Non-negative CP decomposition of hyperspectral data

Miguel A. Veganzones, Jeremy Cohen, Rodrigo Cabral-Farias and Pierre Comon

gipsa-lab, CNRS, UMR 5216, F-38420, Grenoble, France
miguel-angel.veganzones@gipsa-lab.fr

ERC AdG-2013-320594 DECODA

UNIVERSITÉ DE GRENOBLE

gipsa-lab

Grenoble | images | parole | signal | automatique | laboratoire

Outline

Outline

Goals

1. Analysis of hyperspectral data by means of Canonical Polyadic (CP) tensor decomposition.
    1.1 Time-series.
    1.2 Multi-angle acquisitions.
    1.3 Conventional images.
2. Physical interpretation of the one-rank factors in terms of spectral unmixing.
3. Applications :
    3.1 Snow cover maps of the Alps : collaboration with LTHE laboratory and MeteoFrance.
    3.2 Analysis of Martian surface : collaboration with IPAG (Mars-ReCo project).

Goals

1. Analysis of hyperspectral data by means of Canonical Polyadic (CP) tensor decomposition.
   1.1 Time-series.
   1.2 Multi-angle acquisitions.
   1.3 Conventional images.
2. Physical interpretation of the one-rank factors in terms of spectral unmixing.
3. Applications :
   3.1 Snow cover maps of the Alps : collaboration with LTHE laboratory and MeteoFrance.
   3.2 Analysis of Martian surface : collaboration with IPAG (Mars-ReCo project).

Goals

1. Analysis of hyperspectral data by means of Canonical Polyadic (CP) tensor decomposition.
   1.1 Time-series.
   1.2 Multi-angle acquisitions.
   1.3 Conventional images.
2. Physical interpretation of the one-rank factors in terms of spectral unmixing.
3. Applications :
   3.1 Snow cover maps of the Alps : collaboration with LTHE laboratory and MeteoFrance.
   3.2 Analysis of Martian surface : collaboration with IPAG (Mars-ReCo project).

Hyperspectral tensors

> Hyperspectral matrix : $\mathbf{X}^{I \times J}$.
  - I : number of pixels (spatial way).
  - J : number of bands (spectral way).

> Hyperspectral tensor : $\boldsymbol{\mathcal{X}}^{I \times J \times K}$.
  - K : number of time acquisitions (temporal way : SNOW project).
  - K : number of angles (angular way : Mars-ReCo project).
  - What about data cubes (rows $\times$ columns $\times$ bands) ?

Hyperspectral tensors

> Hyperspectral matrix : $\mathbf{X}^{I \times J}$.
  - I : number of pixels (spatial way).
  - J : number of bands (spectral way).

> Hyperspectral tensor : $\boldsymbol{\mathcal{X}}^{I \times J \times K}$.
  - K : number of time acquisitions (temporal way : SNOW project).
  - K : number of angles (angular way : Mars-ReCo project).
  - What about data cubes (rows $\times$ columns $\times$ bands) ?

Hyperspectral tensors

> Hyperspectral matrix : $\mathbf{X}^{I \times J}$.
  - I : number of pixels (spatial way).
  - J : number of bands (spectral way).

> Hyperspectral tensor : $\boldsymbol{\mathcal{X}}^{I \times J \times K}$.
  - K : number of time acquisitions (temporal way : SNOW project).
  - K : number of angles (angular way : Mars-ReCo project).
  - What about data cubes (rows $\times$ columns $\times$ bands) ?

Outline

Non-negative CP decomposition

> Formulation :

$$\mathcal{X}_{ijk} \approx \sum_{r=1}^{R} A_{ir} B_{jr} C_{kr} \lambda_r. \quad (1)$$

- $R \in \mathbb{N}_+$: tensor non-negative rank.
- $\mathbf{A}^{I \times R}$: spatial factors.
- $\mathbf{B}^{J \times R}$: spectral factors.
- $\mathbf{C}^{K \times R}$: temporal/angular factors.
- $\mathbf{\Lambda}^{R \times R}$: scaling diagonal matrix.
- **Everything is non-negative !**

> Compact representation : $\boldsymbol{\mathcal{X}} \approx (\mathbf{A}, \mathbf{B}, \mathbf{C})\, \mathbf{\Lambda}$.

