Computation on Partitionned Matrices in Chemometrics and Sensometrics

Mohamed Hanafi & David Legland

mohamed.hanafi@oniris-nantes.fr
david.legland@nantes.inra.fr

École Nationale Vétérinaire, Agroalimentaire et de l'Alimentation Nantes Atlantique

Institut National de la Recherche Agronomique

SUMMARY

- Motivation and aims.
- Issues and contributions.
- Conclusions and perspectives.

1.1. Foreword

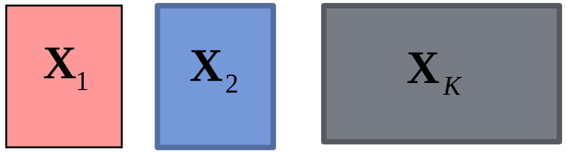
- Does not consider chemometrics (sensometrics) from its purpose (Extract knowledge from data) but its two main components (data and methods)
- The data sets are conceptualized as matrices and methods use intensively matrix algebra.
- The matrix algebra is an essential component of chemometrics (sensometrics).

1.2. Starting point…

- Increasingly, the matrices involved in chemometrics (sensometrics) are associated with one or more partitions.

- The partitions are an important aspect of data and methods in chemometrics (sensometrics)

  - The partition can be an important part of the data sets themselves. (Example multiblock data)
  - The partition can be a part of the method (clustering methods)
  - The partition can be latent which emerged during the implementation of algorithms (eg. Cross-validation procedure).



Multiblocs data

Incomplete data

1.3. Hypothesis and Aims

THE TREE THAT HIDES THE FOREST

tree is matrix algebra.
the forest is partitioned matrix algebra.

Aims

- To formalize a framework for computing on partitioned matrices.
- To introduce rules and concepts to perform computation on partitioned matrices.
- To study the contribution of this framework in chemometrics (sensometrics).

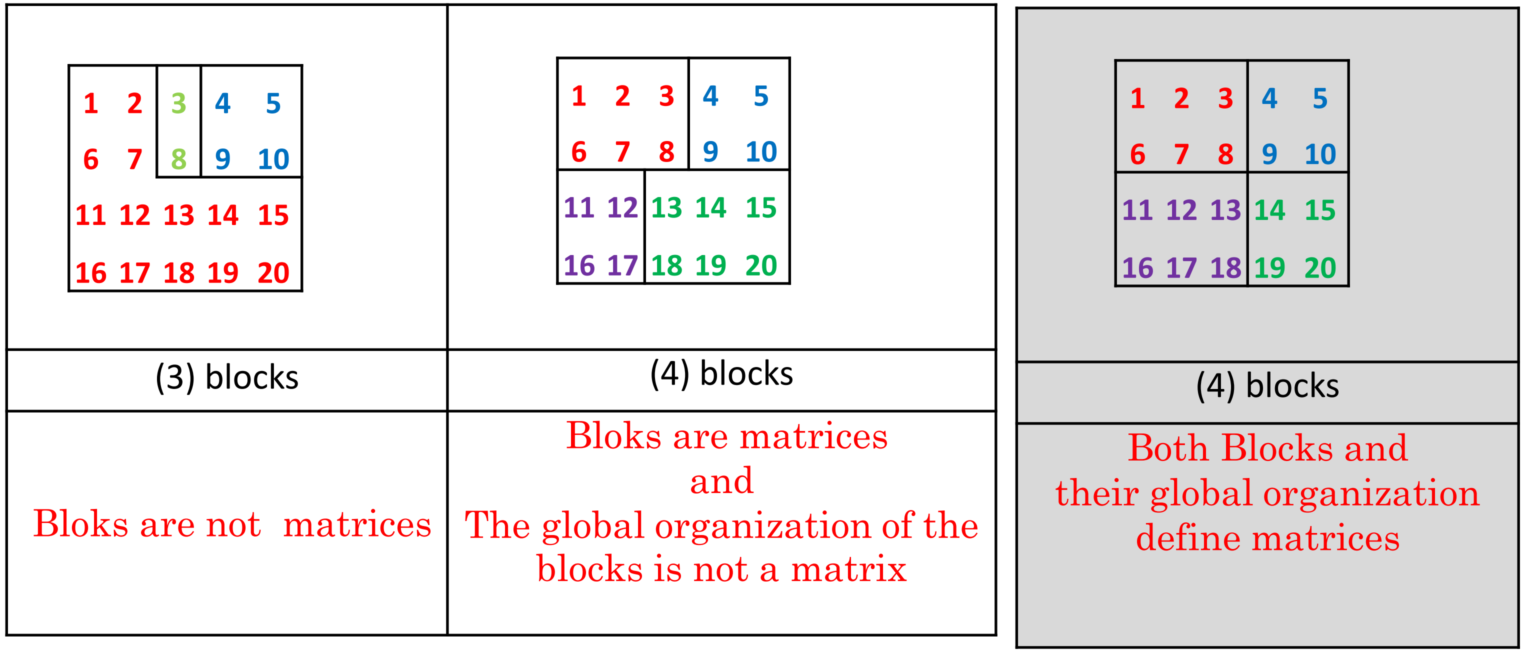
Interests.

