Sparse component analysis for unsupervised multichannel image decomposition

Ivica Kopriva

Ruđer Bošković Institute

e-mail: ikopriva@irb.hr ikopriva@gmail.com
Web: http://www.lair.irb.hr/ikopriva/

July 11 2014.

Acknowledgments:

Croatian Science Foundation Grant 9.01/232 "Nonlinear component analysis with applications in chemometrics and pathology"**.**

Ministry of Science, Education and Sports, Republic of Croatia, Grant 098-0982903-2558 "Multispectral data analysis".

Talk outline

- Instantaneous blind source separation (BSS):– problem definition and overview of main methods
- underdetermined BSS (uBSS) and sparse component analysis (SCA):
  - asymptotic results from compressed sensing theory
  - SCA by data clustering and L_p-norm minimization
  - SCA by sparseness constrained non-negative matrix factorization (NMF)
- Applications in hyper-, multispectral and magnetic resonance image decomposition.

Blind separation of sources

- Linear problems
  - Dynamic problems
  - Static problems
    - Over- and determined problems
      - Principal component analysis
      - Independent component analysis
      - Dependent component analysis
      - Nonnegative matrix factorization
      - Nonnegative tensor factorization
    - Underdetermined problems
      - Sparse component analysis:
      - * clustering + lp ($0<p\leq1$) min
      - * Hierarchical nonnegative matrix factorization
- Nonlinear problems

Blind Source Separation – linear static problem

Recovery of signals from their multichannel linear superposition using minimum of *a priori* information i.e. multichannel measurements only.

Problem:

$\mathbf{X}=\mathbf{A}\mathbf{S}$ $\mathbf{X}\in R^{NxT}$, $\mathbf{A}\in R^{NxM}$, $\mathbf{S}\in R^{MxT}$

N - number of sensors/mixtures;
M - unknown number of sources
T - number of samples/observations

Goal: find **S**, **A** and number of sources M based on **X** only.

A. Hyvarinen, J. Karhunen, E. Oja, "Independent Component Analysis," John Wiley, 2001.
A. Cichocki, S. Amari, "Adaptive Blind Signal and Image Processing," John Wiley, 2002.
P. Comon, C. Jutten, editors, "Handbook of Blind Source Separation," Elsevier, 2010.

Blind Source Separation – linear static problem

X=**AS** and **X**=**ATT**$^{-1}$**S** are equivalent for any square invertible matrix **T**. There are infinitely many pairs (**AT**, **T**$^{-1}$**S**) satisfying linear mixture model **X**=**AS**. Solutions unique up to permutation and scaling indeterminacies, **T**=**PΛ**, are meaningful. For such solutions constraints must be imposed on **A** and/or **S**.

Independent component analysis (**ICA**) solves BSS problem provided that: source signals **S** are statistically independent and non-Gaussian; mixing matrix **A** is full column rank i.e. M≤N.

Dependent component analysis (DCA) improves accuracy of ICA when sources are not statistically independent. Linear high-pass filtering type of preprocessing transform is applied row-wise to **X**: L(**X**)=**A**L(**S**). ICA is applied to L(**X**) to estimate **A** and L(**S**). **S** is estimated from **S**≈**A**$^{-1}$**X**.

Matlab implementation of many ICA algorithms can be found in the ICALAB:
http://www.bsp.brain.riken.go.jp/ICALAB/

Blind Source Separation – linear static problem

Sparse component analysis (SCA) solves BSS problem imposing sparseness constraints on source signals **S**. M can be less than, equal to or greater than N.

Thus, SCA can be used to solve underdetermined BSS problems where number of source signals is greater than number of mixtures.

Nonnegative matrix factorization (NMF) solves BSS problem imposing nonnegativity, sparseness, smoothness or constraints on source signals. NMF algorithms that enforce sparse decomposition of **X** can be seen as SCA algorithms.

Matlab implementation of many NMF algorithms can be found in the NMFLAB:
http://www.bsp.brain.riken.jp/ICALAB/nmflab.html

A. Cichocki, R. Zdunek, A. H. Phan, S. Amari, Nonnegative Matrix and Tensor Factorizations-Applications to Exploratory Multi-way Data Analysis and Blind Source Separation, John Wiley, 2009.

Underdetermined BSS

- SCA-based solution of the uBSS problem is obtained in two stages:

  1) estimate basis or mixing matrix **A** using data clustering.

  2) estimating sources, with estimated **A**, one at a time $\mathbf{s}_t$, t=1,…,T or simultaneously solving underdetermined linear systems of equations $\mathbf{x}_t = \mathbf{A}\mathbf{s}_t$. Provided that $\mathbf{s}_t$ is sparse enough, solution is obtained at the minimum of L_p-norm, $\|\mathbf{s}_t\|_p$, $0 \le p \le 1$.

Here: $\|\mathbf{s}_t\|_p = \left(\sum_{m=1}^{M} |s_{mt}|^p \right)^{1/p}$.

- NMF-based solution yields **A** and **S** simulatneously through sparseness and nonnegativity constrained factorization of **X**.

When uBSS problems can(not) be solved?

Let us focus on underdetermined linear system:

$$\mathbf{x}=\mathbf{A}\mathbf{s},\ \mathbf{x}\in R^N,\ \mathbf{s}\in R^M,\ M>N$$

Let **s** be K-sparse i.e. $K=||\mathbf{s}||_0$.

Provided that **A** is random, with entries from Gaussian or Bernoulli distributions, compressed sensing theory has established necessary and sufficient condition on N, M and K to obtain, with probability one, unique solution at the minimum of L_1-norm of **s,** ref. [a]:

$$N\approx K\log(M/K)$$

a) Candès E, Tao T. Near optimal signal recovery from random projections: universal encoding strategy?. *IEEE Trans. Information Theory* 2006; **52**: 5406-5425.

When uBSS problems can(not) be solved?

When L_p-norm of **s** is minimized, the condition on number of measurements N is:

$$N \geq C_1(p)K + pC_2(p)K\log(M/K),$$

where $C_1(p)$ and $C_2(p)$ are norm-dependent constants, ref [a]:

a) Chartran R, Staneva V. Restricted isometry properties and nonconvex compressive sensing. *Inverse Problems* 2008; 24: 035020 (14 pages).

Hence, $\lim_{p\to 0} N \geq C_1(0)K$. Thus, for p=0 number of measurements N does not depend on M!!!! That explains good results of L_0-norm constrained algorithms when compared against L_1-norm constrained algorithms when K is increasing, ref [b, c]:

b) Pehaz R, Pernkopf, F. Sparse nonnegative matrix factorization with ℓ^0-constraints. *Neurocomputing*. 2012; **80**: 38-46.
c) Mohimani H, Babie-Zadeh B, Jutten C. A Fast Approach for Overcomplete Sparse Decomposition Based on ℓ_0 Smoothed Norm. *IEEE Trans. Sig. Proc*. 2009; **57**(1): 289-301.

uBSS – L_p norm minimization: $0< p \leq 1$

•Signal **s** is K-sparse if it has K non-zero components, i.e. $K=||\mathbf{s}||_0$.

•If uBSS problem is not sparse in original domain it ought to be transformed in domain where enough level of sparseness can be achieved: $T(\mathbf{x})=\mathbf{A}T(\mathbf{s})$.

•Time-frequency and time-scale (wavelet) bases are employed for this purpose quite often.

•In addition to sparseness requirement on **s** certain degree of incoherence of the mixing matrix **A** is required as well. Mutual coherence is defined as the largest absolute and normalized inner product between different columns in **A**, what reads as

$$\mu\ \mathbf{A} = \max_{1\leq i,j\leq M \text{ and } i\neq j} \frac{\left|\mathbf{a}_i^T \mathbf{a}_j\right|}{\left\|\mathbf{a}_i\right\|\left\|\mathbf{a}_j\right\|}$$

uBSS – L_p norm minimization: $0< p \leq 1$

The mutual coherence provides a worst case measure of similarity between the basis vectors. It indicates how much two closely related vectors may confuse any pursuit algorithm (solver of the underdetermined linear system of equations). The worst-case perfect recovery condition for **s** relates sparseness requirement on **s** and coherence of **A**, ref. [a,b]:

$$\|\mathbf{s}\|_0 < \frac{1}{2}\left(1+\frac{1}{\mu\ \mathbf{A}}\right)$$

a) R. Gribonval and M. Nielsen, "Sparse representations in unions of bases," *IEEE Transactions on Information Theory* **49**, 3320-3325 (2003).
b) J. A. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Transactions on Information Theory* **50**, 2231-2242 (2004).

