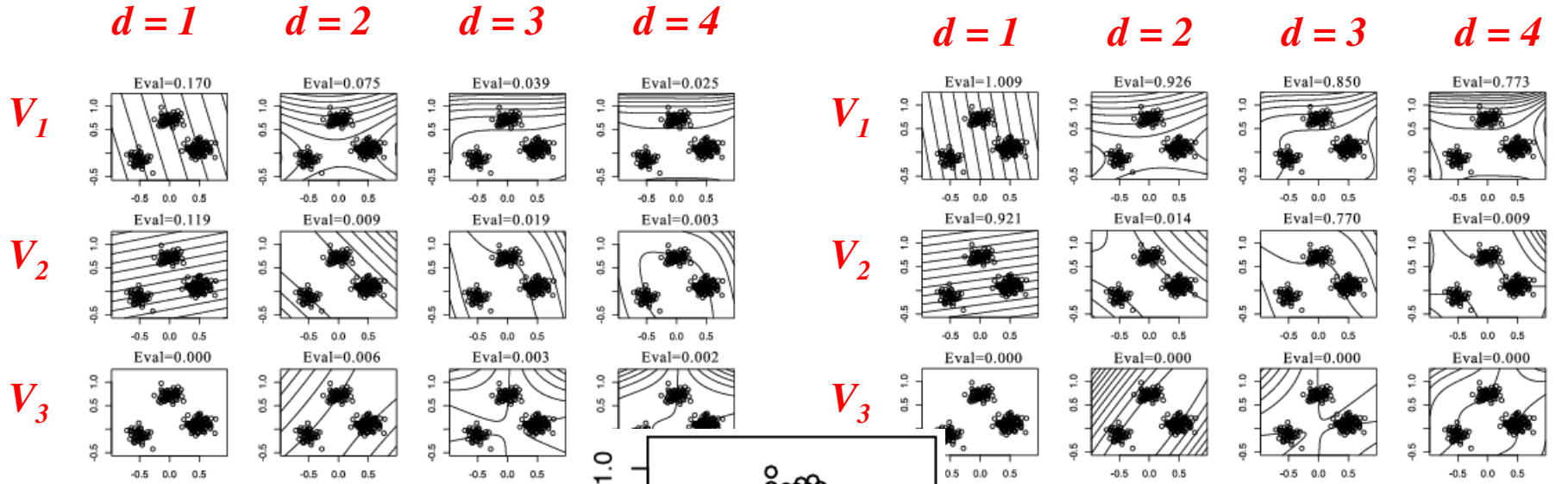


## KPCA

## KSIR

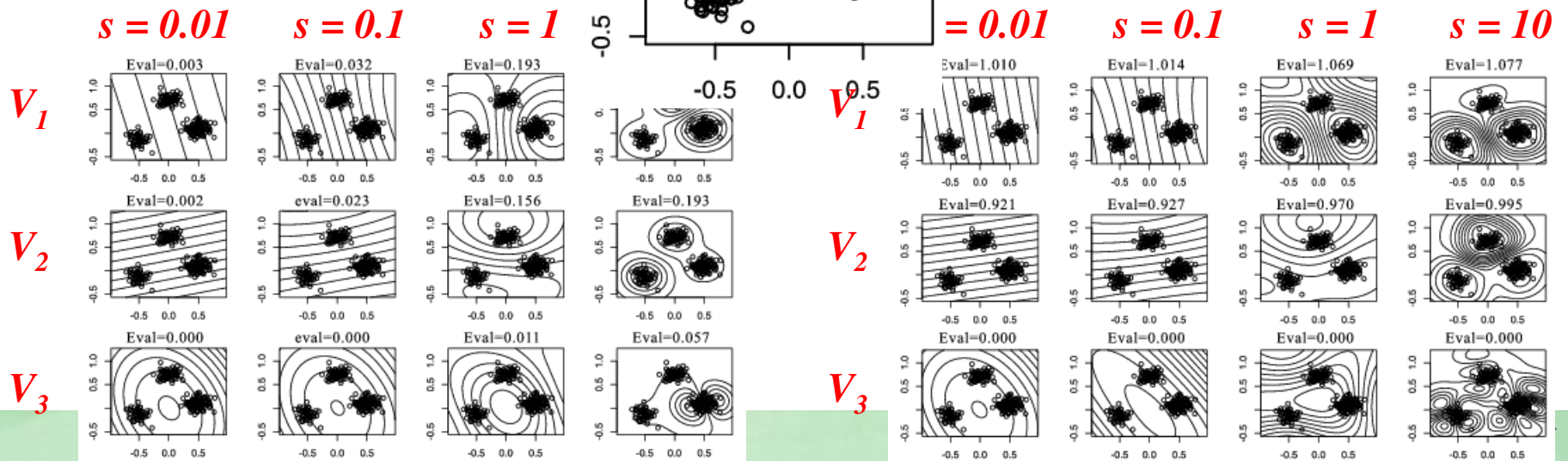


# Visualization (2): Three Clusters Data



**KPCA**

**KSIR**





# Visualization (3): Li Data Model (6.3)

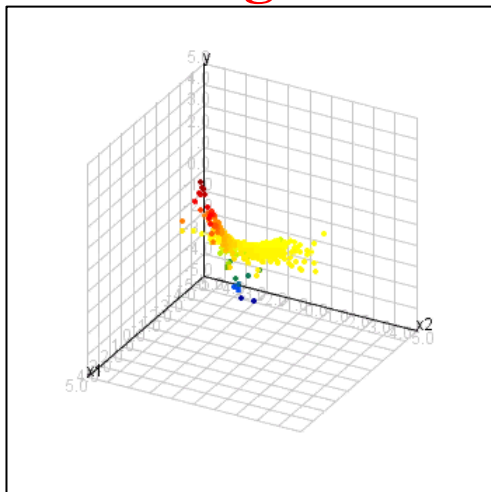
$$y = \frac{x_1}{0.5 + (x_2 + 1.5)^2} + \sigma \cdot \epsilon$$

$x_1, x_2, x_3, \dots, x_p$  standard normal

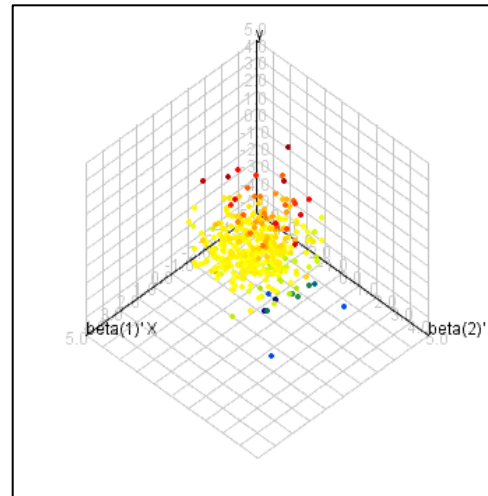
take  $p = 10$  with  $\sigma = 0.5$ .

$n = 400$ .

