CCA with Its Applications
(Part II)

Chen Songcan (陈松灿)
s.chen@nuaa.edu.cn
http://parnec.nuaa.edu.cn

Nanjing University of Aeronautics & Astronautics
MLA’06, Nanjing
Given $X$ set $\{x_1, \ldots, x_n\} \in \mathbb{R}^p$, and $Y$ set $\{y_1, \ldots, y_n\} \in \mathbb{R}^q$, CCA aims to simultaneously seek $w_x \in \mathbb{R}^p$, $w_y \in \mathbb{R}^q$, to ensure [Bor99] (specially use for feature extraction).
CCA formulation

\[
\max_{w_x, w_y} \frac{\text{cov}(w_x^T x, w_y^T y)}{\sqrt{\text{var}(w_x^T x) \text{var}(w_y^T y)}}
\]

\[
= \max_{w_x, w_y} \frac{\sum_{i=1}^{n} w_x^T x_i y_i^T w_y}{\sqrt{\sum_{i=1}^{n} w_x^T x_i x_i^T w_x} \sqrt{\sum_{i=1}^{n} w_y^T y_i y_i^T w_y}}
\]

\[
\Rightarrow \begin{pmatrix}
XX^T \\
YX^T
\end{pmatrix}
\begin{pmatrix}
w_x \\
w_y
\end{pmatrix} = \lambda \begin{pmatrix}
XY^T \\
YX^T
\end{pmatrix}
\begin{pmatrix}
w_x \\
w_y
\end{pmatrix}
\]

where \( X = [x_1, ..., x_n], Y = [y_1, ..., y_n] \)
CCA Alternative formulation

\[
\begin{align*}
\min_{w_x, w_y} & \sum_{i=1}^{n} \left\| w_x^T (x_i - \bar{x}) - w_y^T (y_i - \bar{y}) \right\|^2 \\
\text{s.t.} & \quad \sum_{i=1}^{n} \left\| w_x^T (x_i - \bar{x}) \right\|^2 = 1, \quad \sum_{i=1}^{n} \left\| w_y^T (y_i - \bar{y}) \right\|^2 = 1
\end{align*}
\]

\[
\begin{align*}
\max w_x^T \cdot \sum_{i=1}^{n} \sum_{j=1}^{n} (x_i - x_j)(y_i - y_j)^T \cdot w_y \\
\text{s.t.} & \quad w_x^T \cdot \sum_{i=1}^{n} \sum_{j=1}^{n} (x_i - x_j)(x_i - x_j)^T \cdot w_x = 1 \\
& \quad w_y^T \cdot \sum_{i=1}^{n} \sum_{j=1}^{n} (y_i - y_j)(y_i - y_j)^T \cdot w_y = 1
\end{align*}
\]

Basis of our LPCCA
Applications

- Image retrieval [HSS04], image segmentation [LGD05], image de-noising [He04], image analysis [Nie02], feature detection of image [And00]
- Text categorization, text mining, text retrieval, image-text retrieval, machine translation [For04]
- Computer vision [KSE05]
- Pattern recognition [SZL05]
- Biometrics [YVN03]
- Signal processing [SS06]
- Location estimation in wireless sensor networks [PKY06]
- Facial expression recognition [ZZZ06]
- Multi-view SVM design [FHM05]
New Applications

- location estimation in wireless sensor networks

New Applications

- facial expression recognition (1:happiness, sadness, surprise, angry, disgust, 6:fear)

Landmark points vs. Semantic ratings (estimation) (JAFFE database)

