Applications of the Principal Components Transform concept

António Barros & D.N. Rutledge
Universidade de Aveiro & Institut National Agronomique Paris-Grignon

_ Very often one has, for a single sample, a signal with thousands of points, *e.g.* 2D-NMR, 2D-Fluorescence, Images etc.

_ The analysis of these complex signals is often only possible using chemometric methods to extract meaningful information.

_ The main concern in this case is the constraint of available computer resources, in particularly memory, and computation speed.

Hence…

_ There is a growing need for data treatment methods for such very wide data sets which usually contain a **large number** of objects and a **very large number** of variables.

_ Is often better to perform calculations in the **PC-space**, rather than in the **original space**.

_ Conceptually the PCT is similar to FT (Fourier Transform):

PCA is performed to create a new domain (PC-space)

FT: time domain → frequency domain
PCT: original domain → PC domain

Calculations are simplified in this new domain

FT: convolution, noise reduction, etc.
PCT: MVA on a smaller set of dimensions (PCs)

Results are back-transformed into the original space

"Inverse FT": frequency domain → time domain.
"Inverse PCT": PC domain → original domain.

_ PCT framework will be shown in:

:: Partial Least Squares regression (PCT-PLS1)

:: Segmented PCT-PLS1

:: Two-Dimensional Correlation Spectroscopy (PCT-2DCOS)

:: Outer-Product PCT-PCA

The motivation

_ **PLS** is one of the most widely used regression techniques.

_ **PLS** is known as a soft-modelling technique.
i.e. no *a priori* assumption is made about the model structure.

_ **PLS** needs a reliable estimation of the predictive ability.
i.e. a major concern is to avoid over- or under-fitting (robustness).

_ **PLS** applied to very wide datasets can make huge demands on computer resources, especially memory.

The model:

$\mathbf{y}_{(n,1)} = \mathbf{X}_{(n,m)}\ \mathbf{b}_{PLS1(m,1)} + \mathbf{f}_{(n,1)}$, where m >> n
(original space)

The PCT approach:

1. Decomposition
   $\mathbf{X}_{(n,m)} = \mathbf{T}_{X(n,\ k)}\ \mathbf{P}^{T}_{X(k,m)} + \mathbf{E}_{(n,m)}$
   (NIPALS or SVD)

2. PCT-PLS
   $\mathbf{y}_{(n,1)} = \mathbf{T}_{X(n,k)}\ \mathbf{b}_{PCT\text{-}PLS1(k,1)} + \mathbf{f}_{(n,1)}$
   (PC space)

   as k << m
   – increase the speed of predictive power assessing

Barros AS, Rutledge DN, "Principal Components Transform-Partial Least Squares regression (PCT-PLS): A novel method to accelerate cross-validation in PLS regression", Chemom. Intell. Lab. Syst., 73, 245-255, 2004.

PCT-PLS1 / PLS1 relationships

PCA	PCT-PLS1	PLS1
$\mathbf{T}_X$	$\mathbf{T}_{PCT\text{-}PLS1}$	$\mathbf{T}_{PLS1}$
$\mathbf{P}_X$	$\mathbf{T}_{yPCT\text{-}PLS1}$	$\mathbf{T}_{yPLS1}$
	$\mathbf{P}_{PCT\text{-}PLS1}$	$\mathbf{P}_{PLS1}$
	$\mathbf{P}_{yPCT\text{-}PLS1}$	$\mathbf{P}_{yPLS1}$
	$\mathbf{W}_{PCT\text{-}PLS1}$	$\mathbf{W}_{PLS1}$
	$\mathbf{b}_{PCT\text{-}PLS1}$	$\mathbf{b}_{PLS1}$

PCA	PCT-PLS1		PLS1
$\mathbf{T}_X$	$\mathbf{T}_{PCT\text{-}PLS1}$	=	$\mathbf{T}_{PLS1}$
$\mathbf{P}_X$	$\mathbf{T}_{yPCT\text{-}PLS1}$	=	$\mathbf{T}_{yPLS1}$
	$\mathbf{P}_X\mathbf{P}^T_{PCT\text{-}PLS1}$	=	$\mathbf{P}_{PLS1}$
	$\mathbf{P}_{yPCT\text{-}PLS1}$	=	$\mathbf{P}_{yPLS1}$
	$\mathbf{P}_X\mathbf{W}^T_{PCT\text{-}PLS1}$	=	$\mathbf{W}_{PLS1}$
	$\mathbf{P}_X\mathbf{b}^T_{PCT\text{-}PLS1}$	=	$\mathbf{b}_{PLS1}$

Matlab code snippet

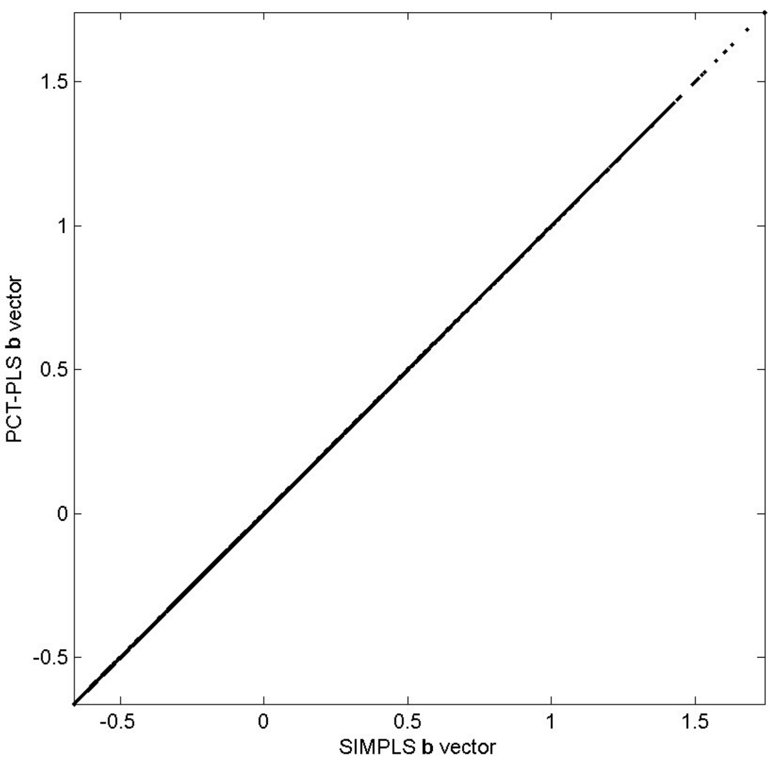
```
X = load('xdata.txt');
y = load('ydata.txt');

[U S V] = SVD( X*X');
T = U * sqrt(S);

% cross-validation : recover the optimal
% number of Latent Variables (lv)
[press lv] = pls1cv( T, y );

% optional - if one needs to look at the
% b coefficients in the original space

lvopt = find( y == min(y));

[bpct b0pct] = pls1( T, y, lvopt);
bpls1 = V * bpct';
```

