C06

核方法
Kernel Method

吳漢銘
國立臺北大學 統計學系

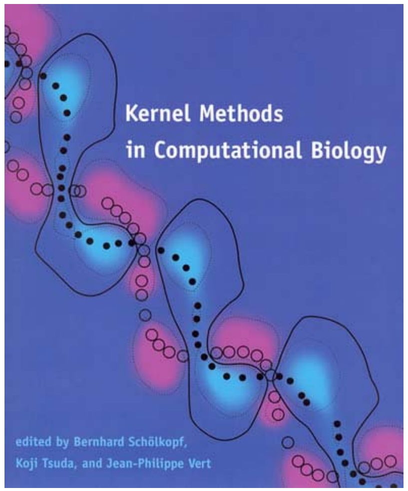
http://www.hmwu.idv.tw

本章大綱

- Kernel Methods, Kernel Trick
- Kernel Data and Its Properties

- PCA/SIR in the Euclidean Space
- Kernel PCA, Kernel SIR in a Non-linear Feature Space

- Relations Towards Other Methods
- KSIR for Nonlinear Dimensional Reduction
- Experiments on Classification



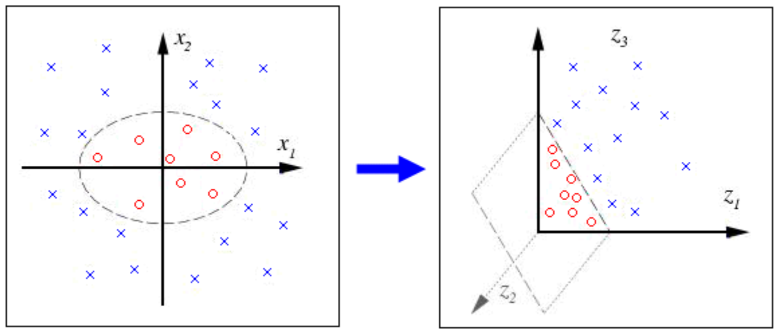
核方法 (Kernel Methods)

- 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***", they argue that a *positive definite kernel* is identical to a *dot product* in the feature space.

$$\Phi : \mathbb{R}^2 \to \mathbb{R}^3$$
$$(x_1, x_2) \mapsto (z_1, z_2, z_3) := (x_1^2, \sqrt{2}\, x_1 x_2, x_2^2)$$

- Boser et al (1992), to construct ***SVMs***, a generalization of the so-called optimal hyperplane algorithm.



Input Space **Feature Space**

- 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 20 years, there have seen a large number of ***kernelization*** of various algorithms. (PCA, LDA, CCA, PLS,…)

Prepare Kernel Data

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

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

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

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

$\mathbf{x}_i$

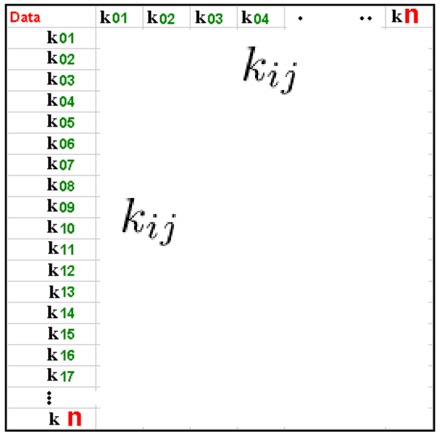
$\mathbf{x}_j$

Data	V1	V2	V3	V4	...	Vp
x01	-0.48	-0.42	0.87	0.92		-0.18
x02	-0.39	-0.58	1.08	1.21		-0.33
x03	0.87	0.25	-0.17	0.18		-0.44
x04	1.57	1.03	1.22	0.31		-0.49
x05	-1.15	-0.86	1.21	1.62		0.16
x06	0.04	-0.12	0.31	0.16		-0.06
x07	2.95	0.45	-0.40	-0.66		-0.38
x08	-1.22	-0.74	1.34	1.50		0.29
x09	-0.73	-1.08	-0.79	-0.02		0.44
x10	-0.58	-0.40	0.13	0.58		0.02
x11	-0.50	-0.42	0.66	1.05		0.06
x12	-0.86	-0.29	0.42	0.46		0.10
x13	-0.16	0.29	0.17	-0.28		-0.55
x14	-0.36	-0.03	-0.03	-0.08		-0.25
x15	-0.72	-0.85	0.54	1.04		0.24
x16	-0.78	-0.52	0.26	0.20		0.48
x17	0.60	-0.55	0.41	0.45		-0.66
⋮						
xn	-2.29	-0.64	0.77	1.60		0.55

?

$\mathbf{\Phi}(\mathbf{x}_i)$

$\mathbf{\Phi}(\mathbf{x}_j)$



- 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.

A real-valued comparison function $k : \mathcal{X} \times \mathcal{X} \rightarrow R$ is used, and data set $\mathbf{X}_{[n \times p]}$ is represented by the $n \times n$ matrix of pairwise comparisons $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$.

- The representation as a square matrix does not depend on the nature of the objects to be analyzed.
- 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 $\mathbf{x}_1, \cdots, \mathbf{x}_n$ in $\mathcal{X}$, and any choice of real numbers $c_1, \cdots, c_n$ in R.

Kernel as Inner Product

The inner product between vectors is the first kernel we encounter.

(called **linear kernel**).

$\mathcal{X} = R^p$ object $\mathbf{x}_i = (x_{i1}, \cdots, x_{ip})^t$.

symmetric and positive definite

One is tempted to compare such vectors using their inner product:

$$\text{for any } \mathbf{x}_i, \mathbf{x}_j \in R^p,\ k_L(\mathbf{x}_i, \mathbf{x}_j) := \mathbf{x}_i^T \mathbf{x}_j = \sum_{t=1}^{p} x_{it} x_{jt}.$$

Represent objects $\mathbf{x} \in \mathcal{X}$ as a vector $\phi(\mathbf{x}) \in R^p$,

defining a kernel for any $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$ by $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$.

Theorem: for any kernel k on a space $\mathcal{X}$, there exists a Hilbert space $\mathcal{F}$ and a mapping $\phi : \mathcal{X} \to \mathcal{F}$ 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 $u, v \in \mathcal{F}$. (Aronszajn 1950)

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 $\mathcal{F}$.

A Hilbert space is a vector space endowed with a dot product that is complete for the norm induced. R^p with the classical inner product is an example of a finite-dimensional Hilbert space.

David Hilbert (01/23/1862 – 02/14/1943)
German mathematician

Reproducing Kernel Hilbert Space

Linear kernel and their associated functional space:

Let k be a kernel on a space $\mathcal{X}$, to show k is associated with a set of real-valued functions on $\mathcal{X}$, $\mathcal{H}_k \subset \{f : \mathcal{X} \to R\}$, endowed with a structure of Hilbert space.

$\mathcal{X} = R^p$ the functional space is f: $R^p \to R$ the associated norm is

$$\mathcal{H}_k = \{f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}, \mathbf{w} \in R^p\} \qquad \| f \|_{\mathcal{H}_k} = \| \mathbf{w} \| \text{ for } f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}.$$

The set $\mathcal{H}_k$ is defined as the set of function $f : \mathcal{X} \to R$ of the form $f(\mathbf{x}) = \sum_{i=1}^n \alpha_i k(\mathbf{x}_i, \mathbf{x})$, for $n > 0$, a finite number of points $\mathbf{x}_1, \cdots, \mathbf{x}_n \in \mathcal{X}$, and $\mathbf{w}$ finite number of weights $\alpha_1, \cdots, \alpha_n \in R$, together with their limits under the norm $\| f \|^2_{\mathcal{H}_k} := \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j)$.

$\mathcal{H}_k$ is a Hilbert space, with a dot product defined for two elements $f(\mathbf{x}) = \sum_{i=1}^n \alpha_i k(\mathbf{x}_i, \mathbf{x})$ and $g(\mathbf{x}) = \sum_{j=1}^m \alpha'_j k(\mathbf{x}'_j, \mathbf{x})$ by $\langle f, g\rangle = \sum_{i=1}^n \sum_{j=1}^m \alpha_i \alpha'_j k(\mathbf{x}_i, \mathbf{x}'_j)$.

The value $f(\mathbf{x})$ of a function $f \in \mathcal{H}_k$ at a point $\mathbf{x} \in \mathcal{X}$ can be expressed as a dot product in $\mathcal{H}_k$, $f(\mathbf{x}) = \langle f, k(\mathbf{x}, \cdot)\rangle$.

taking $f(\cdot) = k(\mathbf{x}, \cdot)$: the reproducing property valid for any $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \langle k(\mathbf{x}_i, \cdot), k(\mathbf{x}_j, \cdot)\rangle.$$

The functional space $\mathcal{H}_k$ is usually called the reproducing kernel Hilbert space (RKHS) associated with k.

