First and second order derivatives for CP and INDSCAL

Jorge Tendeiro

IOPS Summer Conference

11 June 2010

Outline

1. **CP, INDSCAL**
2. **Motivation: how to catch flies?**
3. **Some optimization background (back to High School!, and beyond)**
4. **The equivalence problem (from motivation to application)**

1. **CP, INDSCAL**

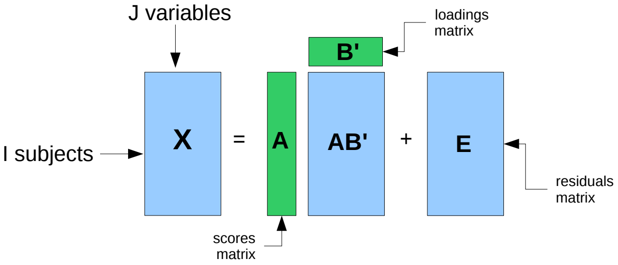
2. Motivation: how to catch flies?

3. Some optimization background (back to High School!, and beyond)

4. The equivalence problem (from motivation to application)

Two-way array = data matrix

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



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

Goal

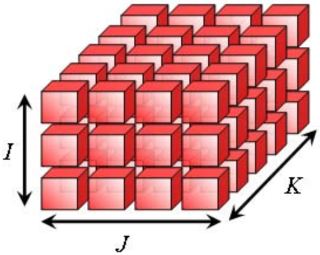
Representation of variables in low-dimension space:

$$R = \#\text{ columns of } \mathbf{A} \text{ and } \mathbf{B} \quad < \quad J = \#\text{ columns of } \mathbf{X}$$

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

So **X** is approximated by a sum of R rank-1 matrices:

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



Three-way arrays

- generalize matrix structure to 3D
- formal concept
- easy to generalize to *n*-way

$\underline{\mathbf{X}}$: $I \times J \times K$ array (I=subjects, J=variables, K=situations)
Number of components: R

Model

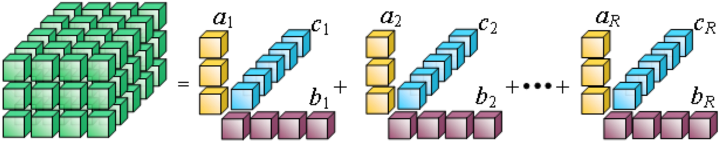
$$\mathbf{X}_k = \mathbf{A}\mathbf{C}_k\mathbf{B}' + \mathbf{E}_k$$ PCA ...

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

$\mathbf{A}$	$(I \times R)$:	"subjects" matrix
$\mathbf{B}$	$(J \times R)$:	"variables" matrix
$\mathbf{C}$	$(K \times R)$:	"situations" matrix
$\mathbf{C}_k$	$(R \times R)$:	$\mathrm{Diag}(\mathbf{c}_{k\cdot})$

Minimize loss function:

$$f_{\mathrm{CP}}(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \sum_{k=1}^{K} \|\mathbf{X}_k - \mathbf{A}\mathbf{C}_k\mathbf{B}'\|^2$$



Parallel proportional profiles (Cattell 1944)

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

CANDECOMP/PARAFAC (CP)

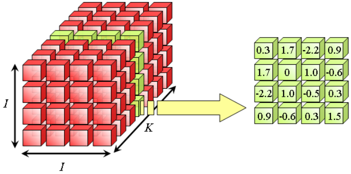
Similarities between PCA and CP

- 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 can be hard

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



Model

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

...

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

Minimize loss function:

$$f_{\text{IND}}(\mathbf{A}, \mathbf{C}) = \sum_{k=1}^{K} \|\mathbf{S}_k - \mathbf{A}\mathbf{C}_k\mathbf{A}'\|^2$$

1. CP, INDSCAL
2. **Motivation: how to catch flies?**
3. Some optimization background (back to High School!, and beyond)
4. The equivalence problem (from motivation to application)

Kruskal, Harshman, Lundy (1983, 1985):

$$\underline{\mathbf{X}} = \left[\begin{array}{cc|cc} 1 & 0 & 0 & 1 \\ 0 & -1 & 1 & 0 \end{array}\right]$$

Random starts of CP with $r = 2$ components invariably give $f = 2$.

It must be the global minimum.

No!

Ten Berge, Kiers, De Leeuw (1988)

$\inf(f)=1$

It must be LOCAL minima.

No!

What we found

All solutions with $f = 2$ are NOT minima (not even local!).

How can you reach such a conclusion?

Some optimization background

Goal

Given a scalar function $f : \mathbb{R}^n \longrightarrow \mathbb{R}$, how to find extremes (minima, maxima)?