> Optimization problem :

$$\begin{array}{ll} \text{minimize} & \|\boldsymbol{\mathcal{X}} - (\mathbf{A}, \mathbf{B}, \mathbf{C})\, \mathbf{\Lambda}\|_F^2 \\ \text{w.r.t.} & \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{\Lambda} \\ \text{subject to} & \mathbf{A} \succeq 0, \mathbf{B} \succeq 0, \mathbf{C} \succeq 0 \end{array} \quad (2)$$

Non-negative CP decomposition

> Formulation :

$$\mathcal{X}_{ijk} \approx \sum_{r=1}^{R} A_{ir} B_{jr} C_{kr} \lambda_r. \quad (1)$$

- $R \in \mathbb{N}_+$: tensor non-negative rank.
- $\mathbf{A}^{I \times R}$: spatial factors.
- $\mathbf{B}^{J \times R}$: spectral factors.
- $\mathbf{C}^{K \times R}$: temporal/angular factors.
- $\mathbf{\Lambda}^{R \times R}$: scaling diagonal matrix.
- **Everything is non-negative !**

> Compact representation : $\boldsymbol{\mathcal{X}} \approx (\mathbf{A}, \mathbf{B}, \mathbf{C})\,\mathbf{\Lambda}$.

> Optimization problem :

$$\begin{array}{ll} \text{minimize} & \|\boldsymbol{\mathcal{X}} - (\mathbf{A}, \mathbf{B}, \mathbf{C})\,\mathbf{\Lambda}\|_F^2 \\ \text{w.r.t.} & \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{\Lambda} \\ \text{subject to} & \mathbf{A} \succeq 0, \mathbf{B} \succeq 0, \mathbf{C} \succeq 0 \end{array} \quad (2)$$

Non-negative CP decomposition

> Formulation :

$$\mathcal{X}_{ijk} \approx \sum_{r=1}^{R} A_{ir} B_{jr} C_{kr} \lambda_r. \tag{1}$$

- $R \in \mathbb{N}_+$: tensor non-negative rank.
- $\mathbf{A}^{I \times R}$: spatial factors.
- $\mathbf{B}^{J \times R}$: spectral factors.
- $\mathbf{C}^{K \times R}$: temporal/angular factors.
- $\mathbf{\Lambda}^{R \times R}$: scaling diagonal matrix.
- **Everything is non-negative !**

> Compact representation : $\boldsymbol{\mathcal{X}} \approx (\mathbf{A}, \mathbf{B}, \mathbf{C})\, \mathbf{\Lambda}$.

> Optimization problem :

$$\begin{array}{cc} \text{minimize} & \|\boldsymbol{\mathcal{X}} - (\mathbf{A}, \mathbf{B}, \mathbf{C})\, \mathbf{\Lambda}\|_F^2 \\ \text{w.r.t.} & \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{\Lambda} \\ \text{subject to} & \mathbf{A} \succeq 0, \mathbf{B} \succeq 0, \mathbf{C} \succeq 0 \end{array} \tag{2}$$

Properties

> Best lower non-negative rank approximates always exist $\rightarrow$ The problem is well-posed.

> There exist upper bounds to the tensor rank, R_o, that ensure uniqueness -> But only for exact decompositions !

> (Recently proved) If condition $R \leq R_o$ holds true $\rightarrow$ Almost always the best lower non-negative rank approximate is unique.

Properties

- Best lower non-negative rank approximates always exist $\rightarrow$ The problem is well-posed.
- There exist upper bounds to the tensor rank, R_o, that ensure uniqueness -> But only for exact decompositions !
- (Recently proved) If condition $R \leq R_o$ holds true $\rightarrow$ Almost always the best lower non-negative rank approximate is unique.

Properties

> Best lower non-negative rank approximates always exist $\rightarrow$ The problem is well-posed.

