Problèmes d'optimisation

- Introduction
- Existence
- Unicité
- Calcul d'une décomposition orthogonale d'un tenseur symétrique
- Calcul d'une décomposition inversible
- Calcul d'une décomposition générale
- Conclusions

Tenseur

Définitions

- Tenseur representé par un tableau $\mathbf{T} = \{ T_{ij..k} \}$ dans une base canonique

Tenseur

Définitions

- Tenseur representé par un tableau $\mathbf{T} = \{T_{ij..k}\}$ dans une base canonique
- *Ordre* de $\mathbf{T} \stackrel{\text{def}}{=} \#$ de modes $= \#$ indices

Tenseur

Définitions

- Tenseur representé par un tableau $\mathbf{T} = \{ T_{ij..k} \}$ dans une base canonique
- *Ordre* de $\mathbf{T} \stackrel{\text{def}}{=} \#$ de modes $= \#$ indices
- *Dimension* $K_\ell \stackrel{\text{def}}{=}$ intervalle de variation du ℓ*eme* indice

Tenseur

Définitions

- Tenseur representé par un tableau $\mathbf{T} = \{T_{ij..k}\}$ dans une base canonique
- *Ordre* de $\mathbf{T} \stackrel{\text{def}}{=} \#$ de modes $= \#$ indices
- *Dimension* $K_\ell \stackrel{\text{def}}{=}$ intervalle de variation du ℓ*eme* indice
- $\mathbf{T}$ est *Cubique* quand toutes les dimensions $K_\ell = K$ sont égales

Tenseur

Définitions

- Tenseur representé par un tableau $\mathbf{T} = \{T_{ij..k}\}$ dans une base canonique
- *Ordre* de $\mathbf{T} \stackrel{\text{def}}{=} \#$ de modes $= \#$ indices
- *Dimension* $K_\ell \stackrel{\text{def}}{=}$ intervalle de variation du ℓ*eme* indice
- $\mathbf{T}$ est *Cubique* quand toutes les dimensions $K_\ell = K$ sont égales
- $\mathbf{T}$ est *Symétrique* s'il est cubique et que ses composantes ne changent pas par permutation *quelconque* des indices

Une fonction objectif très simple

- On veut minimiser l'erreur de modèle:

$$\sum_{ij..k} |\, T_{ij..k} - \sum_{p} \lambda_p \, A_{ip} \, B_{jp} \, ..C_{kp}|^2$$

Une fonction objectif très simple

- On veut minimiser l'erreur de modèle:

$$\sum_{ij..k} |\, T_{ij..k} - \sum_p \lambda_p \, A_{ip} \, B_{jp} \, ..C_{kp}|^2$$

- Autre écriture:

$$||\mathbf{T} - \sum_p \lambda_p \, \boldsymbol{a}(p) \otimes \boldsymbol{b}(p) \otimes .. \otimes \boldsymbol{c}(p)||^2$$

Problèmes

- Existence
- Unicité
- Calcul

Problèmes

- Existence (topologie)
- Unicité
- Calcul

Problèmes

- Existence (topologie)
- Unicité (liée au rang)
- Calcul

Problèmes

- Existence (topologie)
- Unicité (liée au rang)
- Calcul (4 exemples)

Tensor rank

CanD - Parafac

Tensor rank

CanD - Parafac

- Any tensor or array **T**, of dimensions $I \times J \times \cdots \times K$ can always be decomposed as

$$\mathbf{T} = \sum_q \lambda_q \, \boldsymbol{u}^{(q)} \otimes \boldsymbol{v}^{(q)} \otimes \ldots \otimes \boldsymbol{w}^{(q)}$$

Tensor rank

CanD - Parafac

- Any tensor or array **T**, of dimensions $I \times J \times \cdots \times K$ can always be decomposed as

$$\mathbf{T} = \sum_q \lambda_q \, \boldsymbol{u}^{(q)} \otimes \boldsymbol{v}^{(q)} \otimes \ldots \otimes \boldsymbol{w}^{(q)}$$

- The *tensor rank* of **T** is the minimal number of "rank-1 terms" such that equality holds.

Tensor rank

CanD - Parafac

- Any tensor or array **T**, of dimensions $I \times J \times \cdots \times K$ can always be decomposed as

$$\mathbf{T} = \sum_q \lambda_q \, \boldsymbol{u}^{(q)} \otimes \boldsymbol{v}^{(q)} \otimes \ldots \otimes \boldsymbol{w}^{(q)}$$

- The *tensor rank* of **T** is the minimal number of "rank-1 terms" such that equality holds.
- This yields the *Canonical Decomposition* (CanD), sometimes referred to as *Parafac*, or *Outer Product decomposition*.

Connection between symmetric tensors and homogeneous polynomials

- For instance cubics in n variables:

$$p(\mathbf{x}) = \sum_{i,j,k=1}^{n} T_{ijk}\, x_i\, x_j\, x_k$$

Connection between symmetric tensors and homogeneous polynomials

- For instance cubics in n variables:

$$p(\mathbf{x}) = \sum_{i,j,k=1}^{n} c_{ijk}\, T_{ijk}\, x_i\, x_j\, x_k$$

$c_{ijk} \stackrel{\text{def}}{=}$ multinomial coefficient $d!\,[\mu(d_1)..\mu(d_n)]^{-1}$

Connection between symmetric tensors and homogeneous polynomials

- For instance cubics in n variables:

$$p(\mathbf{x}) = \sum_{i,j,k=1}^{n} c_{ijk}\, T_{ijk}\, x_i\, x_j\, x_k$$

$c_{ijk} \stackrel{\text{def}}{=}$ multinomial coefficient $d!\,[\mu(d_1)..\mu(d_n)]^{-1}$

- **Waring problem** A polynomial of degre d can be decomposed into a sum of r dth-powers of linear forms:

$$p(\mathbf{x}) = \sum_{p=1}^{r} L_p(\mathbf{x})^d$$

Topologie

- L'ensemble des tenseurs de rang au plus r n'est fermé que pour $r \leq 1$ et r maximal (tout l'espace)

Topologie

- L'ensemble des tenseurs de rang au plus r n'est fermé que pour $r \leq 1$ et r maximal (tout l'espace)
- Le rang maximal des tenseurs d'ordre et de dimensions données est très mal connu