Kernel Trick

The Hilbert space $\mathcal{H}_k$ is one possible feature space associated with the kernel k, when we consider the mapping $\phi : \mathcal{X} \to \mathcal{H}$ defined by $\phi(\mathbf{x}) := k(\mathbf{x}, \cdot)$.

- The ***kernel Trick*** was first published in the 1964 paper *Theoretical foundations of the potential function method in pattern recognition learning*.
- Any algorithm for vectorial data that can be expressed only in terms of ***dot products*** between vectors can be performed implicitly in the feature space associated with any kernel, by replacing each dot product by a kernel evaluation.
- It is a very convenient trick to transform *linear* methods, such as LDA or PCA into *nonlinear* methods, by simply replacing the classic dot product by a more general kernel.
- The kernel trick transforms any algorithm that solely dependents on the dot product between two vectors. Wherever 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.

$$\boxed{k(x, x') = \langle \Phi(x), \Phi(x') \rangle}$$

Kernel Data: Properties

- Raw data on Euclidean space $\boldsymbol{R}^p$
  - Kernel data on a RKHS $\boldsymbol{H}_k$
- Via a specific statistical notion of classical approach on $\boldsymbol{R}^p$
  - Kernel approach on $\boldsymbol{H}_k$, which is exactly the classical procedure on kernel data.

- **Main goal**: Parallel to the classical multivariate statistical analysis, we aim to develop an analysis tool in the Gaussian reproducing kernel Hilbert space.

- **Main advantage**: Nonparametric approach with "parametric-plus" computing load.
  parametric: classical multivariate analysis procedures.
  plus: kernel data preparation.

- **Kernel map can bring the data distribution to better elliptical symmetry.** Kernel data are (with empirical and theoretical justification)
  - Better elliptically symmetrically distributed.
  - Better approximately normal (Gaussian)

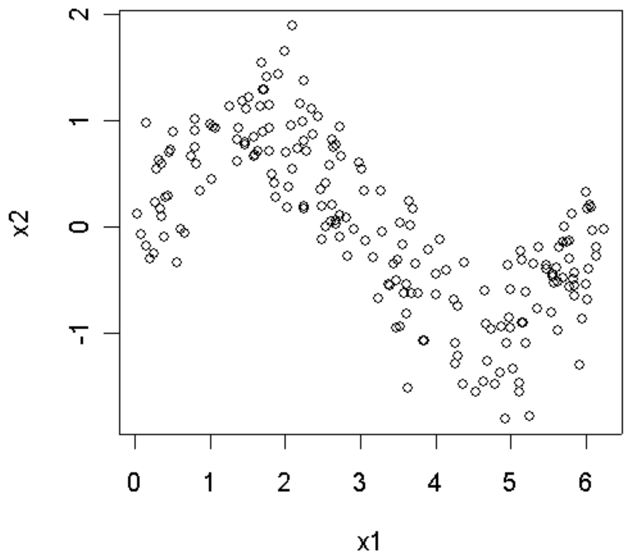
Example: Better Elliptical Symmetry

- Kernel map can bring the data distribution to better elliptical symmetry.

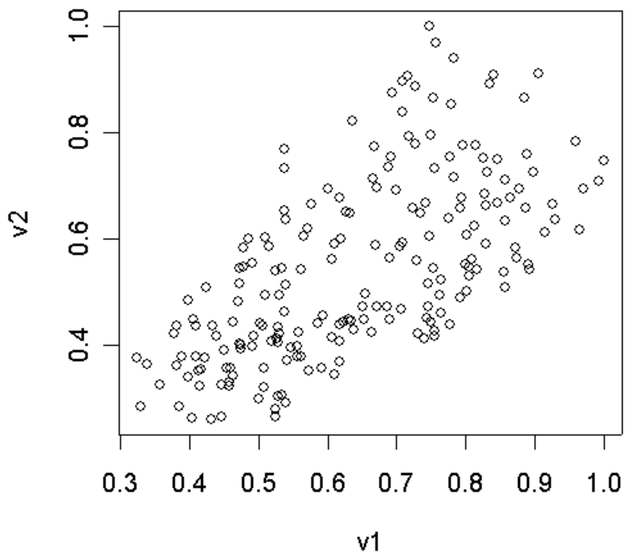
A random sample $\mathbf{X}$ of size 200 consisting of $\{\mathbf{x}_i = (x_{i1}, \cdots, x_{i5}), i = 1, \cdots, 200\}$, where $x_{i1}, x_{i3}, x_{i4}, x_{i5} \sim \text{uniform}(0, 2\pi)$, and $x_{i2} = \sin(x_{i1}) + \epsilon_i$,

$$\epsilon_i \sim N(0, \sigma^2) \text{ with } \sigma = 0.4.$$

- Using Gaussian kernel with scale=0.05.
- The raw data is scaled to have unit variance of each column before transformation

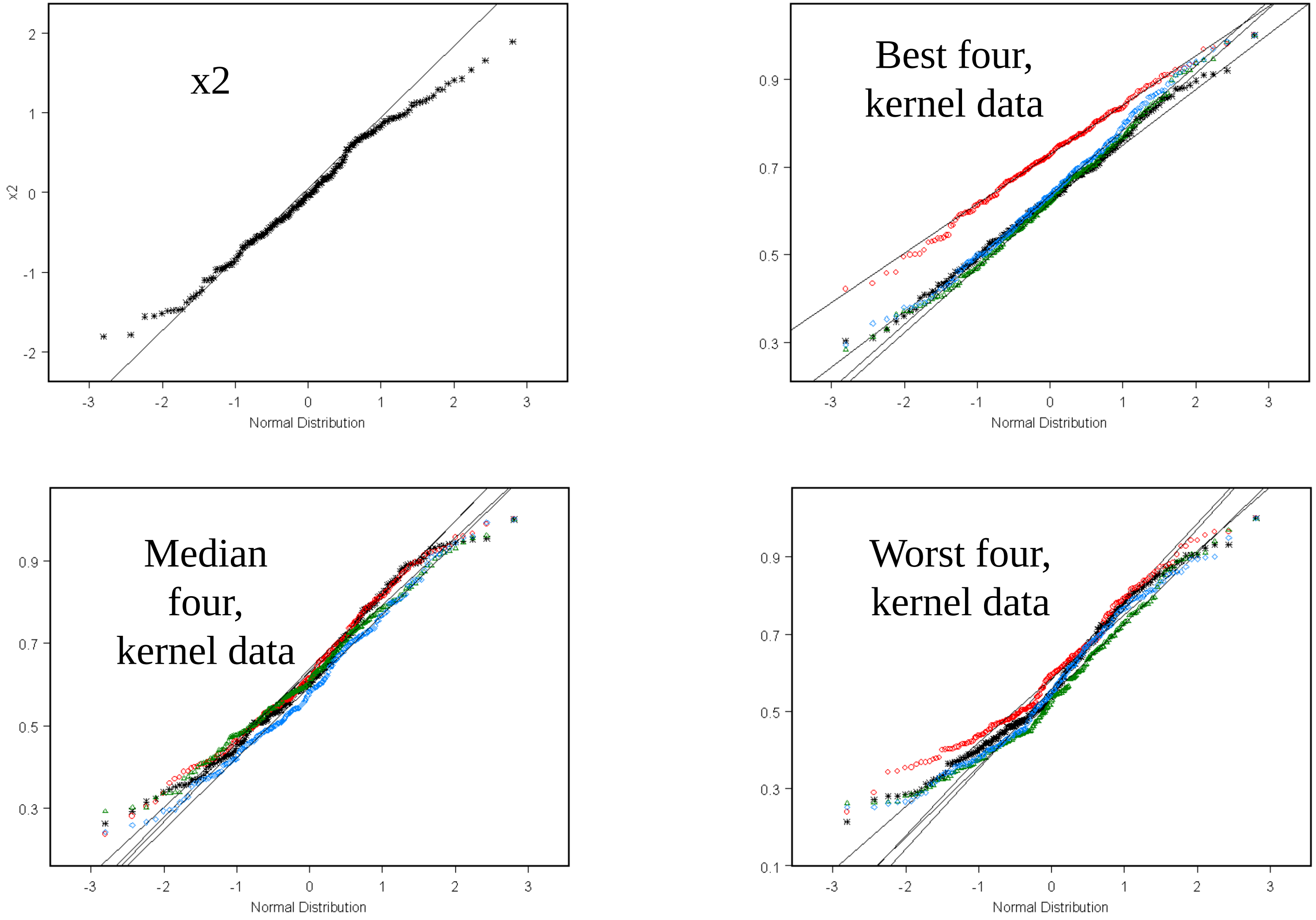


Scatterplot (x1, x2)