- A framework that provides powerful tools for fast prototyping algorithms.
- A framework that offers effective tools to disseminate and implement chemometrics (sensometrics) algorithms and methods.
- A framework inspired by chemometrics (sensometrics) for new developments in mathematics

2.1. Issue (1)

How to choose an appropriate class from several kinds of partitionned matrices ?

- Partitionned Matrix= Matrix + Partition of the positions of its elements.
- Block-matrices are partitioned matrices where the partition of its elements can be described as a partition on the rows or columns or both at once.



Several kinds of partitionned matrices

$$\mathbf{X} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \\ 16 & 17 & 18 & 19 & 20 \end{bmatrix}$$

2.2. Contribution (1)

Definitions and Notations for block-Matrices

- Partition of an integer.
  - A partition on an integer *n* in *k* parts is a decomposition of *n* as : $n = \sum_{i=1}^{k} \tau_i \quad (\tau_i \neq 0)$
  - $\mathbf{P}_k(n)$ denotes the set of partitions of n in k parts.

- Partition of a pair of integers.
  - A partition of a pair of integers *(n,p)* is an element of $\mathbf{P}_{(k,\bar{k})}(n,p) = \mathbf{P}_k(n) \times \mathbf{P}_{\bar{k}}(p)$

((4),(7)),
((1,1,1,1),(7)),
((1,1,1,1),(1,1,1,1,1,1,1)),
((2,2),(2,3,2)),

Some partitions of (4,7)

- Block-Matrix.
  - An element of $\tilde{\mathbf{M}}_{\mathbf{P}_{(k,\bar{k})}(n,p)} = \mathbf{M}_{(n,p)} \times \mathbf{P}_{(k,\bar{k})}(n,p)$ is called block-matrix
  - A block-matrix is a pair of a matrix and a partition of its dimension.
  - A partition of the dimension of a matrix is called Block-dimension.

  - Notation for block-matrix $\underbrace{\mathbf{X}}_{(P_k(n),P_{\bar{k}}(p))}$

2.3. Issue (2)

How to handle block-matrix on scientific software (here R or Matlab)?

BlockMatrix toolbox

David Legland.

https://github.com/dlegland/BlockMatrixToolbox

BlokBerry Package

Gaston Sanchez

https://github.com/gastonstat/blockberry

Contribution (2) : Two toolboxes for Block-Matrices.

```
>> A = BlockMatrix(reshape(1:20, [5 4]), {[3 2], [2 2]})

A =

BlockMatrix object with 5 rows and 4 columns
  row dims: 3 2
  col dims: 2 2

        1           6           11          16
        2           7           12          17
        3           8           13          18
        4           9           14          19
        5          10           15          20

>> getMatrix(A)

ans =

     1     6    11    16
     2     7    12    17
     3     8    13    18
     4     9    14    19
     5    10    15    20

>> blockDimensions(A)

ans =

BlockDimensions object with 2 dimensions
( (3, 2), (2, 2) )

>>
```