New Applications

- **Multi-view SVM design**

\[
\begin{align*}
\min L & = \frac{1}{2} \| w_A \|^2 + \frac{1}{2} \| w_B \|^2 + C^A \sum_{i=1}^{\ell} \xi_i^A + C^B \sum_{i=1}^{\ell} \xi_i^B + D \sum_{i=1}^{\ell} \eta_i \\
\text{such that} & \quad |\langle w_A, \phi_A(x_i) \rangle + b_A - \langle w_B, \phi_B(x_i) \rangle - b_B| \leq \eta_i + \epsilon \\
& \quad y_i (\langle w_A, \phi_A(x_i) \rangle + b_A) \geq 1 - \xi_i^A \\
& \quad y_i (\langle w_B, \phi_B(x_i) \rangle + b_B) \geq 1 - \xi_i^B \\
& \quad \xi_i^A \geq 0, \quad \xi_i^B \geq 0, \quad \eta_i \geq 0 \quad \text{all for} \quad 1 \leq i \leq \ell.
\end{align*}
\]

CCA alternative formulation

<table>
<thead>
<tr>
<th></th>
<th>Motorbike</th>
<th>Bicycle</th>
<th>People</th>
<th>Car</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM 1</td>
<td>94.05</td>
<td>91.58</td>
<td>91.58</td>
<td>87.95</td>
</tr>
<tr>
<td>SVM 2</td>
<td>91.15</td>
<td>91.15</td>
<td>90.57</td>
<td>86.21</td>
</tr>
<tr>
<td>KCCA + SVM</td>
<td>94.19</td>
<td>90.28</td>
<td>90.57</td>
<td>88.68</td>
</tr>
<tr>
<td>SVM 2K</td>
<td>94.34</td>
<td>93.47</td>
<td>92.74</td>
<td>90.13</td>
</tr>
</tbody>
</table>

Our previous work

- Locality preserving CCA (LPCCA)
- Kernelized LPCCA
- Their applications in pose estimation and data visualization
Our previous work--LPCCA

- **Motive:** CCA is linear, and is insufficient to reveal the nonlinear correlation between two sets X and Y of real-world variables.

- **Objective:**

  \[
  \max_{w_x, w_y} w_x^T \cdot \sum_{i=1}^{n} \sum_{j=1}^{n} S^x_{ij} (x_i - x_j) S^y_{ij} (y_i - y_j)^T \cdot w_y
  \]

  \[
  \text{s.t. } w_x^T \cdot \sum_{i=1}^{n} \sum_{j=1}^{n} S^{x2}_{ij} (x_i - x_j)(x_i - x_j)^T \cdot w_x = 1
  \]

  \[
  w_y^T \cdot \sum_{i=1}^{n} \sum_{j=1}^{n} S^{y2}_{ij} (y_i - y_j)(y_i - y_j)^T \cdot w_y = 1
  \]

  \[
  \Rightarrow \begin{pmatrix}
  XS_{xy} Y^T \\
  YS_{yx} X^T
  \end{pmatrix}
  \begin{pmatrix}
  w_x \\
  w_y
  \end{pmatrix}
  = \lambda_{LP}
  \begin{pmatrix}
  XS_{xx} X^T \\
  YS_{yy} Y^T
  \end{pmatrix}
  \begin{pmatrix}
  w_x \\
  w_y
  \end{pmatrix}
  \]

Our previous work

CCA                     KCCA               LPCCA

Error distribution histograms

CCA                  KCCA               LPCCA

KLPCCA, TNN in revision

Error distribution histograms
Our new work

- CIPCA--Class-information-incorporated PCA
- DCCA-- Discriminant CCA
- DCCAM– DCCA with missing samples
- Multi-Kernel learning
- Beyond KCCA
CIPCA

- **Motive**
  1. utilize the class information for feature extraction
  2. CIPCA and CCA can be unified into a framework

\[
\max_w \frac{w^T C_{z z} w}{w^T D w}, \quad \text{where} \quad D = \begin{bmatrix} X X^T \\ Y Y^T \end{bmatrix} \text{for CCA}, \quad D = I \text{ for CIPCA}
\]

- **Objective:**
  \[
  \max_W tr \left( W^T C_{z z} W \right) \implies \hat{y} = W_Y W_X^+ x
  \]

\[
z = \begin{bmatrix} x \\ y \end{bmatrix} \in \mathbf{R}^{D+c}
\]