Kernel data Scatterplot

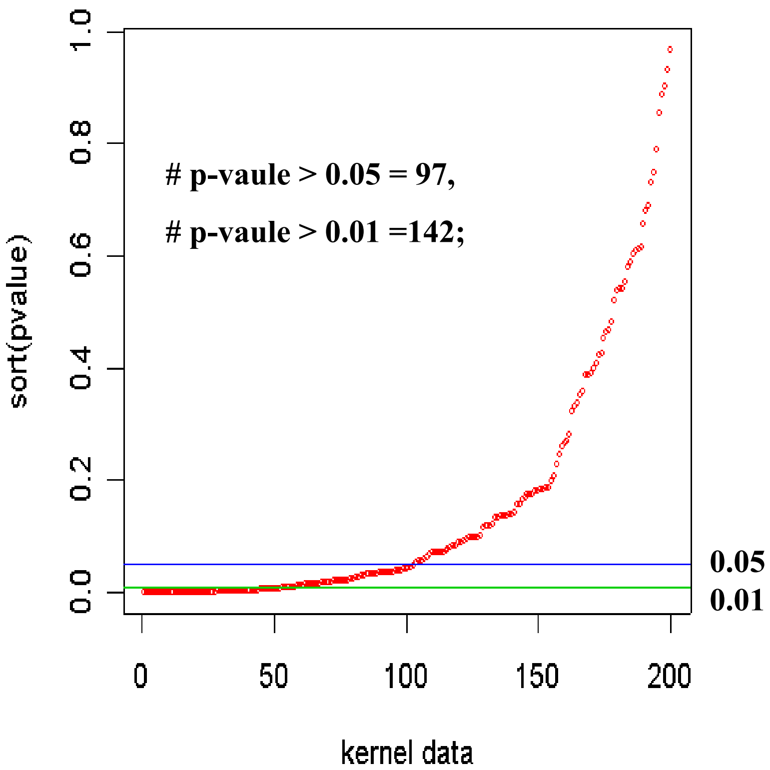
Example: Normal Probability Plot



Example: Justification of Gaussianity

Empirical Justification of Gaussianity:

Kolmogorov-Smirnov Test: H_0: The data follow a normal distribution



Prepare Your Data to Do the Above Empirical Justification

Theoretical Justification of Gaussianity

Kernel data $\{\sqrt{\sigma_n^p}\Gamma_j\}_{j=1}^n$ projected along the random direction h

$$\sqrt{\sigma_n^p}\langle h, \Gamma_1\rangle_{\mathcal{H}_n}, \cdots, \sqrt{\sigma_n^p}\langle h, \Gamma_n\rangle_{\mathcal{H}_n}.$$

Let $\theta_n(h)$ be the empirical distribution of this sequence, assigning probability mass n^{-1} to each $\sqrt{\sigma_n^p}\langle h, \Gamma_j\rangle_{\mathcal{H}_n}$.

Theorem *Under some conditions, as $n \to \infty$, the empirical distribution $\theta_n(h)$ converges weakly to $N(0, \tau^2)$ in probability.*

For details:
Huang, S.Y., Hwang, C. R. and Lin, M.H. Kernel Fisher's Discriminant Analysis in Gaussian Reproducing Kernel Hilbert Space.

PCA in the Euclidean Space

Centered Observations: column vectors $x_i \in \mathbb{R}^N, i = 1, \ldots, m$
(Centered meaning: $\sum_{i=1}^m x_i = 0$)

PCA finds the principal axes by diagonalizing the covariance matrix

$$C = \frac{1}{m}\sum_{j=1}^m x_j x_j^\mathsf{T}$$

Note that C is positive definite, and thus can be diagonalized with nonnegative eigenvalues.

$$\lambda \boldsymbol{v} = C\boldsymbol{v}$$

$$C\boldsymbol{v} = \frac{1}{m}\sum_{j=1}^m x_j x_j^\mathsf{T} \boldsymbol{v} = \lambda \boldsymbol{v}$$

Show that $(\boldsymbol{x}\boldsymbol{x}^T)\boldsymbol{v} = (\boldsymbol{x} \cdot \boldsymbol{v})\boldsymbol{x}$

$$\begin{aligned}\boldsymbol{v} &= \tfrac{1}{m\lambda}\textstyle\sum_{j=1}^m x_j x_j^\mathsf{T} \boldsymbol{v} \\ &= \tfrac{1}{m\lambda}\textstyle\sum_{j=1}^m (x_j \cdot \boldsymbol{v}) x_j\end{aligned}$$

$(x_j \cdot \boldsymbol{v})$ is just a scalar

$$\boldsymbol{v} = \sum_{i=1}^m \alpha_i x_i$$

Kernel PCA

$$\Phi : \mathcal{X} \to \mathcal{H}, \mathrm{x} \mapsto \Phi(\mathrm{x})$$

$$\sum_{k=1}^{m} \Phi(x_k) = 0$$

$$\bar{C} = \frac{1}{M} \sum_{j=1}^{M} \Phi(\mathbf{x}_j)\Phi(\mathbf{x}_j)^\top,$$

$$\lambda \mathbf{V} = C\mathbf{V}$$

$$\lambda(\Phi(\mathbf{x}_k) \cdot \mathbf{V}) = (\Phi(\mathbf{x}_k) \cdot \bar{C}\mathbf{V})$$

$$\mathbf{V} = \sum_{i=1}^{M} \alpha_i \Phi(\mathbf{x}_i).$$

$$\lambda \sum_{i=1}^{M} \alpha_i (\Phi(\mathbf{x}_k) \cdot \Phi(\mathbf{x}_i)) =$$

$$\frac{1}{M} \sum_{i=1}^{M} \alpha_i \left(\Phi(\mathbf{x}_k) \cdot \sum_{j=1}^{M} \Phi(\mathbf{x}_j)\right)(\Phi(\mathbf{x}_j) \cdot \Phi(\mathbf{x}_i))$$

$$K_{ij} := (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)),$$

$$M\lambda K \boldsymbol{\alpha} = K^2 \boldsymbol{\alpha},$$

$$M\lambda \boldsymbol{\alpha} = K\boldsymbol{\alpha}$$

$$(\mathbf{V}^k \cdot \Phi(\mathbf{x})) = \sum_{i=1}^{M} \alpha_i^k (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}))$$

Kernel PCA: kpca {kernlab}

kernlab: Kernel-Based Machine Learning Lab

```
> library(kernlab)
> rbf <- rbfdot(sigma = 0.05) #Radial Basis kernel function
> rbf
Gaussian Radial Basis kernel function.
 Hyperparameter : sigma =  0.05
> KX <- kernelMatrix(kernel=rbf, x=as.matrix(iris[,1:4])) # calculate kernel matrix
> dim(KX)
[1] 150 150
```

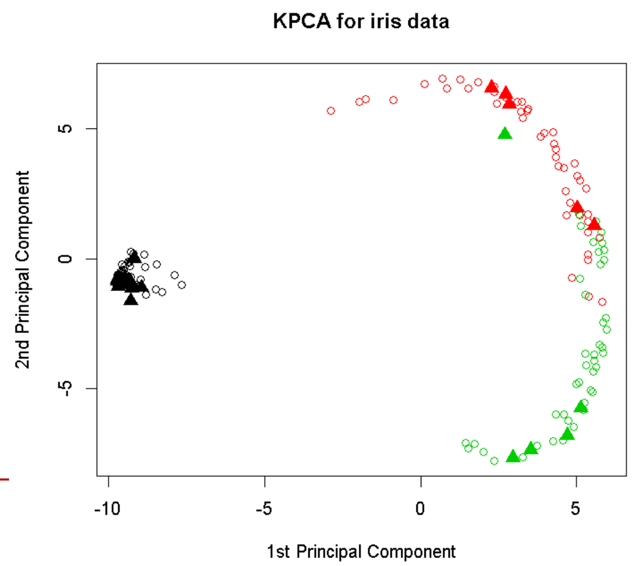
- rbfdot (Radial Basis kernel function)
- polydot (Polynomial kernel function
- vanilladot (Linear kernel function)
- tanhdot (Hyperbolic tangent kernel function)

```
test <- sample(1:150, 20)
iris.kpca <- kpca(~., data=iris[-test, -5], kernel="rbfdot", kpar=list(sigma=0.2),
features=2)

# print the principal component vectors
pcv(iris.kpca)

# plot the data projection on the components
plot(rotated(iris.kpca), col=as.integer(iris[-test, 5]),
     xlab="1st Principal Component",
     ylab="2nd Principal Component",
     main="KPCA for iris data")

# embed remaining points
emb <- predict(iris.kpca, as.matrix(iris[test, -5]))
points(emb, col=iris[test, 5], pch=17, cex=1.5, asp=1)
```



SIR in the Euclidean Space

- Li (1991) introduced the following model

$$y = f(\beta_1'\mathbf{x}, \cdots, \beta_K'\mathbf{x}, \epsilon).$$

Li, K. C. (1991). Sliced inverse regression for dimensional reduction (with discussion). *JASA* **86**, 316-342.

y is a univariate variable.

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

β's are vectors with dimension $p \times 1$.

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

f is an arbitrary function.