1	6	11	16
2	7	12	17
3	8	13	18
4	9	14	19
5	10	15	20

Working with a block-matrix

```
>> rowBlock=blockDimensions(A,1)

rowBlock =

IntegerPartition object with 2 terms
     (3,  2)
>> colBlock=blockDimensions(A,2)

colBlock =

IntegerPartition object with 2 terms
     (2,  2)
>> size(A)

ans =

     5     4

>> blockNumber(A)

ans =

     4

>> blockSize(A)

ans =

     2     2
```

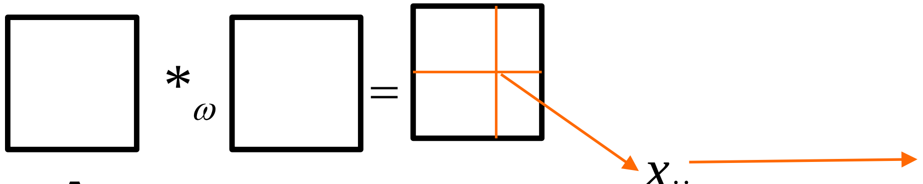
Sizes of block-matrices

```
A =

BlockMatrix object with 5 rows and 4 columns
  row dims: 3 2
  col dims: 2 2

        1            6           11           16
        2            7           12           17
        3            8           13           18
        4            9           14           19
        5           10           15           20

>> reveal(A)
     2  2
  3  +  +
  2  +  +
>> A11=A{1,1}

A11 =

     1     6
     2     7
     3     8

>> A22=A{2,2}

A22 =

    14    19
    15    20

>> getBlock(A,1,1)

ans =

     1     6
     2     7
     3     8
```

Almost basic manipulation operations have been extended to block-matrices

2.4. Issue (3)

To what extend matrix product can be extended to block matrix ?

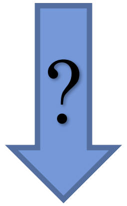


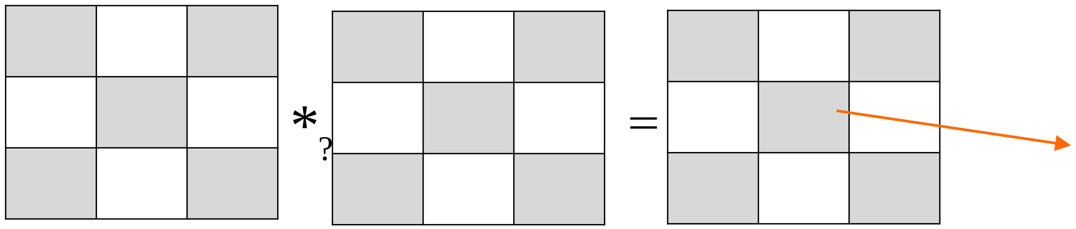
$$\underbrace{\mathbf{A}}_{(n_\mathbf{A},\, p_\mathbf{A})} *_\omega \underbrace{\mathbf{B}}_{(n_\mathbf{B},\, p_\mathbf{B})} = \underbrace{\mathbf{X}}_{(n_\mathbf{X}\, p_\mathbf{X})}$$

x_{ij}

$*_\mathrm{s}$	$x_{ij} = ab_{ij}$
$*_\mathrm{h}$	$x_{ij} = a_{ij} b_{ij}$
$*_\mathrm{u}$	$x_{ij} = \sum_{k=1}^{k=p_\mathbf{A}} a_{ik} b_{kj}$

What is the appropriate notation?
How to define the Block result ?
What are the validation rules for these products ?
What'is the Block-dimension of the result ?