## Results

<table>
<thead>
<tr>
<th>Dataset</th>
<th>PCA</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>LDA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image</td>
<td>95.89</td>
<td>96.02</td>
<td>77.34</td>
<td>92.51</td>
<td>89.62</td>
<td>77.91</td>
</tr>
<tr>
<td>Splice</td>
<td>71.38</td>
<td>70.38</td>
<td>83.37</td>
<td>70.94</td>
<td>73.64</td>
<td>79.08</td>
</tr>
<tr>
<td>Banana</td>
<td>86.36</td>
<td>86.36</td>
<td>54.64</td>
<td>71.83</td>
<td>71.83</td>
<td>61.66</td>
</tr>
<tr>
<td>B.-Cancer</td>
<td>67.04</td>
<td>67.13</td>
<td>54.84</td>
<td>60.71</td>
<td>60.43</td>
<td>64.95</td>
</tr>
<tr>
<td>Diabetis</td>
<td>69.45</td>
<td>69.87</td>
<td>75.57</td>
<td>72.32</td>
<td>73.93</td>
<td>68.70</td>
</tr>
<tr>
<td>F.-Solar</td>
<td>60.91</td>
<td>60.94</td>
<td>62.75</td>
<td>62.81</td>
<td>62.81</td>
<td>61.10</td>
</tr>
<tr>
<td>German</td>
<td>70.07</td>
<td>70.03</td>
<td>75.64</td>
<td>70.75</td>
<td>73.60</td>
<td>67.77</td>
</tr>
<tr>
<td>Heart</td>
<td>77.02</td>
<td>76.76</td>
<td>82.59</td>
<td>77.20</td>
<td>79.83</td>
<td>76.88</td>
</tr>
<tr>
<td>Ringnorm</td>
<td>65.63</td>
<td>65.02</td>
<td>74.78</td>
<td>65.74</td>
<td>68.69</td>
<td>68.22</td>
</tr>
<tr>
<td>Thyroid</td>
<td>95.68</td>
<td>95.67</td>
<td>85.76</td>
<td>94.13</td>
<td>92.44</td>
<td>82.55</td>
</tr>
<tr>
<td>Titanic</td>
<td>67.00</td>
<td>67.00</td>
<td>67.69</td>
<td>73.66</td>
<td>73.66</td>
<td>67.04</td>
</tr>
<tr>
<td>Twonorm</td>
<td>93.45</td>
<td>93.11</td>
<td>97.32</td>
<td>93.30</td>
<td>94.64</td>
<td>96.47</td>
</tr>
<tr>
<td>Waveform</td>
<td>84.43</td>
<td>84.25</td>
<td>80.34</td>
<td>84.48</td>
<td>85.24</td>
<td>81.43</td>
</tr>
</tbody>
</table>
Our new work

- CIPCA
- DCCA
- DCCAM
- Multi-Kernel learning
- Beyond KCCA
DCCA- modified feature extractor

- **Motive**
  1) for feature fusion and multimodal recognition

Multiple Biometrics [JRS04]  Applications

face and fingerprint

Multiple biometrics
DCCA

- **Motive**
  2) in CCA, the correlation between $(x_i, y_i)$ is insufficient to discriminate between classes
  3) in DCCA, we take correlation as similarity metric, aims to maximize the within-class correlation, minimize the between-class correlation

- **Consider the correlation between classes**

![Diagram showing the correlation between classes](image-url)
DCCA

- maximize the within-class correlation
- minimize the between-class correlation

Objective function

$$\max_{w_x, w_y} \frac{w_x^T (C_w - \eta C_b) w_y}{(w_x^T C_{xx} w_x \cdot w_y^T C_{yy} w_y)^{1/2}}$$

where $\eta$ is a tunable parameter.
DCCA

- **within-class correlation**

\[ C_w = \sum_{i=1}^{c} \sum_{k=1}^{n_i} \sum_{l=1}^{n_i} x^{(i)}_{k} y^{(i)T}_{l} = XAY^{T} \]