➡ The β's are referred to effective dimension reduction ($e.d.r.$) or projection directions.

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

SIR: Algorithm

$$
\begin{array}{c|c}
 & \begin{array}{cccccc} y & X_1 & X_2 & & \cdots & X_p \end{array} \\
\begin{array}{c} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \\ \vdots \\ \\ \mathbf{x}_N \end{array} &
\left[\begin{array}{c|ccccc}
10 & 1 & 4 & & & 2 \\
13 & 3 & 5 & & & 3 \\
9 & 0 & 3 & & & 2 \\
17 & 5 & 5 & & \cdots & 6 \\
\vdots & \vdots & \vdots & & & \vdots \\
12 & 2 & 4 & & & 3 \\
11 & 2 & 5 & & & 2
\end{array}\right]
\end{array}
$$

Sample Mean

$$\bar{\mathbf{x}} = \frac{\sum_{i=1}^{N} \mathbf{x}_i}{N}$$

$$
\begin{array}{c|c|c}
 & \begin{array}{cccccc} y & X_1 & X_2 & & \cdots & X_p \end{array} & \\
\begin{array}{c} \bar{\mathbf{x}}(1) \\ \\ \bar{\mathbf{x}}(2) \\ \\ \bar{\mathbf{x}}(3) \\ \\ \vdots \\ \\ \bar{\mathbf{x}}(N) \end{array} &
\left[\begin{array}{c|ccccc}
9 & 0 & 3 & & & 2 \\
10 & 1 & 4 & & & 2 \\
\hline
11 & 2 & 5 & & & 2 \\
12 & 2 & 4 & & \cdots & 3 \\
\hline
\vdots & \vdots & \vdots & & & \vdots \\
13 & 3 & 5 & & & 3 \\
17 & 5 & 5 & & & 6
\end{array}\right] &
\begin{array}{c} 1 \\ \\ 2 \\ \\ \vdots \\ \\ H \end{array}
\end{array}
$$

Sliced Mean

$$\bar{\mathbf{x}}_h = n_h^{-1} \sum_{(i) \in \mathrm{slice}_h} \mathbf{x}_{(i)}$$

$$
\begin{array}{c|c}
 & \begin{array}{cccc} X_1 & X_2 & \cdots & X_p \end{array} \\
\begin{array}{c} \bar{\mathbf{x}}_1 \\ \bar{\mathbf{x}}_2 \\ \vdots \\ \bar{\mathbf{x}}_H \end{array} &
\left[\begin{array}{cccc}
0.5 & 3.5 & \cdots & 2.0 \\
2.0 & 4.5 & & 2.5 \\
\vdots & \vdots & & \vdots \\
4.0 & 5.0 & & 4.5
\end{array}\right]
\end{array}
$$

$$\hat{\Sigma}_W \hat{\beta}_i = \hat{\lambda}_i \hat{\Sigma}_X \hat{\beta}_i$$

$$\hat{\lambda}_1 \geq \hat{\lambda}_1 \geq \cdots \geq \hat{\lambda}_p$$

$$
\begin{array}{c|c}
 & \begin{array}{cccc} \mathrm{SIR}_1 & \mathrm{SIR}_2 & \cdots & \mathrm{SIR}_K \end{array} \\
\begin{array}{c} y \\ \hline 10 \\ 13 \\ 9 \\ 17 \\ 12 \\ 11 \end{array} &
\left[\begin{array}{cccc}
\hat{\beta}_1 \mathbf{x}_1 & \hat{\beta}_2 \mathbf{x}_1 & \cdots & \hat{\beta}_K \mathbf{x}_1 \\
\hat{\beta}_1 \mathbf{x}_2 & \hat{\beta}_2 \mathbf{x}_2 & \cdots & \hat{\beta}_K \mathbf{x}_2 \\
 & & & \\
 & & & \\
 & & & \\
\hat{\beta}_1 \mathbf{x}_N & \hat{\beta}_2 \mathbf{x}_N & \cdots & \hat{\beta}_K \mathbf{x}_N
\end{array}\right]
\end{array}
$$

Sample Covariance Matrix

$$\hat{\Sigma}_X = \sum_{i=1}^{N} N^{-1} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T$$

Weighted Covariance Matrix for the Slice Means

$$\hat{\Sigma}_W = \sum_{h=1}^{H} \frac{n_h}{N} (\bar{\mathbf{x}}_h - \bar{\mathbf{x}})(\bar{\mathbf{x}}_h - \bar{\mathbf{x}})^T$$

SIR: Theorem

Linear Design Condition (L.D.C.)

For any b in R^p,

the conditional expectation $E(b'\mathbf{x}|\beta_1'\mathbf{x},\cdots,\beta_K'\mathbf{x})$ is linear in $\beta_1'\mathbf{x},\cdots,\beta_K'\mathbf{x}$;

- that is, for some constants $c_0, c_1, \cdots, c_k$,
  $E(b'\mathbf{x}|\beta_1'\mathbf{x},\cdots,\beta_K'\mathbf{x}) = c_0 + c_1\beta_1'\mathbf{x} + \cdots + c_k\beta_K'\mathbf{x}$.

THEOREM:
under regular conditions, the centered inverse regression curve $E[\mathbf{x}|y] - E[\mathbf{x}]$ is contained in the linear subspace spanned by $\beta_k\Sigma_{\mathrm{X}}$ $(k = 1, \cdots, K)$.

COROLLARY 3.1 (Li, 1991)
Assume that $\mathbf{x}$ has been standardized to $\mathbf{z}$. Then under the model and (3.1), the standardized inverse regression curve $E(\mathbf{z}|y)$ is contained in the linear space generated by the standardized *e.d.r.* directions $\theta_1\ \theta_2\ \cdots\ \theta_K$

The SIR directions $\mathbf{v_i}$ falls into the *e.d.r* space.

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

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

- We will show that even if $\mathcal{F}$ has arbitrarily large dimensionality, for certain choices of ϕ, we can still perform SIR in $\mathcal{F}$.

- Assume for the moment that our data mapped into feature space, $\phi(\mathbf{x}_1), \cdots, \phi(\mathbf{x}_n)$, is centered, i.e. $\sum_{i=0}^{n} \phi(\mathbf{x}_i) = 0$.

KSIR: Algorithm

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

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

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

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

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

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

- The equivalent system $\lambda\langle\phi(\mathbf{x}_k), \Sigma_{\mathbf{zz}}\boldsymbol{\beta}\rangle = \langle\phi(\mathbf{x}_k), \Sigma_{\mathbf{wz}}\boldsymbol{\beta}\rangle$, for all $k = 1, \cdots, n$.
- there exists $\alpha_1, \cdots, \alpha_n$ such that $\boldsymbol{\beta} = \sum_{i=1}^{n}\alpha_i\phi(\mathbf{x}_i)$.

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

KSIR (conti.)

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

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

KSIR (conti.)

$$
\begin{aligned}
&\langle \phi(\mathbf{x}_k), \Sigma_{\mathbf{wz}}\boldsymbol{\beta}\rangle \\
= &\ \langle \phi(\mathbf{x}_k), \{\sum_{h=1}^{H} p_h \bar{\phi}(\mathbf{m}_h)\bar{\phi}(\mathbf{m}_h)^T\}\{\sum_{i=1}^{n}\alpha_i\phi(\mathbf{x}_i)\}\rangle \\
= &\ \sum_{i=1}^{n}\alpha_i \langle \phi(\mathbf{x}_k), \sum_{h=1}^{H} p_h\bar{\phi}(\mathbf{m}_h)\rangle\langle\bar{\phi}(\mathbf{m}_h), \phi(\mathbf{x}_i)\rangle \\
= &\ \sum_{i=1}^{n}\alpha_i \sum_{h=1}^{H} \frac{\sum_{j=1}^{n}\mathbf{K}_{kj}\delta_h(y_j)}{n}\frac{\sum_{j=1}^{n}\mathbf{K}_{ji}\delta_h(y_j)}{\sum_{j=1}^{n}\delta_h(y_j)} \\
= &\ \frac{1}{n}\sum_{i=1}^{n}\alpha_i \sum_{h=1}^{H} \frac{\sum_{j=1}^{n}\mathbf{K}_{kj}\delta_h(y_j)}{\sqrt{\sum_{j=1}^{n}\delta_h(y_j)}}\frac{\sum_{j=1}^{n}\mathbf{K}_{ji}\delta_h(y_j)}{\sqrt{\sum_{j=1}^{n}\delta_h(y_j)}}, \quad \forall k = 1,\cdots,n \\
\Rightarrow &\ \frac{1}{n}\mathbf{K}\mathbf{E}_H\mathbf{K}\boldsymbol{\alpha}
\end{aligned}
$$

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

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

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

$$
\Sigma_{\mathbf{wz}}\boldsymbol{\beta} = \lambda\Sigma_{\mathbf{zz}}\boldsymbol{\beta} \;\Rightarrow\; \lambda\,\mathbf{K}\mathbf{K}\boldsymbol{\alpha} = \mathbf{K}\mathbf{E}_H\mathbf{K}\boldsymbol{\alpha} \;\Rightarrow\; \lambda\mathbf{K}\boldsymbol{a} = \mathbf{E}_H\mathbf{K}\boldsymbol{a}
$$

Normalization and Projection

Let $\lambda_1 \geq \cdots \geq \lambda_n$ denote the eigenvalues, and $\boldsymbol{\alpha}_1, \cdots, \boldsymbol{\alpha}_n$ the corresponding complete set of eigenvectors, with λ_t being the first nonzero eigenvalues.