CV is much faster in $\mathbf{T}_{(n,\ k)}$ than in $\mathbf{X}_{(n,\ m)}$ as m >> k

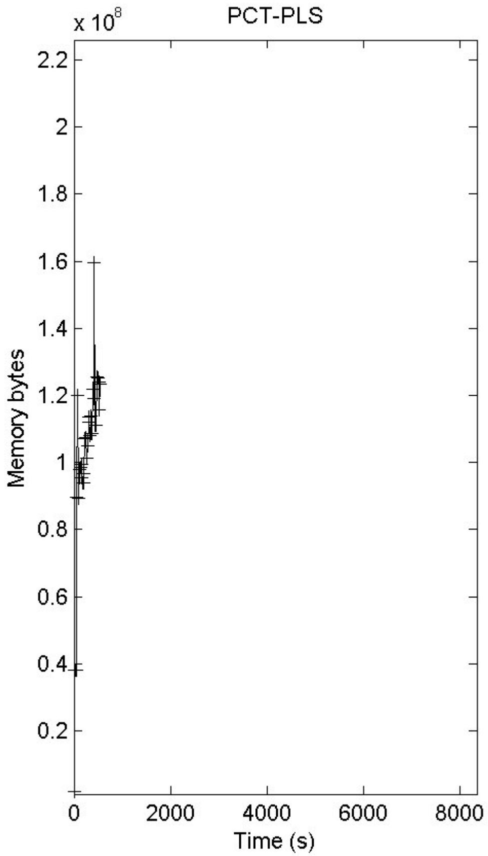
To recover the **b** vector in the original space using the optimal number of LVs

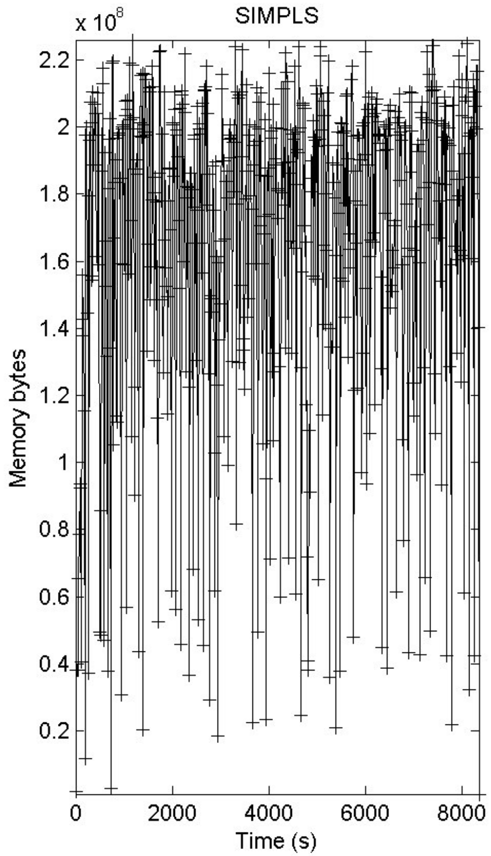
Dataset with 450702 variables

b vectors relationship



Memory allocation profile





(the **b** vectors are the same for both approaches)

_ PCT framework will be shown in:

:: Partial Least Squares regression (PCT-PLS1)

:: Segmented PCT-PLS1

:: Two-Dimensional Correlation Spectroscopy (PCT-2DCOS)

:: Outer-Product PCT-PCA

Let :

$$\mathbf{X}_{(n,\ m)} = \mathbf{T}_{X(n,\ k)}\ \mathbf{P}^{T}_{X(k,\ m)}$$

$$\mathbf{X}_{(n,\ m)} = [\mathbf{X}_{1(n,m1)}|\mathbf{X}_{2(n,m2)}|\ \ldots\ |\mathbf{X}_{q(n,mq)}] =$$
$$= [\mathbf{T}_{1(n,k1)}\mathbf{P}^{T}_{1(k1,m1)}|\mathbf{T}_{2(n,k2)}\ \mathbf{P}^{T}_{2(k2,m2)}|\ldots|\mathbf{T}_{q(n,kq)}\mathbf{P}^{T}_{q(kq,mq)}]$$

where m1 + m2 + ... + mq = m

Concatenating the $\mathbf{T}_q$ matrices:

$$\mathbf{Q}_{(n,\ k1+k2+\ldots+kq)} = [\mathbf{T}_{1(n,k1)}|\mathbf{T}_{2(n,k2)}|\ldots|\mathbf{T}_{q(n,kq)}]$$

Q can be decomposed as:

$$\mathbf{Q}_{(n,\ k1+k2+\ldots+kq)} = \mathbf{T}_{PCT(n,h)}\mathbf{P}^{T}_{PCT(h,\ k1+k2+\ldots+kq)}$$

or

$$\mathbf{Q}_{(n,\ k1+k2+\ldots+kq)} = \mathbf{T}_{PCT(n,h)}\,[\ \mathbf{P}^{T}_{PCT1(h,\ k1)}|\ \mathbf{P}^{T}_{PCT2(h,\ k2)}\ |\ \ldots\ |\ \mathbf{P}^{T}_{PCTq(h,\ kq)}]$$

$\mathbf{Q}_{(n,\ k1+k2+...+kq)} = \mathbf{T}_{PCT(n,h)}\mathbf{P}^{T}_{PCT(h,\ k1+k2+...+kq)}$

PCT-PLS1:

$\mathbf{y}_{(n,1)} = \mathbf{T}_{PCT(n,h)}\ \mathbf{b}_{PCT(h,1)} + \mathbf{f}_{(n,1)}$

- assess model dimensionality
- explore scores, etc.

