SIMFEAT: A Simple Feature Extraction Toolbox

Emma Izquierdo-Verdiguier, Luis Gómez-Chova and Gustavo Camps-Valls

Image Processing Laboratory (IPL) – Universitat de València. Spain.

Motivation

- Feature selection/extraction is essential before classification or regression
- High number of correlated features gives rise to:
  - Collinearity
  - Overfitting
- Linear methods offer interpretability $\sim$ knowledge discovery
- Linear algorithms are commonly used: PCA, PLS, CCA, ...
- Linear algorithms fail when data distributions are curved

Motivation

- Feature selection/extraction is essential before classification or regression
- High number of correlated features gives rise to:
  - Collinearity
  - Overfitting
- Linear methods offer interpretability $\sim$ knowledge discovery
- Linear algorithms are commonly used: PCA, PLS, CCA, ...
- Linear algorithms fail when data distributions are curved

Outline

- PCA is widely used
- PCA is not optimal for supervised problems
- PLS is a good alternative to PCA, yet *suboptimal* in MSE sense
- Orthonormalized PLS (OPLS) is optimal in MSE sense
- Real problems typically show non-linear relations

Notation preliminaries

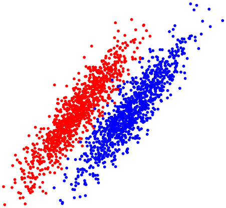
Data	$\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{l}$, $\mathbf{x}_i \in \mathbb{R}^N$, $\mathbf{y}_i \in \mathbb{R}^M$.
Input Data Matrix	$\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_l]^\top$
Label Matrix	$\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_l]^\top$
Number of projections	n_p
Projected Inputs	$\mathbf{X}' = \mathbf{X}\mathbf{U}$
Projected Outputs	$\mathbf{Y}' = \mathbf{Y}\mathbf{V}$
Projection matrices	$\mathbf{U}$ $(N \times n_p)$, and $\mathbf{V}$ $(M \times n_p)$
Covariance	$\mathbf{C}_{xy} = E\{(\mathbf{x} - \boldsymbol{\mu}_x)(\mathbf{y} - \boldsymbol{\mu}_y)\} \sim \frac{1}{l}\mathbf{X}^\top\mathbf{Y}$
Frobenius norm of a matrix	$\|A\|_F^2 = \sum_{ij} a_{ij}^2$

Notation preliminaries

1. $tr(\mathbf{AB}) = tr(\mathbf{BA})$
2. $(\mathbf{AB})^\top = \mathbf{B}^\top \mathbf{A}^\top$, $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$
3. $\frac{\partial(\mathbf{a}^\top \mathbf{b})}{\partial \mathbf{a}} = \mathbf{b}$, $\frac{\partial(\mathbf{b}^\top \mathbf{a})}{\partial \mathbf{a}} = \mathbf{b}^\top$
4. $\mathbf{CV} = \mathbf{VD} \longrightarrow$ `[V D] = eig(C);`
5. $\mathbf{C} = \mathbf{UAV}^\top \longrightarrow$ `[U A V] = svd(C);`
6. Orthogonal transform = rotation $\rightarrow \mathbf{U}^{-1} = \mathbf{U}^\top$
7. Download `matrixcookbook.pdf` !

Notation preliminaries

- Imagine a classification problem in which labels matter (a lot!)
- "Blind" feature extraction is not a good choice.
- Let's see what happens with different methods ...



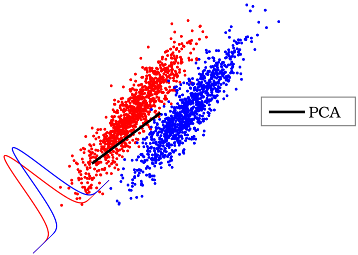
Principal Component Analysis (PCA)

"Find projections that maximize the variance of the projected data"

PCA: maximize: $\mathrm{Tr}\{\|\mathbf{XU}\|_F^2\} = \mathrm{Tr}\{(\mathbf{XU})^\top(\mathbf{XU})\} = \mathrm{Tr}\{\mathbf{U}^\top\mathbf{C}_{xx}\mathbf{U}\}$

subject to: $\mathbf{U}^\top\mathbf{U} = \mathbf{I}$

```
» Cxx = X'*X;
» [U D] = eigs(Cxx,np);
» Xproj = X*U;
```