We normalize $\boldsymbol{\alpha}_1, \cdots, \boldsymbol{\alpha}_n$ by requiring that the corresponding vectors in $\mathcal{F}$ be normalized: $\langle \boldsymbol{\beta}_k, \boldsymbol{\beta}_k \rangle = 1$ for all $k = 1, \cdots, t$.

Normalization Condition:

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

Projections on the eigenvectors $\boldsymbol{\beta}_k$ in $\mathcal{F}$, $k = 1, \cdots, t$:

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

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

Centering in Feature Space

The mapped data is centered in $\mathcal{F}$, $\sum_{i=1}^{n} \phi(\mathbf{x}_i) = 0$.

- The points $\tilde{\phi}(\mathbf{x}_i) := \phi(\mathbf{x}_i) - \frac{1}{n}\sum_{i=1}^{n} \phi(\mathbf{x}_i)$ will be centered.
- Define $\tilde{\mathbf{K}} := \langle \tilde{\phi}(\mathbf{x}_i), \tilde{\phi}(\mathbf{x}_i) \rangle$ in $\mathcal{F}$.

$$\boxed{\tilde{\mathbf{K}} = \mathbf{K} - I_n\mathbf{K} - \mathbf{K}I_n + I_n\mathbf{K}I_n, \ (I_n)_{ij} = 1/n.}$$

For Training Data

$$K_{tr} \leftarrow \mathrm{kernelMatrix}(poly, \boldsymbol{X}_{tr})$$

$$K_{tr.c} \leftarrow K_{tr} - \mathbf{1}_{tr}K_{tr} - K_{tr}\mathbf{1}_{tr} + \mathbf{1}_{tr}K_{tr}\mathbf{1}_{tr}$$

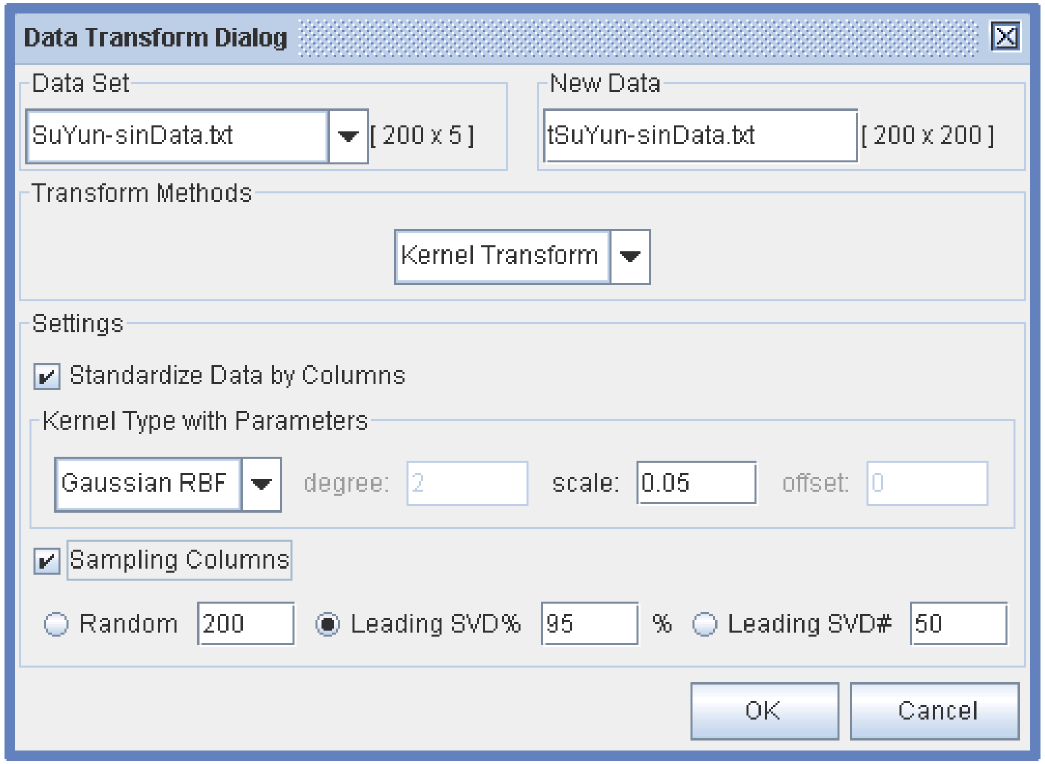
For Testing Data

$$K_{te} \leftarrow \mathrm{kernelMatrix}(poly, \boldsymbol{X}_{te}, \boldsymbol{X}_{tr})$$

$$K_{te.c} \leftarrow K_{te} - \mathbf{1}_{te}K_{tr} - K_{te}\mathbf{1}_{tr} + \mathbf{1}_{te}K_{tr}\mathbf{1}_{tr}$$

Reduced Features

- we are not working in the **full feature space**, but just in a comparably small linear subspace of it, whose dimension equals at most the number of observations.
- Working in a space whose dimension equals the number of observations can pose difficulties.
- To deal with these, one can either use only a subset of the extracted features, or use some other form of capacity control or regularization.



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

Relations Towards Other Methods

SIR vs. KSIR

- KSIR generalizes SIR to a nonlinear one by kernelization of the SIR algorithm.
- It finds nonlinear d.r. subspace, a central d.r. subspace in H_k
- A semiparametric method.
- **SIR**: spectrum analysis of cov(E[x|y]) wrt cov(x)
- **KSIR**: spectrum analysis of a generalized association measure.

KSIR vs. KPCA

PCA
eigenvalue problem

$$\lambda \boldsymbol{v} = C\boldsymbol{v}$$

covariance matrix

$$C = \frac{1}{m}\sum_{j=1}^{m} x_j x_j^\top$$

kernel PCA
eigenvalue problem

$$\boldsymbol{K\alpha} = \lambda\boldsymbol{\alpha}$$

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

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

Relations Towards Other Methods

KSIR vs. KFDA

$$\max_a \frac{\boldsymbol{a}^t \Sigma_B \boldsymbol{a}}{\boldsymbol{a}^t \Sigma_W \boldsymbol{a}} \Rightarrow \Sigma_B \boldsymbol{a} = \gamma_i \Sigma_W \boldsymbol{a}, \quad \gamma_1 \geq \gamma_2 \geq \cdots \geq \gamma_p$$

$$\Rightarrow \Sigma_{\mathbf{xx}} = \Sigma_B + \Sigma_W \Rightarrow \Sigma_B \boldsymbol{a}_i = \frac{\gamma_i}{1+\gamma_i} \Sigma_{\mathbf{xx}} \boldsymbol{a}_i$$

$$\Sigma_{\mathbf{wx}} \boldsymbol{\beta}_j = \lambda_j \Sigma_{\mathbf{xx}} \boldsymbol{\beta}_j$$

$$\Rightarrow \lambda_i = \gamma/(1+\gamma) \text{ and } \boldsymbol{a}_i \propto \boldsymbol{\beta}_i,$$

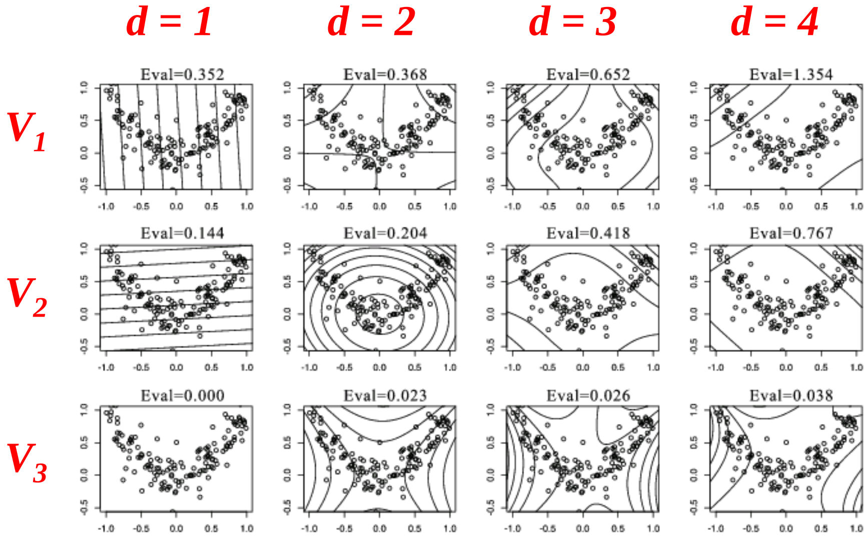
Chen, C. H., and Li, K. C. (2001)