$\mathbf{X}_{ij} = ?$

$$\underbrace{\mathbf{A}}_{\left(P_{k_\mathbf{A}}(n_\mathbf{A}), P_{\bar{k}_\mathbf{A}}(p_\mathbf{A})\right)} *_? \underbrace{\mathbf{B}}_{\left(P_{k_\mathbf{B}}(n_\mathbf{B}), P_{\bar{k}_\mathbf{B}}(p_\mathbf{B})\right)} = \underbrace{\mathbf{X}}_{\left(P_{k_\mathbf{X}}(n_\mathbf{X}), P_{\bar{k}_\mathbf{X}}(p_\mathbf{X})\right)} = ?$$

2.5. Contribution (3)

Introduce and unify (9+6=15) products for block-matrices

$$\underbrace{\mathbf{X}}_{\left(P_{k_{\mathbf{X}}}(n_{\mathbf{X}}),P_{\bar{k}_{\mathbf{X}}}(p_{\mathbf{X}})\right)} = \underbrace{\mathbf{A}}_{\left(P_{k_{\mathbf{A}}}(n_{\mathbf{A}}),P_{\bar{k}_{\mathbf{A}}}(p_{\mathbf{A}})\right)} *_{(\omega_1,\omega_2)} \underbrace{\mathbf{B}}_{\left(P_{k_{\mathbf{B}}}(n_{\mathbf{B}}),P_{\bar{k}_{\mathbf{B}}}(p_{\mathbf{B}})\right)}$$

Notation	Block definition	remark
$*_{(s,s)}$	$\mathbf{X}_{ij} = a_{11} *_s \mathbf{B}_{ij}$	Well known
$*_{(s,h)}$	$\mathbf{X}_{ij} = \mathbf{A}_{11} *_h \mathbf{B}_{ij}$	New
$*_{(s,u)}$	$\mathbf{X}_{ij} = \mathbf{A}_{11} *_u \mathbf{B}_{ij}$	M. Günther, and L. Klotz (2012)
$*_{(s,k)}$	$\mathbf{X}_{ij} = \mathbf{A}_{11} *_k \mathbf{B}_{kj}$	Koning Neudecker, Wansbeek(1991)
$*_{(h,s)}$	$\mathbf{X}_{ij} = a_{ij} *_s \mathbf{B}_{ij}$	New
$*_{(h,h)}$	$\mathbf{X}_{ij} = \mathbf{A}_{ij} *_h \mathbf{B}_{ij}$	Well known
$*_{(h,u)}$	$\mathbf{X}_{ij} = \mathbf{A}_{ij} *_u \mathbf{B}_{ij}$	Horn, Mathias and Nakamura (1991)
$*_{(h,k)}$	$\mathbf{X}_{ij} = \mathbf{A}_{ij} *_k \mathbf{B}_{ij}$	R. A. Horn et al (1992)
$*_{(u,s)}$	$\mathbf{X}_{ij} = \sum_{k=1}^{k_A} a_{ik} *_s \mathbf{B}_{kj}$	New

Notation	Names	remark
$*_{(u,h)}$	$\mathbf{X}_{ij} = \sum_{k=1}^{k_A} \mathbf{A}_{ik} *_h \mathbf{B}_{kj}$	New
$*_{(u,u)}$	$\mathbf{X}_{ij} = \sum_{k=1}^{k_A} \mathbf{A}_{ik} *_u \mathbf{B}_{kj}$	Well known
$*_{(u,k)}$	$\mathbf{X}_{ij} = \sum_{k=1}^{k_A} \mathbf{A}_{ik} *_k \mathbf{B}_{kj}$	W. De Launey and J. Seberry (1994)
$*_{(k,s)}$	$\mathbf{X}_{ij} = a_{11} *_s \mathbf{B}_{ij}$	= kronecker
$*_{(k,h)}$	$\mathbf{X}_{ij} = \mathbf{A}_{ij} *_h \mathbf{B}_{11}$	⇔ $*_{(s,h)}$
$*_{(k,u)}$	$\mathbf{X}_{ij} = \mathbf{A}_{ij} *_u \mathbf{B}_{11}$	⇔ $*_{(s,u)}$