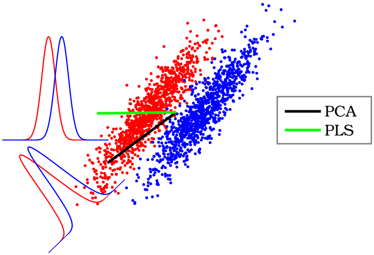


Partial Least Squares (PLS)

"Find directions of maximum input-output cross-covariance"

$$\text{PLS:} \quad \text{maximize:} \quad \mathrm{Tr}\{(\mathbf{XU})^\top(\mathbf{YV})\} = \mathrm{Tr}\{\mathbf{U}^\top \mathbf{C}_{xy}\mathbf{V}\}$$
$$\text{subject to:} \quad \mathbf{U}^\top\mathbf{U} = \mathbf{V}^\top\mathbf{V} = \mathbf{I}$$

```
» Cxy=X'*Y;
» [U D V] = svds(Cxy,np);
» Xproj = X*U;
```



Orthonormalized Partial Least Squares (OPLS)

"Choose the projection matrix **U** *that minimizes the MSE error ..."*

OPLS: find: $\mathbf{U} = \arg\min\{\|\mathbf{Y} - \mathbf{X}'\mathbf{W}\|_F^2\}$

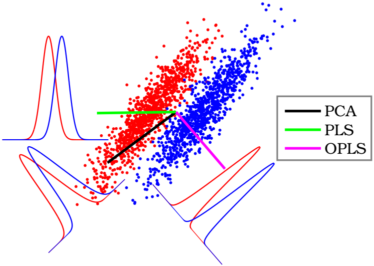
where: $\mathbf{X}' = \mathbf{X}\mathbf{U}, \qquad \mathbf{W} = (\mathbf{X}'^\top \mathbf{X}')^{-1}\mathbf{X}'\mathbf{Y}$

Orthonormalized Partial Least Squares (OPLS)

"... which can be rewritten as **[Worsley98]***:"*

$$\text{OPLS:} \quad \text{maximize:} \quad \mathrm{Tr}\{\mathbf{U}^\top \mathbf{C}_{xy}\mathbf{C}_{xy}^\top \mathbf{U}\}$$
$$\text{subject to:} \quad \mathbf{U}^\top \mathbf{C}_{xx}\mathbf{U} = \mathbf{I}$$

```
» Cxy=X'*Y;Cxx=X'*X;
» [U,D] = eigs((Cxy)*(Cxy'),Cxx,np);
» [U,D] = eigs(inv(Cxx)*(Cxy)*(Cxy'),np);
» Xproj = X*U;
```



Canonical Correlation Analysis (CCA)

Unlike PCA or PLS, CCA looks for directions of max I/O correlation:

CCA: $$\mathbf{u}, \mathbf{v} = \arg\max_{\mathbf{u},\mathbf{v}} \frac{(\mathbf{u}^\top \mathbf{C}_{xy} \mathbf{v})^2}{\mathbf{u}^\top \mathbf{C}_{xx} \mathbf{u} \; \mathbf{v}^\top \mathbf{C}_{yy} \mathbf{v}}$$

CCA(2): $$\mathbf{u}, \mathbf{v} = \arg\max_{\mathbf{u},\mathbf{v}} \; \mathbf{u}^\top \mathbf{C}_{xy} \mathbf{v}$$

$$\text{subject to: } \mathbf{u}^\top \mathbf{C}_{xx} \mathbf{u} = \mathbf{v}^\top \mathbf{C}_{yy} \mathbf{v} = 1$$

CCA(3): $$\mathbf{U}, \mathbf{V} = \arg\max_{\mathbf{U},\mathbf{V}} \; \mathrm{Tr}\{\mathbf{U}^\top \mathbf{C}_{xy} \mathbf{V}\}$$