In: I. F. Gorodnitsky and B. D. Rao, "Sparse signal reconstruction from limited data using FOCUSS, a re-weighted minimum norm algorithm," *IEEE Trans. Signal Process*., vol.45, no.3, pp. 600–616, Mar. 1997.

another uniqueness theorem has been stated. If **A** has unique representation property, that is if all $N\times N$ sub-matrices are full rank, the unique solution of **x=As** exists if: $\|\mathbf{s}\|_0 \leq N/2$.

uBSS – L_p norm minimization: $0< p \leq 1$

Solving underdetermined system of linear equations **x**=**As** amounts to solving:

$$\hat{\mathbf{s}}(t) = \arg\min_{\mathbf{s}(t)} \|\mathbf{s}(t)\|_0 \quad \text{s.t. } \hat{\mathbf{A}}\mathbf{s}(t) = \mathbf{x}(t) \quad \forall t = 1,\ldots,T$$

or for problems with noise or approximation error:

$$\hat{\mathbf{s}}(t) = \arg\min_{\mathbf{s}(t)} \frac{1}{2}\left\|\hat{\mathbf{A}}\mathbf{s}(t) - \mathbf{x}(t)\right\|_2^2 + \lambda\|\mathbf{s}(t)\|_0 \qquad \forall t = 1,\ldots,T$$

$$\hat{\mathbf{s}}(t) = \arg\min_{\mathbf{s}(t)} \|\mathbf{s}(t)\|_0 \quad \text{s.t. } \left\|\hat{\mathbf{A}}\mathbf{s}(t) - \mathbf{x}(t)\right\|_2^2 \leq \varepsilon \qquad \forall t = 1,\ldots,T$$

Direct minimization of L_0–norm of **s** is combinatorial problem that is NP-hard. For larger dimension M it becomes computationally infeasible.

uBSS – L_1 norm minimization

Replacement of L_0-norm by L_1-norm is done quite often. That is known as *convex relaxation* of the minimum L_0-norm problem. It leads to linear program:

$$\hat{\mathbf{s}}(t) = \arg\min_{\mathbf{s}(t)} \sum_{m=1}^{\hat{M}} s_m \quad t \quad \text{s.t. } \hat{\mathbf{A}}\mathbf{s}(t) = \mathbf{x}(t) \quad \forall t = 1,\ldots, \quad \text{s.t. } \mathbf{s}(t) \geq 0$$

L_1-regularized least square problem ref.[a,b]:

$$\hat{\mathbf{s}}(t) = \arg\min_{\mathbf{s}(t)} \frac{1}{2}\left\|\hat{\mathbf{A}}\mathbf{s}(t) - \mathbf{x}(t)\right\|_2^2 + \lambda\left\|\mathbf{s}(t)\right\|_1 \qquad \forall t = 1,\ldots,T$$

and L_2-regularized linear problem [b,c]:

$$\hat{\mathbf{s}}(t) = \arg\min_{\mathbf{s}(t)} \left\|\mathbf{s}(t)\right\|_1 \quad \text{s.t. } \left\|\hat{\mathbf{A}}\mathbf{s}(t) - \mathbf{x}(t)\right\|_2^2 \leq \varepsilon \qquad \forall t = 1,\ldots,T$$

a) S..J. Kim, K. Koh, M. Lustig, S. Boyd, D. Gorinevsky, "An Interior-Point Method for Large-Scale -Regularized Least Squares," *IEEE Journal of Selected Topics in Signal Processing* **1**, 606-617 (2007), **http://www.stanford.edu/~boyd/l1_ls/.**
b) E. van den Berg, M.P. Friedlander, "Probing the Pareto Frontier for Basis Pursuit Solutions," SIAM J. Sci. Comput. 31, 890-912 (2008).
c) M.A.T. Figuiredo, R.D. Nowak, S.J. Wright, "Gradient Projection for Sparse Reconstruction: Application to Compressed Sensing and Other Inverse Problems," *IEEE Journal on Selected Topics in Signal Processing* **1**, 586-597 (2007).

Iterative soft/hard thresholding

L_1-regularized least square problem:

$$\hat{\mathbf{s}}(t) = \arg\min_{\mathbf{s}(t)} \frac{1}{2}\left\|\hat{\mathbf{A}}\mathbf{s}(t) - \mathbf{x}(t)\right\|_2^2 + \lambda\left\|\mathbf{s}(t)\right\|_1 \qquad \forall t = 1,\ldots,T$$

can be reformulated within analytic soft thresholding representation theory [a, b]:

$$B(\mathbf{s}^{(k)}(t)) = \mathbf{s}^{(k)} + \mathbf{A}^T \quad \mathbf{x}(t) - \mathbf{A}\mathbf{s}^{(k)}(t)$$

$$s_m^{(k+1)}(t) = \begin{cases} B(\mathbf{s}^{(k)}(t))_m - sign(B(\mathbf{s}^{(k)}(t))_m)\lambda/2, & \left|B(\mathbf{s}^{(k)}(t))_m\right| > \lambda/2 \\ 0, \quad otherwise \end{cases}$$

where $\lambda=\sigma^2$ provided that error term (noise) has normal distribution. Otherwise some kind of cross-validation (trial and error) needs to be applied.

a) D. L. Donoho, Denoising by soft-thresholding, *IEEE Trans. Information Theory*, **41** (1995), 613-627.
b) I. Daubechies, M. Defrise, D.M. Christine, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint, *Comm. Pure and Appl. Math*., **LVII** (2004) 1413-1457.

Iterative soft/hard thresholding

L_0-regularized least square problem:

$$\hat{\mathbf{s}}(t) = \arg\min_{\mathbf{s}(t)} \frac{1}{2}\left\|\hat{\mathbf{A}}\mathbf{s}(t) - \mathbf{x}(t)\right\|_2^2 + \lambda\left\|\mathbf{s}(t)\right\|_0 \qquad \forall t = 1,\ldots,T$$

can be reformulated within analytic hard thresholding representation theory [a]:

$$s_m^{(k+1)}(t) = \begin{cases} s_m^{(k)}(t) - sign(s_m^{(k)}(t))\lambda/2, & \left|s_m^{(k)}(t)\right| > \lambda/2 \\ 0, \quad otherwise \end{cases}$$

where $\lambda=\sigma^2$ provided that error term (noise) has normal distribution. Otherwise some kind of cross-validation (trial and error) needs to be applied.

a) R. Chartrand, V. Staneva, Restricted isometry properties and nonconvex compressive sensing, *Inverse Problems*, **24** (2008) 1-14.

Iterative soft/hard thresholding

Fast Iterative Shrinkage Thresholding (Fast_IST) algorithm:

Beck, M. Teboulle, "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems," *SIAM J. Image. Sci.*, Vol**. 2**, No. 1, pp. 183-202, 2009.

MATLAB code is freely available for download at:

http://ie.technion.ac.il/Home/Users/becka.html

This algorithm uses L_1–based regularization of least square approximation problem.

$$\hat{\mathbf{S}} = \min_{\mathbf{S}} \left\{ \frac{1}{2} \left\| \hat{\mathbf{A}}\mathbf{S} - \mathbf{X} \right\|_F^2 + \lambda \left\| \mathbf{S} \right\|_1 \right\}$$

The method can be easily implemented in batch mode to solve all the *T* equations simultaneously. The method also shrinks to zero small nonzero elements of **S** that are influenced by noise. Regularization parameter λ has to be determined through cross-validation or experience.

Estimation of mixing matrix: clustering

F. M. Naini, G. H. Mohimani, M. Babaie-Zadeh, C. Jutten, "Estimating the mixing matrix in sparse component analysis (SCA) based on partial k-dimensional subspace clustering," *Neurocomputing*, vol. 71, pp. 2330-2343, 2008.

- Assuming unit L_2-norm of $\boldsymbol{a}_m$ we can parameterize column vectors in 3D space by means of azimuth and elevation angles

$$\mathbf{a}_m = [\cos(\varphi_m)\sin(\theta_m) \;\; \sin(\varphi_m)\sin(\theta_m) \;\; \cos(\theta_m)]^{\mathrm{T}}$$

- Due to nonnegativity constraints both angles are confined in $[0,\pi/2]$. Now estimation of $\mathbf{A}$ and M is obtained by means of data clustering algorithm:

- We remove all data points close to the origin for which applies: $\left|\mathbf{x}(t)\right|_2 \le \varepsilon \quad t=1}^{T$ where ε represents some predefined threshold.

- Normalize to unit L_2-norm remaining data points $\mathbf{x}(t)$, i.e., $\mathbf{x}\ t \rightarrow \mathbf{x}\ t \ / \left|\mathbf{x}\ t\ \right|_2 \quad t=1}^{\bar{T}$

Estimation of mixing matrix: clustering

• Calculate function $f(\mathbf{a})$:

$$f\ \mathbf{a} = \sum_{t=1}^{\bar{T}} \exp\left(-\frac{d^2\ \mathbf{x}(t), \mathbf{a}}{2\sigma^2} \right)$$

where $d\ \mathbf{x}(t), \mathbf{a} = \sqrt{1 - \mathbf{x}(t) \cdot \mathbf{a}^{\ 2}}$ and $\mathbf{x}(t) \cdot \mathbf{a}$ denotes inner product. Parameter σ is called dispersion. If set to sufficiently small value, in our experiments this turned out to be $\sigma \approx 0.05$, the value of the function $f(\mathbf{a})$ will approximately equal the number of data points close to $\mathbf{a}$. Thus by varying mixing angles $0 \leq \varphi, \theta \leq \pi/2$ we effectively cluster data.