However…

How to reconstruct the $\mathbf{b}_{PLS}$ vector?

- Following the PCT-PLS one knows that:

$\mathbf{b}_{PLS(m,1)} = \mathbf{P}_{X(m,h)}\ \mathbf{b}_{PCT(h,1)}$

but $\mathbf{P}_X$ can be very wide…

From:

$$\mathbf{Q}_{(n,\ k1+k2+...+kq)} = \mathbf{T}_{PCT(n,h)} [\ \mathbf{P}^{T}_{PCT1(h,\ k1)} \mid \mathbf{P}^{T}_{PCT2(h,\ k2)} \mid \dots \mid \mathbf{P}^{T}_{PCTq(h,\ kq)}]$$

and

$$\mathbf{X}_{(n,\ m)} = [\mathbf{T}_{1(n,k1)}\mathbf{P}^{T}_{1(k1,m1)} \mid \mathbf{T}_{2(n,k2)}\ \mathbf{P}^{T}_{2(k2,m2)} \mid \dots \mid \mathbf{T}_{q(n,kq)}\mathbf{P}^{T}_{q(kq,mq)}]$$

and

$$\mathbf{y}_{(n,1)} = \mathbf{T}_{PCT(n,h)}\ \mathbf{b}_{PCT(h,1)} + \mathbf{f}_{(n,1)}$$

One can shows that:

$$\mathbf{b}_{PLS1(m1,1)} = \mathbf{P}_{1(m1,k1)}\mathbf{P}^{T}_{PCT1(k1,h)}\mathbf{b}_{PCT(h,1)}$$

$$\mathbf{b}_{PLS2(m2,1)} = \mathbf{P}_{2(m2,k2)}\mathbf{P}^{T}_{PCT2(k2,h)}\mathbf{b}_{PCT(h,1)}$$

...

$$\mathbf{b}_{PLSq(mq,1)} = \mathbf{P}_{q(mq,kq)}\mathbf{P}^{T}_{PCTq(kq,h)}\mathbf{b}_{PCT(h,1)}$$

$$\mathbf{b}_{PLS(m,1)} = [\mathbf{b}_{PLS1(m1,1)} \mid \mathbf{b}_{PLS2(m2,1)} \mid \dots \mid \mathbf{b}_{PLSq(mq,1)}]$$

- concatenation by rows

1. To increase the performance of SegPCT:
   instead of

$$\mathbf{X}_{(n,\ m)} = [\mathbf{X}_{1(n,m1)}|\mathbf{X}_{2(n,m2)}|\ \ldots\ |\mathbf{X}_{q(n,mq)}]$$

   one can use the kernel $\mathbf{XX}^T$ approach for each segment:

$$[\ \mathbf{X}_1\mathbf{X}^T{}_1|\ \mathbf{X}_2\mathbf{X}^T{}_2\ |\ \ldots|\ \mathbf{X}_q\mathbf{X}^T{}_q]$$

   to recover the scores and the loadings.

2. The scores and loadings in the original-variable space are reconstructed independently

   as such:

   to assess the model dimensionality the loadings in the original-variable space does not have to be reconstructed.

Matrix size	**PCT-PLS1**		**SegPCT-PLS1**		
	Time (s)	**Memory* (Mbytes)**	**Time (s)**	**Memory* (Mbytes)**	**Segment size**
[100, 100000]	49	39 (65)	106	7.6 (15.4)	1000
[100, 250000]	132	98 (99)	194	5.4 (15.4)	2500
[100, 500000]	298	195 (197)	302	6.3 (15.4)	5000
[100, 750000]	891	221 (292)	413	7.3 (15.4)	7500
[100, 1000000]	2185	220 (391)	518	8.4 (15.4)	10000

(*) Memory values of the working set of the algorithms (usage of the main memory to perform the calculations)
Values between parentheses are due to the amount of allocated virtual memory.

→ For very wide matrices SegPCT-PLS1 is more efficient (speed and memory) than PCT-PLS1.

→ For moderated wide matrices the PCT-PLS1 should be used instead of SegPCT-PLS1.

_ PCT framework will be shown in:

:: Partial Least Squares regression (PCT-PLS1)

:: Segmented PCT-PLS1

:: Two-Dimensional Correlation Spectroscopy (PCT-2DCOS)

:: Outer-Product PCT-PCA

The motivation

_ **2DCOS** is spectral technique for evaluating 2-way datasets obtained when a sample is subject to an external sequential perturbation

_ **2DCOS** detects in-phase (synchronous) and out-of-phase (asynchronous) correlations between spectral intensity variations

_ **2DCOS** emphasises or detect important variations that cannot be detect in the 1D spectrum.