Command Window

New to MATLAB? See resources for Getting Started.

```
A =

BlockMatrix object with 5 rows and 4 columns
  row dims: 3 2
  col dims: 2 2

        1           6          11          16
        2           7          12          17
        3           8          13          18
        4           9          14          19
        5          10          15          20

>> B = BlockMatrix(reshape(1:12, [4 3]), {[2 2], [2 1]})

B =

BlockMatrix object with 4 rows and 3 columns
  row dims: 2 2
  col dims: 2 1

        1           5           9
        2           6          10
        3           7          11
        4           8          12

>> X = blockProduct_hu(A,B)

X =

BlockMatrix object with 5 rows and 3 columns
  row dims: 3 2
  col dims: 2 1

       13          41         259
       16          52         278
       19          63         297
       48         100         382
       55         115         405

>>
```

A

The 15 products are available in Block Matrix Toolbox

B

$$\mathbf{X} = \mathbf{A} *_{(h,u)} \mathbf{B}$$

```
Command Window
New to MATLAB? See resources for Getting Started.

>> A = BlockMatrix(reshape(1:36, [6 6]), {[3 3], [2 2 2]})

A =

BlockMatrix object with 6 rows and 6 columns
  row dims: 3 3
  col dims: 2 2 2

        1            7           13           19           25           31
        2            8           14           20           26           32
        3            9           15           21           27           33
        4           10           16           22           28           34
        5           11           17           23           29           35
        6           12           18           24           30           36

>> B = BlockMatrix(magic(6), {[3 3], [2 2 2]})

B =

BlockMatrix object with 6 rows and 6 columns
  row dims: 3 3
  col dims: 2 2 2

       35            1            6           26           19           24
        3           32            7           21           23           25
       31            9            2           22           27           20
        8           28           33           17           10           15
       30            5           34           12           14           16
        4           36           29           13           18           11

>> X = blockProduct_hh(A, B)

X =

BlockMatrix object with 6 rows and 6 columns
  row dims: 3 3
  col dims: 2 2 2

       35            7           78          494          475          744
        6          256           98          420          598          800
       93           81           30          462          729          660
```

A

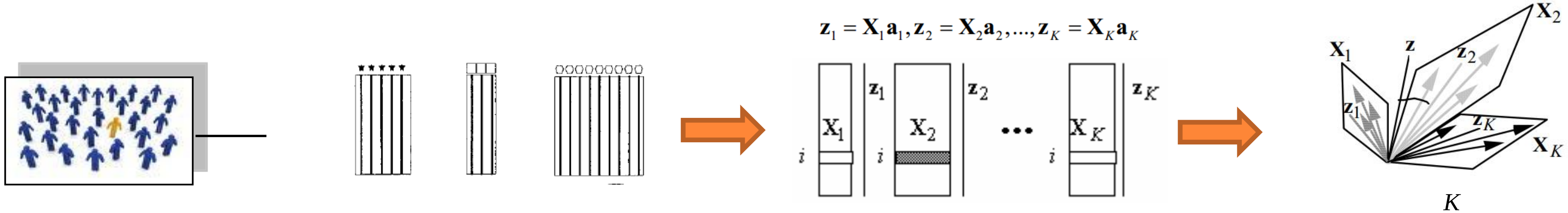
B

The 15 products are available in Block Matrix Toolbox

$$\mathbf{X} = \mathbf{A} *_{(h,h)} \mathbf{B}$$

2.6. Issue (4) how the framework is useful ?

Un example : fast prototyping algorithms for multiblock methods



$$\max_{\|\mathbf{a}_k\|=\|\mathbf{z}\|=1} \sum_{k=1}^{K} \langle \mathbf{z}_k, \mathbf{z} \rangle$$

(MaxBet)

$$\max_{\|\mathbf{a}_k\|=1} \sum_{k,l=1}^{K} \operatorname{cov}(\mathbf{X}_k\mathbf{a}_k, \mathbf{X}_k\mathbf{a}_k) \Leftrightarrow \max_{\|\mathbf{a}_k\|=\|\mathbf{z}\|=1} \sum_{k=1}^{K} \operatorname{cov}(\mathbf{X}_k\mathbf{a}_k, \mathbf{z})$$