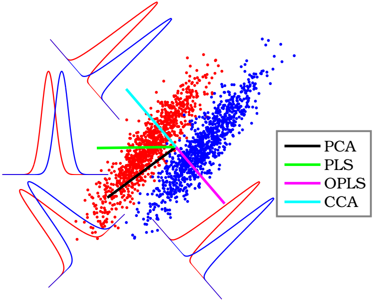
$$\text{subject to: } \mathbf{U}^\top \mathbf{C}_{xx} \mathbf{U} = \mathbf{V}^\top \mathbf{C}_{yy} \mathbf{V} = \mathbf{I}$$

Canonical Correlation Analysis (CCA)

Unlike PCA or PLS, CCA looks for directions of max I/O correlation:

$$\begin{pmatrix} \mathbf{0} & \mathbf{C}_{xy} \\ \mathbf{C}_{xy}^{\top} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix} = \lambda \begin{pmatrix} \mathbf{C}_{xx} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{yy} \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix}$$

```
» A = [0 Cxy; Cxy' 0];   B = [Cxx 0; 0 Cyy];
» [UV D] = eigs(A,B,np); u = UV(1:d/2,:);
» Xproj = X*u;
```



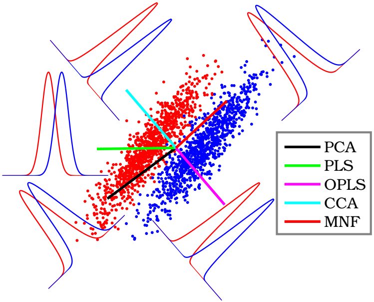
Minimum Noise Fraction (MNF)

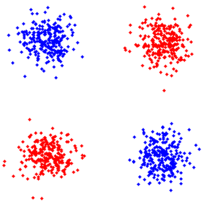
"Maximize the signal-to-noise ratio of the projections:"

$$\text{MNF:} \qquad \text{maximize:} \quad \mathrm{Tr}\left\{\frac{(\mathbf{XU})^\top(\mathbf{XU})}{(\mathbf{NU})^\top(\mathbf{NU})}\right\}$$

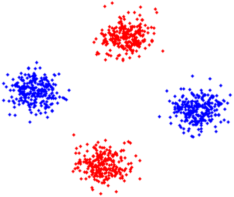
$$\text{subject to:} \quad \mathbf{X} = \mathbf{S} + \mathbf{N}, \qquad \mathbf{S}^\top\mathbf{N} = \mathbf{S}\mathbf{N}^\top = \mathbf{0}$$

```
» N=noise(X,10);
» [U,D] = eigs(X'*X,N'*N,np);
» Xproj = X*U;
```

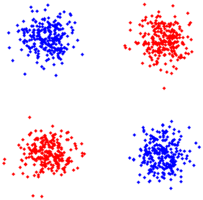




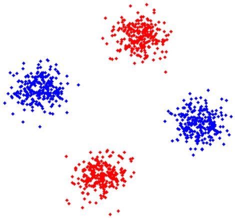
Original data



PCA



Original data



OPLS

Kernel methods for non-linear feature extraction

1. Map the points in $\mathcal{X}$ to a higher dimensional space $\mathcal{H}$:

$$\mathbf{X} \rightarrow \mathbf{\Phi}$$

2. Express projection matrix $\mathbf{U}$ in $\mathcal{H}$ as a linear combination of mapped data

$$\mathbf{U} = \mathbf{\Phi}^\top \mathbf{A}$$

3. Replace the dot (scalar) products by a kernel function:

$$\mathbf{K} = \mathbf{\Phi}\mathbf{\Phi}^\top$$

4. Express your new algorithm as a function of $\mathbf{K}$ and solve it for $\mathbf{A}$
5. Compute projections

$$\mathcal{P}(\mathbf{X}^*) = \mathbf{\Phi}^*\mathbf{U} = \mathbf{\Phi}^*\ \mathbf{\Phi}^\top\mathbf{A} = \mathbf{K}(\mathbf{X}^*, \mathbf{X})\mathbf{A}$$

Kernel Principal Component Analysis (KPCA)

- *"Find projections maximizing the variance of the projected data in $\mathcal{H}$"*
- Apply the representer's theorem: $\mathbf{U} = \mathbf{\Phi}^\top \mathbf{A}$ where $\mathbf{A} = [\alpha_1, \ldots, \alpha_n]^\top$

$$\text{KPCA:} \qquad \begin{aligned} \text{maximize:} \quad & \text{Tr}\{\mathbf{A}^\top \mathbf{K}\mathbf{K}\mathbf{A}\} \\ \text{subject to:} \quad & \mathbf{A}^\top \mathbf{K}\mathbf{A} = \mathbf{I} \end{aligned}$$