Topologie

- L'ensemble des tenseurs de rang au plus r n'est fermé que pour $r \leq 1$ et r maximal (tout l'espace)
- Le rang maximal des tenseurs d'ordre et de dimensions données est très mal connu
- Il faudrait connaitre le rang exact de **T** pouir le décomposer (éviter l'approximation), mais on ne sait pas faire

A sequence of rank-2 tensors converging towards a rank-4

$$\mathsf{T}_\varepsilon = \frac{1}{\varepsilon}\left[(\boldsymbol{u} + \varepsilon\,\boldsymbol{v})^{\otimes 4} - \boldsymbol{u}^{\otimes 4}\right]$$

A sequence of rank-2 tensors converging towards a rank-4

$$\mathbf{T}_\varepsilon = \frac{1}{\varepsilon}\left[(\boldsymbol{u} + \varepsilon\,\boldsymbol{v})^{\otimes 4} - \boldsymbol{u}^{\otimes 4}\right]$$

- In fact, as $\varepsilon \to 0$, it tends to:

$$\mathbf{T}_0 = \boldsymbol{u}\otimes\boldsymbol{u}\otimes\boldsymbol{u}\otimes\boldsymbol{v} + \boldsymbol{u}\otimes\boldsymbol{u}\otimes\boldsymbol{v}\otimes\boldsymbol{u} + \boldsymbol{u}\otimes\boldsymbol{v}\otimes\boldsymbol{u}\otimes\boldsymbol{u} + \boldsymbol{v}\otimes\boldsymbol{u}\otimes\boldsymbol{u}\otimes\boldsymbol{u}$$

which is proportional to the rank-4 symmetric tensor:

$$3\mathbf{T}_0 = 8\,(\boldsymbol{u}+\boldsymbol{v})^{\otimes 4} - 8(\boldsymbol{u}-\boldsymbol{v})^{\otimes 4} - (\boldsymbol{u}+2\boldsymbol{v})^{\otimes 4} + (\boldsymbol{u}-2\boldsymbol{v})^{\otimes 4} \quad (1)$$

where $\boldsymbol{u}$ and $\boldsymbol{v}$ are not collinear.

A sequence of rank-2 tensors converging towards a rank-4

$$\mathbf{T}_\varepsilon = \frac{1}{\varepsilon}\left[(\boldsymbol{u} + \varepsilon\,\boldsymbol{v})^{\otimes 4} - \boldsymbol{u}^{\otimes 4}\right]$$

- In fact, as $\varepsilon \to 0$, it tends to:

  $$\mathbf{T}_0 = \boldsymbol{u}\otimes\boldsymbol{u}\otimes\boldsymbol{u}\otimes\boldsymbol{v} + \boldsymbol{u}\otimes\boldsymbol{u}\otimes\boldsymbol{v}\otimes\boldsymbol{u} + \boldsymbol{u}\otimes\boldsymbol{v}\otimes\boldsymbol{u}\otimes\boldsymbol{u} + \boldsymbol{v}\otimes\boldsymbol{u}\otimes\boldsymbol{u}\otimes\boldsymbol{u}$$

  which is proportional to the rank-4 symmetric tensor:

  $$3\mathbf{T}_0 = 8\,(\boldsymbol{u}+\boldsymbol{v})^{\otimes 4} - 8(\boldsymbol{u}-\boldsymbol{v})^{\otimes 4} - (\boldsymbol{u}+2\boldsymbol{v})^{\otimes 4} + (\boldsymbol{u}-2\boldsymbol{v})^{\otimes 4} \quad (1)$$

  where $\boldsymbol{u}$ and $\boldsymbol{v}$ are not collinear.
- This is the *maximal rank* of 4th order tensors of dimension 2.

Generic rank

- **Informal definition:** A property is *typical* if it holds true on a non-zero-volume set

Generic rank

- **Informal definition:** A property is *typical* if it holds true on a non-zero-volume set
- **Informal definition:** A property is *generic* if is true almost everywhere.

Generic rank

- **Informal definition:** A property is *typical* if it holds true on a non-zero-volume set
- **Informal definition:** A property is *generic* if is true almost everywhere.
- There can be *several* typical ranks, but only *a single* generic rank.

Generic rank

- **Informal definition:** A property is *typical* if it holds true on a non-zero-volume set
- **Informal definition:** A property is *generic* if is true almost everywhere.
- There can be *several* typical ranks, but only *a single* generic rank.
- In the complex field, all typical ranks are generic
  Tensors with *entries randomly drawn* according to a continuous pdf are generic.

Generic rank

- **Informal definition:** A property is *typical* if it holds true on a non-zero-volume set
- **Informal definition:** A property is *generic* if is true almost everywhere.
- There can be *several* typical ranks, but only *a single* generic rank.
- In the complex field, all typical ranks are generic
  Tensors with *entries randomly drawn* according to a continuous pdf are generic.
- Importance in practice: noisy measurements

Problemes ouverts concernant l'unicité

- Unicité (à une permutation près) est *presque* toujours garantie si le rang est sous-générique

Problemes ouverts concernant l'unicité

- Unicité (à une permutation près) est *presque* toujours garantie si le rang est sous-générique
- Il existe des conditions suffisantes [Kruskal'77] mais elles sont trop restrictives (rang $\ll$ générique)

Problemes ouverts concernant l'unicité

- Unicité (à une permutation près) est *presque* toujours garantie si le rang est sous-générique
- Il existe des conditions suffisantes [Kruskal'77] mais elles sont trop restrictives (rang $\ll$ générique)
- Il n'y a jamais unicité si le rang est sur-génerique

Problemes ouverts concernant l'unicité

- Unicité (à une permutation près) est *presque* toujours garantie si le rang est sous-générique
- Il existe des conditions suffisantes [Kruskal'77] mais elles sont trop restrictives (rang $\ll$ générique)
- Il n'y a jamais unicité si le rang est sur-génerique
- Cas des tenseurs non symétriques: rang générique?

