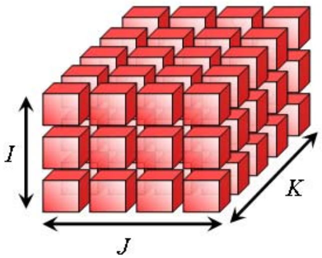
Three-way PCA

Jorge Tendeiro

University of Groningen

25 March 2010 / JOCLAD



Idea

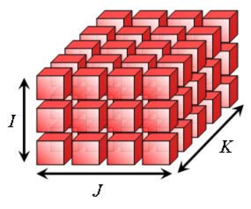
- three-way arrays: generalize matrix structure to 3D
- loaf-of-bread structure

Examples of three-way data

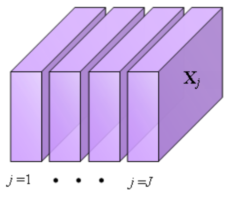
- different anxiety measures, different circumstances, various subjects
- sales of different products, in different shops, in different weeks
- job requirements for various jobs, according to various job analysts

SLICES of a three-way array

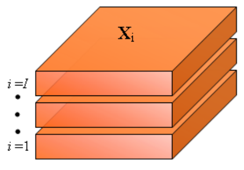
Three-way array



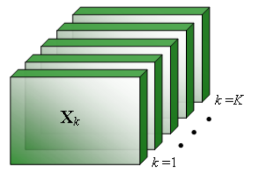
Lateral slices (X_j)



Horizontal slices (X_i)

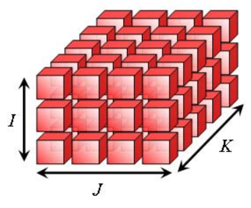


Frontal slices (X_k)

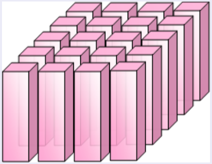


FIBERS of a three-way array

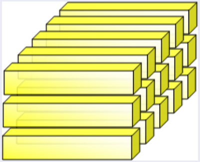
Three-way array



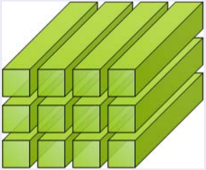
Vertical fibers ($\mathbf{x}_{jk}$)



Horizontal fibers ($\mathbf{x}_{ik}$)

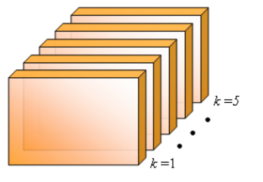


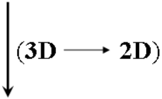
Depth fibers ($\mathbf{x}_{ij}$)



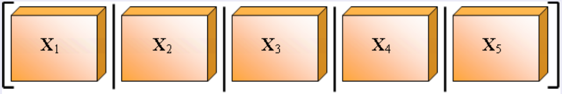
Unfolding/matricizing a three-way array

Frontal slices ($\mathbf{X}_k$)



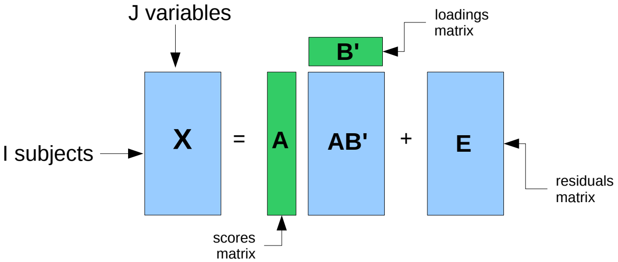


Matricizing $\underline{\mathbf{X}}$



Two-way array = data matrix

Scores of subjects (rows) on variables (columns).



$$\mathbf{X} = \mathbf{AB}' + \mathbf{E}$$

Goal

Representation of variables in low-dimension space:

R = # columns of **A** and **B** < J = # columns of **X**

X : $I \times J$ matrix of rank J; **AB'** : $I \times J$ matrix of rank R

So **X** is approximated by a matrix with lower rank, ie, **X** is approximated by a sum of R rank-1 matrices:

$$\mathbf{X} \simeq \mathbf{a}_1\mathbf{b}_1' + \cdots + \mathbf{a}_R\mathbf{b}_R'$$

X: $I \times J \times K$ array (I=subjects, J=variables, K=situations)

Goal: find components for subjects, variables and situations.