> There exist upper bounds to the tensor rank, R_o, that ensure uniqueness -> But only for exact decompositions !

> (Recently proved) If condition $R \leq R_o$ holds true $\rightarrow$ Almost always the best lower non-negative rank approximate is unique.

Properties

> Best lower non-negative rank approximates always exist $\rightarrow$ The problem is well-posed.

> There exist upper bounds to the tensor rank, R_o, that ensure uniqueness -> But only for exact decompositions !

> (Recently proved) If condition $R \leq R_o$ holds true $\rightarrow$ Almost always the best lower non-negative rank approximate is unique.

Algorithms

- > The CP optimization problem is highly non-convex.
- > Yet many algorithms provide rather precise computation.
- > These algorithms can be divided into two main classes :
  - *All-at-once gradient-based descent, e.g.* : all CP parameters are updated at the same time using a gradient scheme and non-negativity constraints are implemented through barriers or soft penalizations.
  - *Alternating minimization* : the cost function is minimized in an alternating way for each factor ($\mathbf{A}$, $\mathbf{B}$ or $\mathbf{C}$) while the others are fixed.
- > Drawback : These algorithms are computationally expensive $\rightarrow$ Useless for large tensors.
  - Proposed solution : compression-based NN-CP decomposition.

Algorithms

- > The CP optimization problem is highly non-convex.
- > Yet many algorithms provide rather precise computation.
- > These algorithms can be divided into two main classes :
  - *All-at-once gradient-based descent, e.g.* : all CP parameters are updated at the same time using a gradient scheme and non-negativity constraints are implemented through barriers or soft penalizations.
  - *Alternating minimization* : the cost function is minimized in an alternating way for each factor ($\mathbf{A}$, $\mathbf{B}$ or $\mathbf{C}$) while the others are fixed.
- > Drawback : These algorithms are computationally expensive $\rightarrow$ Useless for large tensors.
  - Proposed solution : compression-based NN-CP decomposition.

Algorithms

- The CP optimization problem is highly non-convex.
- Yet many algorithms provide rather precise computation.
- These algorithms can be divided into two main classes :
  - *All-at-once gradient-based descent, e.g.* : all CP parameters are updated at the same time using a gradient scheme and non-negativity constraints are implemented through barriers or soft penalizations.
  - *Alternating minimization* : the cost function is minimized in an alternating way for each factor ($\mathbf{A}$, $\mathbf{B}$ or $\mathbf{C}$) while the others are fixed.
- Drawback : These algorithms are computationally expensive $\rightarrow$ Useless for large tensors.
  - Proposed solution : compression-based NN-CP decomposition.

Algorithms

> The CP optimization problem is highly non-convex.

> Yet many algorithms provide rather precise computation.

> These algorithms can be divided into two main classes :

- *All-at-once gradient-based descent, e.g.* : all CP parameters are updated at the same time using a gradient scheme and non-negativity constraints are implemented through barriers or soft penalizations.
- *Alternating minimization* : the cost function is minimized in an alternating way for each factor ($\mathbf{A}$, $\mathbf{B}$ or $\mathbf{C}$) while the others are fixed.

> Drawback : These algorithms are computationally expensive $\rightarrow$ Useless for large tensors.

- Proposed solution : compression-based NN-CP decomposition.

- The CP optimization problem is highly non-convex.
- Yet many algorithms provide rather precise computation.
- These algorithms can be divided into two main classes :
  - *All-at-once gradient-based descent, e.g.* : all CP parameters are updated at the same time using a gradient scheme and non-negativity constraints are implemented through barriers or soft penalizations.
  - *Alternating minimization* : the cost function is minimized in an alternating way for each factor ($\mathbf{A}$, $\mathbf{B}$ or $\mathbf{C}$) while the others are fixed.
- Drawback : These algorithms are computationally expensive $\rightarrow$ Useless for large tensors.
  - Proposed solution : compression-based NN-CP decomposition.