Notation based on matrix algebra	Notation based on Block-matrix algebra
$\mathbf{a}_k^{(s+1)} = \dfrac{\sum_{l=1}^{K} \mathbf{X}_l^{'}\mathbf{X}_k\mathbf{a}_k^{(s)}}{\left\| \sum_{l=1}^{K} \mathbf{X}_l^{'}\mathbf{X}_k\mathbf{a}_k^{(s)} \right\|} (1 \le k \le K)$	$\mathbf{A} = (\mathbf{X}^{'} *_{(u,u)} \mathbf{X})$ $\mathbf{a}^{(s+1)} = \left[\left\| \mathbf{A} *_{(h,u)} \mathbf{a}^{(s)} \right\|^{-1_h} \right] *_{(h,s)} \left[\mathbf{A} *_{(h,u)} \mathbf{a}^{(s)} \right] (1 \le k \le K)$

1. Van de Geer, J.P. (1984). Linear relations among *k sets of variables*. Psychometrika*, 49, 79–94.*
2. Hanafi,M., Ten Berge, J. (2003). Psychometrika ,vol. 68, NO. 1, 97–103.

Without BlockMatrix Toolbox

```
function [q,init,lamda]= maxbet(data,vec,t,tol)
a=data'*data
k=length(vec) ;%nombre de blok%
[n,p]=size(a);%colonne de xx
i1=0;
i2=0;
lamda=zeros(k,1);
%t=rand(p,1);
   i1=0;
   i2=0;
    for i=1:k,
        i1=i1+1;
        i2=i2+vec(i);
        t(i1:i2,:)=t(i1:i2,:)/norm(t(i1:i2,:));
        i1=i2;
        end
init=t;
   residu=1;
   q=zeros(p,1);
   while (residu>seuil)
        y=a*t;
        i1=0;
    i1=0;
        i2=0;
    for i=1:k,
        i1=i1+1;
        i2=i2+vec(i);
        q(i1:i2,:)=y(i1:i2,:)/norm(y(i1:i2,:));
        i1=i2;
        end
     residu=abs(t'*a*t-q'*a*q);
      t=q;
    end
  i1=0;
  i2=0;
  for i=1:k,
      i1=i1+1;
      i2=i2+vec(i);
      lamda(i,:)=q(i1:i2,:)'*a(i1:i2,:)*q;
      i1=i2;
```

With BlockMatrix Toolbox

```
function [q, iter, resid] = maxbet_procedure2(data, tt, tol)
%MAXBET_PROCEDURE2  MAXBET procedure for multi-block matrices.

% create new BlockMatrix representing the normalized input vectors
qq = blockProduct_hs(1./blockNorm(tt), tt);

AA = blockProduct_uu(data',data);

resid = 1;

iter = 0;

while resid > tol

    iter = iter + 1;

    q = blockProduct_uu(AA,qq);

    q = blockProduct_hs(1./blockNorm(q), q); % block normalization

    resid = norm(blockNorm(q) - blockNorm(qq)); % residual

    qq = q;
end
```

3.1. Conclusions

- Clarify the definition and vocabulary for block matrices.
  - What is a block matrices and what it is not ?
- Introduce the block-dimension of the block-matrices.
  - How to describe the size of a block-matrix?
- Introduce and unify (15) products for the block-matrices.
- Mount connections examples of the proposed framework with chemometrics.
  - Does this framework is useful for chemometrics?
- Provide operational solution for fast prototyping algorithms that involve block-matrices. (Packages)
  - How to handle and work with block-matrices under R or Matlab?

3.2. Persepctives

- There are, at least, 15 different ways to extend intrinsically PCA to block-matrices

$$\mathbf{X} = \mathbf{A} *_{(\omega_1,\omega_2)} \mathbf{B} + \mathbf{R}$$

- In practice, some of the 15 extensions can be relevent in multiblock data analysis perspective

- Identify these extensions is the main perspective of this work.