- Including Lagrange multipliers $\mathbf{\Lambda}$, this problem is equivalent to

$$\mathbf{K}\mathbf{K}\mathbf{A} = \mathbf{K}\mathbf{\Lambda}\mathbf{A} \quad \rightarrow \quad \mathbf{K}\mathbf{A} = \mathbf{\Lambda}\mathbf{A}$$

```
» [A,L] = eigs(K,np);
» Xtestproj = K(Xtest,Xtrain)*A;
```

Kernel Partial Least Squares (KPLS)

- *"Find projections for maximum input-output cross-covariance"*
- Apply the representer's theorem: $\mathbf{U} = \mathbf{\Phi}^\top \mathbf{A}$ where $\mathbf{A} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_n]^\top$

$$\text{KPLS:} \qquad \begin{array}{ll} \text{maximize:} & \text{Tr}\{\mathbf{U}^\top \mathbf{\Phi}^\top \mathbf{Y} \mathbf{V}\} \\ \text{subject to:} & \mathbf{U}^\top \mathbf{U} = \mathbf{V}^\top \mathbf{V} = \mathbf{I} \end{array}$$

- Including Lagrange multipliers $\mathbf{\Lambda}$, this problem is equivalent to

$$\left(\begin{array}{cc} \mathbf{0} & \mathbf{K}_x \mathbf{Y} \\ \mathbf{Y} \mathbf{K}_x & \mathbf{0} \end{array} \right) \left(\begin{array}{c} \boldsymbol{\alpha} \\ \mathbf{v} \end{array} \right) = \lambda \left(\begin{array}{c} \boldsymbol{\alpha} \\ \mathbf{v} \end{array} \right)$$

» $[\alpha \;\; V \;\; D] = svds(K * Y, np)$;

» `Xtestproj = K(Xtest,Xtrain) *` α;

Kernel Orthonormalized Partial Least Squares (KOPLS)

- *"Find optimal projections in MSE terms"*
- Apply the representer's theorem: $\mathbf{U} = \mathbf{\Phi}^\top \mathbf{A}$ where $\mathbf{A} = [\alpha_1, \ldots, \alpha_n]^\top$

$$\text{KOPLS:} \quad \begin{array}{ll} \text{maximize:} & \text{Tr}\{\mathbf{U}^\top \mathbf{\Phi}^\top \mathbf{Y}\mathbf{Y}^\top \mathbf{\Phi}\mathbf{U}\} \\ \text{subject to:} & \mathbf{U}^\top \mathbf{\Phi}^\top \mathbf{\Phi}\mathbf{U} = \mathbf{I} \end{array}$$

- Including Lagrange multipliers $\mathbf{\Lambda}$, this problem is equivalent to

$$\text{KOPLS:} \quad \begin{array}{ll} \text{maximize:} & \text{Tr}\{\mathbf{A}^\top \mathbf{K}_x \mathbf{K}_y \mathbf{K}_x \mathbf{A}\} \\ \text{subject to:} & \mathbf{A}^\top \mathbf{K}_x \mathbf{K}_x \mathbf{A} = \mathbf{I} \end{array}$$

```
» M = Kx*Y*Y'*Kx;   N = Kx*Kx;
» [A D] = eigs(M,N,np);
» Xtestproj = K(Xtest,Xtrain) * A;
```

Kernel Canonical Correlation Analysis (KCCA)

- *"Find projections that maximize input-output correlation"*
- Apply the representer's theorem: $\mathbf{U} = \mathbf{\Phi}^\top \mathbf{A}$ where $\mathbf{A} = [\alpha_1, \ldots, \alpha_n]^\top$
- Including Lagrange multipliers $\mathbf{\Lambda}$, this problem is equivalent to

$$\begin{pmatrix} \mathbf{0} & \mathbf{K}_x\mathbf{Y} \\ \mathbf{Y}\mathbf{K}_x & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{a} \\ \mathbf{v} \end{pmatrix} = \lambda \begin{pmatrix} \mathbf{K}_x\mathbf{K}_x & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_y \end{pmatrix} \begin{pmatrix} \mathbf{a} \\ \mathbf{v} \end{pmatrix}$$

```
» M = [0 Kx*Y;Y*Kx 0];     N = [Kx*Kx 0;0 Cyy];
» [AV D] = eigs(M,N,np);
» a = AV(1:n/2,:);
» Xtestproj = K(Xtest,Xtrain) * a;
```