Three types of points to consider:

1. points in the border of the domain of f;
   Example: $f(x) = x^3$, x between -1 and 1
2. points where f is not twice continuously differentiable;
   Example: $f(x) = |x|$, for $x = 0$
3. **points where f is twice continuously differentiable**

   ↑ our goal

How to optimize $f : \mathbb{R}^n \longrightarrow \mathbb{R}$

Step 1 Compute partial derivatives = 1st order derivatives for each variable, while the others are "constant".
Find stationary points (SPs) by solving the system

$$\begin{cases} \cdots \\ f_i(x_1, \ldots, x_n) = 0 \ , \quad (i = 1, \ldots, n) \\ \cdots \end{cases}$$

Step 2 Analyze 2nd order derivatives = eigenvalues of the **Hessian** matrix:

$$\text{Hess} = \begin{bmatrix} f_{11} & \cdots & f_{1n} \\ \vdots & \ddots & \vdots \\ f_{n1} & \cdots & f_{nn} \end{bmatrix}$$

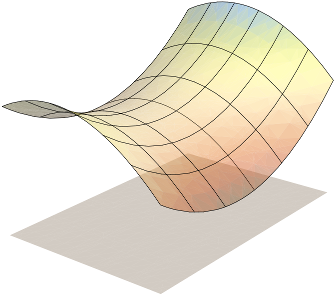
Decision rule:

- Hess is positive definite $\Longrightarrow$ SP is minimum
- Hess is negative definite $\Longrightarrow$ SP is maximum
- Hess is indefinite $\Longrightarrow$ SP is saddle point

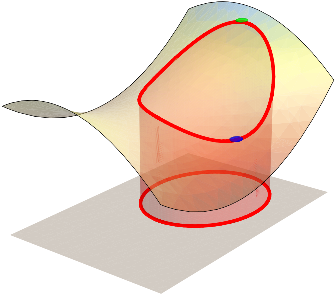
Constrained optimization

Method of Lagrange multipliers

Useful to find maxima/minima of a function subject to constraints



Unconstrained.
No minimum, no maximum.



Constrained to red points.
Minimum, Maximum.

The equivalence problem

$\underline{\mathbf{S}}$: $I \times I \times K$ array with $I \times I$ symmetric frontal slices

- Carroll and Chang (1970) suggested to use CP ▸ ... to fit the INDSCAL ▸ ... model:

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

  and then "hope" that **A** is columnwise proportional to **B** (**A** and **B** equivalent).

- **A** and **B** seem equivalent in practical applications. However, contrived counterexamples do exist.

Result: $\mathbf{A} \neq \mathbf{B}$ is possible at global minima if slices are indefinite (Ten Berge and Kiers, 1991).

Example:

$$\underline{\mathbf{S}} = \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 0 & 2 \\ 0 & -1 & 0 & 0 & -2 & 0 \\ 0 & 0 & 1 & 2 & 0 & 0 \end{array}\right]$$

$$\mathbf{A}^* = \begin{bmatrix} \sqrt{1/3} & \sqrt{0.5} \\ -\sqrt{1/3} & 0 \\ \sqrt{1/3} & -\sqrt{0.5} \end{bmatrix}, \mathbf{B}^* = \begin{bmatrix} \sqrt{1/3} & \sqrt{0.5} \\ \sqrt{1/3} & 0 \\ \sqrt{1/3} & -\sqrt{0.5} \end{bmatrix}, \mathbf{C}^* = \begin{bmatrix} 2 & 2 \\ 0 & -2 \end{bmatrix}.$$

This solution minimizes CP's loss function:

$$f_{\mathrm{CP}}(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \|\mathbf{S}_1 - \mathbf{A}\mathbf{C}_1\mathbf{B}'\|^2 + \|\mathbf{S}_2 - \mathbf{A}\mathbf{C}_2\mathbf{B}'\|^2 \geqslant 5$$

and

$$f_{\mathrm{CP}}(\mathbf{A}^*, \mathbf{B}^*, \mathbf{C}^*) \equiv 5$$

The equivalence problem: $R = 1$

Result: $\mathbf{A} \neq \mathbf{B}$ is only possible at non-optimal points if slices are non-negative definite (Ten Berge and Kiers, 1991).

Example:

$$\underline{\mathbf{S}} = \left[\begin{array}{ccc|ccc} 3 & 1 & 0 & 3 & -1 & 0 \\ 1 & 3 & 0 & -1 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

$$\mathbf{A}^* = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \mathbf{B}^* = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \mathbf{C}^* = \begin{bmatrix} 1 \\ -1 \end{bmatrix}.$$

This solution does not minimize CP's loss function:

$$f_{\text{CP}}(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \|\mathbf{S}_1 - \mathbf{A}\mathbf{C}_1\mathbf{B}'\|^2 + \|\mathbf{S}_2 - \mathbf{A}\mathbf{C}_2\mathbf{B}'\|^2 \geqslant 21$$

but

$$f_{\text{CP}}(\mathbf{A}^*, \mathbf{B}^*, \mathbf{C}^*) \equiv 39$$

These points are, in fact, saddle points (Bennani Dosse and Ten Berge, 2008).