**Orig**

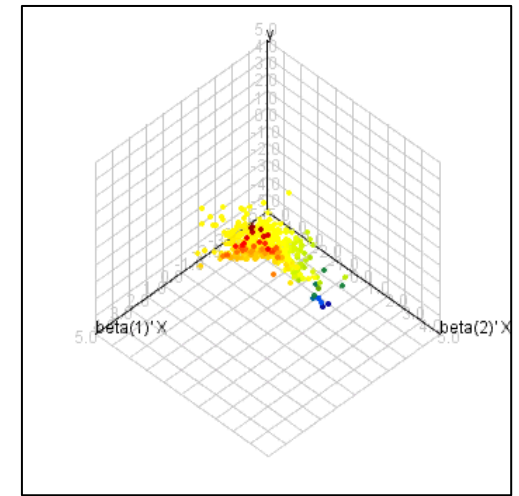


**PCA**



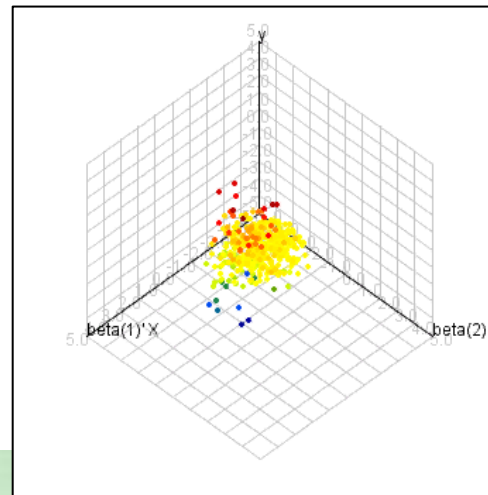
**SIR**

H=13

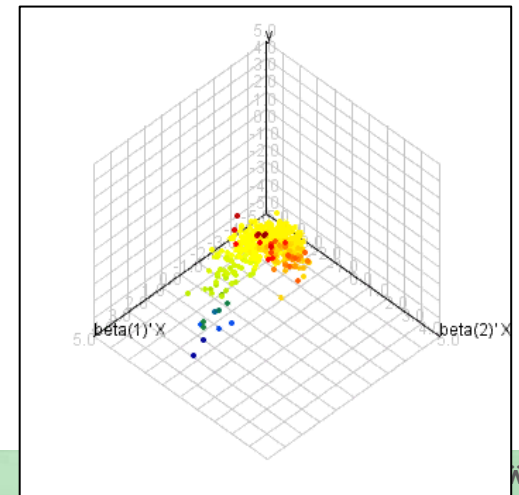


**KPCA**

Gaussian s=0.05



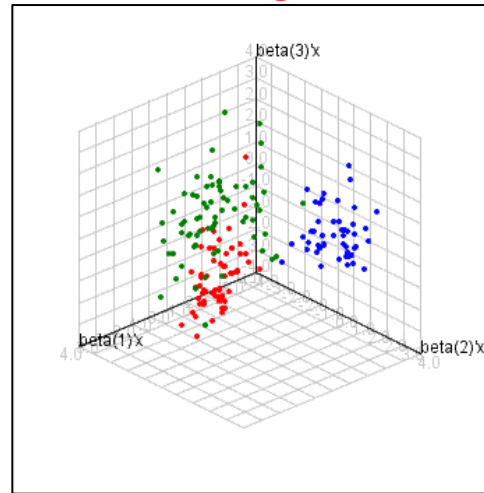
**KSIR**



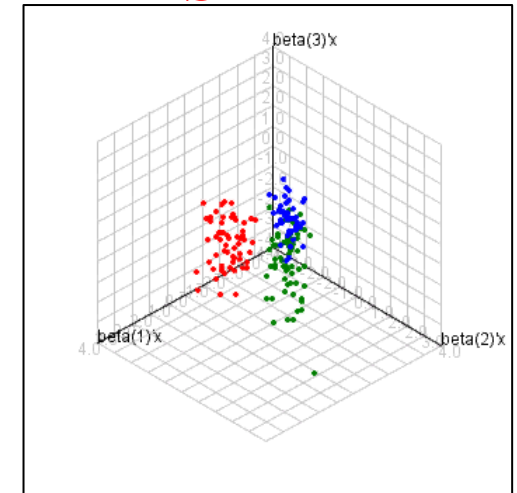
# Visualization (4): Wine Data

- Wine data (n=178) are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars.
- The analysis determined the quantities of 13 constituents found in each of the three types of wines.