KSIR vs. KCCA

Kernel Fisher discriminant Analysis as special case of CCA.
(Kuss, M. and Graepel, T: The Geometry Of Kernel Canonical Correlation Analysis. (108), Max Planck Institute for Biological Cybernetics, Tübingen, Germany (May 2003)

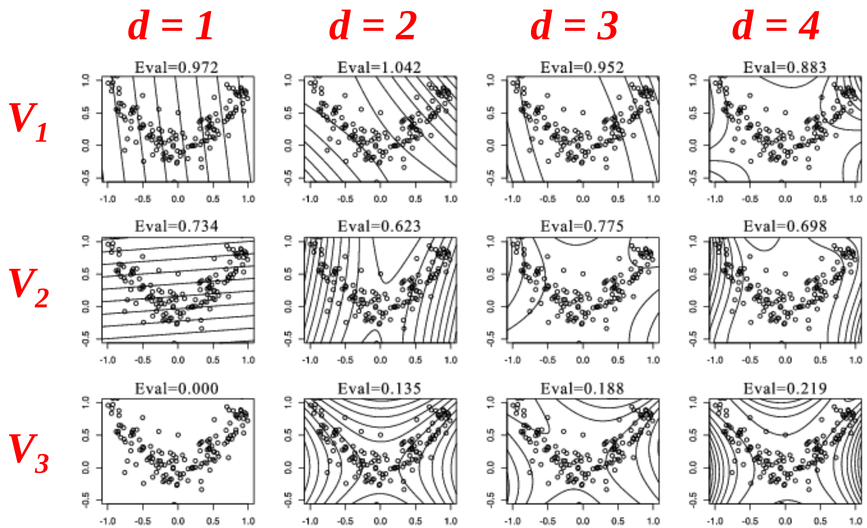
Visualization: Square Data (150x2)

KPCA

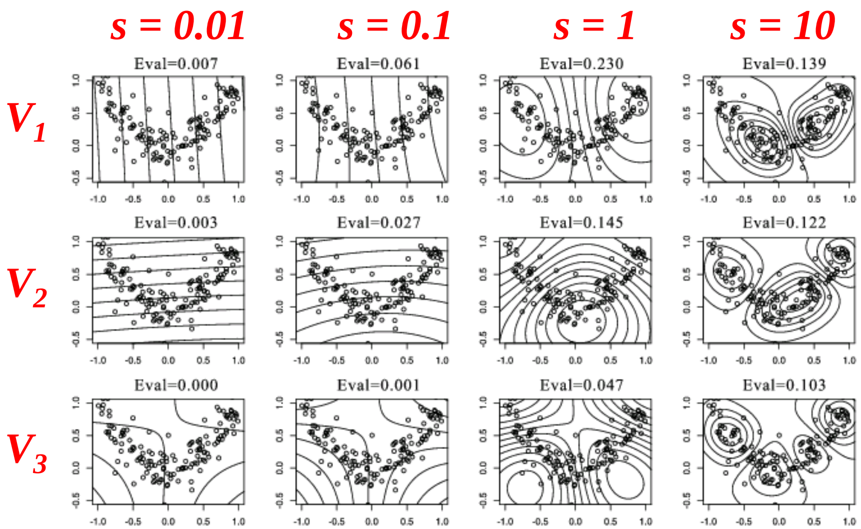


KSIR

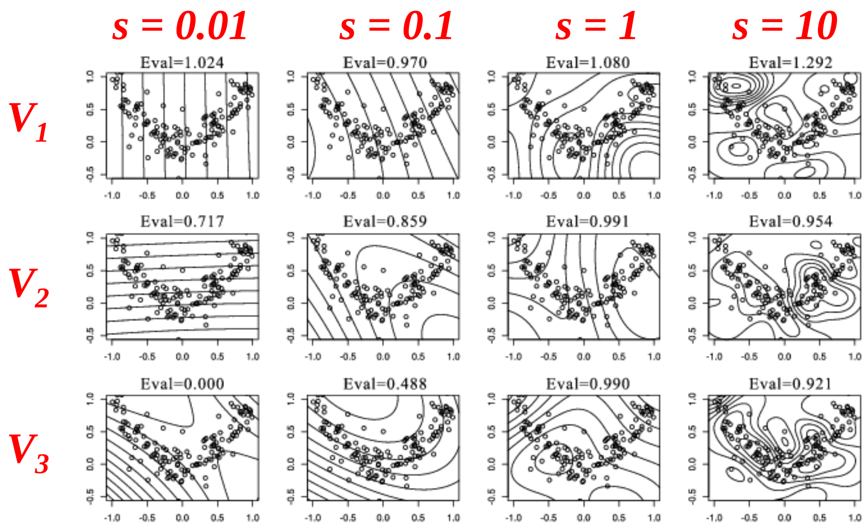
H=8



KPCA



KSIR



Visualization: Three Clusters Data (220x2)