Compression-based NN-CP decomposition

General approach [1] :

1. Compress the original tensor $\boldsymbol{\mathcal{X}}^{I,J,K}$ on a small tensor $\boldsymbol{\mathcal{G}}^{R_1,R_2,R_3}$.
   - Supposition : the compressed tensor $\boldsymbol{\mathcal{G}}$ contains almost the same information as the original tensor $\boldsymbol{\mathcal{X}}$.
2. Compute the CP decomposition of $\boldsymbol{\mathcal{G}}$.
   - Consider non-negativity constraints of $\boldsymbol{\mathcal{X}}$ in the optimization problem.
   - Estimate the compressed factors $\mathbf{A}_c^{R_1 \times R}$, $\mathbf{B}_c^{R_2 \times R}$ and $\mathbf{C}_c^{R_3 \times R}$.
3. Uncompress the estimated one-rank factors $\mathbf{A}_c$, $\mathbf{B}_c$ and $\mathbf{C}_c$ to obtain the original factors $\mathbf{A}^{I \times R}$, $\mathbf{B}^{J \times R}$ and $\mathbf{C}^{K \times R}$.

[1] Cohen, J., Cabral-Farias, R., Comon, P. ; "Fast Decomposition of Large Nonnegative Tensors," IEEE Signal Processing Letters, 22(7), pp. 862-866, 2015.

Compression-based NN-CP decomposition

General approach [1] :

1. Compress the original tensor $\mathcal{X}^{I,J,K}$ on a small tensor $\mathcal{G}^{R_1,R_2,R_3}$.
   - Supposition : the compressed tensor $\mathcal{G}$ contains almost the same information as the original tensor $\mathcal{X}$.
2. Compute the CP decomposition of $\mathcal{G}$.
   - Consider non-negativity constraints of $\mathcal{X}$ in the optimization problem.
   - Estimate the compressed factors $\mathbf{A}_c^{R_1\times R}$, $\mathbf{B}_c^{R_2\times R}$ and $\mathbf{C}_c^{R_3\times R}$.
3. Uncompress the estimated one-rank factors $\mathbf{A}_c$, $\mathbf{B}_c$ and $\mathbf{C}_c$ to obtain the original factors $\mathbf{A}^{I\times R}$, $\mathbf{B}^{J\times R}$ and $\mathbf{C}^{K\times R}$.

---

[1] Cohen, J., Cabral-Farias, R., Comon, P. ; "Fast Decomposition of Large Nonnegative Tensors," IEEE Signal Processing Letters, 22(7), pp. 862-866, 2015.

Compression-based NN-CP decomposition

General approach [1] :

1. Compress the original tensor $\boldsymbol{\mathcal{X}}^{I,J,K}$ on a small tensor $\boldsymbol{\mathcal{G}}^{R_1,R_2,R_3}$.
   - Supposition : the compressed tensor $\boldsymbol{\mathcal{G}}$ contains almost the same information as the original tensor $\boldsymbol{\mathcal{X}}$.
2. Compute the CP decomposition of $\boldsymbol{\mathcal{G}}$.
   - Consider non-negativity constraints of $\boldsymbol{\mathcal{X}}$ in the optimization problem.
   - Estimate the compressed factors $\mathbf{A}_c^{R_1 \times R}$, $\mathbf{B}_c^{R_2 \times R}$ and $\mathbf{C}_c^{R_3 \times R}$.
3. Uncompress the estimated one-rank factors $\mathbf{A}_c$, $\mathbf{B}_c$ and $\mathbf{C}_c$ to obtain the original factors $\mathbf{A}^{I \times R}$, $\mathbf{B}^{J \times R}$ and $\mathbf{C}^{K \times R}$.

[1] Cohen, J., Cabral-Farias, R., Comon, P. ; "Fast Decomposition of Large Nonnegative Tensors," IEEE Signal Processing Letters, 22(7), pp. 862-866, 2015.