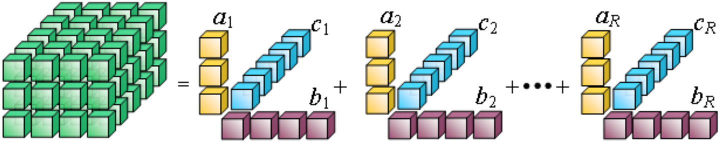
$$\mathbf{X}_k = \mathbf{A}\mathbf{C}_k\mathbf{B}' + \mathbf{E}_k$$

▸ PCA

▸ ...

$\mathbf{C}_k$=diagonal matrix holding k-th row of **C**

$$\mathbf{X}_k \simeq c_{k1}\mathbf{a}_1\mathbf{b}_1' + \cdots + c_{kR}\mathbf{a}_R\mathbf{b}_R'$$



Parallel proportional profiles (Cattell 1944)

To simultaneously analyse several matrices together, find a set of common factors ($\mathbf{A}$, $\mathbf{B}$) that can be fitted with different weights ($\mathbf{C}_k, k = 1, \ldots, K$) to many data matrices at the same time.

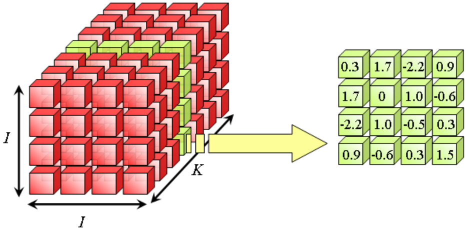
Similarities between PCA and CP

- CP decomposes an array as a sum of rank one arrays
- rank($\underline{\mathbf{X}}$)=minimum number of rank one arrays for which CP gives perfect fit

Differences between PCA and CP

- only iterative algorithm for CP
- CP is usually unique
- preprocessing three-way data hard

S: $I \times I \times K$ array with symmetric slices (set of correlation matrices, for example)



INDSCAL is CP with the constraint $\mathbf{A} = \mathbf{B}$:

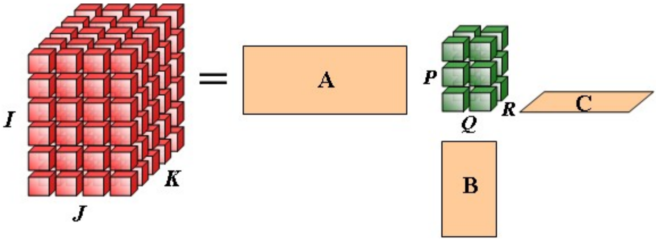
$$\boxed{\mathbf{S}_k = \mathbf{A}\mathbf{C}_k\mathbf{A}' + \mathbf{E}_k}$$

X: $I \times J \times K$ array (I=subjects, J=variables, K=situations)

Goal: find components for subjects, variables and situations.

$$\mathbf{X}_k = \mathbf{A}\left(\sum_{r=1}^{R} c_{kr}\mathbf{G}_r\right)\mathbf{B}' + \mathbf{E}_k$$

CP



Features of Tucker3

- **A**, **B**, **C** can have different number of components
- All components can interact (one per mode)
- Tucker3 decomposes $\underline{\mathbf{X}}$ as a sum of rank-1 arrays (built from **A**,**B**,**C**), weighted by the entries in core array $\underline{\mathbf{G}}$

Tucker3 – freedom of rotation

PCA's freedom of rotation (motivation)

S nonsingular

$$\begin{aligned}\tilde{\mathbf{X}} &= \mathbf{A}\mathbf{B}' \\ &= (\mathbf{A}\mathbf{S})(\mathbf{S}^{-1}\mathbf{B}')\end{aligned}$$

It is possible to "shuffle"(=invertible linear combination) the columns of **A** and **B** such that $\mathbf{A}\mathbf{B}'$ remains equal.

Tucker3's freedom of rotation

It is possible to "shuffle" the columns of **A**, **B**, **C** and preserve the fit, letting $\underline{\mathbf{G}}$ absorb the compensations (and vice-versa).

Simplifying three-way arrays

Goal

Given array $\underline{\mathbf{X}}$, find nonsingular linear combinations of the slices of $\underline{\mathbf{X}}$ (in any direction possible) such that most of the entries of the array become zero.

Why?