Kernel Minimum Noise Fraction (KMNF)

- *"Find projections that maximize SNR in $\mathcal{H}$"*
- Apply the representer's theorem: $\mathbf{U} = \mathbf{\Phi}^\top \mathbf{A}$ where $\mathbf{A} = [\alpha_1, \ldots, \alpha_n]^\top$
- Including Lagrange multipliers $\mathbf{\Lambda}$, this problem is equivalent to

$$\text{KMNF:} \qquad \text{maximize:} \quad \mathrm{Tr}\left\{ \frac{\mathbf{A}^\top \mathbf{K}^2 \mathbf{A}}{\mathbf{A}^\top \mathbf{K}_{xn} \mathbf{K}_{xn} \mathbf{A}} \right\}$$

$$\text{subject to:} \quad \mathbf{A}^\top \mathbf{K}_{xn} \mathbf{K}_{xn} \mathbf{A} = \mathbf{I}$$

```
» Kxx=kernel('rbf',X,X,sigma);
» Kxx=kernelcentering(Kxx);
» N=noise(X,10);
» Kxn=kernel('rbf',X,N,sigma);
» Kxn=kernelcentering(Kxn);
» [A D] = eigs(Kxx*Kxx',Kxn*Kxn',np);
» Xtestproj = K(Xtest,Xtrain) * A;
```

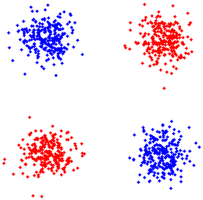
Kernel Entropy Components Analysis (KECA)

- *"Find projections with maximum entropy in $\mathcal{H}$"*
- Apply the representer's theorem: $\mathbf{U} = \mathbf{\Phi}^\top \mathbf{A}$ where $\mathbf{A} = [\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_n]^\top$
- Including Lagrange multipliers $\mathbf{\Lambda}$, this problem is equivalent to

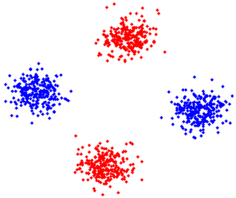
$$\text{KECA:} \qquad \mathbf{K} = \mathbf{A}\mathbf{D}\mathbf{A}^\top \quad \text{s.t. maximum Rényi entropy of } \mathbf{\Phi}\mathbf{U}$$

```
» K = kernel('rbf',X,X,sigma);
» [A,D] = eigs(K);
» lambda = diag(D);
» d=size(A,1);
» for t=1:size(A,2)
      m(t)=lambda(t)*(A(:,t)'*ones(d,1))^2;
  end
» [val,ind] = sort(m,'descend');
» A = A(:,ind);
» Xtestproj = K(Xtest,Xtrain) * A;
```

An illustrative example (cont'd)

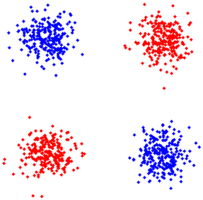


Original data

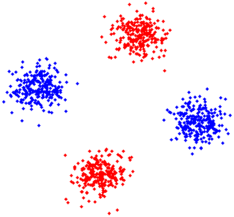


PCA

An illustrative example (cont'd)

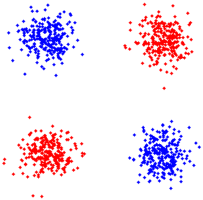


Original data

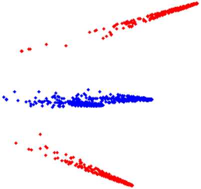


OPLS

An illustrative example (cont'd)

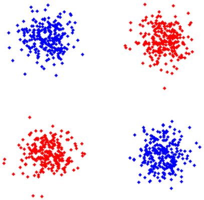


Original data



KPCA

An illustrative example (cont'd)