- **between-class correlation**

\[ C_b = \sum_{i=1}^{c} \sum_{j=1}^{c} \sum_{k=1}^{n_i} \sum_{l=1}^{n_j} x^{(i)}_{k} y^{(j)T}_{l} \]

\[
= (XI_{n})^{T} - XAY^{T} \\
= -XAY^{T} \\
= -C_w
\]

is a block, diagonal matrix of size \( n \)-by-\( n \).
Objective function

\[
\max_{w_x, w_y} \frac{(1 + \eta) \cdot w_x^T C_w w_y}{(w_x^T C_{xx} w_x \cdot w_y^T C_{yy} w_y)^{1/2}} = \max_{w_x, w_y} \frac{w_x^T C_w w_y}{(w_x^T C_{xx} w_x \cdot w_y^T C_{yy} w_y)^{1/2}}
\]

\[
\Rightarrow \begin{pmatrix}
XAY^T \\
YAX^T
\end{pmatrix} \begin{pmatrix}
w_x \\
w_y
\end{pmatrix} = \lambda \begin{pmatrix}
XX^T \\
YY^T
\end{pmatrix} \begin{pmatrix}
w_x \\
w_y
\end{pmatrix}
\]
DCCA Results on toy problem

\[ y_i = W^T x_i + b + \varepsilon_i \]

\[ w = \begin{bmatrix} 0.6 & -\sqrt{\frac{5}{2}} \\ 0.8 & \sqrt{\frac{5}{2}} \end{bmatrix} \quad b = [1, 1]^T \quad \varepsilon_i \text{ the imposed Gaussian noise} \]
Hypertext categorization
(WebKB dataset)

<table>
<thead>
<tr>
<th>Unimodal Recognition</th>
<th>Multimodal Recognition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>Recognition Accuracy</td>
</tr>
<tr>
<td></td>
<td>fulltext</td>
</tr>
<tr>
<td>Naïve Bayes</td>
<td>0.9083</td>
</tr>
<tr>
<td>(k\text{-NN})</td>
<td><strong>0.9448</strong></td>
</tr>
<tr>
<td>CMV</td>
<td>0.9098</td>
</tr>
</tbody>
</table>

Note: PR1 and PR2 correspond to features in parallel and in serial, respectively.
## Handwritten numeral recognition (Multiple Feature database)

<table>
<thead>
<tr>
<th>#</th>
<th>X</th>
<th>Y</th>
<th>Recognition accuracy using different methods</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>DCCA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>PR1</td>
</tr>
<tr>
<td>1</td>
<td>mfeat_fac</td>
<td>mfeat_fou</td>
<td>0.9560</td>
</tr>
<tr>
<td>2</td>
<td>mfeat_fac</td>
<td>mfeat_kar</td>
<td>0.9752</td>
</tr>
<tr>
<td>3</td>
<td>mfeat_fac</td>
<td>mfeat_mor</td>
<td>0.9077</td>
</tr>
<tr>
<td>4</td>
<td>mfeat_fac</td>
<td>mfeat_pix</td>
<td>0.9718</td>
</tr>
<tr>
<td>5</td>
<td>mfeat_fac</td>
<td>mfeat_zer</td>
<td>0.9589</td>
</tr>
<tr>
<td>6</td>
<td>mfeat_fou</td>
<td>mfeat_kar</td>
<td>0.9393</td>
</tr>
<tr>
<td>7</td>
<td>mfeat_fou</td>
<td>mfeat_mor</td>
<td>0.8089</td>
</tr>
<tr>
<td>8</td>
<td>mfeat_fou</td>
<td>mfeat_pix</td>
<td>0.9373</td>
</tr>
<tr>
<td>9</td>
<td>mfeat_fou</td>
<td>mfeat_zer</td>
<td>0.8367</td>
</tr>
<tr>
<td>10</td>
<td>mfeat_kar</td>
<td>mfeat_mor</td>
<td>0.8928</td>
</tr>
<tr>
<td>11</td>
<td>mfeat_kar</td>
<td>mfeat_pix</td>
<td>0.9493</td>
</tr>
<tr>
<td>12</td>
<td>mfeat_kar</td>
<td>mfeat_zer</td>
<td>0.9383</td>
</tr>
<tr>
<td>13</td>
<td>mfeat_mor</td>
<td>mfeat_pix</td>
<td>0.8799</td>
</tr>
<tr>
<td>14</td>
<td>mfeat_mor</td>
<td>mfeat_zer</td>
<td>0.7943</td>
</tr>
<tr>
<td>15</td>
<td>mfeat_pix</td>
<td>mfeat_zer</td>
<td>0.9310</td>
</tr>
</tbody>
</table>