Orthogonal decomposition of a symmetric tensor

If **Q** orthogonal, the two problems are equivalent:

1. Direct:

$$\min_{\mathbf{Q},\mathbf{\Lambda}} ||\mathcal{C}_{ijk\ell} - \sum_{p=1}^{P} Q_{ip}\, Q_{jp}\, Q_{kp}\, \Lambda_{ppp}||^2$$

2. Inverse:

$$\min_{\mathbf{Q},\mathbf{\Lambda}} ||\sum_{ijk} Q_{ip}\, Q_{jq}\, Q_{kr}\, \mathcal{C}_{ijk} - \Lambda_{ppp}\, \delta_{pqr}||^2$$

Proof.
The Frobenius norm is invariant under orthogonal change of coordinates. □

Orthogonal decomposition of a symmetric tensor

If **Q** orthogonal, the two problems are equivalent:

1. Direct:

$$\min_{\mathbf{Q},\boldsymbol{\Lambda}} ||\mathcal{C}_{ijk\ell} - \sum_{p=1}^{P} Q_{ip}\, Q_{jp}\, Q_{kp}\, \Lambda_{ppp}||^2$$

2. Inverse:

$$\min_{\mathbf{Q},\boldsymbol{\Lambda}} ||\sum_{ijk} Q_{ip}\, Q_{jq}\, Q_{kr}\, \mathcal{C}_{ijk} - \Lambda_{ppp}\, \delta_{pqr}||^2$$

Proof.
The Frobenius norm is invariant under orthogonal change of coordinates. □

- Existence is guaranteed

Pair sweeping (1)

- If we assume the inverse approach, then one can compute a sequence of best plane rotations,

$$\mathbf{G} \stackrel{\text{def}}{=} \frac{1}{\sqrt{1+\theta^2}} \begin{pmatrix} 1 & \theta \\ -\theta & 1 \end{pmatrix}$$

maximizing: $\Upsilon(\mathbf{Q}) \stackrel{\text{def}}{=} \sum_i T_{iii}^2$, with
$T_{pqr} = \sum_{ijk} Q_{ip}\, Q_{jq}\, Q_{kr}\, C_{ijk}$.

Pair sweeping (1)

- If we assume the inverse approach, then one can compute a sequence of best plane rotations,

$$\mathbf{G} \stackrel{\text{def}}{=} \frac{1}{\sqrt{1+\theta^2}} \begin{pmatrix} 1 & \theta \\ -\theta & 1 \end{pmatrix}$$

  maximizing: $\Upsilon(\mathbf{Q}) \stackrel{\text{def}}{=} \sum_i T_{iii}^2$, with
  $T_{pqr} = \sum_{ijk} Q_{ip}\, Q_{jq}\, Q_{kr}\, C_{ijk}$.
- At each iteration, we have a single unknown, θ, that can be imposed to lie in $(-1, 1]$.

Pair sweeping (2)

At every iteration, the absolute maximum wrt θ is obtained algebraically. Example for 3rd order:

- We have $\Upsilon \stackrel{\text{def}}{=} (C_{111})^2 + (C_{222})^2$

Pair sweeping (2)

At every iteration, the absolute maximum wrt θ is obtained algebraically. Example for 3rd order:

- We have $\Upsilon \stackrel{\text{def}}{=} (C_{111})^2 + (C_{222})^2$
- Then Υ is a rational function in θ:

$$\Upsilon(\theta) = (\theta + \frac{1}{\theta})^{-3} \sum_{i=1}^{3} a_i \left(\theta^i - (-\theta)^{-i}\right)$$

Pair sweeping (2)

At every iteration, the absolute maximum wrt θ is obtained algebraically. Example for 3rd order:

- We have $\Upsilon \stackrel{\text{def}}{=} (C_{111})^2 + (C_{222})^2$
- Then Υ is a rational function in θ:

$$\Upsilon(\theta) = (\theta + \frac{1}{\theta})^{-3} \sum_{i=1}^{3} a_i \, \left(\theta^i - (-\theta)^{-i}\right)$$

- Denote $\xi = \theta - 1/\theta$. stationary points are roots of a degre-2 trinomial:

$$\omega(\xi) = d_2\,\xi^2 + d_1\,\xi - 4\,d_2$$

where coefficients a_i and d_i are given functions of $\mathbf{C}$ at previous iteration.

Pair sweeping (2)

At every iteration, the absolute maximum wrt θ is obtained algebraically. Example for 3rd order:

- We have $\Upsilon \stackrel{\text{def}}{=} (C_{111})^2 + (C_{222})^2$
- Then Υ is a rational function in θ:

$$\Upsilon(\theta) = (\theta + \frac{1}{\theta})^{-3} \sum_{i=1}^{3} a_i \, \left(\theta^i - (-\theta)^{-i}\right)$$

- Denote $\xi = \theta - 1/\theta$. stationary points are roots of a degre-2 trinomial:

$$\omega(\xi) = d_2\,\xi^2 + d_1\,\xi - 4\,d_2$$

  where coefficients a_i and d_i are given functions of $\mathbf{C}$ at previous iteration.
- θ is obtained from ξ by rooting a 2nd degree trinomial.

Pair sweeping (3)

- La procédure de balayage n'est attractive que si le coût de calcul d'une rotation plane est très réduit

Pair sweeping (3)

- La procédure de balayage n'est attractive que si le coût de calcul d'une rotation plane est très réduit
- La solution en θ est calculable par radicaux pour des tenseurs d'ordre 3 (2 polynômes de degré 2 en cascade).

Pair sweeping (3)

- La procédure de balayage n'est attractive que si le coût de calcul d'une rotation plane est très réduit
- La solution en θ est calculable par radicaux pour des tenseurs d'ordre 3 (2 polynômes de degré 2 en cascade).
- **Bonne nouvelle:** la solution en θ reste calculable par radicaux pour des tenseurs d'ordre 4 (racines d'un polynôme de degré 4, puis d'un autre de degré 2).

Pair sweeping (3)

- La procédure de balayage n'est attractive que si le coût de calcul d'une rotation plane est très réduit
- La solution en θ est calculable par radicaux pour des tenseurs d'ordre 3 (2 polynômes de degré 2 en cascade).
- **Bonne nouvelle:** la solution en θ reste calculable par radicaux pour des tenseurs d'ordre 4 (racines d'un polynôme de degré 4, puis d'un autre de degré 2).
- **Mauvaise nouvelle:** marche bien en pratique, maus convergence vers le maximum *absolu* n'est toujours pas théoriquement prouvée.

Two different formulations:

1. Direct: look for **A**, **B**, **C**:

$$\min_{\mathbf{A},\mathbf{B},\mathbf{C}} ||T_{ijk} - \sum_{p=1}^{P} A_{ip}\, B_{jp}\, C_{kp}||^2$$

   i.e. decompose **T** into a sum of P rank-one terms

2. Inverse: look for **A**, **B**, **C**:

$$\min_{\mathbf{A},\mathbf{B},\mathbf{C}} \sum_{mnp \neq ppp} |\sum_{ijk} A_{mi}\, B_{nj}\, C_{pk}\ T_{ijk}|^2$$

   i.e. try to diagonalize **T** by linear transforms

Un problème mal posé

- L'ensemble des matrices inversibles est ouvert. Chercher un minimum sur cet ensemble est donc mal posé, et pourrait conduire à un infimum non inversible.