The synchronous spectrum:

$$\mathbf{\Phi}_{(v1,\ v2)}$$

:: represents the similarity between two, v1 and v2, separated spectral intensity variations as a function of the perturbation

The Asynchronous spectrum:

$$\mathbf{\Psi}_{(v1,\ v2)}$$

:: describes the dissimilarity between two, v1 and v2, separated spectral intensity variations as a function of the perturbation

$$\mathbf{\Phi}_{(m,\ m)} = \mathbf{X}^{T}_{(m,\ n)}\ \mathbf{X}_{(n,\ m)}$$

$$\mathbf{\Psi}_{(m,\ m)} = \mathbf{X}^{T}_{(m,\ n)}\ \mathbf{H}_{(n,\ n)}\ \mathbf{X}_{(n,\ m)}$$

$$\mathbf{H}_{jk} = \begin{cases} 0 & \text{if } j = k \\ 1/\pi(k - j) & \text{otherwise} \end{cases}$$

H : Hilbert-Noda transform matrix

The PCT approach for the synchronous spectrum

$\Phi_{(m,\ m)} = \mathbf{X}^{T}{}_{(m,\ n)}\ \mathbf{X}_{(n,\ m)}$

:: decomposition of **X** as $\mathbf{X} = \mathbf{TP}^{T}$

$\Phi_{(m,\ m)} = \mathbf{P}_{(m,\ k)}\ \mathbf{T}^{T}{}_{(k,\ n)}\ \mathbf{T}_{(n,\ k)}\ \mathbf{P}^{T}{}_{(k,\ m)}$

:: pre-multiplying by $\mathbf{P}^{T}$ and post-multiplying by **P** and as $\mathbf{P}^{T}\mathbf{P} = \mathbf{I}$ then:

$\mathbf{P}^{T}{}_{(k,\ m)}\ \Phi_{(m,\ m)}\ \mathbf{P}_{(m,\ k)} = \mathbf{T}^{T}{}_{(k,\ n)}\ \mathbf{T}_{(n,\ k)}$

:: or

$\Phi_{PCT(m,\ m)} = \mathbf{T}^{T}{}_{(k,\ n)}\ \mathbf{T}_{(n,\ k)}$ → one could perform 2DCOS on the scores

$\Phi_{(m,\ m)} = \mathbf{P}_{(m,\ k)}\ \Phi_{PCT(m,\ m)}\ \mathbf{P}^{T}{}_{(k,\ m)}$ → 2DCOS in the original space

The PCT approach for the asynchronous spectrum

$\Psi_{(m,\ m)} = \mathbf{X}^{T}_{(m,\ n)}\ \mathbf{H}_{(n,\ n)}\ \mathbf{X}_{(n,\ m)}$

:: decomposition of **X** as $\mathbf{X} = \mathbf{T}\mathbf{P}^{T}$

$\Psi_{(m,\ m)} = \mathbf{P}_{(m,\ k)}\ \mathbf{T}^{T}_{(k,\ n)}\ \mathbf{H}_{(n,\ n)}\mathbf{T}_{(n,\ k)}\ \mathbf{P}^{T}_{(k,\ m)}$

:: pre-multiplying by $\mathbf{P}^{T}$ and post-multiplying by **P** and as $\mathbf{P}^{T}\mathbf{P} = \mathbf{I}$ then:

$\mathbf{P}^{T}_{(k,\ m)}\ \Phi_{(m,\ m)}\ \mathbf{P}_{(m,\ k)} = \mathbf{T}^{T}_{(k,\ n)}\ \mathbf{H}_{(n,\ n)}\ \mathbf{T}_{(n,\ k)}$