## PCA

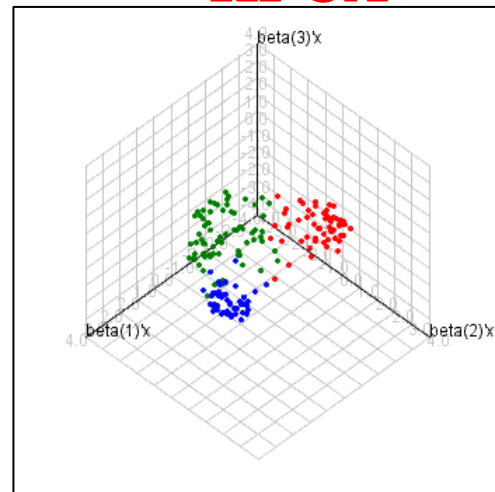


## SIR

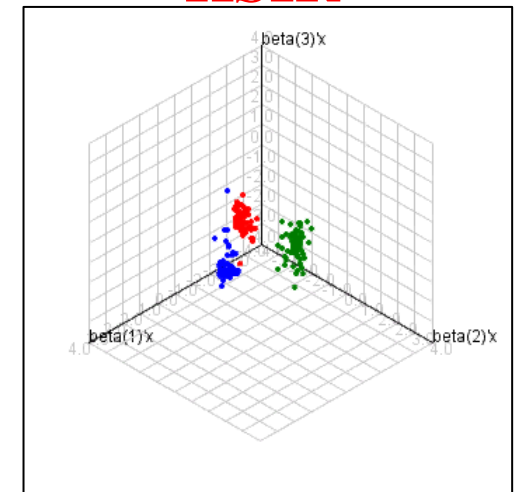


Gaussian  $s=0.05$

## KPCA



## KSIR

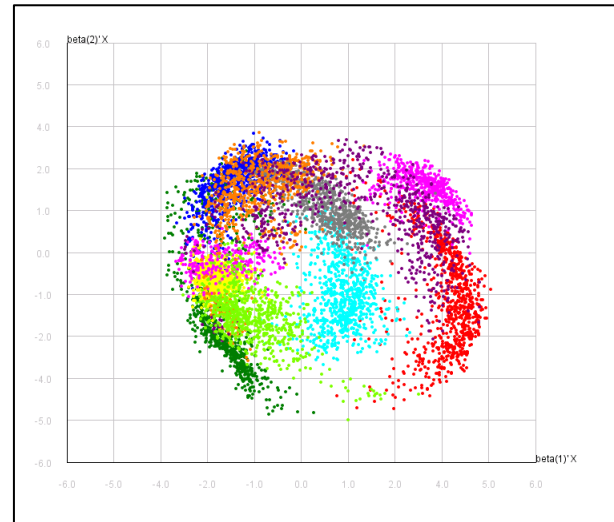


# Visualization (5): Pendigit Data

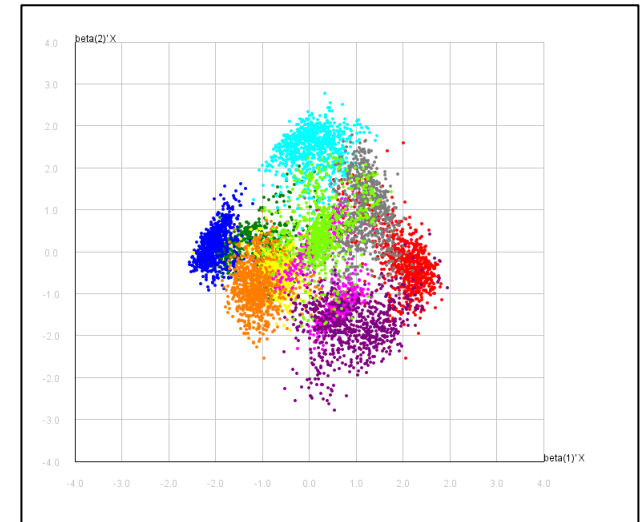
- Pen-based recognition of handwritten Digits
- 7494 instances, 16 attributes
- 10 classes

■ 0	: 780
■ 1	: 779
■ 2	: 780
■ 3	: 719
■ 4	: 780
■ 5	: 720
■ 6	: 778
■ 7	: 719
■ 8	: 719
■ 9	: 719