Note: PR1 and PR2 correspond to features in parallel and in serial, respectively.

http://www.ics.uci.edu/~mlearn/MLSummary.html

The works about DCCA have been submitted.
Our new work

- CIPCA
- DCCA
- DCCAM
- Multi-Kernel learning
- Beyond KCCA
Motive

1) deal with missing samples due to sensor failure, high cost, etc.
2) “missing samples” means “not pairwise” for X and Y
3) differ from “missing data” (or “incomplete data”)

Missing data

Missing sample
Usual strategies for missing data

1) expectation maximization [DHS00]
2) substitution of class mean for missing data
3) iterative estimation of the missing data [WM01]
4) local least squares imputation (估算) [KGP05]

Our solution

directly deal with the training set with missing samples
Objective

\[
\max_{w_x, w_y} \frac{w_x^T (C_w - \eta C_b) w_y}{(w_x^T C_{xx} w_x \cdot w_y^T C_{yy} w_y)^{1/2}}
\]

\[
\Rightarrow \begin{pmatrix} XAY^T & w_x \\ YAX^T & w_y \end{pmatrix} = \lambda \begin{pmatrix} XX^T & w_x \\ YY^T & w_y \end{pmatrix}
\]

\[
A = \begin{bmatrix}
I_{m_1 \times n_1} \\
& \ddots \\
& & I_{m_i \times n_i} \\
& & & \ddots \\
& & & & I_{m_k \times n_k}
\end{bmatrix} \in \mathbb{R}^{m \times n}
\]

is a block, diagonal matrix of size \(m\)-by-\(n\).
DCCAM Results on Toy problem

(a) Data
(b) CCA
(c) PLS
(d) DCCAM

Circled samples are missing ones. For (b) CCA and (c) PLS, the un-pairwise samples are deleted.
DCCAM Results on Real datasets

Multiple Feature Database (handwritten numerals), 10 classes, 100 pairs per class. For each class, we

1) delete 10 samples from X, and delete 10 samples from Y.
2) delete 20 samples from X, and delete 20 samples from Y.
3) delete 30 samples from X, and delete 30 samples from Y.
4) delete 40 samples from X, and delete 40 samples from Y.
5) delete 50 samples from X, and delete 50 samples from Y.
6) delete 60 samples from X, and delete 60 samples from Y.

Note: the deleted samples are NOT pairwise.
DCCAM Results on Real datasets

- Adjustment of the comparative algorithms
  For CCA and PLS, the following algorithm is adjusted to match the case of “missing samples”

1) delete the rest un-pairwise samples
   (named as CCA-I, PLS-I) and DCCA-I.

2) substitution of class mean vector for missing samples
   (named as CCA-II, PLS-II) and DCCA-II.

3) iterative estimation of the missing samples
   (named as CCA-III, PLS-III).

Note that DCCA-I and –II are just for comparison, and no DCCA-III is introduced.
DCCAM Results on Real datasets

X-axis: the # of missing samples per class. i.e., 10, 20, ... 50, 60.