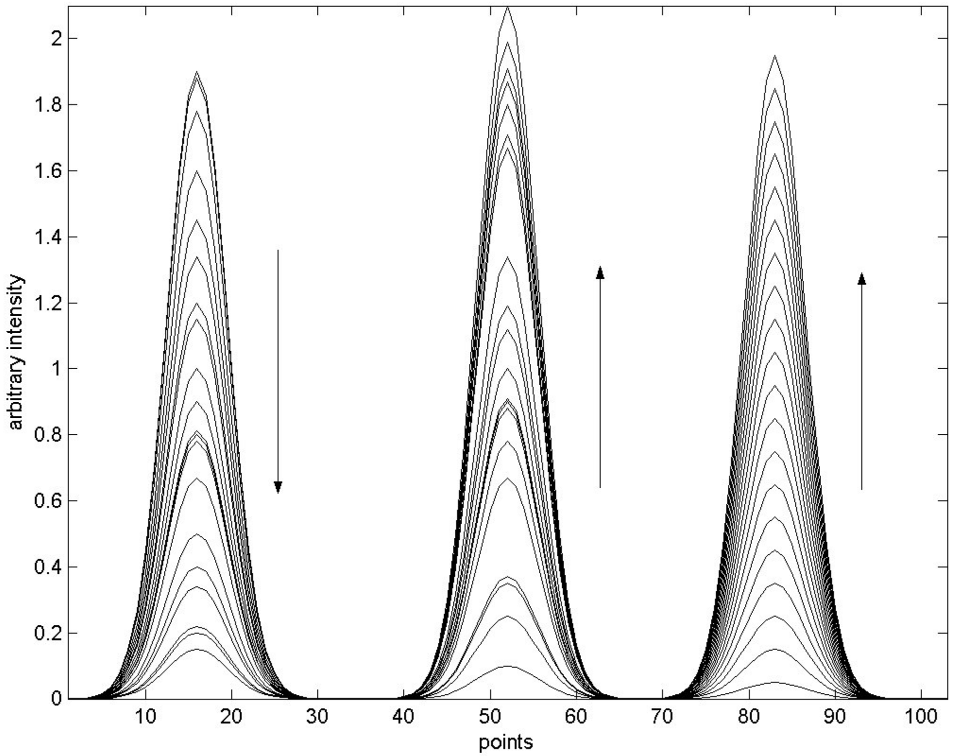
:: or

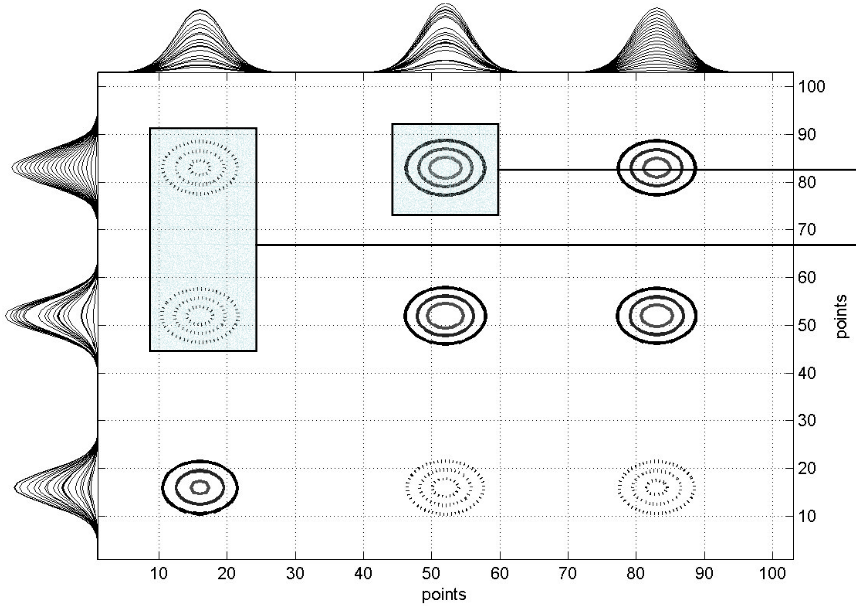
$\Psi_{PCT(m,\ m)} = \mathbf{T}^{T}_{(k,\ n)}\ \mathbf{H}_{(n,\ n)}\ \mathbf{T}_{(n,\ k)}$ → one could perform 2DCOS on the scores

$\Psi_{(m,\ m)} = \mathbf{P}_{(m,\ k)}\ \Psi_{PCT(m,\ m)}\ \mathbf{P}^{T}_{(k,\ m)}$ → 2DCOS in the original space

Simulated dataset

:: band 17 decreases at a given rate
:: bands 53 and 83 increases at different rates





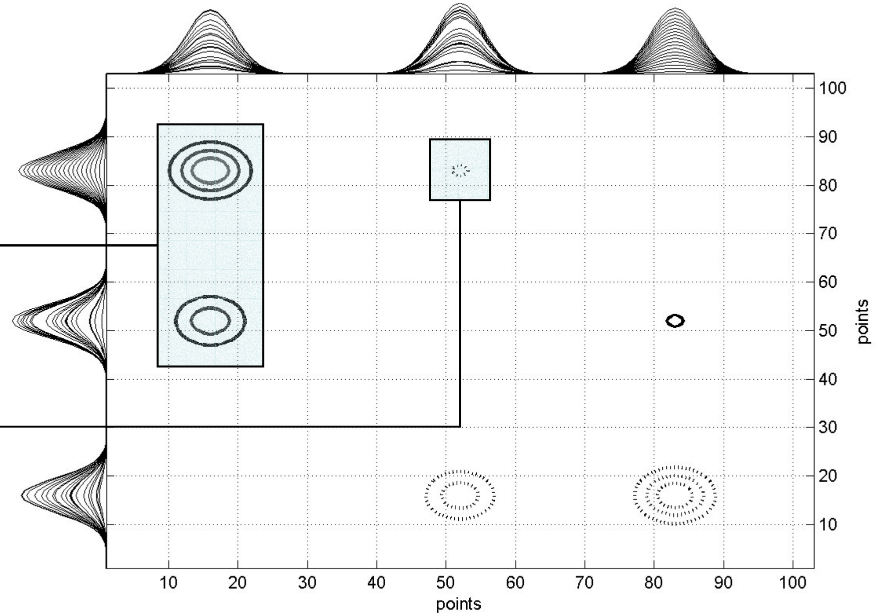
band 53 change in the same direction as band 83

band 17 change in the opposite direction of bands 53 and 83

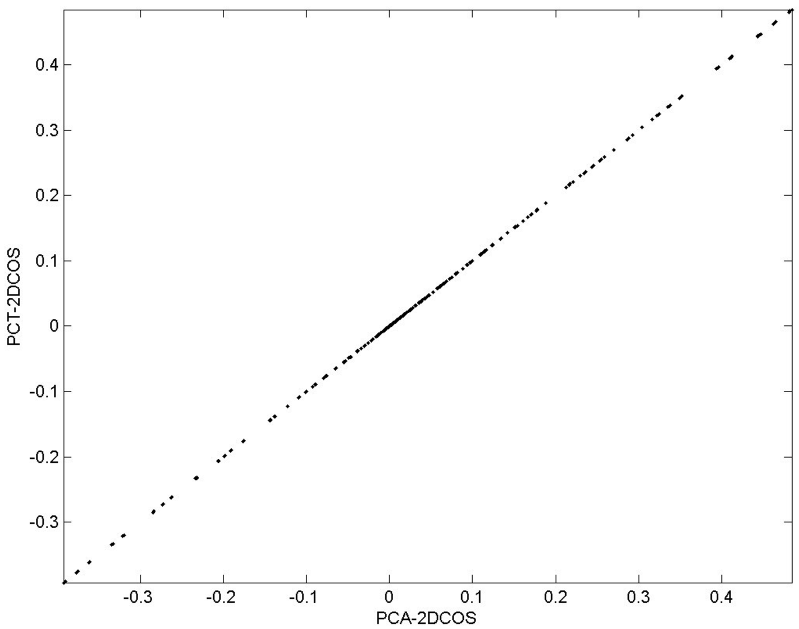
Synchronous spectrum

band 17 occurs predominantly after intensity variations of bands 53 and 83.

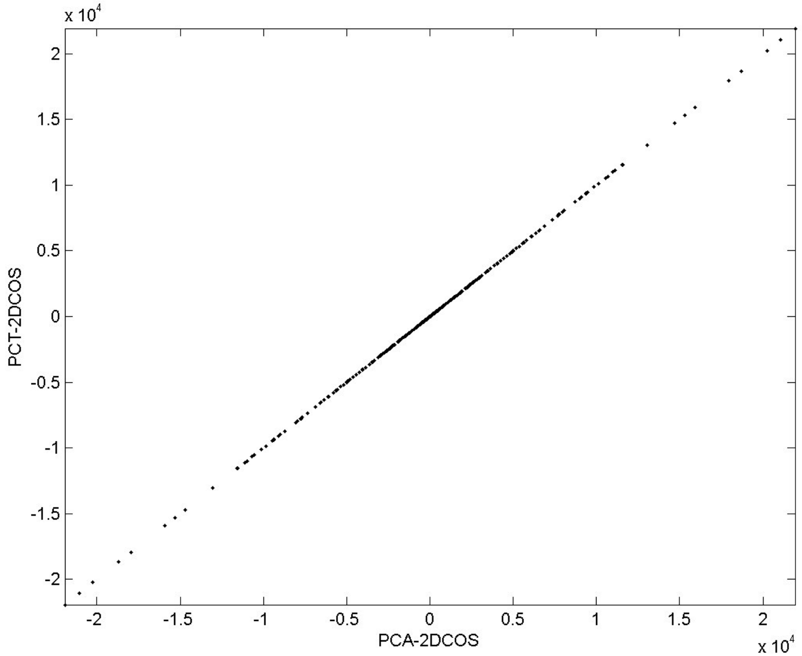
band 53 occurs predominantly after intensity variations of band 83.