Un problème mal posé

- L'ensemble des matrices inversibles est ouvert. Chercher un minimum sur cet ensemble est donc mal posé, et pourrait conduire à un infimum non inversible.
- Plusieurs algorithmes existent et ne s'affranchissent pas de ce problème

Un problème mal posé

- L'ensemble des matrices inversibles est ouvert. Chercher un minimum sur cet ensemble est donc mal posé, et pourrait conduire à un infimum non inversible.
- Plusieurs algorithmes existent et ne s'affranchissent pas de ce problème
- Approche directe: La triangularisation conjointe permet d'estimer les facteurs orthogonal et triangulaire d'une matrice inversible, l'un après l'autre

Joint triangularization of matrix slices

1. From $\mathbf{T}$, determine a collection of matrices (e.g. matrix slices), $\mathbf{T}[k]$, satisfying $\mathbf{T}[k] = \mathbf{A}\,\mathbf{D}[k]\,\mathbf{B}^{\mathsf{T}}$, $\mathbf{D}[k] \stackrel{\text{def}}{=} \mathbf{diag}\{C_{k,:}\}$.

Joint triangularization of matrix slices

1. From $\mathbf{T}$, determine a collection of matrices (e.g. matrix slices), $\mathbf{T}[k]$, satisfying $\mathbf{T}[k] = \mathbf{A}\,\mathbf{D}[k]\,\mathbf{B}^{\mathsf{T}}$, $\mathbf{D}[k] \stackrel{\text{def}}{=} \mathbf{diag}\{C_{k,:}\}$.
2. Compute the Generalized Schur decomposition [VandP96]:
   $\mathbf{Q}\,\mathbf{T}[k]\,\mathbf{Z} = \mathbf{R}[k]$, where $\mathbf{R}[k]$ are upper-triangular

Joint triangularization of matrix slices

1. From $\mathbf{T}$, determine a collection of matrices (e.g. matrix slices), $\mathbf{T}[k]$, satisfying $\mathbf{T}[k] = \mathbf{A}\,\mathbf{D}[k]\,\mathbf{B}^\mathsf{T}$, $\mathbf{D}[k] \stackrel{\text{def}}{=} \mathbf{diag}\{C_{k,:}\}$.
2. Compute the Generalized Schur decomposition [VandP96]:
   $\mathbf{Q}\,\mathbf{T}[k]\,\mathbf{Z} = \mathbf{R}[k]$, where $\mathbf{R}[k]$ are upper-triangular
3. Since $\mathbf{Q}^\mathsf{T}\,\mathbf{R}[k]\,\mathbf{Z}^\mathsf{T} = \mathbf{A}\,\mathbf{D}[k]\,\mathbf{B}^\mathsf{T}$, matrices $\mathbf{R}' \stackrel{\text{def}}{=} \mathbf{Q}\,\mathbf{A}$ and $\mathbf{R}'' \stackrel{\text{def}}{=} \mathbf{B}^\mathsf{T}\,\mathbf{Z}$ are upper triangular, and can be assumed to have a unit diagonal. Hence $\mathbf{R}'$ and $\mathbf{R}''$ can be computed by solving from the bottom to the top the triangular system [Del04], two entries R'_{ij} and R''_{ij} at a time:

$$\mathbf{R}[k] = \mathbf{R}'\,\mathbf{D}[k]\,\mathbf{R}''$$

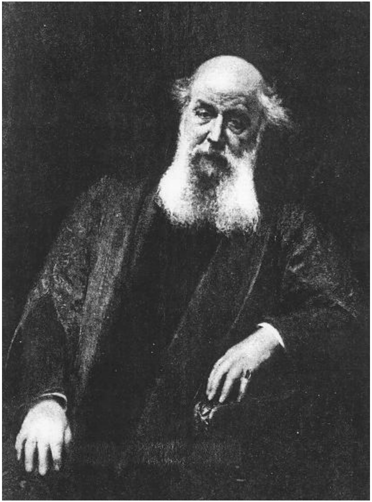
Joint triangularization of matrix slices

1. From $\mathbf{T}$, determine a collection of matrices (e.g. matrix slices), $\mathbf{T}[k]$, satisfying $\mathbf{T}[k] = \mathbf{A}\,\mathbf{D}[k]\,\mathbf{B}^{\mathsf{T}}$, $\mathbf{D}[k] \stackrel{\text{def}}{=} \textbf{diag}\{C_{k,:}\}$.
2. Compute the Generalized Schur decomposition [VandP96]:
   $\mathbf{Q}\,\mathbf{T}[k]\,\mathbf{Z} = \mathbf{R}[k]$, where $\mathbf{R}[k]$ are upper-triangular
3. Since $\mathbf{Q}^{\mathsf{T}}\,\mathbf{R}[k]\,\mathbf{Z}^{\mathsf{T}} = \mathbf{A}\,\mathbf{D}[k]\,\mathbf{B}^{\mathsf{T}}$, matrices $\mathbf{R}' \stackrel{\text{def}}{=} \mathbf{Q}\,\mathbf{A}$ and $\mathbf{R}'' \stackrel{\text{def}}{=} \mathbf{B}^{\mathsf{T}}\,\mathbf{Z}$ are upper triangular, and can be assumed to have a unit diagonal. Hence $\mathbf{R}'$ and $\mathbf{R}''$ can be computed by solving from the bottom to the top the triangular system [Del04], two entries R'_{ij} and R''_{ij} at a time:
   $$\mathbf{R}[k] = \mathbf{R}'\,\mathbf{D}[k]\,\mathbf{R}''$$
4. Compute $\mathbf{A} = \mathbf{R}'\,\mathbf{Q}^{\mathsf{T}}$ and $\mathbf{B}^{\mathsf{T}} = \mathbf{R}''\,\mathbf{Z}^{\mathsf{T}}$