• Number of peaks of the function $f(\mathbf{a})$ corresponds with the estimated number of materials M. Locations of the peaks correspond with the estimates of the mixing angles $\hat{\varphi}_m, \hat{\theta}_m \ _{m=1}^{M}$, i.e., mixing vectors $\hat{\mathbf{a}}_m \ \hat{\varphi}_m, \hat{\theta}_m \ _{m=1}^{M}$.

Estimation of the mixing matrix: clustering

• hierarchical clustering by MATLAB function `clusterdata`. It is assumed that number of clusters (sources) is given (known). The method is deterministic and memory demanding.

•*k*-means clustering by MATLAB function `kmeans`. It is assumed that a number of clusters *M* (corresponds with number of sources) is given. *k*-means clustering is a first order method and it is sensitive on initial choice of cluster centers (centroids).

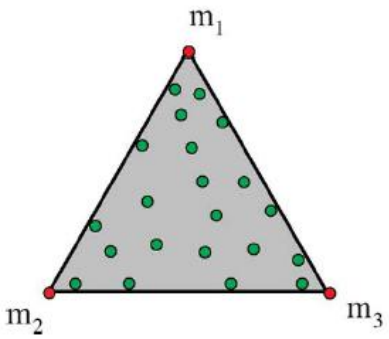
Mixing matrix estimation

N. Gillis and S.A. Vavasis, "Fast and Robust Recursive Algorithms for Separable Nonnegative Matrix Factorization", 2012. **http://arxiv.org/abs/1208.1237**

MATLAB Code: **https://sites.google.com/site/nicolasgillis/code.**

This method estimates the mixing matrix by generalizing some hyperspectral unmixing algorithms based on pure pixels (single source points) assumption.

The algorithm is recursive and fast (Fast_SepNMF), i.e. it estimates one mixing vector at a time. There are no parameters required to be chosen *a priori* or to be tuned. The method works when data matrix is approximately separable, i.e. pure pixels do not exist.The method identifies 'M' columns of data matrix whose convex hull has encompases the data.



Mixing matrix must be full rank, i.e. number of components has to be less than or equal to the number of mixtures.

Mixing matrix estimation

G. H. Ritter, G. Urcid, "A lattice matrix method for hyperspectral image unmixing," *Information Sciences*, vol. 181, pp. 1787-1803,2011.

MATLAB code:
http://www.ehu.es/ccwintco/index.php/Endmember_Induction_Algorithms_(EIAs)_for_MATLAB_and_SCILAB

Autonomous endmember determination algorithm using lattice associate memory (LAM) theory.

Unlike may methods in hyperspectral image analysis it does not assume/require existence of pure pixels. Instead, it searches for the least contaminated pixels.

Also, it does not require number of endmembers (sources) to be known in advance.

Nonnegative matrix factorization (NMF)

Nonnegative matrix factorization

Many BSS problems arising in imaging, chemo- and/or bioinformatics are described by superposition of non-negative latent variables (sources):

$$\mathbf{X} = \mathbf{AS} \quad \mathbf{X} \in \mathbb{R}_{0+}^{N \times T}, \ \mathbf{A} \in \mathbb{R}_{0+}^{N \times M} \ \textit{and} \ \mathbf{S} \in \mathbb{R}_{0+}^{M \times T}$$

where *N* represents number of sensors, *M* represents number of sources and *T* represents number of observations.

Thus, solution of related decomposition problem can be obtained by imposing non-negativity constraints on **A** and **S**, to narrow down number of possible decomposition of **X**. This leads to NMF algorithms.

Due to non-negativity constraints some other constraints (statistical independence) can be relaxed/replaced in applications where they are not fulfilled.

Nonnegative matrix factorization

Modern approaches to NMF problems have been initiated by Lee-Seung' Nature paper, ref. [a], where it is proposed to estimate **A** and **S** through alternative minimization procedure of the two possibly different cost functions:

Set Randomly initialize: $\mathbf{A}^{(0)}$, $\mathbf{S}^{(0)}$,

For *k=1,2,…, until convergence* ***do***

Step 1: $$\mathbf{S}^{(k+1)} = \arg\min_{s_{mt}\geq 0} D_{\mathbf{S}}\left(\mathbf{X}\middle\|\mathbf{A}^{(k)}\mathbf{S}\right)_{\mathbf{S}^{(k)}}$$

Step 2: $$\mathbf{A}^{(k+1)} = \arg\min_{a_{nm}\geq 0} D_{\mathbf{A}}\left(\mathbf{X}\middle\|\mathbf{A}\mathbf{S}^{(k+1)}\right)_{\mathbf{A}^{(k)}}$$

If both cost functions represent squared Euclidean distance (Froebenius norm) we obtain alternating least square (ALS) approach to NMF.

a) D D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature* **401** (6755), 788-791 (1999).

Nonnegative matrix factorization

ALS-based NMF:

$$\mathbf{A}^*, \mathbf{S}^* = \arg\min_{\mathbf{A},\mathbf{S}} D\left(\mathbf{X} \| \mathbf{AS}\right) = \frac{1}{2}\left\|\mathbf{X} - \mathbf{AS}\right\|_F^2 \quad s.t.\ \mathbf{A} \geq \mathbf{0}, \mathbf{S} \geq \mathbf{0}$$

- Minimization of the square of Euclidean norm of approximation error **E=X-AS** is, from the maximum likelihood viewpoint, justified only if error distribution is Gaussian:

$$p\left(\mathbf{X} \middle| \mathbf{A}, \mathbf{S}\right) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\left\|\mathbf{X} - \mathbf{AS}\right\|_2^2}{2\sigma^2}\right)$$

- In many instances non-negativity constraints imposed on **A** and **S** do not suffice to obtain solution that is unique up to standard BSS indeterminacies: permutation and scaling.

Nonnegative matrix factorization

In relation to original Lee-Seung NMF algorithm additional constraints are necessary to obtain factorization unique up to permutation and scaling. Generalization that involves constraints is given in [a]:

$$D\ \mathbf{X}\|\mathbf{AS}\ = \frac{1}{2}\|\mathbf{X}-\mathbf{AS}\|_F^2 + \alpha_{\mathbf{S}} J_{\mathbf{S}}(\mathbf{S}) + \alpha_{\mathbf{A}} J_{\mathbf{A}}(\mathbf{A})$$

where $J_{\mathbf{S}}(\mathbf{S}) = \sum_{m,t} s_{mt}$ and $J_{\mathbf{A}}(\mathbf{A}) = \sum_{n,m} a_{nm}$ are sparseness constraints that correspond with L_1-norm of **S** and **A** respectively. $\alpha_{\mathbf{S}}$ and $\alpha_{\mathbf{A}}$ are regularization constants. Gradient components in matrix form are:

$$\frac{\partial D\ \mathbf{A},\mathbf{S}}{\partial a_{nm}} = \left[-\mathbf{X}\mathbf{S}^{\mathrm{T}} + \mathbf{A}\mathbf{S}\mathbf{S}^{\mathrm{T}}\right]_{nm} + \alpha_{\mathbf{A}} \frac{\partial J_{\mathbf{A}}(\mathbf{A})}{\partial a_{nm}}$$

$$\frac{\partial D\ \mathbf{A},\mathbf{S}}{\partial s_{mt}} = \left[-\mathbf{A}^{\mathrm{T}}\mathbf{X} + \mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{S}\right]_{mt} + \alpha_{\mathbf{S}} \frac{\partial J_{\mathbf{S}}(\mathbf{S})}{\partial s_{mt}}$$

a) A. Cichocki, R. Zdunek, and S. Amari, "Csiszár's Divergences for Non-negative Matrix Factorization: Family of New Algorithms," *LNCS* **3889,** 32-39 (2006).

Nonnegative matrix factorization

Since NMF problem deals with non-negative variables the idea is to automatically ensure non-negativity of **A** and **S** through learning. That can be achieved by multiplicative learning equations:

$$\mathbf{A} \leftarrow \mathbf{A} \otimes \frac{\nabla_{\mathbf{A}}^{-} D(\mathbf{A},\mathbf{S})}{\nabla_{\mathbf{A}}^{+} D(\mathbf{A},\mathbf{S})} \qquad \mathbf{S} \leftarrow \mathbf{S} \otimes \frac{\nabla_{\mathbf{S}}^{-} D(\mathbf{A},\mathbf{S})}{\nabla_{\mathbf{S}}^{+} D(\mathbf{A},\mathbf{S})}$$

where $\otimes$ denotes entry-wise multiplication, $\nabla_{\mathbf{A}}^{-} D(\mathbf{A},\mathbf{S})$ and $\nabla_{\mathbf{A}}^{+} D(\mathbf{A},\mathbf{S})$ denote respectively negative and positive part of the gradient $\nabla_{\mathbf{A}} D(\mathbf{A},\mathbf{S})$. Likewise, $\nabla_{\mathbf{S}}^{-} D(\mathbf{A},\mathbf{S})$ and $\nabla_{\mathbf{S}} D(\mathbf{A},\mathbf{S})$ are negative and positive part of the gradient $\nabla_{\mathbf{S}}^{+} D(\mathbf{A},\mathbf{S})$.