Original data

KOPLS

Method	Maximized Function	Constraints	Solved Problem	Features
PCA	$\mathrm{Tr}\{\mathbf{U}^\top \mathbf{C}_{xx}\mathbf{U}\}$	$\mathbf{U}^\top\mathbf{U}=\mathbf{I}$	$\mathbf{C}_{xx}\mathbf{U}=\mathbf{U}\boldsymbol{\Lambda}$	$\tilde{\mathbf{X}}\mathbf{U}$
	PCA projects linearly the input data onto the directions of largest input variance.			
MNF	$\mathrm{Tr}\{(\mathbf{U}^\top \mathbf{C}_{xx}\mathbf{U})/(\mathbf{U}^\top \mathbf{C}_{nn}\mathbf{U})\}$	$\mathbf{U}^\top\mathbf{C}_{nn}\mathbf{U}=\mathbf{I}$	$\mathbf{C}_{xx}\mathbf{U}=\mathbf{C}_{nn}\mathbf{U}\boldsymbol{\Lambda}$	$\tilde{\mathbf{X}}\mathbf{U}$
	MNF maximizes the ratio between the signal and the noise variances for all the features.			
PLS	$\mathrm{Tr}\{\mathbf{U}^\top \mathbf{C}_{xy}\mathbf{V}\}$	$\mathbf{U}^\top\mathbf{U}=\mathbf{V}^\top\mathbf{V}=\mathbf{I}$	$\mathbf{C}_{xy}=\mathbf{U}\boldsymbol{\Lambda}\mathbf{V}^\top$	$\tilde{\mathbf{X}}\mathbf{U}$
	PLS finds directions of maximum covariance between the projected input and desired output $\mathbf{Y}$.			
OPLS	$\mathrm{Tr}\{\mathbf{U}^\top \mathbf{C}_{xy}\mathbf{C}_{xy}^\top\mathbf{U}\}$	$\mathbf{U}^\top\mathbf{C}_{xx}\mathbf{U}=\mathbf{I}$	$\mathbf{C}_{xy}\mathbf{C}_{xy}^\top\mathbf{U}=\mathbf{C}_{xx}\mathbf{U}\boldsymbol{\Lambda}$	$\tilde{\mathbf{X}}\mathbf{U}$
	OPLS finds optimal directions for performing linear regression of $\tilde{\mathbf{Y}}$ on the projected input data.			
CCA	$\mathrm{Tr}\{\mathbf{U}^\top \mathbf{C}_{xy}\mathbf{V}\}$	$\mathbf{U}^\top\mathbf{C}_{xx}\mathbf{U}=\mathbf{I}\ \mathbf{V}^\top\mathbf{C}_{yy}\mathbf{V}=\mathbf{I}$	$\begin{pmatrix}\mathbf{0} & \mathbf{C}_{xy}\\ \mathbf{C}_{xy}^\top & \mathbf{0}\end{pmatrix}\begin{pmatrix}\mathbf{U}\\ \mathbf{V}\end{pmatrix}=\begin{pmatrix}\mathbf{C}_{xx} & \mathbf{0}\\ \mathbf{0} & \mathbf{C}_{yy}\end{pmatrix}\begin{pmatrix}\mathbf{U}\\ \mathbf{V}\end{pmatrix}\boldsymbol{\Lambda}$	$\tilde{\mathbf{X}}\mathbf{U}$
	CCA finds projections of the input and output data matrices such that each column of $\tilde{\mathbf{Y}}$ can be reconstructed from the corresponding column of $\tilde{\mathbf{X}}$ with minimum square loss.			