Joint triangularization of matrix slices

1. From $\mathbf{T}$, determine a collection of matrices (e.g. matrix slices), $\mathbf{T}[k]$, satisfying $\mathbf{T}[k] = \mathbf{A}\,\mathbf{D}[k]\,\mathbf{B}^\mathsf{T}$, $\mathbf{D}[k] \stackrel{\text{def}}{=} \mathbf{diag}\{C_{k,:}\}$.
2. Compute the Generalized Schur decomposition [VandP96]:
   $\mathbf{Q}\,\mathbf{T}[k]\,\mathbf{Z} = \mathbf{R}[k]$, where $\mathbf{R}[k]$ are upper-triangular
3. Since $\mathbf{Q}^\mathsf{T}\,\mathbf{R}[k]\,\mathbf{Z}^\mathsf{T} = \mathbf{A}\,\mathbf{D}[k]\,\mathbf{B}^\mathsf{T}$, matrices $\mathbf{R}' \stackrel{\text{def}}{=} \mathbf{Q}\,\mathbf{A}$ and $\mathbf{R}'' \stackrel{\text{def}}{=} \mathbf{B}^\mathsf{T}\,\mathbf{Z}$ are upper triangular, and can be assumed to have a unit diagonal. Hence $\mathbf{R}'$ and $\mathbf{R}''$ can be computed by solving from the bottom to the top the triangular system [Del04], two entries R'_{ij} and R''_{ij} at a time:
$$\mathbf{R}[k] = \mathbf{R}'\,\mathbf{D}[k]\,\mathbf{R}''$$
4. Compute $\mathbf{A} = \mathbf{R}'\,\mathbf{Q}^\mathsf{T}$ and $\mathbf{B}^\mathsf{T} = \mathbf{R}''\,\mathbf{Z}^\mathsf{T}$
5. Compute matrix $\mathbf{C}$ from $\mathbf{T}$, $\mathbf{A}$ and $\mathbf{B}$ by solving the over-determined linear system $\mathbf{C}\cdot\{(\mathbf{B}\odot\mathbf{A})^\mathsf{T}\} = \mathbf{T}_{K\times JI}$

Binary case (1)



James Joseph Sylvester **(1814–1897)**

Binary case (2)

Sylvester's theorem in $\mathbb{R}$

Binary case (2)

Sylvester's theorem in $\mathbb{R}$

- A binary quantic $p(x, y) = \sum_{i=0}^{d} \gamma_i \, c(i) \, x^i \, y^{d-i}$ can be decomposed in $\mathbb{R}[x, y]$ into a sum of r powers as

$$p(x, y) = \sum_{j=1}^{r} \lambda_j \, (\alpha_j \, x + \beta_j \, y)^d \text{ if and only if:}$$

Binary case (2)

Sylvester's theorem in $\mathbb{R}$

- A binary quantic $p(x,y) = \sum_{i=0}^{d} \gamma_i \, c(i) \, x^i \, y^{d-i}$ can be decomposed in $\mathbb{R}[x,y]$ into a sum of r powers as $p(x,y) = \sum_{j=1}^{r} \lambda_j \, (\alpha_j \, x + \beta_j \, y)^d$ if and only if:
  1. the form $q_c(x,y) = \prod_{j=1}^{r} (\beta_j \, x - \alpha_j \, y) = \sum_{l=0}^{r} g_l \, x^l \, y^{r-l}$ satisfies

$$\begin{bmatrix} \gamma_0 & \gamma_1 & \cdots & \gamma_r \\ \gamma_1 & \gamma_2 & \cdots & \gamma_{r+1} \\ \vdots & & & \vdots \\ \gamma_{d-r} & & \cdots & \gamma_d \end{bmatrix} \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_r \end{bmatrix} = 0.$$

Binary case (2)

Sylvester's theorem in $\mathbb{R}$

- A binary quantic $p(x,y) = \sum_{i=0}^{d} \gamma_i \, c(i) \, x^i \, y^{d-i}$ can be decomposed in $\mathbb{R}[x,y]$ into a sum of r powers as
$$p(x,y) = \sum_{j=1}^{r} \lambda_j \, (\alpha_j \, x + \beta_j \, y)^d \text{ if and only if:}$$

1. the form $q_c(x,y) = \prod_{j=1}^{r} (\beta_j \, x - \alpha_j \, y) = \sum_{l=0}^{r} g_l \, x^l \, y^{r-l}$ satisfies
$$\begin{bmatrix} \gamma_0 & \gamma_1 & \cdots & \gamma_r \\ \gamma_1 & \gamma_2 & \cdots & \gamma_{r+1} \\ \vdots & & & \vdots \\ \gamma_{d-r} & & \cdots & \gamma_d \end{bmatrix} \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_r \end{bmatrix} = 0.$$
2. $q_c(x,y)$ has distinct real roots.

Binary case (2)

Sylvester's theorem in $\mathbb{R}$

- A binary quantic $p(x,y) = \sum_{i=0}^{d} \gamma_i\, c(i)\, x^i\, y^{d-i}$ can be decomposed in $\mathbb{R}[x,y]$ into a sum of r powers as
  $p(x,y) = \sum_{j=1}^{r} \lambda_j\, (\alpha_j\, x + \beta_j\, y)^d$ if and only if:
  1. the form $q_c(x,y) = \prod_{j=1}^{r} (\beta_j\, x - \alpha_j\, y) = \sum_{l=0}^{r} g_l\, x^l\, y^{r-l}$ satisfies

$$\begin{bmatrix} \gamma_0 & \gamma_1 & \cdots & \gamma_r \\ \gamma_1 & \gamma_2 & \cdots & \gamma_{r+1} \\ \vdots & & & \vdots \\ \gamma_{d-r} & & \cdots & \gamma_d \end{bmatrix} \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_r \end{bmatrix} = 0.$$

  2. $q_c(x,y)$ has distinct real roots.
- Valid even in non generic cases.