When gradients converge to zero corrective terms converge to one. Since learning equations include multiplications and divisions of non-negative terms, non-negativity is ensured automatically.

Nonnegative matrix factorization

Multiplicative learning rules for NMF based on regularized squared L_2-norm of the approximation are obtained as:

$$\mathbf{A} \leftarrow \mathbf{A} \otimes \frac{\left[\mathbf{X}\mathbf{S}^{\mathrm{T}} - \alpha_{\mathbf{A}} \dfrac{\partial J_{\mathbf{A}}(\mathbf{A})}{\partial \mathbf{A}}\right]_{+}}{\mathbf{A}\mathbf{S}\mathbf{S}^{\mathrm{T}} + \varepsilon \mathbf{1}_{NM}} \qquad \mathbf{S} \leftarrow \mathbf{S} \otimes \frac{\left[\mathbf{A}^{\mathrm{T}}\mathbf{X} - \alpha_{\mathbf{S}} \dfrac{\partial J_{\mathbf{S}}(\mathbf{S})}{\partial \mathbf{S}}\right]_{+}}{\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{S} + \varepsilon \mathbf{1}_{MT}}$$

where $[x]_+=\max\{\varepsilon,x\}$ with small ε. For L_1-norm based regularization, derivatives of sparseness constraints in above expressions are equal to 1, i.e.:

$$\mathbf{A} \leftarrow \mathbf{A} \otimes \frac{\left[\mathbf{X}\mathbf{S}^{\mathrm{T}} - \alpha_{\mathbf{A}} \mathbf{1}_{NM}\right]_{+}}{\mathbf{A}\mathbf{S}\mathbf{S}^{\mathrm{T}} + \varepsilon \mathbf{1}_{NM}} \qquad \mathbf{S} \leftarrow \mathbf{S} \otimes \frac{\left[\mathbf{A}^{\mathrm{T}}\mathbf{X} - \alpha_{\mathbf{S}} \mathbf{1}_{MT}\right]_{+}}{\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{S} + \varepsilon \mathbf{1}_{MT}}$$

Hierarchical ALS NMF

Local or hierarchical ALS NMF algorithms were derived in [a, b, c]. They employ minimization of the global cost function to learn the mixing matrix and minimization of set of local cost functions to learn the sources. Global cost function can for example be squared Euclidean norm:

$$D\left(\mathbf{X}\|\mathbf{AS}\right)=\frac{1}{2}\left\|\mathbf{X}-\mathbf{AS}\right\|_F^2+\alpha_{\mathbf{A}}J_{\mathbf{A}}(\mathbf{A})$$

Local cost functions can be also squared Euclidean norms

$$D^{(m)}\left(\mathbf{X}^{(m)}\|\mathbf{a}_m\underline{\mathbf{s}}_m\right)=\frac{1}{2}\left\|\mathbf{X}^{(m)}-\mathbf{a}_m\underline{\mathbf{s}}_m\right\|_F^2+\alpha_{\mathbf{s}}^{(m)}J_{\mathbf{S}}(\underline{\mathbf{s}}_m)+\alpha_{\mathbf{a}}^{(m)}J_{\mathbf{a}}(\mathbf{a}_m)\quad m=1,\ldots,M$$

$$\mathbf{X}^{(m)}=\mathbf{X}-\sum_{j\neq m}\mathbf{a}_j\underline{\mathbf{s}}_j$$

a) A. Cichocki, R. Zdunek, S.I. Amari, Hierarchical ALS Algorithms for Nonnegative Matrix Factorization and 3D Tensor Factorization, *LNCS* **4666** (2007) 169-176

b) A. Cichocki, A-H. Phan, R. Zdunek, and L.-Q. Zhang, "Flexible component analysis for sparse, smooth, nonnegative coding or representation," *LNCS* **4984**, 811-820 (2008).

c) A. Cichocki, R. Zdunek, S. Amari, Nonnegative Matrix and Tensor Factorization, *IEEE Sig. Proc. Mag*. **25** (2008) 142-145.

Hierarchical ALS NMF

Minimization of above cost functions in ALS manner with L_1-based sparseness constraints imposed on **A** and/or **S** yields

$$\underline{\mathbf{s}}_m \leftarrow \left[\mathbf{a}_m^{\mathrm{T}}\mathbf{X}^{(m)} - \alpha_{\mathbf{s}}^{(m)}\mathbf{1}_{1\times T}\right]_+ \quad \begin{matrix} M \\ m=1 \end{matrix}$$

$$\mathbf{A} \leftarrow \left[\ \mathbf{X}\mathbf{S}^{\mathrm{T}} - \alpha_{\mathbf{A}}\mathbf{1}_{N\times M} \quad \mathbf{S}\mathbf{S}^{\mathrm{T}} + \lambda\mathbf{I}_M\ \right]_+^{-1}$$

$$\mathbf{a}_m \leftarrow \mathbf{a}_m \Big/ \left\|\mathbf{a}_m\right\|_2 \quad \begin{matrix} M \\ m=1 \end{matrix}$$

where $\mathbf{I}_{1\times T}$ is an M×M identity matrix, $\mathbf{1}_{1\times T}$ and $\mathbf{1}_{N\times M}$ are row vector and matrix with all entries equal to one and $[\xi]_+=\max\{\varepsilon,\xi\}$ (e.g., $\varepsilon=10^{-16}$).

Regularization constant λ changes as a function of the iteration index as $\lambda_k = \lambda_0 \exp\ -k/\tau$ (with $\lambda_0 = 100$ and $\tau = 0.02$ in the experiments).

Multilayer NMF

Additional improvement in the performance of the NMF algorithms is obtained when they are applied in the multilayer mode [a,b], whereas sequential decomposition of the nonnegative matrices is performed as follows.

In the first layer, the basic approximation decomposition is performed:

$$\mathbf{X} \cong \mathbf{A}^{(1)}\mathbf{S}^{(1)} \in \mathbb{R}_{0+}^{N\times T}$$

In the second layer result from the first layer is used to build up new input data matrix for the second layer $\mathbf{X} \leftarrow \mathbf{S}^{(1)} \in \mathbb{R}_{0+}^{M\times T}$. This yields $\mathbf{X}^{(1)} \cong \mathbf{A}^{(2)}\mathbf{S}^{(2)} \in \mathbb{R}_{0+}^{M\times T}$.

After *L* layers data decomposes as follows: $\mathbf{X} \cong \mathbf{A}^{(1)}\mathbf{A}^{(2)}\cdots\mathbf{A}^{(L)}\mathbf{S}^{(L)}$.

a) A. Cichocki, and R. Zdunek, "Multilayer Nonnegative Matrix Factorization," *El. Letters* **42**, 947-948 (2006).
b) A. Cichocki, R. Zdunek, A. H. Phan, S. Amari, *Nonnegative Matrix and Tensor Factorizations-Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*, John Wiley, 2009.

Multi-start initialization for NMF algorithms

Combined optimization of the cost function D**(X||AS)** with respect to **A** and **S** is non-convex optimization problem. Hence, some strategy is necessary to decrease probability that optimization process will get stuck in some local minima. Such procedure is outlined with the following pseudo code: Select R- number of restarts, K_i number of alternating steps, K_f number of final alternating steps.

for r =1,…,R **do**

Initialize randomly $\boldsymbol{A}^{(0)}$ *and* $\boldsymbol{S}^{(0)}$

$\{\boldsymbol{A}^{(r)},\boldsymbol{S}^{(r)}\} \leftarrow nmf\_algorithm(\boldsymbol{X},\boldsymbol{A}^{(0)},\boldsymbol{S}^{(0)},K_i);$

compute $d=D(\boldsymbol{X}||\boldsymbol{A}^{(r)}\boldsymbol{S}^{(r)});$

end

$r_{min}=argmin_{1\le n\le R} d_r;$

$\{\boldsymbol{A},\boldsymbol{S}\} \leftarrow nmf\_algorithm(\boldsymbol{X},\boldsymbol{A}^{(rmin)},\boldsymbol{S}^{(rmin)},K_f);$

Non-negative matrix under-approximation (NMU)

NMF algorithms outlined befor require a priori knowledge of sparseness related regularization (trade off) constant.

A sequential approach to NMF has been recently proposed in [a] by estimating rank-1 one factors $\mathbf{a}_m\mathbf{s}_m$ one at a time. Each time $\mathbf{a}_m\mathbf{s}_m$ is estimated it is removed from $\mathbf{X} \rightarrow \mathbf{X}-\mathbf{a}_m\mathbf{s}_m$. To prevent subtraction from being negative the under-approximation constraint is imposed on $\mathbf{a}_m\mathbf{s}_m$: $\mathbf{a}_m\mathbf{s}_m \leq \mathbf{X}$.