KPCA	$\mathrm{Tr}\{\mathbf{A}^\top \tilde{\mathbf{K}}_{xx}\tilde{\mathbf{K}}_{xx}\mathbf{A}\}$	$\mathbf{A}^\top\tilde{\mathbf{K}}_{xx}\mathbf{A}=\mathbf{I}$	$\tilde{\mathbf{K}}_{xx}\mathbf{A}=\mathbf{A}\boldsymbol{\Lambda}$	$\tilde{\mathbf{K}}_{xx}\mathbf{A}$
	KPCA finds directions of maximum variance of the input data in $\mathcal{H}$.			
KMNF	$\mathrm{Tr}\{(\mathbf{A}^\top \tilde{\mathbf{K}}_{xx}^2\mathbf{A})/(\mathbf{A}^\top\tilde{\mathbf{K}}_{xn}\tilde{\mathbf{K}}_{nx}\mathbf{A})\}$	$\mathbf{A}^\top\tilde{\mathbf{K}}_{xn}\tilde{\mathbf{K}}_{nx}\mathbf{A}=\mathbf{I}$	$\tilde{\mathbf{K}}_{xx}^2\mathbf{A}=\tilde{\mathbf{K}}_{xn}\tilde{\mathbf{K}}_{xn}^\top\mathbf{A}\boldsymbol{\Lambda}$	$\tilde{\mathbf{K}}_{xx}\mathbf{A}$
	KMNF finds directions of maximum signal to noise ratio of the projected data in $\mathcal{H}$.			
KPLS	$\mathrm{Tr}\{\mathbf{A}^\top \tilde{\mathbf{K}}_{xx}\tilde{\mathbf{Y}}\mathbf{V}\}$	$\mathbf{A}^\top\tilde{\mathbf{K}}_{xx}\mathbf{A}=\mathbf{V}^\top\mathbf{V}=\mathbf{I}$	$\tilde{\mathbf{K}}_{xx}\tilde{\mathbf{Y}}=\mathbf{A}\boldsymbol{\Lambda}\mathbf{V}^\top$	$\tilde{\mathbf{K}}_{xx}\mathbf{A}$
	KPLS finds directions of maximum covariance between the input data in $\mathcal{H}$ and the output $\mathbf{Y}$.			
KOPLS	$\mathrm{Tr}\{\mathbf{A}^\top \mathbf{K}_{xx}\mathbf{K}_{yy}\mathbf{K}_{xx}\mathbf{A}\}$	$\mathbf{A}^\top\mathbf{K}_{xx}^2\mathbf{A}=\mathbf{I}$	$\mathbf{K}_{yy}\mathbf{K}_{xx}\mathbf{A}=\mathbf{K}_{xx}\mathbf{A}\boldsymbol{\Lambda}$	$\tilde{\mathbf{K}}_{xx}\mathbf{A}$
	KOPLS extracts features that minimize the residuals of a multiregression approximating $\tilde{\mathbf{Y}}$.			
KCCA	$\mathrm{Tr}\{\mathbf{A}^\top \mathbf{K}_{xx}\tilde{\mathbf{Y}}\mathbf{V}\}$	$\mathbf{A}^\top\mathbf{K}_{xx}^2\mathbf{A}=\mathbf{I}\ \mathbf{V}^\top\mathbf{C}_{yy}\mathbf{V}=\mathbf{I}$	$\begin{pmatrix}\mathbf{0} & \mathbf{K}_{xx}\tilde{\mathbf{Y}}\\ \tilde{\mathbf{Y}}^\top\mathbf{K}_{xx} & \mathbf{0}\end{pmatrix}\begin{pmatrix}\mathbf{A}\\ \mathbf{V}\end{pmatrix}=\begin{pmatrix}\mathbf{K}_{xx}^2 & \mathbf{0}\\ \mathbf{0} & \mathbf{C}_{yy}\end{pmatrix}\begin{pmatrix}\mathbf{A}\\ \mathbf{V}\end{pmatrix}\boldsymbol{\Lambda}$	$\tilde{\mathbf{K}}_{xx}\mathbf{A}$
	KCCA finds directions in feature space of maximum correlation with a certain projection of $\tilde{\mathbf{Y}}$.			