Binary case (3)

Sylvester's theorem in $\mathbb{C}$

Binary case (3)

Sylvester's theorem in $\mathbb{C}$

- A binary quantic $p(x, y) = \sum_{i=0}^{d} c(i)\, \gamma_i\, x^i\, y^{d-i}$ can be written as a sum of dth powers of r distinct linear forms:

  $p(x, y) = \sum_{j=1}^{r} \lambda_j\, (\alpha_j\, x + \beta_j\, y)^d$ if and only if:

Binary case (3)

Sylvester's theorem in $\mathbb{C}$

- A binary quantic $p(x, y) = \sum_{i=0}^{d} c(i)\, \gamma_i\, x^i\, y^{d-i}$ can be written as a sum of dth powers of r distinct linear forms:
  $p(x, y) = \sum_{j=1}^{r} \lambda_j\, (\alpha_j\, x + \beta_j\, y)^d$ if and only if:
  1. there exists a vector $\boldsymbol{g}$ of dimension $r + 1$, with components g_ℓ, such that

$$\begin{bmatrix} \gamma_0 & \gamma_1 & \cdots & \gamma_r \\ \vdots & & & \vdots \\ \gamma_{d-r} & \cdots & \gamma_{d-1} & \gamma_d \end{bmatrix} \boldsymbol{g}^* = \mathbf{0}. \qquad (2)$$

Binary case (3)

Sylvester's theorem in $\mathbb{C}$

- A binary quantic $p(x,y) = \sum_{i=0}^{d} c(i)\,\gamma_i\,x^i\,y^{d-i}$ can be written as a sum of dth powers of r distinct linear forms:
  $p(x,y) = \sum_{j=1}^{r} \lambda_j\,(\alpha_j\,x + \beta_j\,y)^d$ if and only if:
  1. there exists a vector $\boldsymbol{g}$ of dimension $r+1$, with components g_ℓ, such that

$$\begin{bmatrix} \gamma_0 & \gamma_1 & \cdots & \gamma_r \\ \vdots & & & \vdots \\ \gamma_{d-r} & \cdots & \gamma_{d-1} & \gamma_d \end{bmatrix} \boldsymbol{g}^* = \mathbf{0}. \qquad (2)$$

  2. the polynomial $q(x,y) \stackrel{\text{def}}{=} \sum_{\ell=0}^{r} g_\ell\,x^\ell\,y^{r-\ell}$ $\stackrel{\text{def}}{=} \prod_{j=1}^{r}(\beta_j^*\,x - \alpha_j^*\,y)$ admits r distinct roots

Binary case (3)

Sylvester's theorem in $\mathbb{C}$

- A binary quantic $p(x,y) = \sum_{i=0}^{d} c(i)\,\gamma_i\,x^i\,y^{d-i}$ can be written as a sum of dth powers of r distinct linear forms:
  $p(x,y) = \sum_{j=1}^{r} \lambda_j\,(\alpha_j\,x + \beta_j\,y)^d$ if and only if:
  1. there exists a vector $\boldsymbol{g}$ of dimension $r+1$, with components g_ℓ, such that

$$\begin{bmatrix} \gamma_0 & \gamma_1 & \cdots & \gamma_r \\ \vdots & & & \vdots \\ \gamma_{d-r} & \cdots & \gamma_{d-1} & \gamma_d \end{bmatrix} \boldsymbol{g}^* = \mathbf{0}. \qquad (2)$$

  2. the polynomial $q(x,y) \stackrel{\text{def}}{=} \sum_{\ell=0}^{r} g_\ell\,x^\ell\,y^{r-\ell}$ $\stackrel{\text{def}}{=} \prod_{j=1}^{r}(\beta_j^*\,x - \alpha_j^*\,y)$ admits r distinct roots
- Valid even in non generic cases.

Algorithm for rth order symmetric tensors of dimension 2

Start with $r = 1$ ($d \times 2$ matrix) and increase r until it looses its column rank

1	2
2	3
3	4
4	5
5	6
6	7
7	8

$\longrightarrow$

1	2	3
2	3	4
3	4	5
4	5	6
5	6	7
6	7	8

$\longrightarrow$

1	2	3	4
2	3	4	5
3	4	5	6
4	5	6	7
5	6	7	8

Iterative algorithms for under-determined mixtures

Take 3rd order case to illustrate the reasoning. Define

$$\boldsymbol{p} \stackrel{\text{def}}{=} \begin{bmatrix} \text{vec}\{\mathbf{A}^{\mathsf{T}}\} \\ \text{vec}\{\mathbf{B}^{\mathsf{T}}\} \\ \text{vec}\{\mathbf{C}^{\mathsf{T}}\} \end{bmatrix}, \quad \text{and the gradient } \boldsymbol{g} = \begin{bmatrix} \boldsymbol{g}_{\mathbf{A}} \\ \boldsymbol{g}_{\mathbf{B}} \\ \boldsymbol{g}_{\mathbf{C}} \end{bmatrix}$$

1. Alternate Least Squares (ALS)
2. Gradient update: $\boldsymbol{p}(k+1) = \boldsymbol{p}(k) - \mu(k)\,\boldsymbol{g}(k)$
3. Newton update: $\boldsymbol{p}(k+1) = \boldsymbol{p}(k) - \mathbf{H}(k)^{-1}\,\boldsymbol{g}(k)$
4. LM update: $\boldsymbol{p}(k+1) = \boldsymbol{p}(k) - [\mathbf{J}(k)\mathbf{J}(k)^{\mathsf{H}} + \lambda(k)\mathbf{I}]^{-1}\,\boldsymbol{g}(k)$
5. BFGS update: $\boldsymbol{p}(k+1) = \boldsymbol{p}(k) - [\mathbf{J}(k)\mathbf{J}(k)^{\mathsf{H}} + \mathbf{M}(k)]^{-1}\,\boldsymbol{g}(k)$
6. Conjugate gradient...

ALS (1)

$$\varepsilon = ||\mathbf{T}_{I\times KJ} - \mathbf{A}(\mathbf{C}\odot\mathbf{B})^{\mathsf{T}}||^2 \qquad (3)$$

Advantage: compact writing of the best matrix **A**, for fixed **B** and **C**, since (3) in quadratic in **A** [?]:

$$\hat{\mathbf{A}} = \mathbf{T}_{I\times KJ}\cdot\{(\mathbf{C}\odot\mathbf{B})^{\mathsf{T}}\}^{\dagger}$$

where † denotes pseudo-inverse.