Asynchronous spectrum



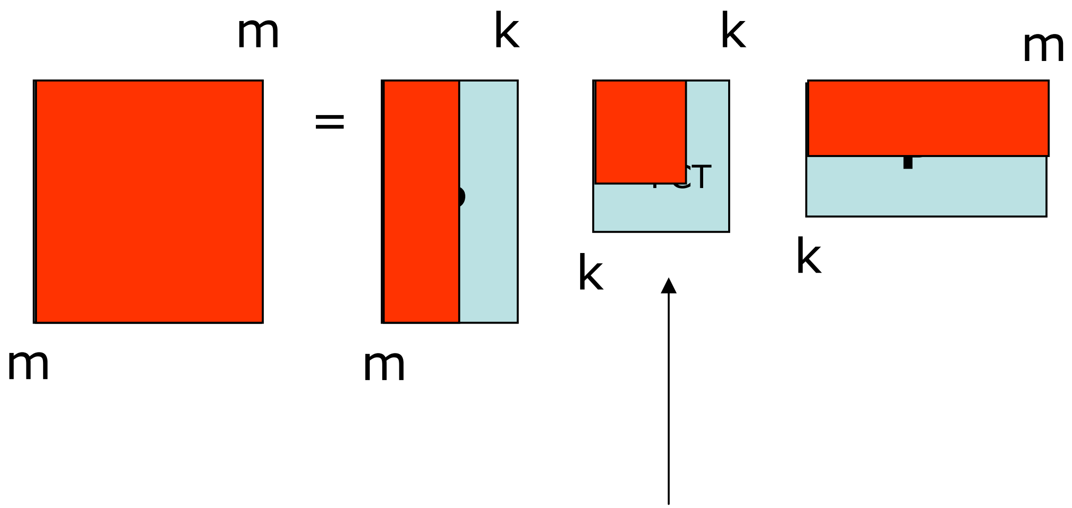
Relationship between synchronous spectra of PCT-2DCOS and PCA-2DCOS



Relationship between synchronous spectra of PCT-2DCOS and PCA-2DCOS

:: PCT-2DCOS allows to build in an interactive way the 2DCOS spectra

For the Asynchronous spectrum:



Just need to calculate this once

_ PCT framework will be shown in:

:: Partial Least Squares regression (PCT-PLS1)

:: Segmented PCT-PLS1

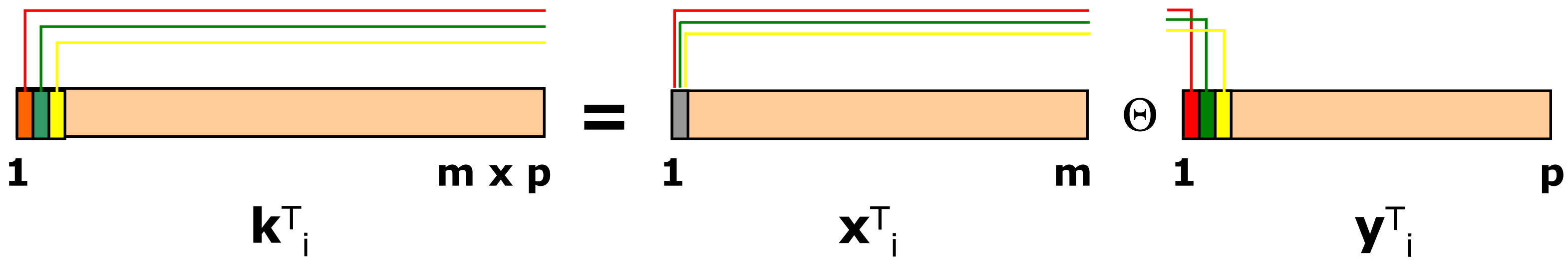
:: Two-Dimensional Correlation Spectroscopy (PCT-2DCOS)

