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.

  - 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 $X_{n \times p} = \{x_i, i = 1, \cdots, n\}, x_i \in \mathbb{R}^p$.

Kernel transformation: $x_i \rightarrow \phi(x_i) := k(x_i, \cdot)$.

Kernel Data: $\{\phi(x_i), i = 1, \cdots, n\}, \phi(\cdot) \in \mathcal{H}_k$.

Kernel Data $K_{n \times n} = \{k_{ij} : k(x_i, x_j), i, j = 1, \cdots, n\}$.

- 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 \( k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R} \) is called a positive definite kernel \( \text{iff} \) it is symmetric, that is, \( k(x_i, x_j) = k(x_j, x_i) \) for any two objects \( x_i, x_j \) in \( \mathcal{X} \), and positive definite, that is, \( \sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j k(x_i, x_j) \geq 0 \) for any \( n > 0 \), any choice of \( n \) objects \( x_1, \ldots, x_n \) in \( \mathcal{X} \), and any choice of real numbers \( c_1, \ldots, c_n \) in \( \mathbb{R} \).
Kernel as Inner Product

Represent objects \( x \in X \) as a vector \( \phi(x) \in \mathbb{R}^p \), defining a kernel for any \( x_i, x_j \in X \) by \( k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle \).

(Aronszajn, 1950)

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

Kernels can all be thought of as dot products in feature space \( F \).

The point \( x \in X \) are viewed as point \( \phi(x) \) in \( F \).
The kernel trick transforms any algorithm that solely depends 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.
SIR in the Euclidean Space

K.C. Li, (1991),

\[ y = f(\beta'_1 x, \ldots, \beta'_K x, \epsilon) \]

\( y \) is a univariate variable. \( f \) is an arbitrary function.
\( \epsilon \) is a random variable independent of \( x \).

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

**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

**Sample Mean**

\[
\bar{x} = \frac{\sum_{i=1}^{N} x_i}{N}
\]

**Sliced Mean**

\[
x_h = n^{-1} \sum_{(i) \in \text{slice}_h} x_{(i)}
\]

**Sample Covariance Matrix**

\[
\hat{\Sigma}_X = \sum_{i=1}^{N} N^{-1}(x_i - \bar{x})(x_i - \bar{x})^T
\]

**Weighted Covariance Matrix for the Sliced Means**

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

**Weighted PCA**

\[
\hat{\Sigma}_W \beta_i = \hat{\lambda}_i \hat{\Sigma}_X \beta_i \quad \hat{\lambda}_1 \geq \hat{\lambda}_2 \geq \cdots \geq \hat{\lambda}_p
\]

**Kernel SIR, Han-Ming Wu**
Kernel SIR in a Non-linear Feature Space

Kernel SIR: Kernelize the SIR algorithm

- first map the data nonlinearity into a feature space \( \mathcal{F} \) by

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

- 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, \( \phi(x_1), \ldots, \phi(x_n) \), is centered, i.e. \( \sum_{i=0}^n \phi(x_i) = 0 \).
KSIR: Algorithm (1)

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

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

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

$$
\Sigma_{wz} = \sum_{h=1}^{H} p_h \tilde{\phi}(m_h)\tilde{\phi}(m_h)^T.
$$

$$
\tilde{\phi}(m_h) = \frac{1}{np_h} \sum_{i=1}^{n} \phi(x_i)\delta_h(y_i)
$$

All solutions $\beta$ lie in span $\{\phi(x_1), \cdots, \phi(x_n)\}$.

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

Define $K := \{k_{ij} = \langle \phi(x_i), \phi(x_j) \rangle\}_{n \times n}$. 
The equivalent system \( \lambda \langle \phi(x_k), \Sigma_{zz}\beta \rangle = \langle \phi(x_k), \Sigma_{wz}\beta \rangle \), for all \( k = 1, \cdots, n \).

\[
\lambda \langle \phi(x_k), \Sigma_{zz}\beta \rangle = \lambda \langle \phi(x_k), \left\{ \frac{1}{n} \sum_{j=1}^{n} \phi(x_j)\phi(x_j)^T \right\} \{ \sum_{i=1}^{n} \alpha_i \phi(x_i) \} \rangle \\
= \lambda \frac{1}{n} \sum_{i=1}^{n} \alpha_i \langle \phi(x_k), \sum_{j=1}^{n} \phi(x_j) \rangle \langle \phi(x_j), \phi(x_i) \rangle \\
= \lambda \frac{1}{n} \sum_{i=1}^{n} \alpha_i \sum_{j=1}^{n} K_{kj} K_{ji}, \ \forall k = 1, \cdots, n \\
\Rightarrow \lambda \frac{1}{n} K K^T \alpha
\]
\[ \langle \phi(x_k), \Sigma_{wz} \beta \rangle \]

\[ = \langle \phi(x_k), \left\{ \sum_{h=1}^{H} p_h \bar{\phi}(m_h) \bar{\phi}(m_h)^T \right\} \{ \sum_{i=1}^{n} \alpha_i \phi(x_i) \} \rangle \]