Similarly:

$$\begin{aligned} ||\mathbf{T}_{J\times IK} - \mathbf{B}(\mathbf{A}\odot\mathbf{C})^{\mathsf{T}}||^2 &\rightarrow \hat{\mathbf{B}} = \mathbf{T}_{J\times IK}\cdot\{(\mathbf{A}\odot\mathbf{C})^{\mathsf{T}}\}^{\dagger} \\ ||\mathbf{T}_{K\times JI} - \mathbf{C}(\mathbf{B}\odot\mathbf{A})^{\mathsf{T}}||^2 &\rightarrow \hat{\mathbf{C}} = \mathbf{T}_{K\times JI}\cdot\{(\mathbf{B}\odot\mathbf{A})^{\mathsf{T}}\}^{\dagger} \end{aligned}$$

ALS (2)

Start with arbitrary $\mathbf{B}(0)$ and $\mathbf{C}(0)$
For $k = 1...kmax$,

- $\mathbf{A}(k+1) = \mathbf{T}_{I\times KJ} \cdot \{(\mathbf{C}(k) \odot \mathbf{B}(k))^\mathsf{T}\}^\dagger$
- $\mathbf{B}(k+1) = \mathbf{T}_{J\times IK} \cdot \{(\mathbf{A}(k+1) \odot \mathbf{C}(k))^\mathsf{T}\}^\dagger$
- $\mathbf{C}(k+1) = \mathbf{T}_{K\times JI} \cdot \{(\mathbf{B}(k+1) \odot \mathbf{A}(k+1))^\mathsf{T}\}^\dagger$

Hence the ALS algorithm also needs that:

$$R \leq \min(JK, IK, IJ)$$

According to Kruskal [?], this inequality is always satisfied.

ALS drawbacks

1. Fairly slow convergence when reaching plateaux
2. May be stuck about local minima

Optimal Line Search: reduction to univariate polynomial

The same principle applies to any iterative algorithm [?]:

- Compute a search direction $[\delta\mathbf{A}, \delta\mathbf{B}, \delta\mathbf{C}]$, which can be the gradient $\boldsymbol{g}$, a direction $\mathbf{H}^{-1}\boldsymbol{g}$, or a difference $[\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}] - [\mathbf{A}(k), \mathbf{B}(k), \mathbf{C}(k)]$...

Optimal Line Search: reduction to univariate polynomial

The same principle applies to any iterative algorithm [?]:

- Compute a search direction $[\delta\mathbf{A}, \delta\mathbf{B}, \delta\mathbf{C}]$, which can be the gradient $\boldsymbol{g}$, a direction $\mathbf{H}^{-1}\boldsymbol{g}$, or a difference $[\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}] - [\mathbf{A}(k), \mathbf{B}(k), \mathbf{C}(k)]$...
- Compute the 6 first coefficients of the 6th degree polynomial $\varepsilon(\mu)$, defined by replacing $[\mathbf{A}, \mathbf{B}, \mathbf{C}]$ by $[\mathbf{A} + \mu\,\delta\mathbf{A}, \mathbf{B} + \mu\,\delta\mathbf{B}, \mathbf{C} + \mu\,\delta\mathbf{C}]$

Optimal Line Search: reduction to univariate polynomial

The same principle applies to any iterative algorithm [?]:

- Compute a search direction $[\delta\mathbf{A}, \delta\mathbf{B}, \delta\mathbf{C}]$, which can be the gradient $\boldsymbol{g}$, a direction $\mathbf{H}^{-1}\boldsymbol{g}$, or a difference $[\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}] - [\mathbf{A}(k), \mathbf{B}(k), \mathbf{C}(k)]$...
- Compute the 6 first coefficients of the 6th degree polynomial $\varepsilon(\mu)$, defined by replacing $[\mathbf{A}, \mathbf{B}, \mathbf{C}]$ by $[\mathbf{A} + \mu\,\delta\mathbf{A}, \mathbf{B} + \mu\,\delta\mathbf{B}, \mathbf{C} + \mu\,\delta\mathbf{C}]$
- Compute the 5 roots of its derivative

Optimal Line Search: reduction to univariate polynomial

The same principle applies to any iterative algorithm [?]:

- Compute a search direction $[\delta\mathbf{A}, \delta\mathbf{B}, \delta\mathbf{C}]$, which can be the gradient $\boldsymbol{g}$, a direction $\mathbf{H}^{-1}\boldsymbol{g}$, or a difference $[\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}] - [\mathbf{A}(k), \mathbf{B}(k), \mathbf{C}(k)]$...
- Compute the 6 first coefficients of the 6th degree polynomial $\varepsilon(\mu)$, defined by replacing $[\mathbf{A}, \mathbf{B}, \mathbf{C}]$ by $[\mathbf{A} + \mu\,\delta\mathbf{A}, \mathbf{B} + \mu\,\delta\mathbf{B}, \mathbf{C} + \mu\,\delta\mathbf{C}]$
- Compute the 5 roots of its derivative
- Select the root μ_o yielding the smallest minumum of w.r.t. (μ)

Optimal Line Search: reduction to univariate polynomial

The same principle applies to any iterative algorithm [?]:

- Compute a search direction $[\delta\mathbf{A}, \delta\mathbf{B}, \delta\mathbf{C}]$, which can be the gradient $\boldsymbol{g}$, a direction $\mathbf{H}^{-1}\boldsymbol{g}$, or a difference $[\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}] - [\mathbf{A}(k), \mathbf{B}(k), \mathbf{C}(k)]$...
- Compute the 6 first coefficients of the 6th degree polynomial $\varepsilon(\mu)$, defined by replacing $[\mathbf{A}, \mathbf{B}, \mathbf{C}]$ by $[\mathbf{A} + \mu\,\delta\mathbf{A}, \mathbf{B} + \mu\,\delta\mathbf{B}, \mathbf{C} + \mu\,\delta\mathbf{C}]$
- Compute the 5 roots of its derivative
- Select the root μ_o yielding the smallest minumum of w.r.t. (μ)
- Update: $\mathbf{A}(k+1) = \mathbf{A}(k) + \mu_o\,\delta\mathbf{A}$, $\mathbf{B}(k+1) = \mathbf{B}(k) + \mu_o\,\delta\mathbf{B}$, $\mathbf{C}(k+1) = \mathbf{C}(k) + \mu_o\,\delta\mathbf{C}$.