Y-axis: the recognition accuracy.
DCCAM Results on Real datasets
DCCAM Results on Real datasets

The works about DCCAM have been submitted.
DCCAM advantages

- Supervised learning method due to the incorporation of the class information
- Tolerance to the missing of samples
- Superior performance to some multimodal recognition algorithms
- Direct processing ability of missing samples and without resorting to artificially compensate the missing samples
- Relative insensitivity to the number of missing samples
- Only involvement of one tunable parameter, \( d \), the final dimensionality of the extracted features. This makes it easy to be manipulated in real applications.
Our new work

- CIPCA
- DCCA
- DCCAM
- Multi-Kernel learning
- Beyond KCCA
Multi-Kernel Learning (MKL)

- **Motive:**
  1. approximate heterogeneous (异质) data sources;
  2. avoid model or kernel parameter selection to some extent;

- The related work [BLJ04, SRS05]

  The conic combinations of multiple kernels

\[ K = \sum_i \alpha_i K_i \]


Multi-Kernel Learning (MKL)

- **Our Method**
  
borrow multiple CCA (mCCA) to fuse multiple kernels

**Essence**: Single dataset $\rightarrow$ Multi-Kernel mapping

+ Multi-CCA (NmCCA) $\rightarrow$ MKL
Multi-Kernel Learning (MKL)

mCCA

\[
J = \min_{W^{(1)}, \ldots, W^{(m)}} \sum_{k,l=1; k \neq l}^m \| S^{(k)} W^{(k)} - S^{(l)} W^{(l)} \|_F \\
\text{s.t.} \quad W^{(k)^T} C_{kk} W^{(k)} = I; \\
\quad w_i^{(k)^T} C_{kl} w_j^{(l)} = 0; \\
\quad k, l = 1, \ldots, m, l \neq k; i, j = 1, \ldots, q, j \neq i;
\]

NmCCA

\[
L = \min_{\omega_l \in \mathbb{R}^{n_l+1}, b_l > 0; l = 1, \ldots, m} \sum_{l=1}^m ((Y_i \omega_l - I_{N \times 1} - b_l)^T (Y_i \omega_l - I_{N \times 1} - b_l) + c_l \omega_l^T \omega_l) \\
+ c_{m+1} \sum_{l=1}^m (Y_i \omega_l - \frac{1}{m} \sum_{j=1}^m Y_j \omega_j)^T (Y_i \omega_l - \frac{1}{m} \sum_{j=1}^m Y_j \omega_j),
\]

Least Squares Criterion

New Multi-CCA (NmCCA)
MKL Results compared to [SRS05]

Classification performance comparison

<table>
<thead>
<tr>
<th>Datasets</th>
<th>MultiK-MHKS</th>
<th>CCA+MHKS1</th>
<th>CCA+MHKS2</th>
<th>MHKS(best)</th>
<th>MHKS(worst)</th>
<th>MKL[21]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Soybean-small</td>
<td>99.57</td>
<td>90.00</td>
<td>88.26</td>
<td>96.52</td>
<td>94.35</td>
<td>97.83</td>
</tr>
<tr>
<td>Balance</td>
<td>93.94</td>
<td>87.88</td>
<td>87.88</td>
<td>90.16</td>
<td>90.10</td>
<td>90.19</td>
</tr>
<tr>
<td>Water</td>
<td>96.21</td>
<td>57.27</td>
<td>63.79</td>
<td>94.85</td>
<td>94.70</td>
<td>93.03</td>
</tr>
<tr>
<td>Sonar</td>
<td>80.65</td>
<td>72.78</td>
<td>72.78</td>
<td>75.83</td>
<td>75.74</td>
<td>75.28</td>
</tr>
<tr>
<td>Wdbc</td>
<td>63.89</td>
<td>63.48</td>
<td>62.96</td>
<td>62.48</td>
<td>59.04</td>
<td>50.81</td>
</tr>
<tr>
<td>Pima-diabetes</td>
<td>71.40</td>
<td>70.83</td>
<td>70.20</td>
<td>69.54</td>
<td>62.91</td>
<td>71.43</td>
</tr>
<tr>
<td>Iris</td>
<td>97.33</td>
<td>92.67</td>
<td>90.53</td>
<td>97.33</td>
<td>97.07</td>
<td>97.33</td>
</tr>
<tr>
<td>Wine</td>
<td>96.89</td>
<td>89.81</td>
<td>93.02</td>
<td>95.19</td>
<td>88.96</td>
<td>79.15</td>
</tr>
</tbody>
</table>
MKL Results compared to [SRS05]