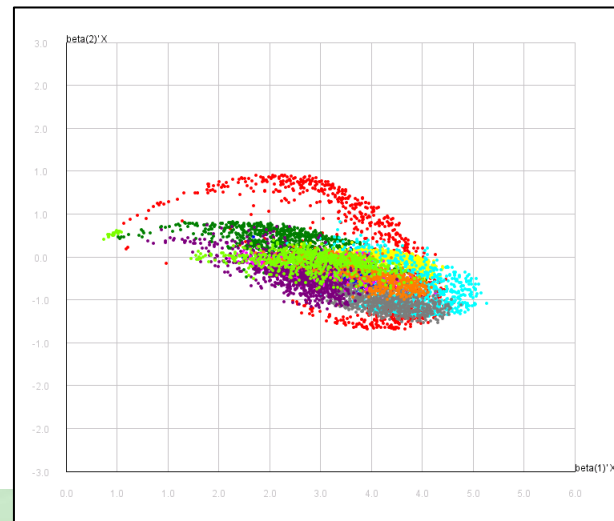
## PCA



## SIR

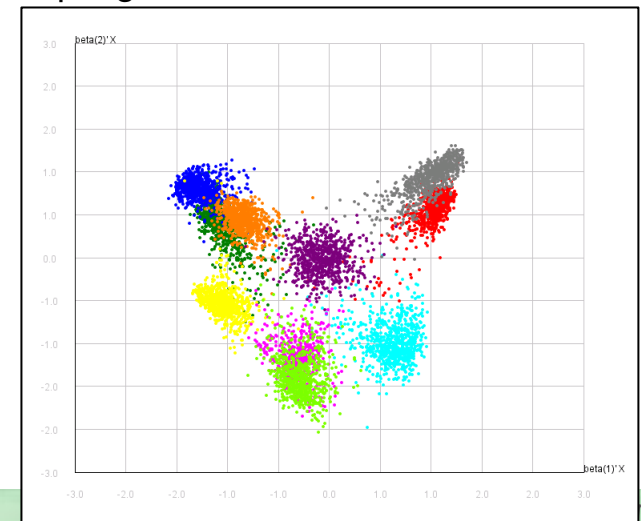


## KPCA



Gaussian 0.05  
Random sampling 200

## KSIR



# Classification (1): UCI Data Sets

Dataset	$n$	$p$	$C$
Wisconsin Breast Cancer (bcw)	683	9	2 (444, 239)
Glass Identification (gls)	214	9	6 (70, 76, 17, 13, 9, 29)
Ionosphere (ion)	351	33	2 (225, 126)
Iris Plants (iri)	150	4	3 (50×3)
BUPA liver disorders (liv)	345	6	2 (145, 200)
Pima Indians Diabetes (pid)	768	8	2 (500, 268)
StatLog image segmentation (seg)	2310	18	7 (330×7)
StatLog vehicle silhouettes (veh)	846	18	4 (212, 217, 218, 199)
Waveform Database Generator (wav)	600	21	3 (200×3)
Wine recognition data (win)	178	13	3 (59, 71, 48)