Statistical reasons:

- Tucker3: simpler core $\underline{\mathbf{G}} \Longrightarrow$ usually simpler interpretation
- constrained Tucker3: distinguish between tautology and non-trivial model

Mathematical reasons:

- typical rank, maximal rank, maximal simplicity

Some results available so far

- Cohen (1974, 1975), MacCallum (1976), Kroonenberg (1983): "diagonalize" frontal slices of $\underline{\mathbf{G}}$ ($P = Q$)
- Kiers (1992): "super-diagonalize" $\underline{\mathbf{G}}$ ($P = Q = R$)
- Kiers (1998): SIMPLIMAX

  $\underline{\mathbf{G}} \longrightarrow$ minimize ssq (m smallest elements)
- Murakami et al. (1998)

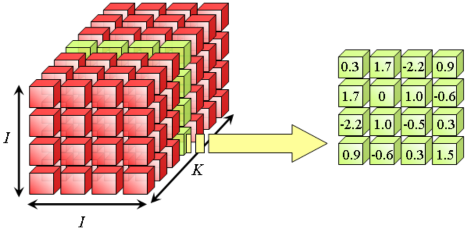
  $P = QR - 1$

  Example: $P = 5, Q = 3, R = 2$

$$\mathbf{G}_a = [\mathbf{G}_1 | \mathbf{G}_2] \longrightarrow \left[\begin{array}{ccc|ccc} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \mu_1 & 0 & 0 & 0 & \mu_2 & 0 \end{array}\right]$$

- also: Ten Berge & Kiers (1999), Ten Berge et al. (2000), Rocci & Ten Berge (2002)

What about symmetric slice arrays?



Symmetric slice $I \times I \times K_{max}$ arrays

- $\{$frontal slices$\}$ $=$ basis for the space of symmetric $I \times I$ matrices
- simple basis for the same space (Rocci & Ten Berge(1994))
  Example: $I = 3$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

- frontal slice mix suffices

Example: $3 \times 3 \times 4$

$$\left[\begin{array}{ccc|ccc|ccc|ccc} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & \alpha & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \delta\alpha & 0 & 0 & 0 & 1 & 0 & 0 \end{array}\right]$$

▸ $3 \times 3 \times 4$

($\delta = 1/-1$ in real/complex case)

Conclusion: weight 8 is always possible
(28 out of 36 entries are zero!)

Example: $4 \times 4 \times 8$

It can almost surely be simplified into one out of two weight 18 arrays.

One of the targets:

$$
\left[\begin{array}{cccc|cccc|cccc|cccc}
\star & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \star & 0 & 0 \\
0 & \star & 0 & 0 & 0 & \star & 0 & 0 & 0 & \star & 0 & 0 & \star & 0 & 0 & 0 \\
0 & 0 & 0 & \star & 0 & 0 & \star & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \star & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \star & 0 & 0 & 0 & 0
\end{array}\right.
$$

$$
\left.\begin{array}{cccc|cccc|cccc|cccc}
0 & 0 & \star & 0 & 0 & 0 & 0 & \star & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \star & 0 & 0 & 0 & 0 & \star \\
\star & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \star & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \star & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \star & 0 & 0
\end{array}\right]
$$

110 out of 128 entries are zero!

Maximal simplicity

Question: can simpler targets be found for the cases previously presented?

Answer:

- $3 \times 3 \times K$: NO (proved)
- $4 \times 4 \times K$: NO(?) (simulation)

Example of application: typical rank

- $\underline{\mathbf{X}}$: symmetric slice $3 \times 3 \times 4$ array
- Ten Berge et al. (2004)

$$\boxed{\text{typical rank } (\underline{\mathbf{X}}) = \{4, 5\}}$$

- rank=4?, rank=5?
  Check if roots of a certain fourth degree polynomial are real and distinct.

Using ▸ $3 \times 3 \times 4$ simple form we can see that:

- rank ($\underline{\mathbf{X}}$)=4 <u>iif</u> $\delta = 1$ and $\alpha > 0$ (and rank is 5 otherwise)
- a CP decomposition is now straightforward

<u>Example</u>: rank=4

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & \sqrt{\alpha} & -\sqrt{\alpha} \\ \sqrt{\alpha} & -\sqrt{\alpha} & 0 & 0 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 0 & 0 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5\sqrt{\alpha^{-1}} & -0.5\sqrt{\alpha^{-1}} \\ 0.5\sqrt{\alpha^{-1}} & -0.5\sqrt{\alpha^{-1}} & 0 & 0 \end{bmatrix}$$

Conclusions. Considerations. Developments

Conclusions

- multi-way = generalization of PCA to higher-dimensions
- rotation freedom of 3PCA allows for simplification

Considerations

- 3PCA core arrays are not “randomly sampled from a continuous distribution”, but do behave as if they were
- valid contribution for Matrix Theory: simultaneous reduction of more than a pair of matrices to sparse forms is scarce

Developments (simplicity

- extend simplicity results to other orders
- address issues like: maximal simplicity, typical rank