Hence, the NMU algorithm is obtained as a solution of:

$$\mathbf{A}^*, \mathbf{S}^* = \arg\min_{\mathbf{A},\mathbf{S}} \frac{1}{2}\left\|\mathbf{X}-\mathbf{AS}\right\|_F^2 \; s.t. \; \mathbf{A} \geq \mathbf{0}, \mathbf{S} \geq \mathbf{0}, \; \mathbf{AS} \leq \mathbf{X}.$$

a) N. Gillis, and F. Glineur, "Using underapproximations for sparse nonnegative matrix factorization," *Patt. Recog.*, vol. 43, pp. 1676-1687, 2010.

Non-negative matrix under-approximation (NMU)

Theorem 1 in [a] proves that number of nonzero entries in **A** and **S** is less than in **X**. Thus, the underapproximation constraint ensures sparse (parts based) factorization of **X**. This, however, does not imply that **A** and **S** obtained by enforcing underapproximation constrain yields the sparseset decomposition of **X**.

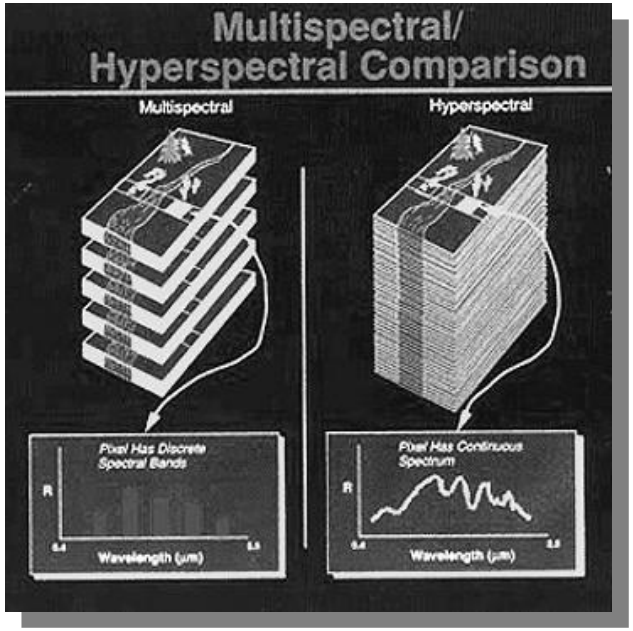
However, since no explicit regularization is used there are no difficulties associated with selecting values of regularization constants.

MATLAB code for NMU algorithm is available at:
https://sites.google.com/site/nicolasgillis/code

a) N. Gillis, and F. Glineur, "Using underapproximations for sparse nonnegative matrix factorization," *Patt. Recog.*, vol. 43, pp. 1676-1687, 2010.

Multidimensional signals[1,2,3]

A number of data sets is not naturally represented in 2D space but in *N*D, $N \geq 3$, space. Few examples include: multispectral/hyperspectral image, video signal, EEG data, fluorescence spectroscopy data, magnetic resonance image, etc.



Multispectral-hyperspectral image (3D tensor)

$$\underline{\mathbf{X}} \in \mathbb{R}_{0+}^{I_1 \times I_2 \times I_3}$$

I_3 spectral images of the size $I_1 \times I_2$ pixels

Multispectral magnetic resonance image (3D tensor)

I_3=3 (PD,T_1 and T_2) images of the size $I_1 \times I_2$ pixels

1. A. Cichocki, R. Zdunek, A.H. Phan, S. Amari, Nonegative Matrix and Tensor Factorizations, John Wiley & Sons, 2009.
2. E. Acar, and B. Yener, "Unsupervised Multiway Data Analysis: A Literature Survey," IEEE Trans. Knowl. Data Eng. **21**, 6 (2009).
3. T.G. Kolda, and B.W. Bader, "Tensor Decompositions and Applications," SIAM Review **51**, 453 (2009).

Multidimensional signals

Very often for the purpose of exploratory data analysis (that includes the BSS methods such as ICA, DCA, SCA or NMF) 3D data are mapped to 2D data. That is known as *matricization*, *unfolding* or *flattening*. For example RGB image tensor (I_3=3) is flattened along mode-3 (spectral mode)

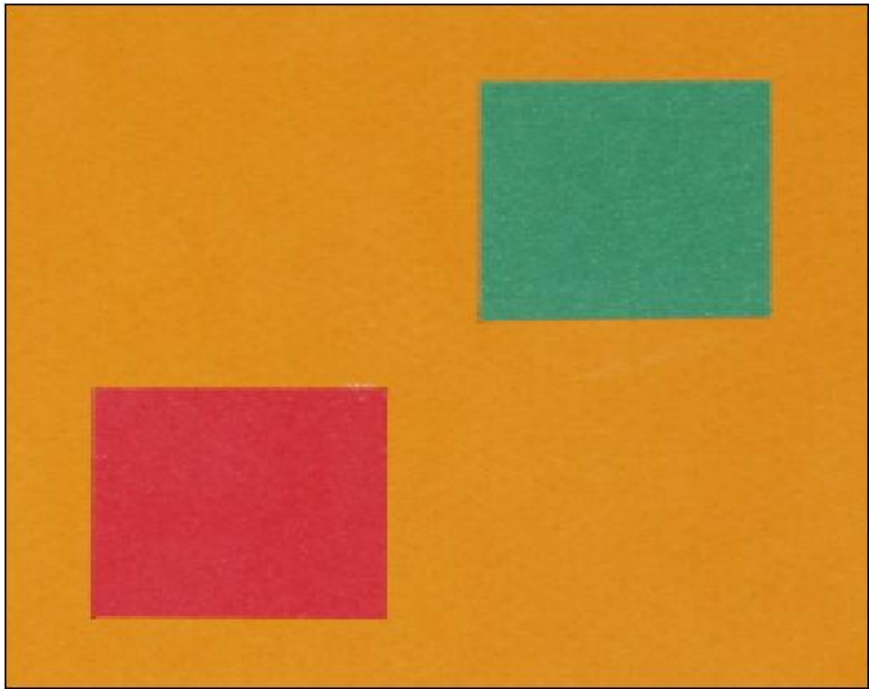
$$\underline{\mathbf{X}} \in \mathbb{R}_{0+}^{I_1 \times I_2 \times I_3} \xrightarrow{3} \mathbf{X}_{(3)} \in \mathbb{R}_{0+}^{I_3 \times I_1 I_2}$$

or in MATLAB notation:

```
for i3=1:3
     X3(i3,:) = reshape(X(i3,:,:),1,I1*I2)
end
```

Meaningful solutions of the BSS (decomposition) problem are characterized by $\mathbf{T}=\mathbf{P}\Lambda$. To obtain them matrix factorization methods such as NMF must impose sparseness constraints on $\mathbf{S}$.

Sparseness constrained unsupervised multichannel image decomposition



Original RGB image

I. Kopriva and A . Cichocki, "Sparse component analysis-based non-probabilistic blind decomposition of low-dimensional multi-spectral images," *Journal of Chemometrics*, vol. **23**, Issue 11, pp. 590-597 (2009).

MATLAB code: http://www.lair.irb.hr/ikopriva/prezentacije-i-izvjetaji.html

Multichannel image and linearan mixing model

X=AS $\mathbf{X} \in \mathbb{R}_{0+}^{N \times T}$, $\mathbf{S} \in \mathbb{R}_{0+}^{M \times T}$, $\mathbf{A} \in \mathbb{R}_{0+}^{N \times M}$ (1)

In imaging spectroscopy (multispectral/RGB image) rows of **X** are vectorized channel images (eg. red, green or blue color), columns of **A** are spectral profiles of objects (tissues, organs) present in image **X**, and rows of **S** are distributions of intensities of objects (tissues, organs) present in image **X**.

By an equivalent interpretation the model (1) is applicable to other types of co-registered multichannel images such as: hyperspectral image, multiphase CT, multispectral magnetic resonance (MR), functional MR image, imaging mass spectrometry, multimodal image obtained by image fusion (PET/CT),…

(u)BSS problem relates to unsupervised decomposition of image **X** into anatomically meaningful components: distributions of intensities of objects present in the image **X**.