*t*-test comparison on MultiK-MHKS with the following algorithms

<table>
<thead>
<tr>
<th>Datasets</th>
<th>CCA+MHKS1</th>
<th>CCA+MHKS2</th>
<th>MHKS(best)</th>
<th>MHKS(worst)</th>
<th>MKL[21]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Soybean-small</td>
<td>5.3906</td>
<td>4.5810</td>
<td>2.2778</td>
<td>3.4330</td>
<td>2.0580</td>
</tr>
<tr>
<td>Balance</td>
<td>10.6106</td>
<td>10.6106</td>
<td>6.8245</td>
<td>6.2137</td>
<td>6.5939</td>
</tr>
<tr>
<td>Water</td>
<td>19.2638</td>
<td>24.3671</td>
<td>1.4443*</td>
<td>1.4128*</td>
<td>2.6877</td>
</tr>
<tr>
<td>Sonar</td>
<td>6.8811</td>
<td>6.8811</td>
<td>4.4371</td>
<td>3.2984</td>
<td>2.3520</td>
</tr>
<tr>
<td>Wdbc</td>
<td>1.0855*</td>
<td>2.5204</td>
<td>2.7287</td>
<td>4.1504</td>
<td>9.8425</td>
</tr>
<tr>
<td>Pima-diabetes</td>
<td>3.0534</td>
<td>2.9210</td>
<td>2.5185</td>
<td>27.5278</td>
<td>1.0000*</td>
</tr>
<tr>
<td>Iris</td>
<td>6.2163</td>
<td>8.9938</td>
<td>0*</td>
<td>0.3612*</td>
<td>0*</td>
</tr>
<tr>
<td>Wine</td>
<td>4.8455</td>
<td>3.8432</td>
<td>2.6056</td>
<td>8.9000</td>
<td>3.8048</td>
</tr>
</tbody>
</table>

‘*’ Denotes that the difference between the two corresponding algorithms is not significant at 5% significance level, i.e. $t$-value $< 1.7341$.

The works about MKL have been submitted.
Our new work

- CIPCA
- DCCA
- DCCAM
- Multi-Kernel learning
- Beyond KCCA
Beyond KCCA

- Proof of “KCCA = 2-KPCA + CCA”
Beyond KCCA

Insight from “KCCA = 2-KPCA + CCA”

- Reveal the essence of KCCA
- What is important …
  1) the original kernel mappings can be generalized to empirical kernel mappings, i.e.,

\[ \phi : x \mapsto \phi(x) \text{ and } \psi : y \mapsto \psi(y) \]

\[ \Rightarrow x \mapsto K_X(X, x) \text{ and } y \mapsto K_Y(Y, y) \]
2) More generally

\[ x \mapsto F_X(X, x) \text{ and } y \mapsto F_Y(Y, y) \]

where both \( F_X \) and \( F_Y \) are non-negative real functions.

- **Advantages**
  - Any types of kernel can be employed,
    - e.g., graph-kernel, string-kernel, tree-kernel, and etc.,
  - even non-kernel functions

The works about Beyond KCCA have been submitted.
References


References


- [KGP05] H. Kim, G.H. Golub and H. Park, Missing value estimation for DNA microarray gene expression data: local least squares imputation, Bioinformatics, vol.21, 187-198, 2005


References


Thanks!