Second-order differential structure

Question

What is the general situation when $R > 1$?

Approach to find an answer

Use simulation (run CP lots of times).

Analyze the first and second-order differential structures of the loss function of CP

$$f_{\text{CP}}(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \sum_{k=1}^{K} \|\mathbf{X}_k - \mathbf{A}\mathbf{C}_k\mathbf{B}'\|^2$$

But how to do this? Number of variables is too big.

Example: array $2 \times 2 \times 2$, $R = 2$ components

$f_{\text{CP}}(\mathbf{A}, \mathbf{B}, \mathbf{C})$ has 12 variables

Second-order differential structure

$$f_{\text{CP}}(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \sum_{k=1}^{K} \|\mathbf{X}_k - \mathbf{A}\mathbf{C}_k\mathbf{B}'\|^2 \tag{1}$$

Procedure

- parameter elimination – express **C** as a function of **A** and **B** (valid at stationary points):

$$\text{row } i \text{ of } \mathbf{C} = (\mathbf{A}'\mathbf{A} * \mathbf{B}'\mathbf{B})^{-1}\text{diag}(\mathbf{A}'\mathbf{X}_i\mathbf{B})$$

- simplify target function (1)
- use matrix differential calculus: the variables to differentitate for are *matrices* **A**, **B**
- constrain **A**,**B**:
  - columns of unit length (identification constraint)
  - orthonormal

Second-order differential structure

Apply the same procedure to INDSCAL's loss function:

$$f_{\text{IND}}(\mathbf{A}, \mathbf{C}) = \sum_{k=1}^{K} \|\mathbf{X}_k - \mathbf{A}\mathbf{C}_k\mathbf{A}'\|^2$$

What was done – for both f_{CP} and f_{IND}:

- Jacobian and Hessian matrices computed in closed form.
- second-order sufficient condition is now available to label SPs.

SVD-approach (Ten Berge, 1988)

- INDSCAL model under orthogonality constraints
- Claim: the algorithm sometimes stops at local optima

But saddle points are possible (for contrived examples).

Example:

$$\underline{\mathbf{S}} = \left[\begin{array}{ccc|ccc} 3 & 1 & 0 & 3 & -1 & 0 \\ 1 & 3 & 0 & -1 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

Global minimum ($f = 1$)

$$\mathbf{A}^* = \begin{bmatrix} \sqrt{0.5} & -\sqrt{0.5} \\ \sqrt{0.5} & \sqrt{0.5} \\ 0 & 0 \end{bmatrix}, \mathbf{C}^* = \begin{bmatrix} 4 & 2 \\ 2 & 4 \end{bmatrix}.$$

Non-optimal SPs: saddle points

$$\mathbf{A}^* = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 0 \end{bmatrix} \qquad \mathbf{C}^* = \begin{bmatrix} 3 & 3 \\ 3 & 3 \end{bmatrix} \qquad f = 5$$

$$\mathbf{A}^* = \begin{bmatrix} \sqrt{0.5} & 0 \\ \sqrt{0.5} & 0 \\ 0 & 1 \end{bmatrix} \qquad \mathbf{C}^* = \begin{bmatrix} 4 & 0 \\ 2 & 1 \end{bmatrix} \qquad f = 20$$

$$\mathbf{A}^* = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \qquad \mathbf{C}^* = \begin{bmatrix} 0 & 3 \\ 1 & 3 \end{bmatrix} \qquad f = 22$$

What happens for randomly generated data?

Simulation study

- 150 $3 \times 3 \times 3$ symmetric slice arrays with Gramian slices
- SVD-approach with $R = 2$ components;
  10 different initializations per array
- the second-order differential structure was analysed in each case

Results:

- saddle points did not occur: there are no indications that the SVD-approach converges to saddle points for randomly generated data
- local optima occurred for $\sim 8\%$ of the arrays

Equivalence problem

- eleven cases considered ($R > 1$ component, arrays with symmetric 3×3 slices)
- two types of arrays: Gramian *vs* non-Gramian slices
- 100 runs per array

Results

- $\mathbf{A} \neq \mathbf{B}$ did not occur for Gramian slices
- $\mathbf{A} \neq \mathbf{B}$ did occur for indefinite slices only in “sick” cases (degenerate)
- saddle points happen rarely

Conclusions

- loss functions of CP and INDSCAL were transformed into "simpler"(=independent variables) optimization functions
- first and second order derivatives were derived
- these tools allow to identify saddle points;
  if there is a saddle point: rerun the algorithm!
- simulation showed that saddle points do not occur frequently, but they do occur with positive probability