Outline

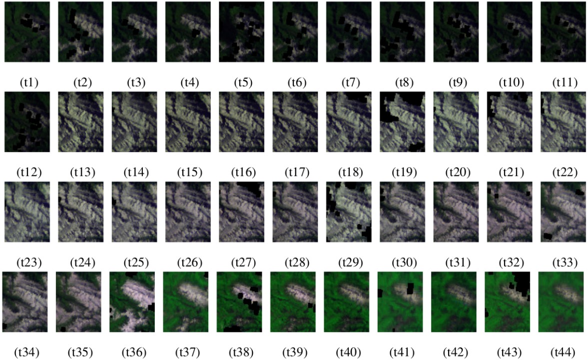
1 Motivation

2 Non-negative CP decomposition

3 Experimental results with time series

4 What about hyperspectral data cubes ?

- > 44 daily MODIS acquisitions with low cloud coverage.
- > Pre-processed to increase spatial resolution (250m).
- > Dimensions : $4800 \times 7 \times 44$.

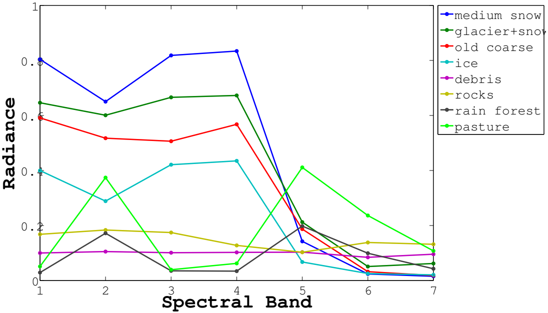


[2] Veganzones, M.A. ; Cohen, J. ; Cabral-Farias, R. ; Chanussot, J. ; Comon, P. ; "Nonnegative tensor CP decomposition of hyperspectral data," IEEE Transactions on Geoscience and Remote Sensing, 54(5), pp. 2577-2588, 2016.

Experimental methodology (I)

> Competing algorithms :
  - Compression-based CP algorithms : CCG and ProCoALS.
  - Conventional CP algorithm : ANLS.
  - Conventional day-basis full-constrained spectral unmixing (FCLSU).

> Groundtruth : 8 spectra (on-field acquisitions).



Experimental methodology (II)

- > We run 50 Monte Carlo runs for each of the algorithms for a set of different rank values in the range $R \in [5, 15]$.
- > For the compression-based CP algorithms, the compressed tensor, $\boldsymbol{\mathcal{X}}_c$, has dimensions $175 \times 7 \times 25$.
- > Quality measures :
  - Reconstruction error : average RMSE.
  - Angular error w.r.t. the groundtruth spectral signatures.
  - Linear Pearson correlation w.r.t. the spatial abundances obtained by the FCLSU.

Experimental methodology (II)

> We run 50 Monte Carlo runs for each of the algorithms for a set of different rank values in the range $R \in [5, 15]$.

> For the compression-based CP algorithms, the compressed tensor, $\boldsymbol{\mathcal{X}}_c$, has dimensions $175 \times 7 \times 25$.

> Quality measures :

- Reconstruction error : average RMSE.
- Angular error w.r.t. the groundtruth spectral signatures.
- Linear Pearson correlation w.r.t. the spatial abundances obtained by the FCLSU.

Experimental methodology (II)

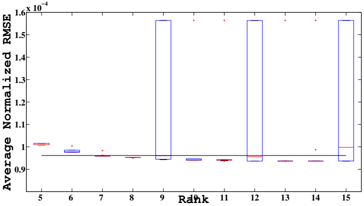
> We run 50 Monte Carlo runs for each of the algorithms for a set of different rank values in the range $R \in [5, 15]$.

> For the compression-based CP algorithms, the compressed tensor, $\boldsymbol{\mathcal{X}}_c$, has dimensions $175 \times 7 \times 25$.

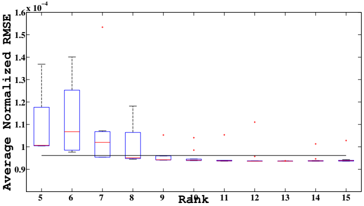
> Quality measures :

- Reconstruction error : average RMSE.
- Angular error w.r.t. the groundtruth spectral signatures.
- Linear Pearson correlation w.r.t. the spatial abundances obtained by the FCLSU.