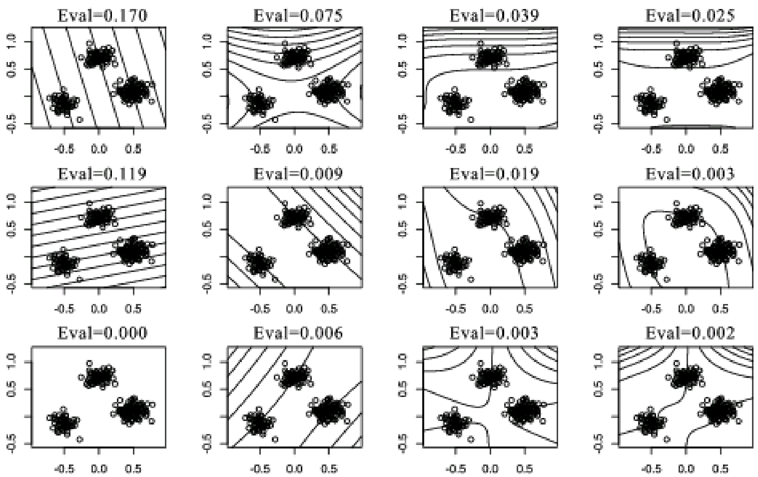
KPCA

d = 1 *d* = 2 *d* = 3 *d* = 4

V_1

V_2

V_3



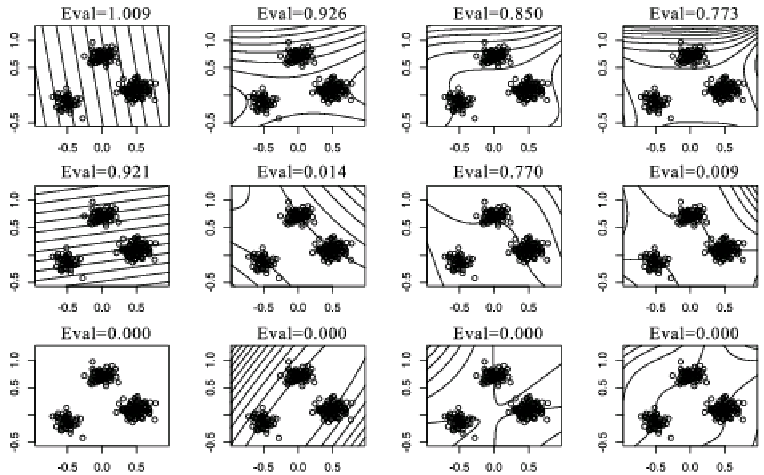
KSIR

d = 1 *d* = 2 *d* = 3 *d* = 4

V_1

V_2

V_3



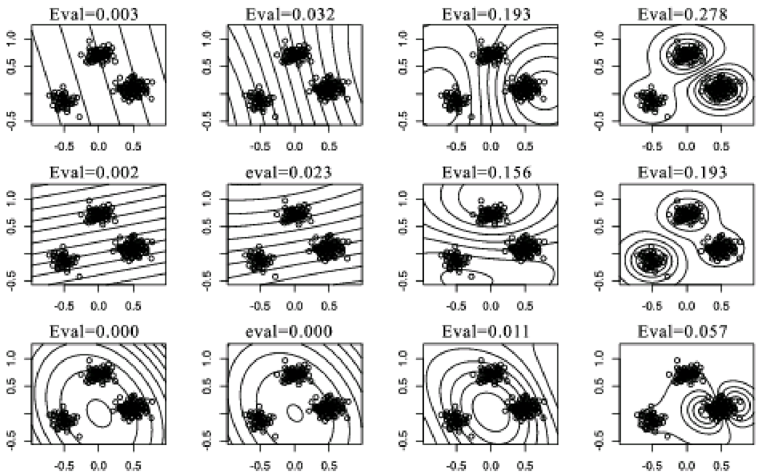
KPCA

s = 0.01 *s* = 0.1 *s* = 1 *s* = 10

V_1

V_2

V_3



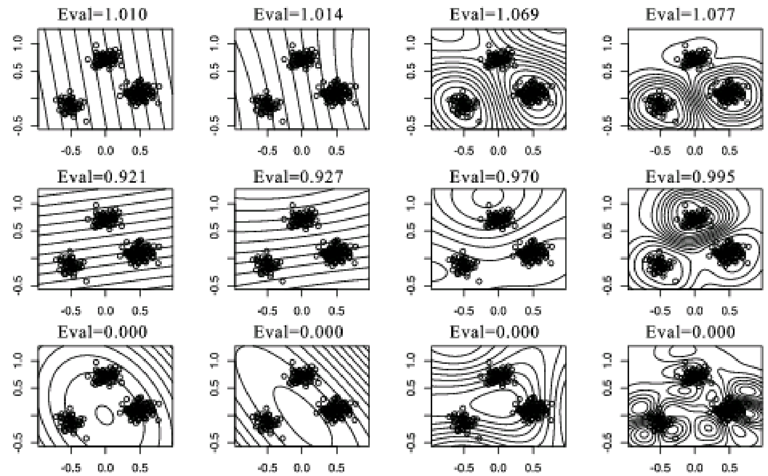
KSIR

s = 0.01 *s* = 0.1 *s* = 1 *s* = 10

V_1

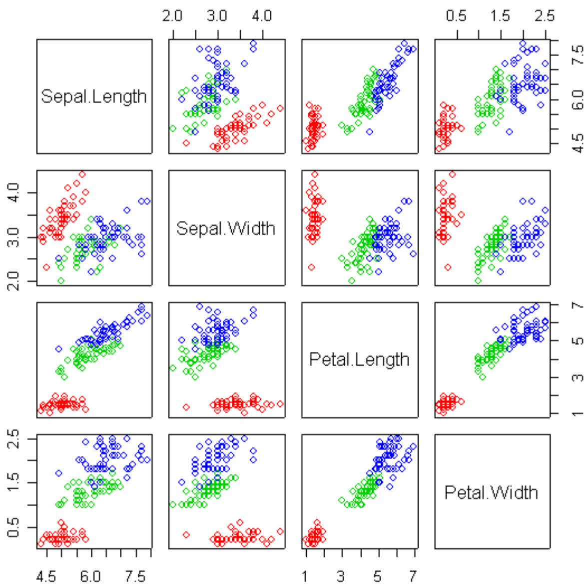
V_2

V_3

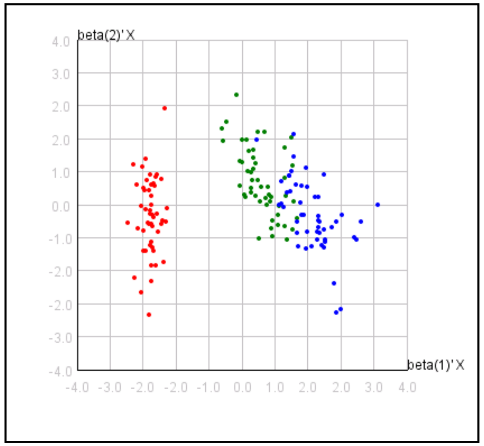


Visualization: Iris Data (150x4)

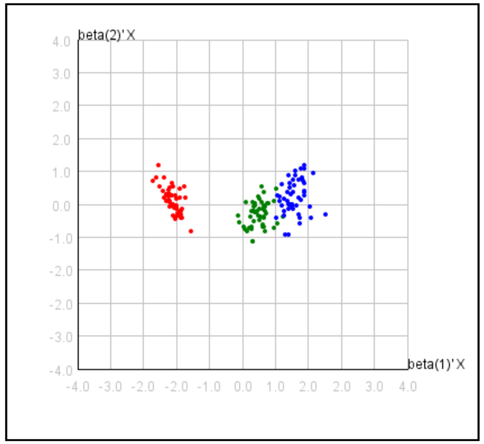
- The sepal length, sepal width, petal length, and petal width are measured in centimeters on 50 iris specimens from each of three species, *Iris setosa*, *I. versicolor*, and *I. virginica*. Fisher (1936)



PCA

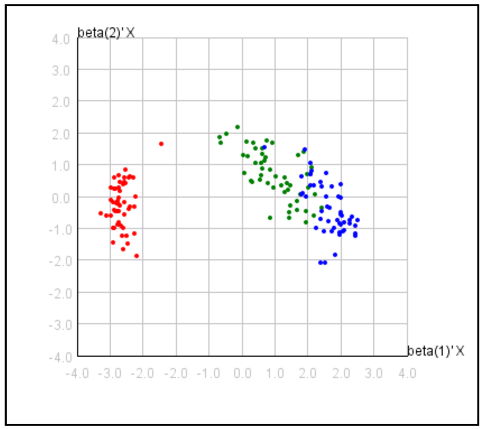


SIR

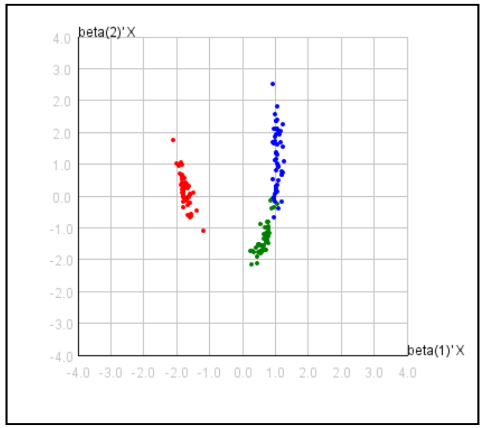


Gaussian s=0.05

KPCA



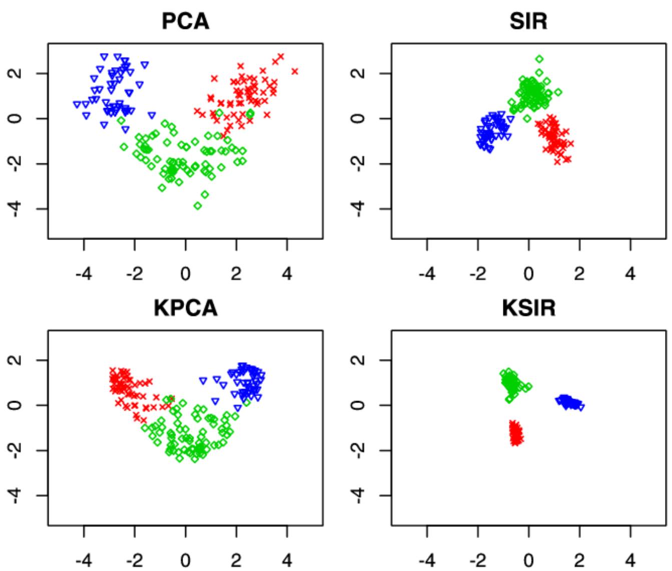
KSIR



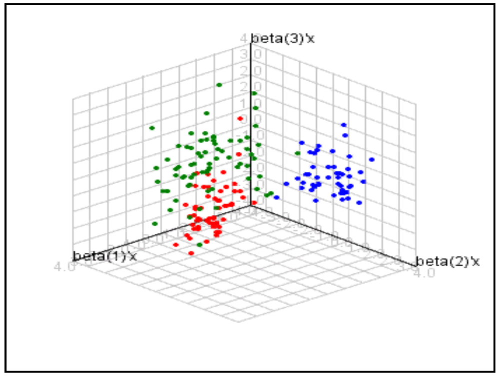
Visualization: Wine Data (178x18)

- 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.
- Past Usage

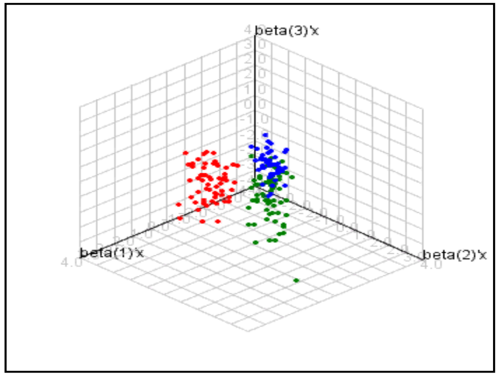
RDA : 100%, QDA 99.4%,
LDA 98.9%, 1NN 96.1%
(z-transformed data, loo)