:: Outer-Product PCT-PCA

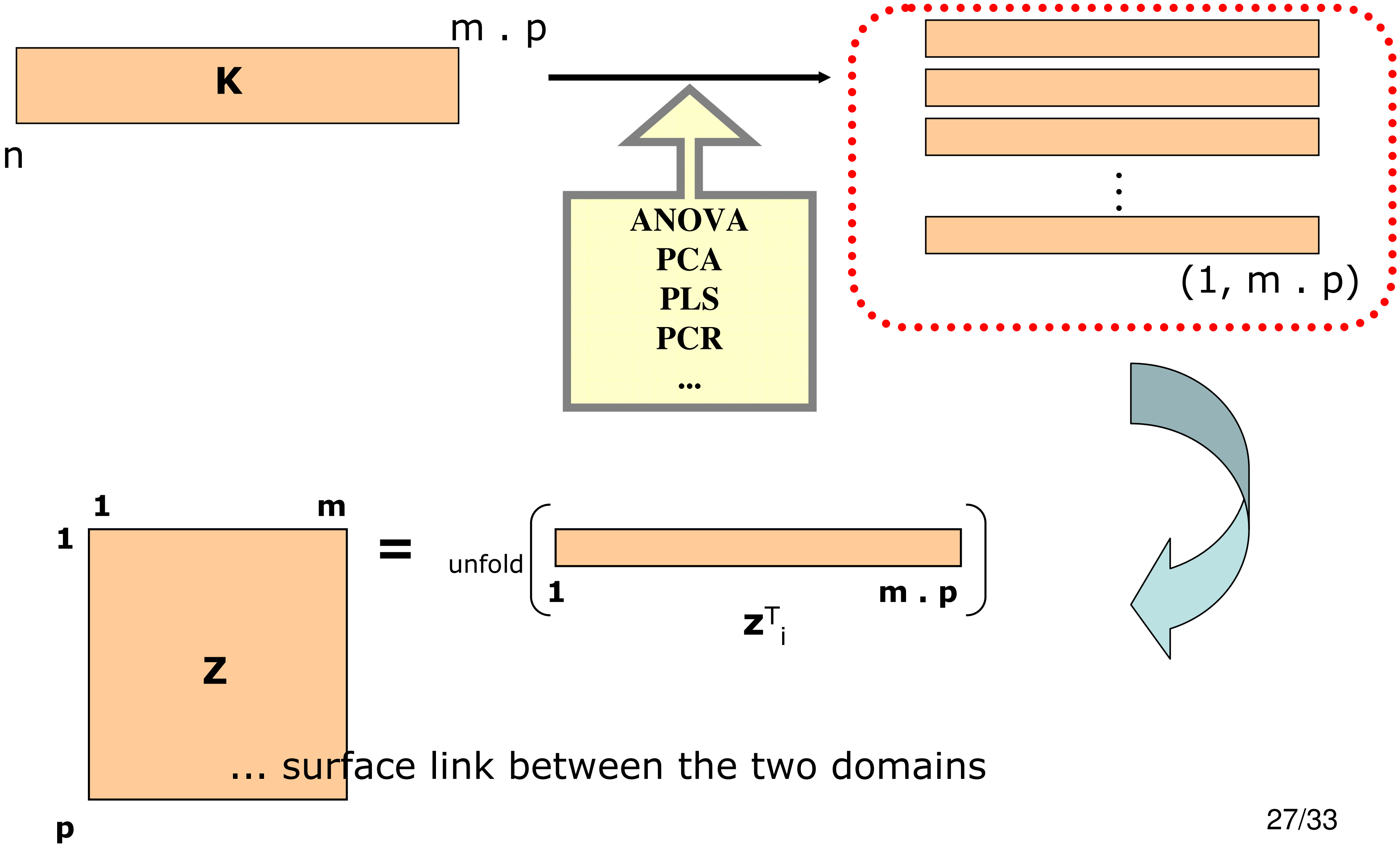
:: The method joins the signals acquired in two different domains by the means of Cartesian product combination between all the variables (points) of both signals.

:: The obtained supra-matrix (**K**) is calculated from the original signal matrices which contains all the information provided by both independent domains.

$$\mathbf{K}_{(n,\ m\ .\ p\ )} = \mathbf{X}_{(n,\ m)}\ \Theta\ \mathbf{Y}_{(n,\ p)}$$



where:
n : number of samples
m : number of variables of domain X
p : number of variables of domain Y
Θ : Outer-Product operator



... surface link between the two domains

:: This technique can produce very wide datasets, which can be very difficult to analyse due to computer resource constraints.

:: Therefore, instead of working in the original-variable space (**K**) one can work in the compressed PC-space (PCT).

- one does not need to calculate the **K** explicitly

PCA decomposition of both domains:

$$\mathbf{X}_{(n,\ m)} = \mathbf{T}_{X(n,\ kX)}\ \mathbf{P}^{T}{}_{X(kX,\ m)}$$

$$\mathbf{Y}_{(n,\ p)} = \mathbf{T}_{Y(n,\ kY)}\ \mathbf{P}^{T}{}_{Y(kY,\ p)}$$

Outer Product of the Scores:

$$\mathbf{Q}_{(n,\ kX.\ KY)} = \mathbf{T}_{X(n,\ kX)}\ \Theta\ \mathbf{T}_{Y(n,\ kY)}$$

PCA decomposition of the **Q** matrix (PCT framework):

$$\mathbf{Q}_{(n,\ kX\ .\ kY)} = \mathbf{T}_{PCT(n,\ h)}\ \mathbf{P}^{T}{}_{PCT(h,\ kX\ .\ kY)}$$

Matrix **Q** is much smaller than matrix **K** as (kX . kY) << (m . p)

From the PCT properties it follows that:

$\mathbf{T}_{PCT} = \mathbf{T}_{K}$ $\therefore$ the PCT scores are equal to the scores of the original-variable space (**K**)

For each one of the h PCT-PCs the $\mathbf{P}_{PCT(kX\ .\ kY,\ h)}$ matrix is unfolded as:

$$\mathbf{P}_{PCT1(kX,\ kY)} \leftarrow \mathbf{P}_{PCT(kX\ .\ kY,1)}$$
$$\mathbf{P}_{PCT2(kX,\ kY)} \leftarrow \mathbf{P}_{PCT(kX\ .\ kY,2)}$$
$$\ldots$$
$$\mathbf{P}_{PCTh(kX,\ kY)} \leftarrow \mathbf{P}_{PCT(kX\ .\ kY,h)}$$

to obtain the loadings in the original-variable space:

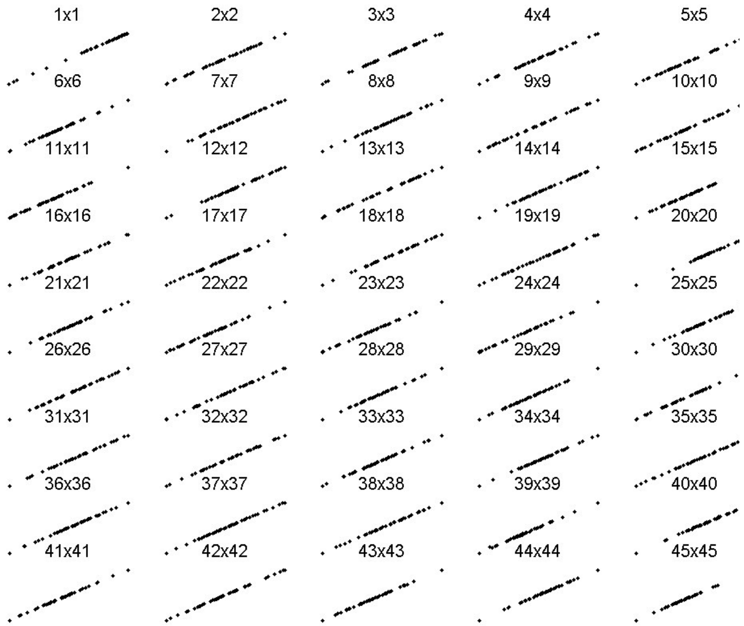
$$\mathbf{P}_{1(m,\ p)} = \mathbf{P}_{X(m,\ kX)}\mathbf{P}_{PCT1(kX,kY)}\mathbf{P}^{T}_{Y(kY,\ p)}$$
$$\mathbf{P}_{2(m,\ p)} = \mathbf{P}_{X(m,\ kX)}\mathbf{P}_{PCT2(kX,kY)}\mathbf{P}^{T}_{Y(kY,\ p)}$$
$$\ldots$$
$$\mathbf{P}_{h(m,\ p)} = \mathbf{P}_{X(m,\ kX)}\mathbf{P}_{PCTh(kX,kY)}\mathbf{P}^{T}_{Y(kY,\ p)}$$

$\mathbf{P}_1$, $\mathbf{P}_2$, …, $\mathbf{P}_h$ are folded-back to:

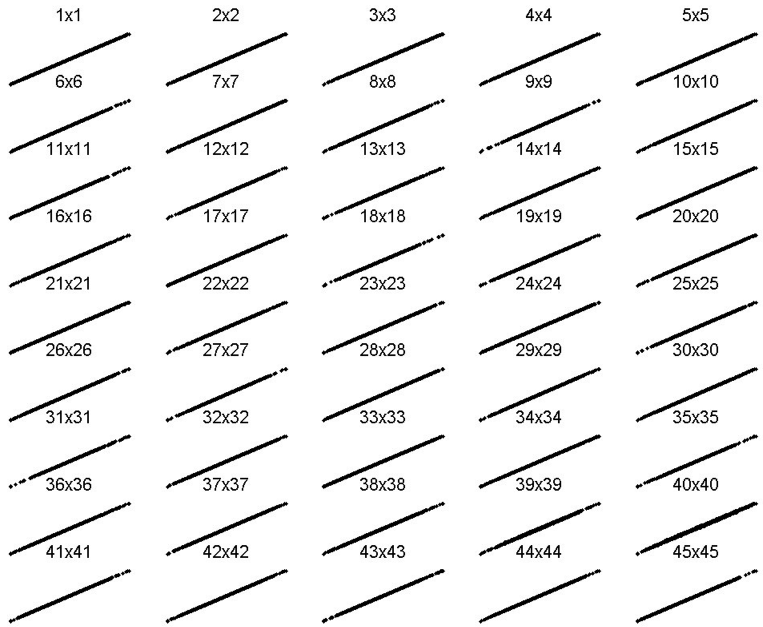
$$\mathbf{P}_{1(m.p,\ 1)},\ \mathbf{P}_{2(m.p,2)},\ \ldots,\ \mathbf{P}_{h(m.p,h)}$$

and concatenated: $\mathbf{P} = [\ \mathbf{P}_1\ |\ \mathbf{P}_2\ |\ \ldots\ |\ \mathbf{P}_h\ ]$

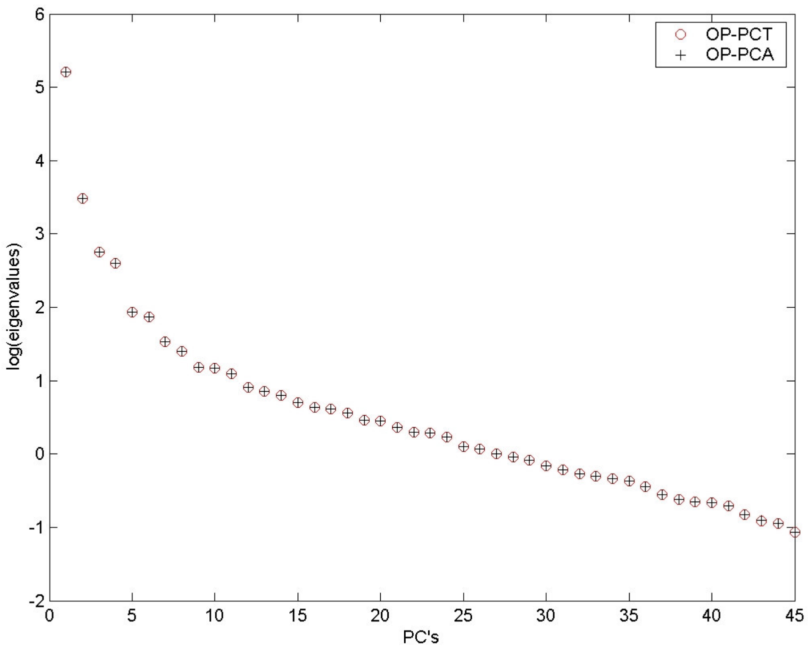
:: OP-PCA decomposition of a $\mathbf{K}_{(45,\ 490625)}$ matrix was compared to the PCT-OP-PCA decomposition of the $\mathbf{Q}_{(45,45*45)}$



Scores scatter plot between the PCs of OP-PCA and PCT-OP-PCA.



Loadings scatter plot between the PCs of OP-PCA and PCT-OP-PCA.



Eigenvalues profiles for
OP-PCA and PCT-OP-PCA

→ OP-PCA took ***324*** s and ***147*** Mbytes.

→ PCT-OP-PCA took **196** s and around **1** Mbyte.

→ The scores, the loadings and the eigenvalues of both approaches are equal.

- The PCT framework seems to be very useful in several MVA contexts.

- The PCT injection into the MVA methods is straightforward.

- The PCT framework allows interactive approaches for modelling.

- The PCT is inherently parallel (for distributed computing)