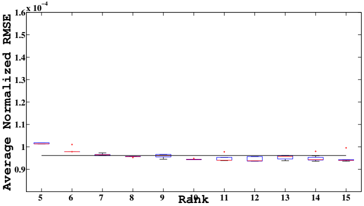
Reconstruction errors



(a) ANLS

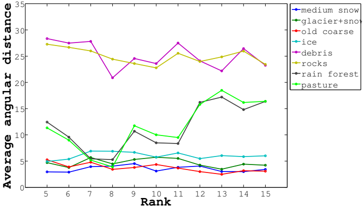


(b) CCG

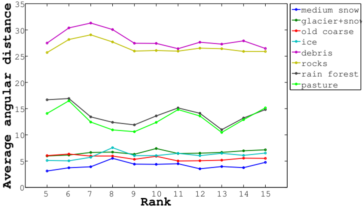


(c) ProCoALS

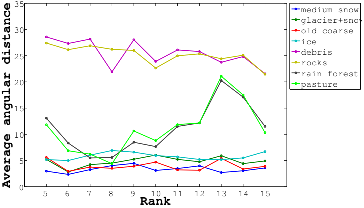
Spectral factors interpretation



(a) ANLS

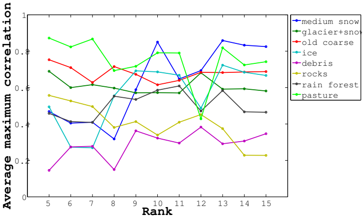


(b) CCG

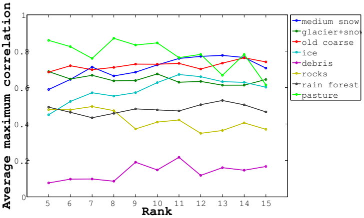


(c) ProCoALS

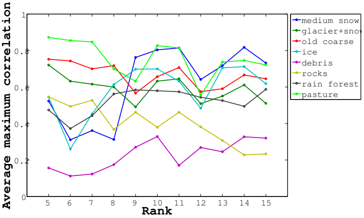
Spatial factors interpretation



(a) ANLS



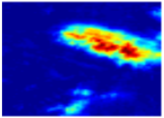
(b) CCG



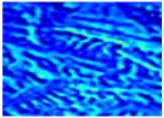
(c) ProCoALS

Spatial factors interpretation

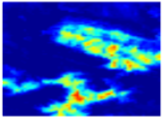
> Best run with rank $R = 8$.



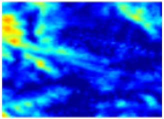
Permanent snow/ice



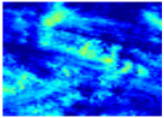
Seasonal snow/ice (type 1)



Seasonal snow/ice (type 2)



Seasonal vegetation (type 1)



Seasonal vegetation (type 2)

Outline

Are hyperspectral data cubes tensors ?

- Hyperspectral images are often provided, not as non-negative matrices, $\mathbf{X} \in \mathbb{R}_{+}^{I \times J}$, but as data cubes, $\boldsymbol{\mathcal{X}} \in \mathbb{R}_{+}^{I_r \times I_c \times J}$, where I_r and I_c denote the number of rows and columns respectively, being $I = I_r I_c$.
- It is possible to think of the data cube as a tensor representation of the image, allowing to apply tensor analysis techniques.
- This is fine ! $\rightarrow$ Employed for de-noising, compression, ...
- ... but, in general, it is not possible to suppose a low rank, that is, a small R.
- Proposed solution : patch-tensors [3].

[3] Veganzones, M.A. ; Cohen, J. ; Cabral-Farias, R. ; Usevich, K., Drumetz, L. ; Chanussot, J. ; Comon, P. ; "Canonical Polyadic decomposition of hyperspectral patch-tensors," 2016 EUSIPCO Conference (submitted).

Are hyperspectral data cubes tensors ?

> Hyperspectral images are often provided, not as non-negative matrices, $\mathbf{X} \in \mathbb{R}_+^{I \times J}$, but as data cubes, $\boldsymbol{\mathcal{X}} \in \mathbb{R}_+^{I_r \times I_c \times J}$, where I_r and I_c denote the number of rows and columns respectively, being $I = I_r I_c$.