Unsupervised decomposition of multispectral images

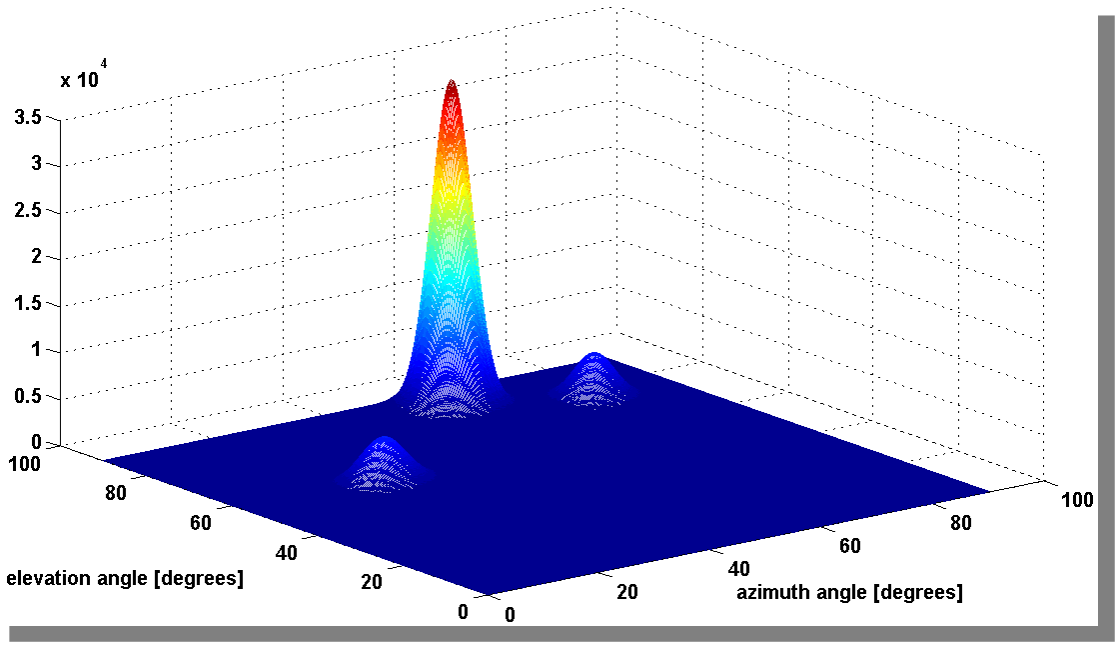
When degree of overlap between objects in spatial domain is very small i.e. $s_m(t)*s_n(t)\approx\delta_{nm}$, it implies $K=\|\mathbf{s}(t)\|_0\approx 1$.

RGB image decomposition problem can be solved with some SCA algorithm, eg. clustering and L_1-norm minimization or NMF algorithm with sparseness constraint.

Estimate of the mixing **A** and number of objects *M* is achieved by clustering.

Unsupervised decomposition of multispectral images

Clustering algorithm is used to estimate number of materials *M*.



Three peaks suggest existence of three materials in the RGB image **i.e. *M*=3**.

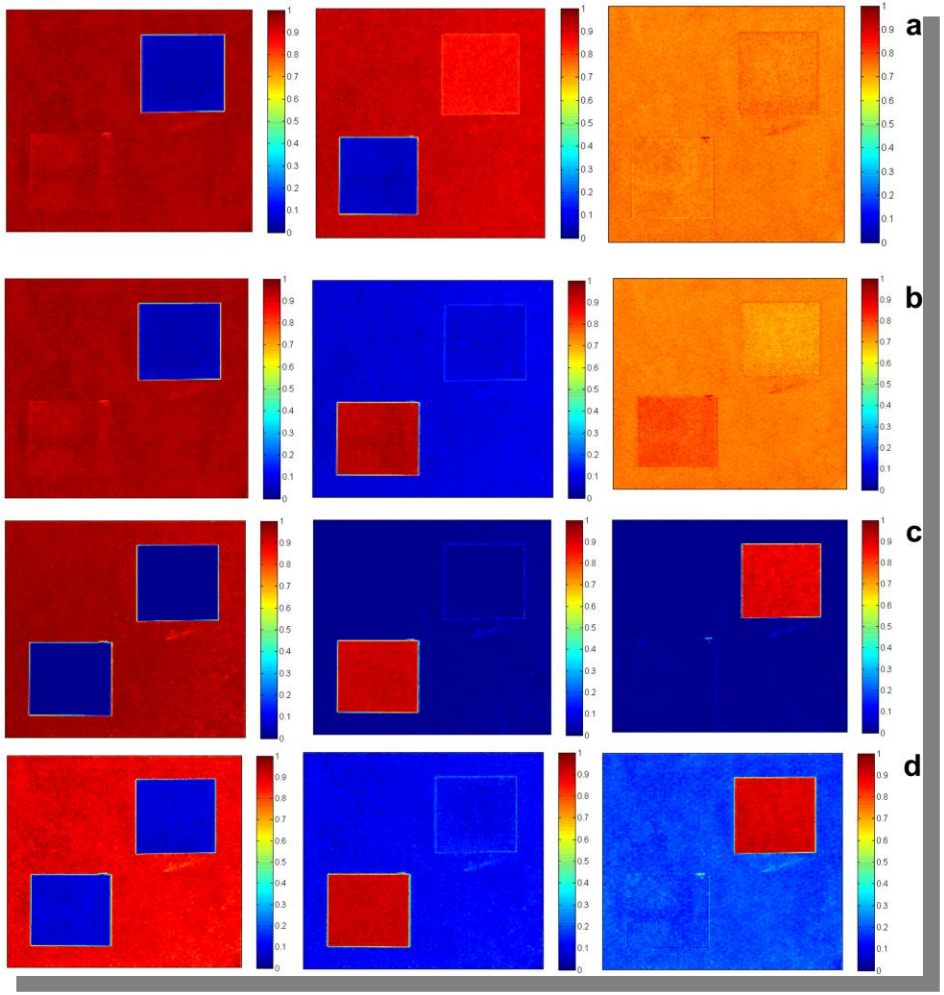
Unsupervised decomposition of multispectral images

Intensity distributions of the materials were extracted by NMF with 25 layers, SCA basedd on linear programming, ICA and DCA methods.

Extracted maps were rescaled to the interval [0,1] where 0 means full absence of the material and 1 means full presence of the material.

This enables visualization of the quality of decomposition process. Zero probability (absence of the material) is visualized with dark blue color and probability one (full presence of the material) is visualized with dark red color.

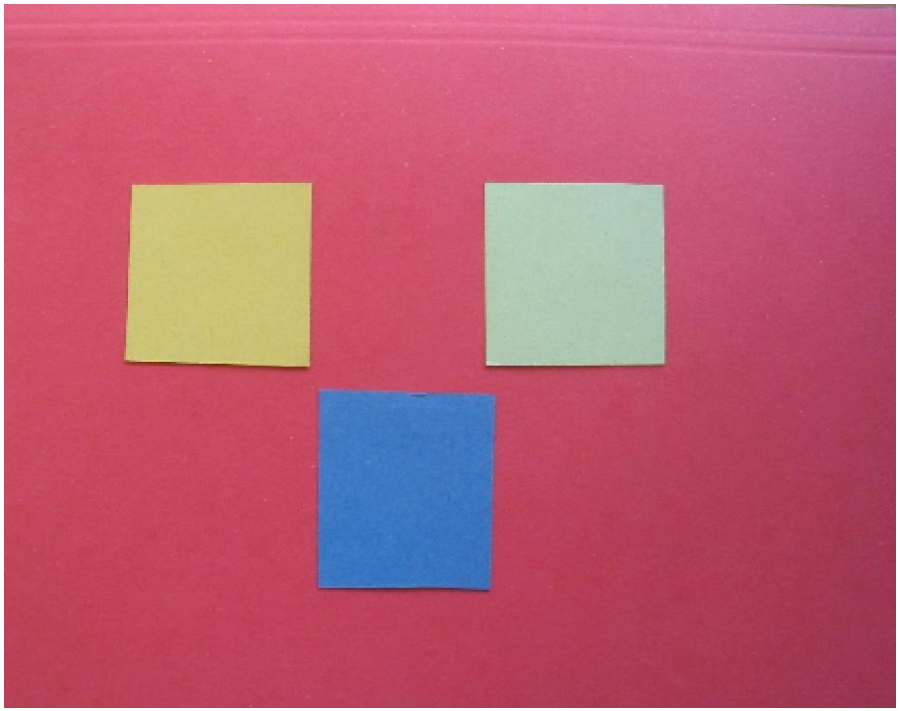
Unsupervised decomposition of multispectral images



a) DCA
b) ICA
c) NMF
d) SCA- linear programming

Unsupervised decomposition of multispectral images

Consider blind decomposition of the RGB image (N=3) composed of four materials (M=4):