-△-	X
-+·	PCA
-×·	SIR
-◇-	KPCA
-▽·	KSIR

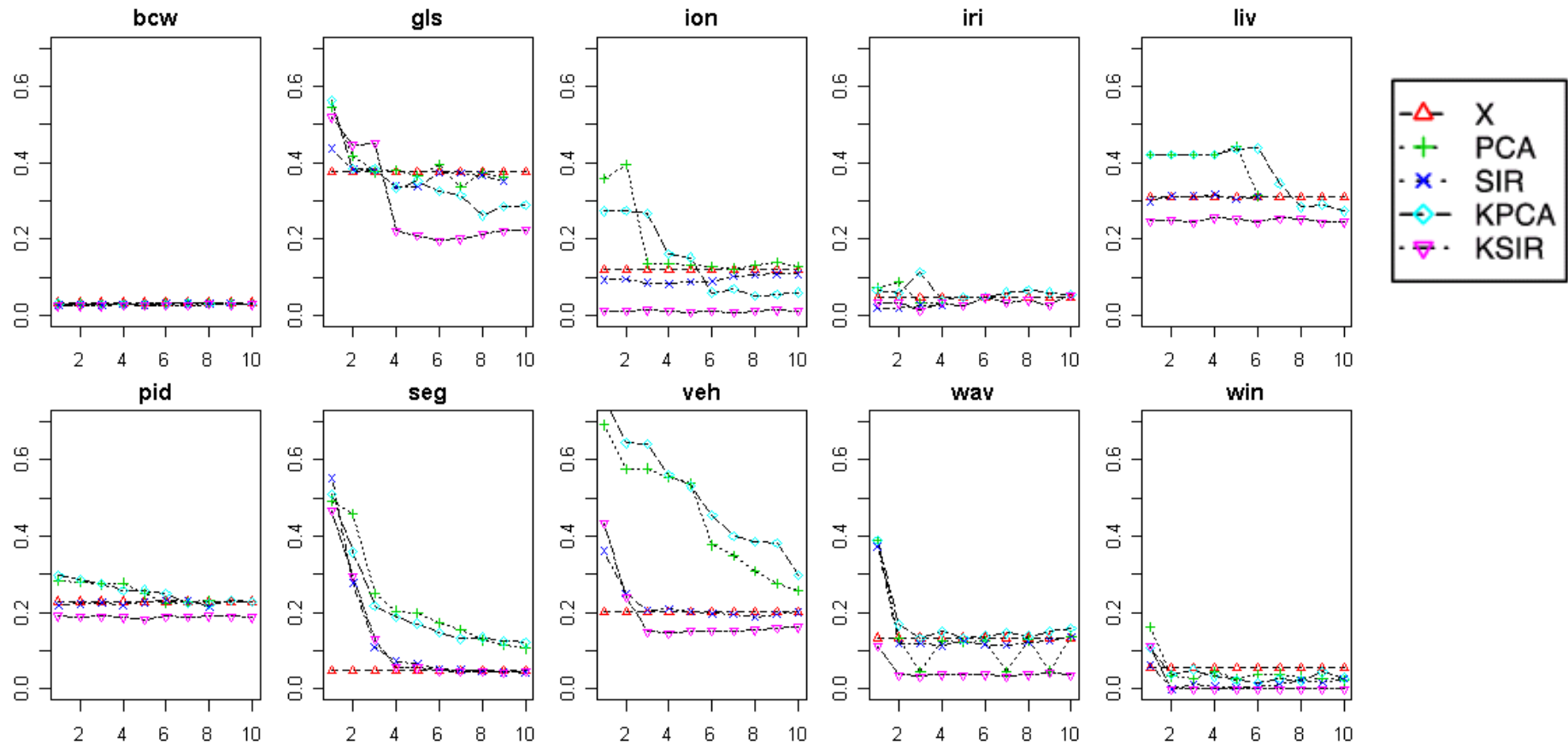
Gaussian 0.05

Random sampling 200

**Linear Support Vector Machine**

**10-fold classification error rates**

# 10-fold Classification ERs: UCI Data Sets



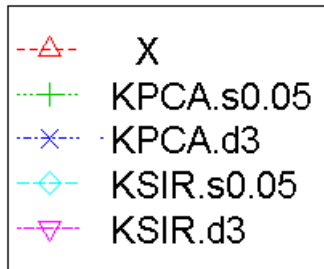
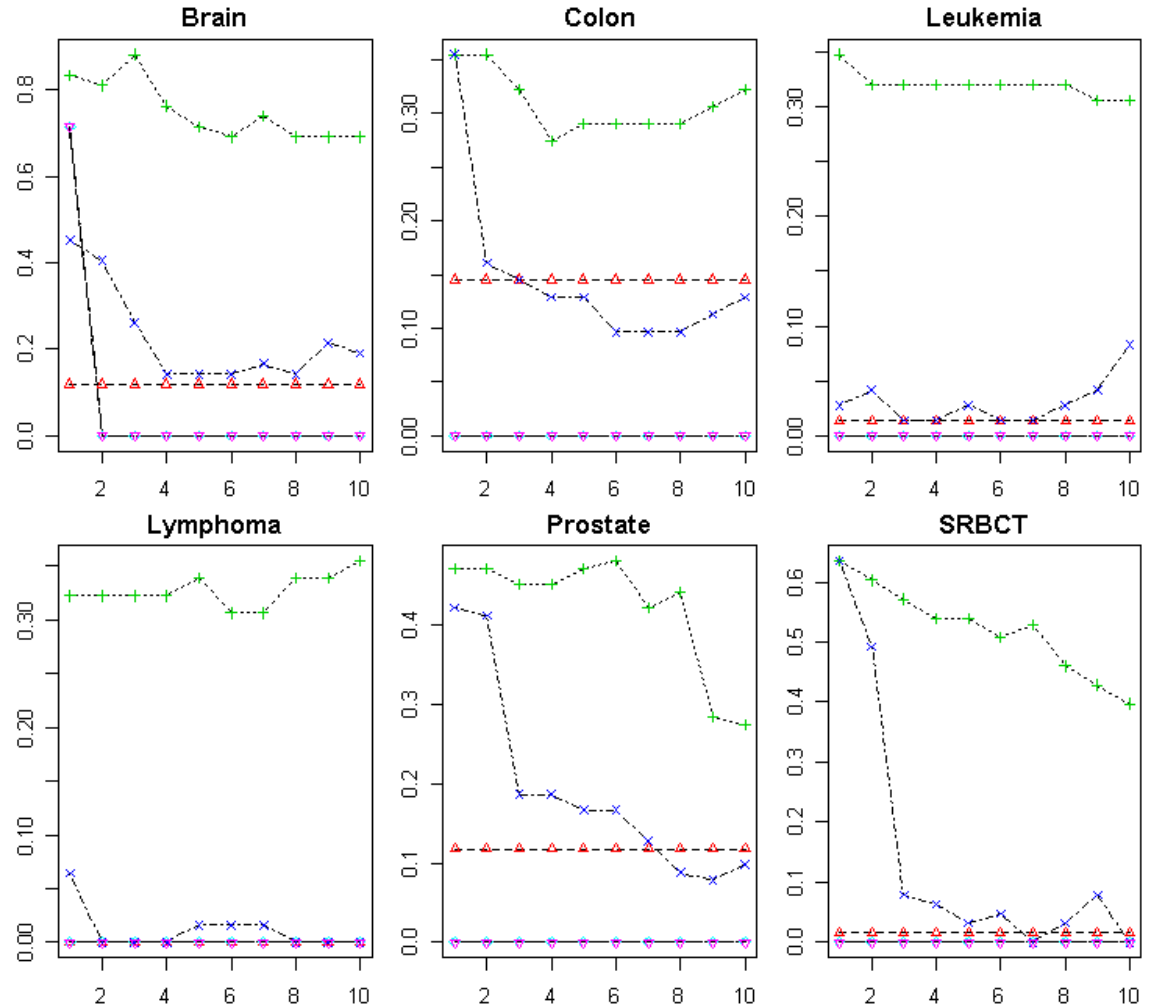
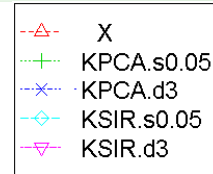
# Classification: Microarray Data Sets

Dataset	Publication	$n$	$p$
Leukemia	Golub <i>et al.</i> (1999)	72	3571
Colon	Alon <i>et al.</i> (1999)	62	2000
Prostate	Singh <i>et al.</i> (2002)	102	6033
Lymphoma	Alizadeh <i>et al.</i> (2000)	62	4026
SRBCT	Khan <i>et al.</i> (2001)	63	2308
Brain	Pomeroy <i>et al.</i> (2002)	42	5597

Dataset	$C$	Response
Leukemia	2 (47, 25)	Subtypes of leukemia
Colon	2 (22, 40)	Tumor/normal tissue
Prostate	2 (50, 52)	Tumor/normal tissue
Lymphoma	3 (42, 9, 11)	Subtypes of lymphoma
SRBCT	4 (23, 20, 12, 8)	Different tumor types
Brain	5 (10, 10, 10, 4, 8)	Different tumor types

## Linear Support Vector Machine

## Leave-one-out classification error rates



# Conclusion and Future Direction

- ❑ Use “*Kernel Trick*” to study the linear algorithm of SIR in the *Feature Space*.
  - ◆ Nonlinear dimension reduction subspace from  $X$  viewpoint
  - ◆ Linear dimension reduction subspace in  $H_k$