\[ = \sum_{i=1}^{n} \alpha_i \langle \phi(x_k), \sum_{h=1}^{H} p_h \bar{\phi}(m_h) \rangle \langle \bar{\phi}(m_h), \phi(x_i) \rangle \]

\[ = \sum_{i=1}^{n} \alpha_i \sum_{h=1}^{H} \frac{\sum_{j=1}^{n} K_{kj} \delta_h(y_j)}{n} \frac{\sum_{j=1}^{n} K_{ji} \delta_h(y_j)}{\sum_{j=1}^{n} \delta_h(y_j)} \]

\[ = \frac{1}{n} \sum_{i=1}^{n} \alpha_i \sum_{h=1}^{H} \frac{\sum_{j=1}^{n} K_{kj} \delta_h(y_j)}{\sqrt{\sum_{j=1}^{n} \delta_h(y_j)}} \frac{\sum_{j=1}^{n} K_{ji} \delta_h(y_j)}{\sqrt{\sum_{j=1}^{n} \delta_h(y_j)}}, \ \forall k = 1, \cdots, n \]

\[ \Rightarrow \frac{1}{n} KE_H K \alpha \]

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

\[ \Sigma_{wz} \beta = \lambda \Sigma_{zz} \beta \quad \Rightarrow \quad \lambda KK \alpha = KE_H K \alpha \quad \Rightarrow \quad \lambda Ka = E_H Ka \]
Normalization and Projection

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

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

**Normalization Condition:**

$$1 = \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i^k \alpha_j^k \langle \phi(x_i), \phi(x_j) \rangle = \langle \alpha^k, K \alpha^k \rangle = \lambda_k \langle \alpha^k, \alpha^k \rangle$$

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

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

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

For Theoretical details:
http://dmlab1.csie.ntust.edu.tw/downloads
KSIR for Nonlinear Dimensional Reduction and Data Visualization

\[ y = f(\beta_1^t \mathbf{x}, \ldots, \beta_B^t \mathbf{x}, \epsilon) \]

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

\[ y = f(\beta_1^t \Phi(\mathbf{x}), \ldots, \beta_B^t \Phi(\mathbf{x}), \epsilon), \text{ where } \beta_b \in \mathbb{R}^d, \ d \leq \infty. \]

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

$$x_2 = x_1^2 + \xi$$

$$x_1 \sim \text{uniformly } [-1, 1]$$

$$\xi \sim \text{normally } \sigma = 0.2$$

$$n = 150.$$

KPCA

$$s = 0.01 \quad s = 0.1 \quad s = 0.1$$

KSIR

$$s = 0.1 \quad s = 1 \quad s = 10$$
Visualization (2): Three Clusters Data

KPCA
s = 0.01  s = 0.1  s = 1
V1  Eval=0.003  Eval=0.032  Eval=0.193
  Eval=0.002  Eval=0.023  Eval=0.156
  Eval=0.002  Eval=0.011  Eval=0.067
V2  Eval=0.002  Eval=0.033  Eval=0.196
  Eval=0.002  Eval=0.023  Eval=0.156
  Eval=0.002  Eval=0.011  Eval=0.067
V3  Eval=0.002  Eval=0.033  Eval=0.196
  Eval=0.002  Eval=0.023  Eval=0.156
  Eval=0.002  Eval=0.011  Eval=0.067

KSIR
s = 0.01  s = 0.1  s = 1  s = 10
V1  Eval=1.019  Eval=1.014  Eval=1.069  Eval=1.077
  Eval=0.977  Eval=0.967  Eval=0.969  Eval=0.959
  Eval=0.977  Eval=0.967  Eval=0.969  Eval=0.959
V2  Eval=0.977  Eval=0.967  Eval=0.969  Eval=0.959
  Eval=0.977  Eval=0.967  Eval=0.969  Eval=0.959
  Eval=0.977  Eval=0.967  Eval=0.969  Eval=0.959
V3  Eval=0.977  Eval=0.967  Eval=0.969  Eval=0.959
  Eval=0.977  Eval=0.967  Eval=0.969  Eval=0.959
  Eval=0.977  Eval=0.967  Eval=0.969  Eval=0.959
$y = \frac{x_1}{0.5 + (x_2 + 1.5)^2} + \sigma \cdot \epsilon$

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

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

$n = 400$. 

PCA

SIR $H=13$

Orig

KPCA Gaussian $s=0.05$

KSIR
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.
Pen-based recognition of handwritten Digits

7494 instances, 16 attributes

10 classes

- PCA
- SIR
- KPCA
- KSIR

Gaussian 0.05
Random sampling 200
Classification (1): UCI Data Sets

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

Gaussian 0.05
Random sampling 200

Linear Support Vector Machine

10-fold classification error rates
10-fold Classification ERs: UCI Data Sets

bcw, gls, ion, iri, liv, pid, seg, veh, wav, win

- X
- PCA
- SIR
- KPCA
- KSIR
Classification: Microarray Data Sets

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

Linear Support Vector Machine

Leave-one-out classification error rates

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         

Kernel SIR, Han-Ming Wu
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).
Thank You!

http://www.hmwu.idv.tw

Kernel SIR, Han-Ming Wu