Optimal Line Search: reduction to univariate polynomial

The same principle applies to any iterative algorithm [?]:

- Compute a search direction $[\delta\mathbf{A}, \delta\mathbf{B}, \delta\mathbf{C}]$, which can be the gradient $\boldsymbol{g}$, a direction $\mathbf{H}^{-1}\boldsymbol{g}$, or a difference $[\hat{\mathbf{A}}, \hat{\mathbf{B}}, \hat{\mathbf{C}}] - [\mathbf{A}(k), \mathbf{B}(k), \mathbf{C}(k)]$...
- Compute the 6 first coefficients of the 6th degree polynomial $\varepsilon(\mu)$, defined by replacing $[\mathbf{A}, \mathbf{B}, \mathbf{C}]$ by $[\mathbf{A} + \mu\,\delta\mathbf{A}, \mathbf{B} + \mu\,\delta\mathbf{B}, \mathbf{C} + \mu\,\delta\mathbf{C}]$
- Compute the 5 roots of its derivative
- Select the root μ_o yielding the smallest minumum of w.r.t. (μ)
- Update: $\mathbf{A}(k+1) = \mathbf{A}(k) + \mu_o\,\delta\mathbf{A}$, $\mathbf{B}(k+1) = \mathbf{B}(k) + \mu_o\,\delta\mathbf{B}$, $\mathbf{C}(k+1) = \mathbf{C}(k) + \mu_o\,\delta\mathbf{C}$.
- Can be executed at every iteration, or less often. Idem for higher orders. Often allows to escape from local minima

Résumé

- Ce n'est pas qu'un problème d'optimisation numérique: il y a des problèmes ouverts sous-jacents (existence, unicité)

Résumé

- Ce n'est pas qu'un problème d'optimisation numérique: il y a des problèmes ouverts sous-jacents (existence, unicité)
- Il existe des algorithmes pointus, moins coûteux. Mais leur conditions d'utilisation sont plus restrictives (rang notamment).

Résumé

- Ce n'est pas qu'un problème d'optimisation numérique: il y a des problèmes ouverts sous-jacents (existence, unicité)
- Il existe des algorithmes pointus, moins coûteux. Mais leur conditions d'utilisation sont plus restrictives (rang notamment).
- Les algorithmes d'optimisation standard les plus bêtes ne peuvent pas être mauvais avec une fonction aussi sympatique qu'un polynôme, mais le pb est la présence de minima locaux.

Matrices vs. Tensors (1)

For matrices:

For tensors:

Matrices vs. Tensors (1)

For matrices:

1. Rank cannot exceed dimensions

For tensors:

Matrices vs. Tensors (1)

For matrices:

1. Rank cannot exceed dimensions

For tensors:

1. Rank can largely exceed dimensions

Matrices vs. Tensors (1)

For matrices:

1. Rank cannot exceed dimensions
2. CAND is not unique (SVD imposes orthogonality)

For tensors:

1. Rank can largely exceed dimensions

Matrices vs. Tensors (1)

For matrices:

1. Rank cannot exceed dimensions
2. CAND is not unique (SVD imposes orthogonality)

For tensors:

1. Rank can largely exceed dimensions
2. CAND is unique for subgeneric ranks

Matrices vs. Tensors (1)

For matrices:

1. Rank cannot exceed dimensions
2. CAND is not unique (SVD imposes orthogonality)
3. Rank is the same in real and complex fields

For tensors:

1. Rank can largely exceed dimensions
2. CAND is unique for subgeneric ranks

Matrices vs. Tensors (1)

For matrices:

1. Rank cannot exceed dimensions
2. CAND is not unique (SVD imposes orthogonality)
3. Rank is the same in real and complex fields

For tensors:

1. Rank can largely exceed dimensions
2. CAND is unique for subgeneric ranks
3. Rank may be different in real and complex fields

Matrices vs. Tensors (1)

For matrices:

1. Rank cannot exceed dimensions
2. CAND is not unique (SVD imposes orthogonality)
3. Rank is the same in real and complex fields
4. Rank of symmetric matrix is the same if CanD is symmetric

For tensors:

1. Rank can largely exceed dimensions
2. CAND is unique for subgeneric ranks
3. Rank may be different in real and complex fields

Matrices vs. Tensors (1)

For matrices:

1. Rank cannot exceed dimensions
2. CAND is not unique (SVD imposes orthogonality)
3. Rank is the same in real and complex fields
4. Rank of symmetric matrix is the same if CanD is symmetric

For tensors:

1. Rank can largely exceed dimensions
2. CAND is unique for subgeneric ranks
3. Rank may be different in real and complex fields
4. Imposing symmetry in the CanD has not yet been proved to lead to same rank

Matrices vs. Tensors (2)

For matrices:

For tensors:

Matrices vs. Tensors (2)

For matrices:

5. Matrices with entries drawn randomly have maximal rank

For tensors:

Matrices vs. Tensors (2)

For matrices:

5. Matrices with entries drawn randomly have maximal rank

For tensors:

5. Generic rank is not maximal in most cases

Matrices vs. Tensors (2)

For matrices:

5. Matrices with entries drawn randomly have maximal rank
6. Maximal rank equals generic rank

For tensors:

5. Generic rank is not maximal in most cases

Matrices vs. Tensors (2)

For matrices:

5. Matrices with entries drawn randomly have maximal rank
6. Maximal rank equals generic rank

For tensors:

5. Generic rank is not maximal in most cases
6. Maximal rank still unlnown for most order/dimensions

Matrices vs. Tensors (2)

For matrices:

5. Matrices with entries drawn randomly have maximal rank
6. Maximal rank equals generic rank
7. The set of matrices of rank at most r is closed, $\forall r$

For tensors:

5. Generic rank is not maximal in most cases
6. Maximal rank still unlnown for most order/dimensions

Matrices vs. Tensors (2)

For matrices:

5. Matrices with entries drawn randomly have maximal rank
6. Maximal rank equals generic rank
7. The set of matrices of rank at most r is closed, $\forall r$

For tensors:

5. Generic rank is not maximal in most cases
6. Maximal rank still unlnown for most order/dimensions
7. The set of tensors of rank at most r is not closed, except for $r = 1$ and r maximal
   - Approximation problem ill-posed!

Matrices vs. Tensors (2)

For matrices:

5. Matrices with entries drawn randomly have maximal rank
6. Maximal rank equals generic rank
7. The set of matrices of rank at most r is closed, $\forall r$
8. subtracting the best rank-1 approximnate decreases rank by 1

For tensors:

5. Generic rank is not maximal in most cases
6. Maximal rank still unlnown for most order/dimensions
7. The set of tensors of rank at most r is not closed, except for $r = 1$ and r maximal
   - Approximation problem ill-posed!

Matrices vs. Tensors (2)

For matrices:

5. Matrices with entries drawn randomly have maximal rank
6. Maximal rank equals generic rank
7. The set of matrices of rank at most r is closed, $\forall r$
8. subtracting the best rank-1 approximnate decreases rank by 1

For tensors:

5. Generic rank is not maximal in most cases
6. Maximal rank still unlnown for most order/dimensions
7. The set of tensors of rank at most r is not closed, except for $r = 1$ and r maximal
   - Approximation problem ill-posed!
8. Subtracting best rank-1 approximate may increase tensor rank