> It is possible to think of the data cube as a tensor representation of the image, allowing to apply tensor analysis techniques.

> This is fine ! $\rightarrow$ Employed for de-noising, compression, ...

> ... but, in general, it is not possible to suppose a low rank, that is, a small R.

> Proposed solution : patch-tensors [3].

[3] Veganzones, M.A. ; Cohen, J. ; Cabral-Farias, R. ; Usevich, K., Drumetz, L. ; Chanussot, J. ; Comon, P. ; "Canonical Polyadic decomposition of hyperspectral patch-tensors," 2016 EUSIPCO Conference (submitted).

Are hyperspectral data cubes tensors ?

- Hyperspectral images are often provided, not as non-negative matrices, $\mathbf{X} \in \mathbb{R}_{+}^{I \times J}$, but as data cubes, $\boldsymbol{\mathcal{X}} \in \mathbb{R}_{+}^{I_r \times I_c \times J}$, where I_r and I_c denote the number of rows and columns respectively, being $I = I_r I_c$.
- It is possible to think of the data cube as a tensor representation of the image, allowing to apply tensor analysis techniques.
- This is fine ! $\rightarrow$ Employed for de-noising, compression, ...
- ... but, in general, it is not possible to suppose a low rank, that is, a small R.
- Proposed solution : patch-tensors [3].

[3] Veganzones, M.A. ; Cohen, J. ; Cabral-Farias, R. ; Usevich, K., Drumetz, L. ; Chanussot, J. ; Comon, P. ; "Canonical Polyadic decomposition of hyperspectral patch-tensors," 2016 EUSIPCO Conference (submitted).

Are hyperspectral data cubes tensors ?

> Hyperspectral images are often provided, not as non-negative matrices, $\mathbf{X} \in \mathbb{R}_+^{I \times J}$, but as data cubes, $\boldsymbol{\mathcal{X}} \in \mathbb{R}_+^{I_r \times I_c \times J}$, where I_r and I_c denote the number of rows and columns respectively, being $I = I_r I_c$.

> It is possible to think of the data cube as a tensor representation of the image, allowing to apply tensor analysis techniques.

> This is fine ! $\rightarrow$ Employed for de-noising, compression, ...

> ... but, in general, it is not possible to suppose a low rank, that is, a small R.

> Proposed solution : patch-tensors [3].

---

[3] Veganzones, M.A. ; Cohen, J. ; Cabral-Farias, R. ; Usevich, K., Drumetz, L. ; Chanussot, J. ; Comon, P. ; "Canonical Polyadic decomposition of hyperspectral patch-tensors," 2016 EUSIPCO Conference (submitted).

Are hyperspectral data cubes tensors ?

- Hyperspectral images are often provided, not as non-negative matrices, $\mathbf{X} \in \mathbb{R}_+^{I \times J}$, but as data cubes, $\boldsymbol{\mathcal{X}} \in \mathbb{R}_+^{I_r \times I_c \times J}$, where I_r and I_c denote the number of rows and columns respectively, being $I = I_r I_c$.
- It is possible to think of the data cube as a tensor representation of the image, allowing to apply tensor analysis techniques.
- This is fine ! $\rightarrow$ Employed for de-noising, compression, ...
- ... but, in general, it is not possible to suppose a low rank, that is, a small R.
- Proposed solution : patch-tensors [3].

[3] Veganzones, M.A. ; Cohen, J. ; Cabral-Farias, R. ; Usevich, K., Drumetz, L. ; Chanussot, J. ; Comon, P. ; "Canonical Polyadic decomposition of hyperspectral patch-tensors," 2016 EUSIPCO Conference (submitted).

Non-negative CP decomposition of hyperspectral data

Miguel A. Veganzones, Jeremy Cohen, Rodrigo Cabral-Farias and Pierre Comon

gipsa-lab, CNRS, UMR 5216, F-38420, Grenoble, France
miguel-angel.veganzones@gipsa-lab.fr

ERC AdG-2013-320594 DECODA