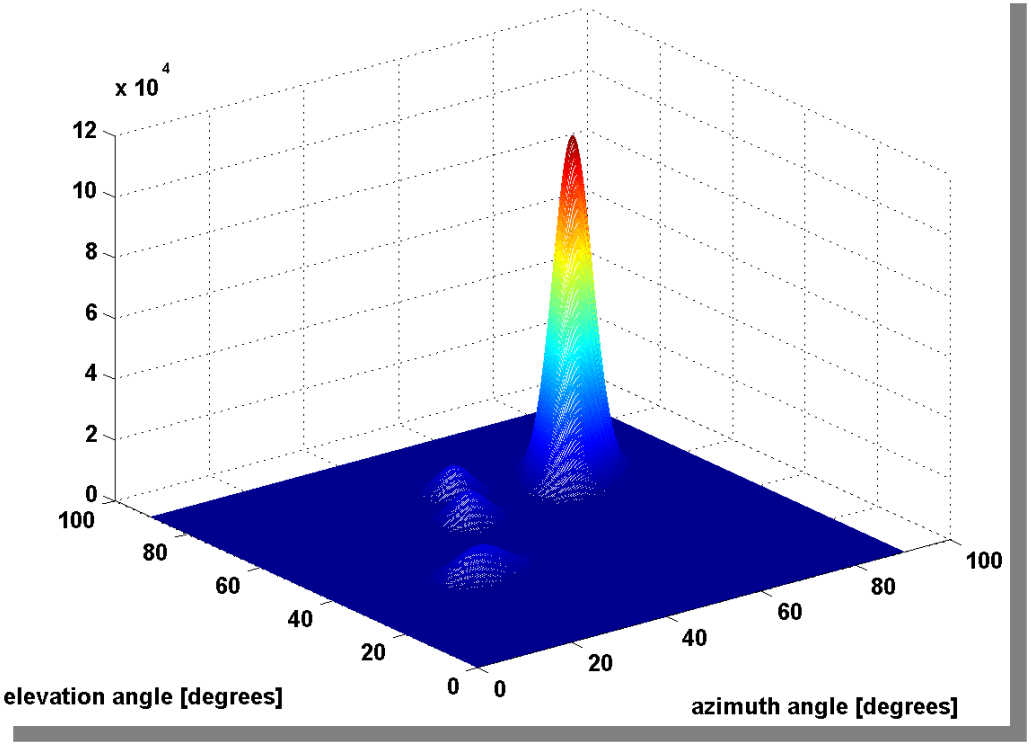


I. Kopriva and A . Cichocki, "Sparse component analysis-based non-probabilistic blind decomposition of low-dimensional multi-spectral images," *Journal of Chemometrics*, vol. **23**, Issue 11, pp. 590-597 (2009).

MATLAB code: http://www.lair.irb.hr/ikopriva/prezentacije-i-izvjetaji.html

Unsupervised decomposition of multispectral images

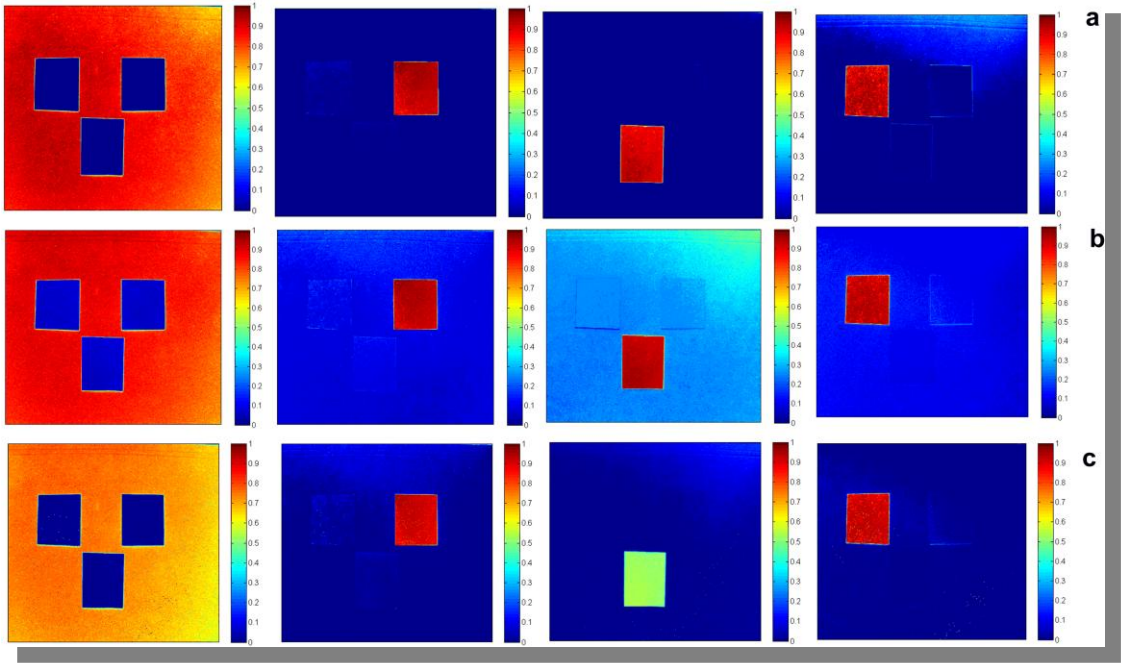
For shown experimental RGB image clustering function is obtained as:



Four peaks suggest existence of four materials in the RGB image **i.e. *M*=4**.

Unsupervised decomposition of multispectral images

Intensity maps of the materials extracted by HALS NMF with 25 layers, linear programming and interior point method, [a], are obtained as:



a) 25 layers HALS NMF; b) Interior point method; c) Linear programming.

a) S. J. Kim, K. Koh, M. Lustig, S. Boyd, D. Gorinevsky, "An Interior-Point Method for Large-Scale L_1 -Regularized Least Squares,"IEEE Journal of Selected Topics in Signal Processing **1**, 606-617 (2007).
http://www.stanford.edu/~boyd/l1_ls/.

Unsupervised decomposition of multispectral images

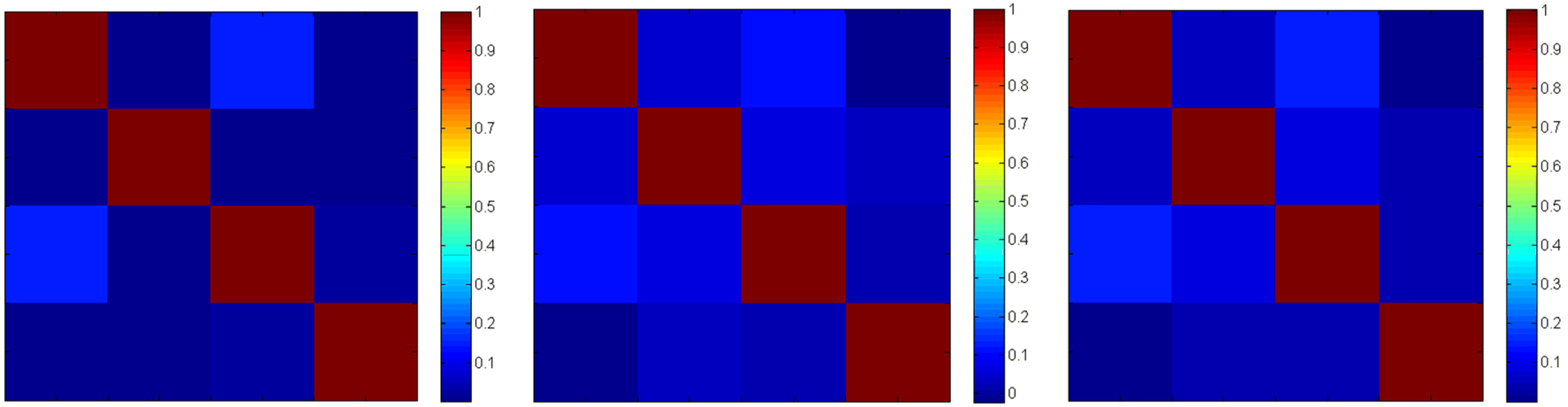
Since materials in the experimental RGB image are orthogonal (they do not overlap in spatial domain) we can evaluate performance of the employed blind image decomposition methods via the correlation matrix defined as $\mathbf{G=SS}^{\mathrm{T}}$. For perfect estimation the correlation matrix will be diagonal and performance is visualized as deviation from diagonal matrix. To quantify decomposition quality numerically we compute the correlation index in dB scale as

$$CR = -10\log_{10}\sum_{\substack{i,j=1 \\ j\neq i}}^{M} g_{ij}^{2}$$

where before calculating correlation matrix **G** rows of **S** are normalized to unit L_2-norm.

Unsupervised decomposition of multispectral images

Correlation matrices



From left to right: 25 layers HALS NMF; Interior point method; c) Linear programming.

CR performance measure in dB

	Multilayer HALS NMF	**Interior-point method**	**Linear program**
CR [dB]	13.67	9.97	7.77
CPU time [s]*	3097	7751	3265

*MATLAB environment on 2.4 GHz Intel Core 2 Quad Processor Q6600 desktop computer with 4GB RAM.

Sparseness constrained NMF for 3D decomposition of multichannel medical images

I. Kopriva, A. Jukić, X. Chen, "Sparseness Constrained Nonnegative Matrix Factorization for Unsupervised 3D Segmentation of Multichannel Images: Demonstration on Multispectral Magnetic Resonance Image of the Brain," *SPIE Medical Imaging Symposium*, Orlando, FL, February 9-14, 2013, Proc. SPIE Vol. **8669**, paper # 119.

MATLAB code: http://www.lair.irb.hr/ikopriva/prezentacije-i-izvjetaji.html

4D tensor model of multi-channel multi-slice image

For 3D decomposition multi-channel and multi-slice image is represented by multilinear mixture model:

$$\underline{\mathbf{X}} \approx \underline{\mathbf{G}} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_3 \mathbf{A}^{(3)} \times_4 \mathbf{A}^{(4)}$$

where $\underline{\mathbf{X}} \in \mathbb{R}_{0+}^{I_1 \times I_2 \times I_3 \times I_4}$ stands for image tensor composed of I_4 channel images, I_3 slices, and $I_1 \times I_2$ pixel (voxel) elements per slice.