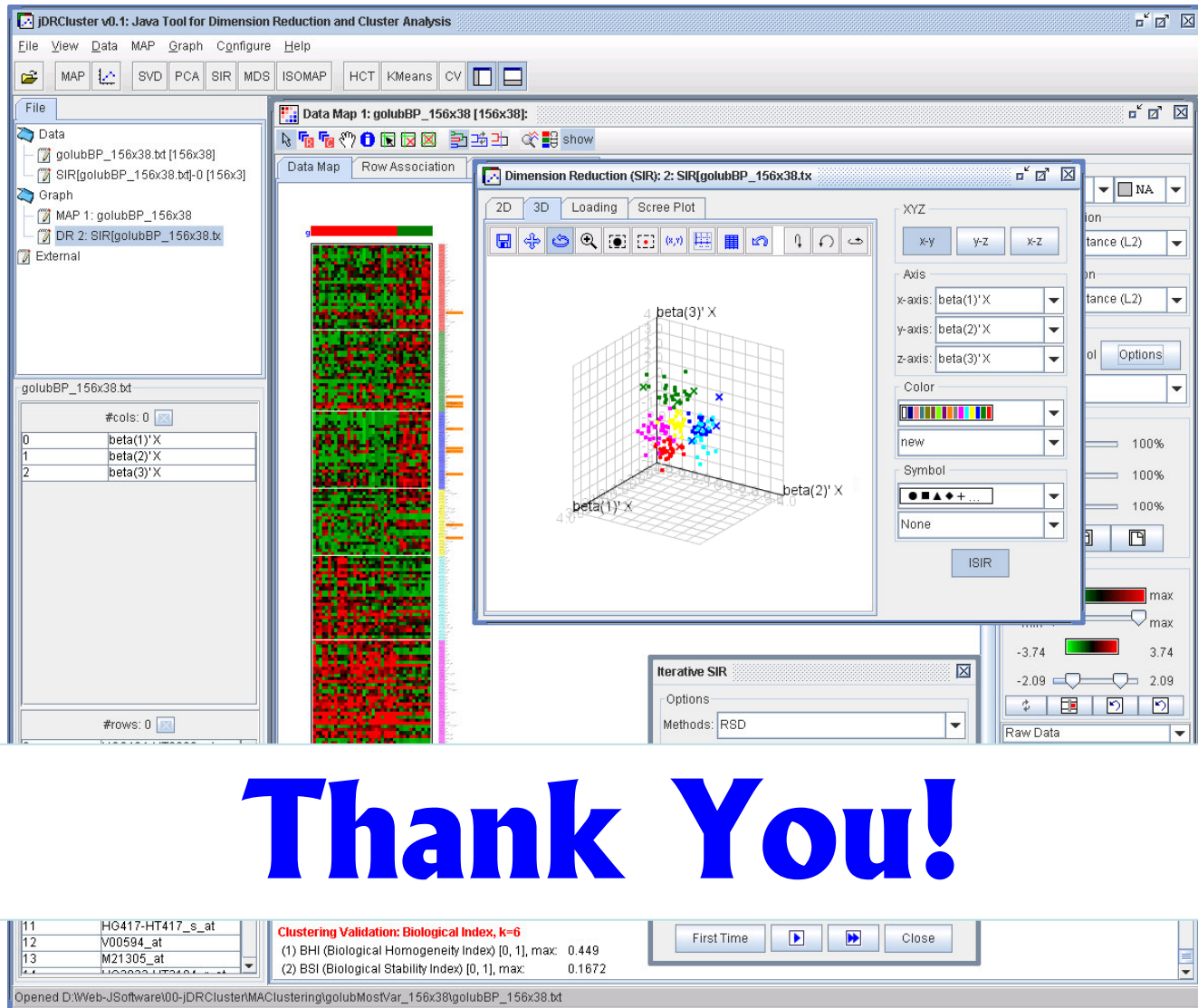
- ❑ Nonlinear Dimension Reduction and Visualization
- ❑ For Classification.
- ❑ Apply to Clustering Problem.

## **SIR/KSIR:**

A tool for **feature extraction** and **data exploratory analysis**.

- ❑ Theoretical Prosperities of Kernel SIR.
- ❑ Selection of Kernel Parameters (model selection).

# jDRCluster: Dimension Reduction and Cluster Analysis



# Thank You!