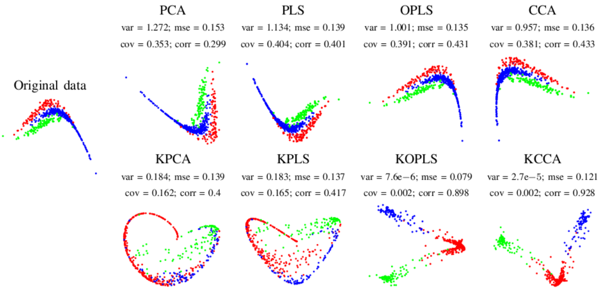


Figura: Projections by different methods in a three-class problem. For the first projection of each method, we show its variance (var), the mean-square-error when the projected data **XU** are used to approximate **Y** (mse), and the largest covariance (cov) and correlation (corr) that can be achieved with any linear projection of the target data.

Conclusions

- Defined the most useful linear and kernel methods for feature extraction
- KPCA is useful in unsupervised learning or as pre-processor
- An excellent alternative to supervised methods is KPLS
- KOPLS is optimal in the MSE sense
- KCCA optimizes correlation but strong regularization needed
- Major problem: non-sparse computationally demanding methods

References

- I. T. Jollife, Principal Component Analysis, Springer, 1986.
- J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*, Cambridge University Press, 2004.
- R. Rosipal and N. Kramer, "Overview and recent advances in partial least squares," *Subspace, Latent Structure and Feature Selection Techniques*, 2006.
- A.A. Green, M. Berman, P. Switzer, and M.D. Craig. A transformation for ordering multispectral data in terms of image quality with implications for noise removal. IEEE TGARS, 26 (1): 65–74, Jan 1988b.
- R. Rosipal and L.J. Trejo, "Kernel partial least squares regression in reproducing kernel Hilbert space," JMLR, vol. 2, pp. 97–123, March 2002.
- J. Arenas-García and G. Camps-Valls, "Eficient Kernel Orthonormalized PLS for Remote Sensing Applications," IEEE TGARS, vol. 46, no. 10, pp. 2872-2881, 2008.
- L. Gómez-Chova, A.A. Nielsen, and G. Camps-Valls. Explicit signal to noise ratio in reproducing kernel Hilbert spaces. In IEEE IGARSS, pages 3570–3570. IEEE, Jul 2011c.
- Robert Jenssen, "Kernel Entropy Component Analysis", IEEE TPAMI, vol. 32, no. 5, May 2010.
- `http://isp.uv.es/soft_feature.htm`

Notes on kernel methods ...

- Centering data in kernel feature spaces is typically assumed

$$\mathbf{K} \leftarrow \mathbf{HKH}$$

where $H_{ij} = \delta_{ij} - \frac{1}{l}$, $\delta_{ij} = 1$ if $i = j$, and zero otherwise.

```
» H = eye(l)-1/l*ones(l,l);
» K = H*K*H;
```

- Estimate the sigma value for the RBF kernel

```
» sigma = mean(pdist(X));
» sigma = median(pdist(X));
```

- Plot the empirical mapping

```
» K = kernel('rbf',X,X,sigma);
» [V D] = eig(K);    % K=PHI*PHI'=V*D*V';
» PHI = V*D^(1/2);
» plot(PHI(:,1),PHI(:,2),'ko');
» plot3(PHI(:,1),PHI(:,2),PHI(:,3),'ko');
```

Nom	Mida	Tipus
gpml-matlab	4 elements	carpeta
methods	11 elements	carpeta
tools	14 elements	carpeta
Contents.m	2,5 KB	script/funció MATLAB
COPYING.txt	1,4 KB	document de text pla
lecture.pdf	1,7 MB	Adobe Acrobat Document
README.txt	5,1 KB	document de text pla
simfeat.m	8,1 KB	script/funció MATLAB

```
» cd methods
» ls
     cca.m kcca.m keca.m kmnf.m kopls.m
     kpca.m kpls.m mnf.m opls.m pca.m pls.m
» help kcca
» np = 10;
» U = pca(X,np);      % a linear unsupervised method
» U = pls(X,Y,np);    % a linear supervised method
» A = kpca(X,np);     % a kernel unsupervised method
» A = kopls(X,Y,np);% a kernel supervised method
» U,A: struct!
```

```
» Xprojtest = Xtest*U.basis;
» Xprojtest = kernel('rbf',Xtest,X,sigma) * A.basis;
» cd ..
» Ypred = predict(Y,Xtest,U,np);
» Ypred = predict(Y,Xtest,A,np);
» cd tools
» ls
» binarize.m gen_eig.m kernel.m
» plotFeatures.m plotKernelFeatures.m figures.m
     estimateSigma.m generate_toydata.m noise.m
     kernelcentering.m predict.m
» simfeat
```