PCA

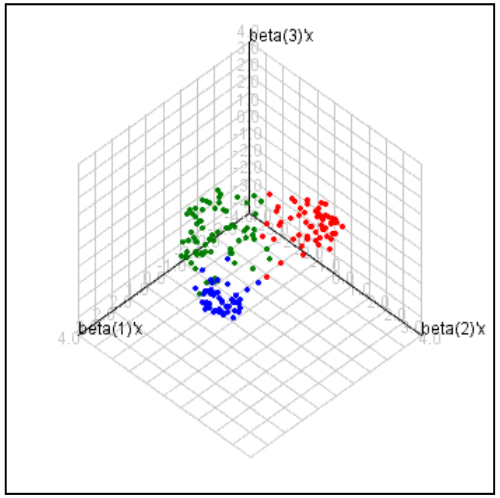


SIR

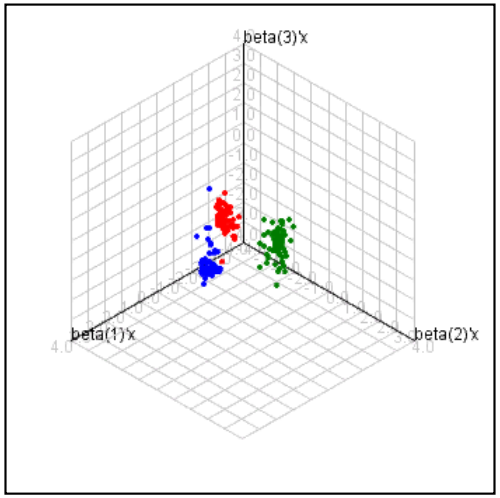


KPCA

Gaussian s=0.05



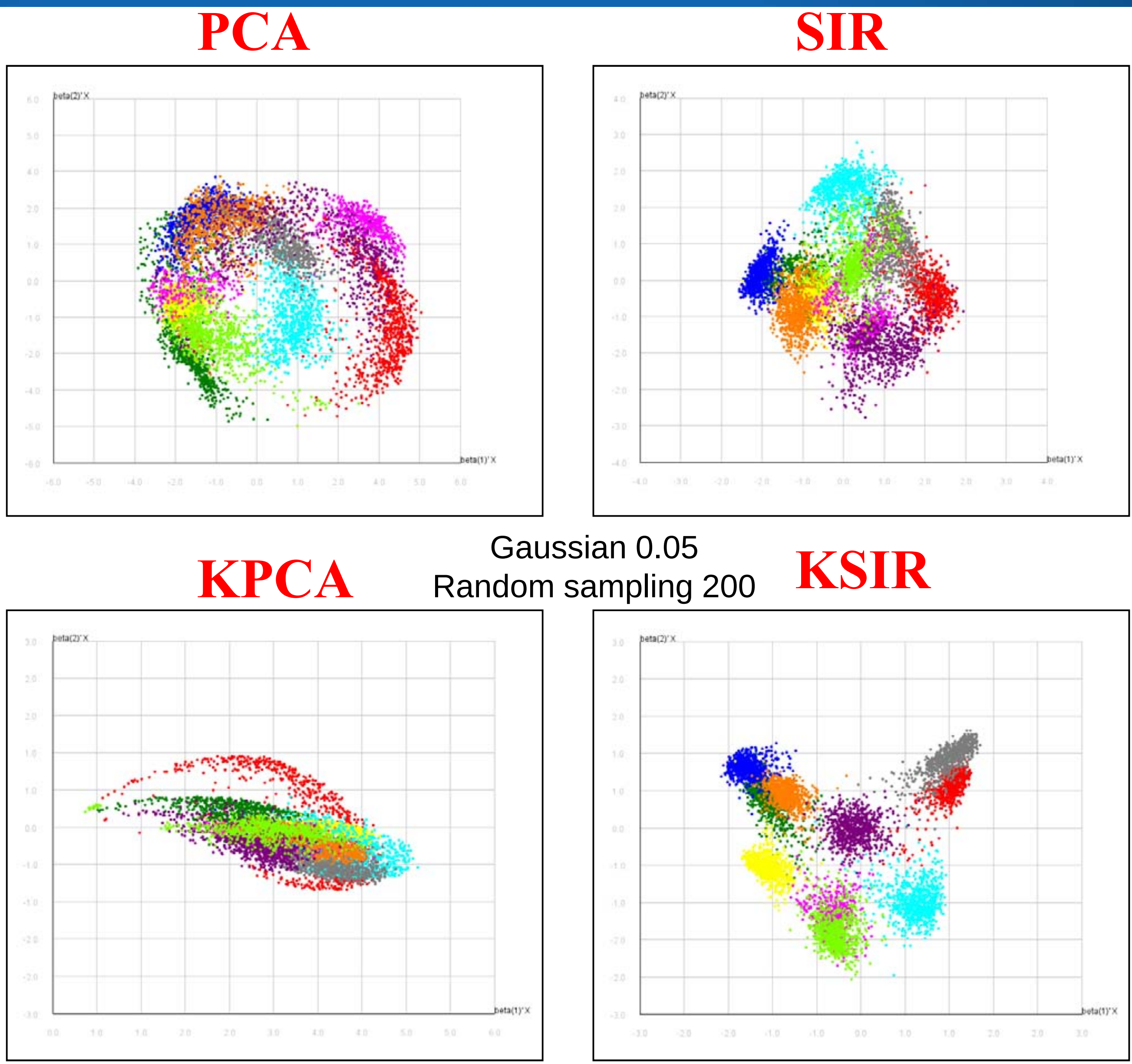
KSIR



Visualization: Pendigit Data (7494x16)

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

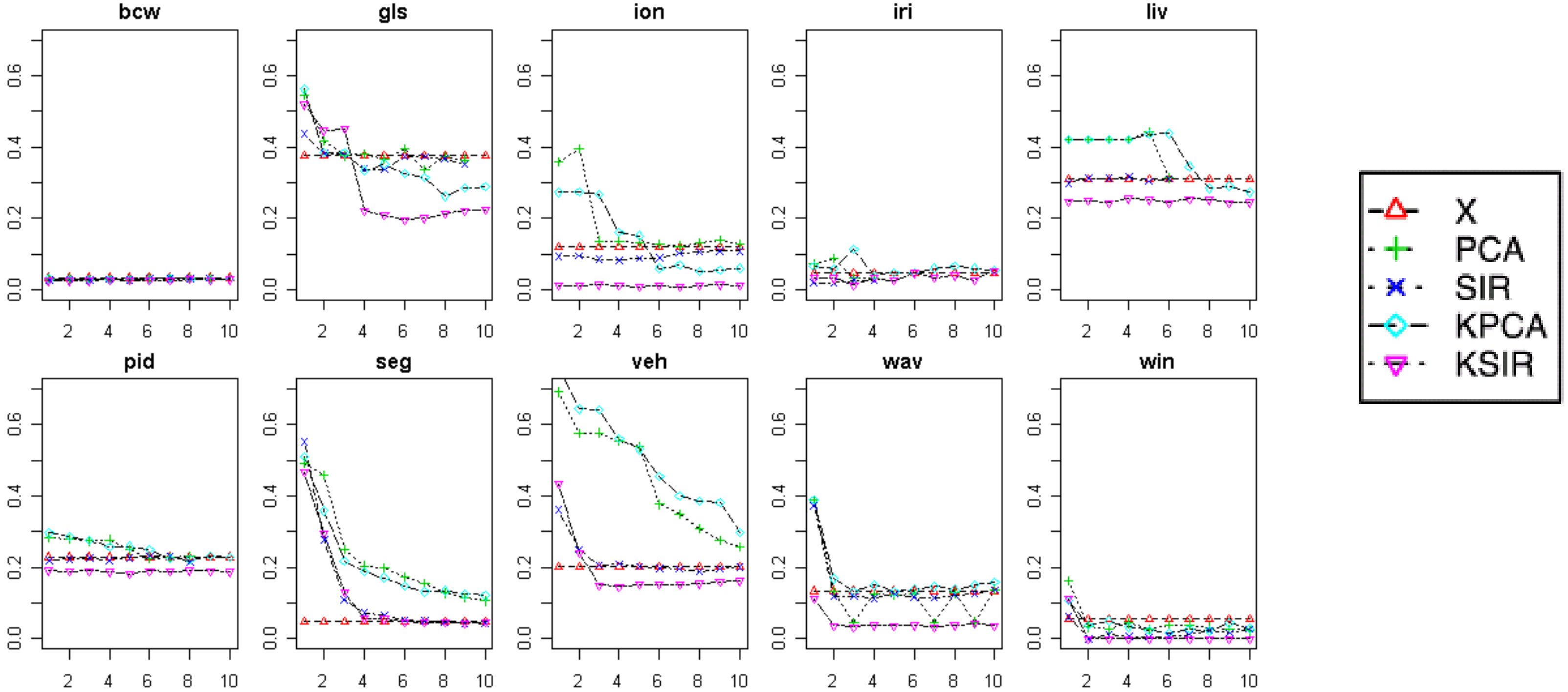
0	: 780
1	: 779
2	: 780
3	: 719
4	: 780
5	: 720
6	: 778
7	: 719
8	: 719
9	: 719



Classification: 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)

Gaussian 0.05
Random sampling 200



Classification: Microarray Data Sets

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