Above model is known as Tucker4 model, [a], where $\underline{\mathbf{G}} \in \mathbb{R}_{0+}^{J_1 \times J_2 \times J_3 \times J_4}$ stands for core tensor and $\mathbf{A}^{(n)} \in \mathbb{R}_{0+}^{I_n \times J_n}{}_{n=1}^{4}$ stand for factor matrices.

Factor matrices associated with first three modes represent directional basis along these modes. They can be used to model source tensor:

$$\underline{\mathbf{S}} = \underline{\mathbf{G}} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_3 \mathbf{A}^{(3)} = \underline{\mathbf{X}} \times_4 \left(\mathbf{A}^{(4)} \right)^{\dagger}$$

a) Tucker, L. R., "Some mathematical notes on three-mode factor analysis," *Psychometrika* 31, 279-311 (1966).

4D tensor model of multi-channel multi-slice image

$\underline{\mathbf{S}} \in \mathbb{R}_{0+}^{I_1 \times I_2 \times I_3 \times J}$ contains 3D intensity distributions of *J* organs (tissues) present in the image.

Matrix $\mathbf{A}^{(4)}$ stands for mixing matrix that in a case of multispectral magnetic resonance image contains in its columns spectral profiles of the tissues present in the image. The image tensor $\underline{\mathbf{X}}$ can be unfoled along mode-4 yielding:

$$\mathbf{X}_{(4)} \approx \mathbf{A}^{(4)} \mathbf{G}_{(4)} \left[\mathbf{A}^{(3)} \otimes \mathbf{A}^{(2)} \otimes \mathbf{A}^{(1)} \right]^{\mathrm{T}} = \mathbf{A}^{(4)} \mathbf{S}_{(4)}$$

3D decomposition is performed applying sparseness constrained factorization of $\mathbf{X}_{(4)}$, for example using the NMU algorithm.

Afterwards, $\mathbf{S}_{(4)}$ is tensorized to get $\underline{\mathbf{S}}$.

3D decomposition of brain tumor

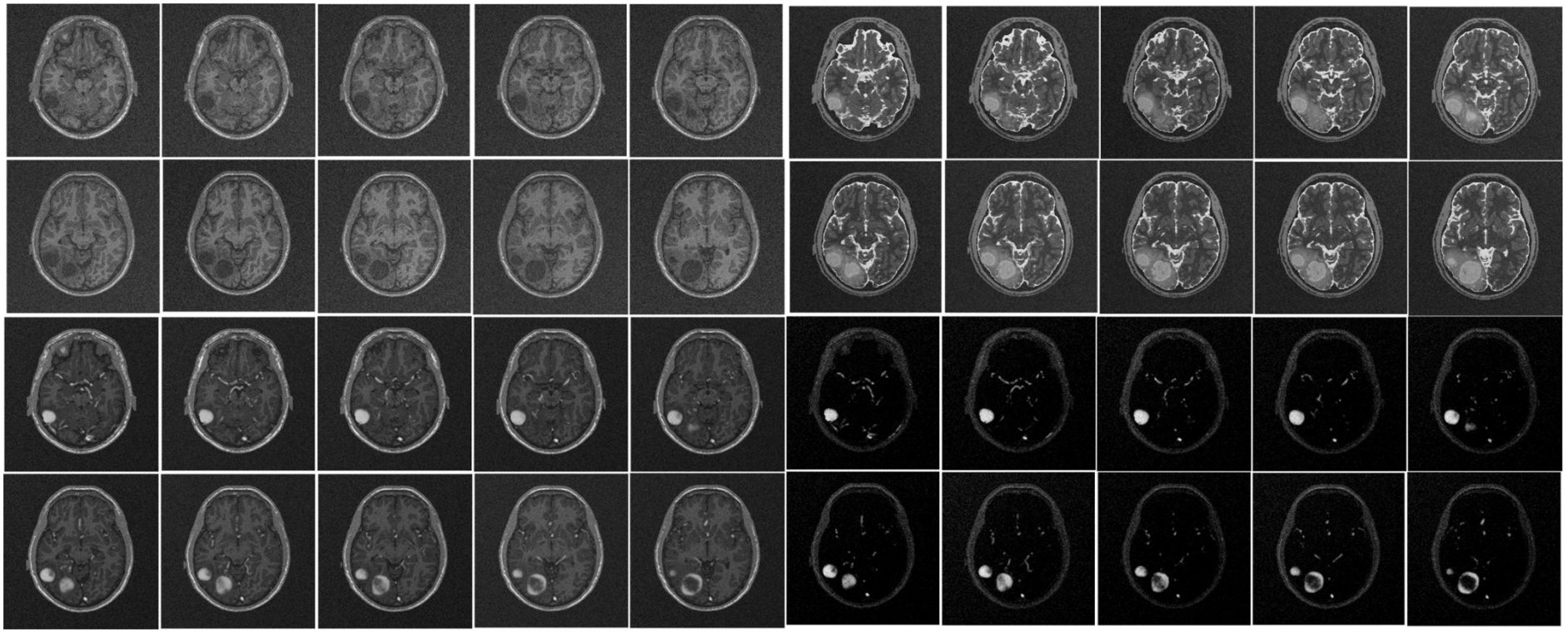
3D decomposition method is demonstrated on extraction of brain tumor from synthetic mMR image. The image is obtained from TumorSim database of the Utah Center for Neuroimage Analysis, [a].

In relation to standard mMR image comprised of T1, T2 and PD images, the PD image has been replaced by T1-weighted image obtained after administration gadolinium contrast agent.

3D decomposition method is applied to slices 50 to 70 of the `TumoSimData_004 dataset`. Thus, I_3=21 slices were segmented jointly. Each slice has 256×256 pixels.

a) The TumorSim database of Utah Center for Neuroimage Analysis: **http://www.nitrc.org/projects/tumorsim/**.

3D decomposition of brain tumor



Every second slice from 52 to 70. T1 image (top left), T2 image (top right), T1_GAD image (bottom left), NMU extracted tumor (bottom right).

3D decomposition of brain tumor

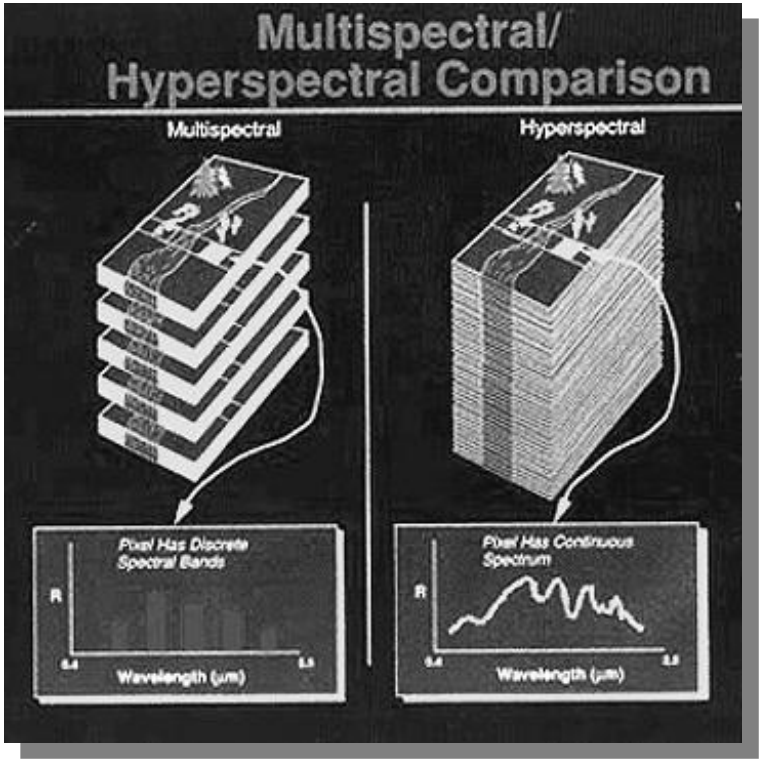
Slice number	50	51	52	53	54	55	56	57	58	59	60
3D Segmentation	**0.7278**	**0.7679**	**0.8387**	**0.8669**	**0.8634**	**0.8512**	**0.8748**	**0.8875**	**0.8876**	**0.8938**	**0.7811**
T1_GAD image	0.1942	0.2280	0.2565	0.2836	0.2940	0.3193	0.3388	0.3626	0.3754	0.3583	0.3536
Slice number	61	62	63	64	65	66	67	68	69	70	
3D Segmentation	**0.7436**	**0.7587**	**0.7061**	**0.7699**	**0.7223**	**0.5672**	**0.5635**	**0.4799**	**0.4060**	**0.4113**	
T1_GAD image	0.3810	0.4137	0.4415	0.4343	0.4221	0.3619	0.3287	0.2851	0.2431	2158	

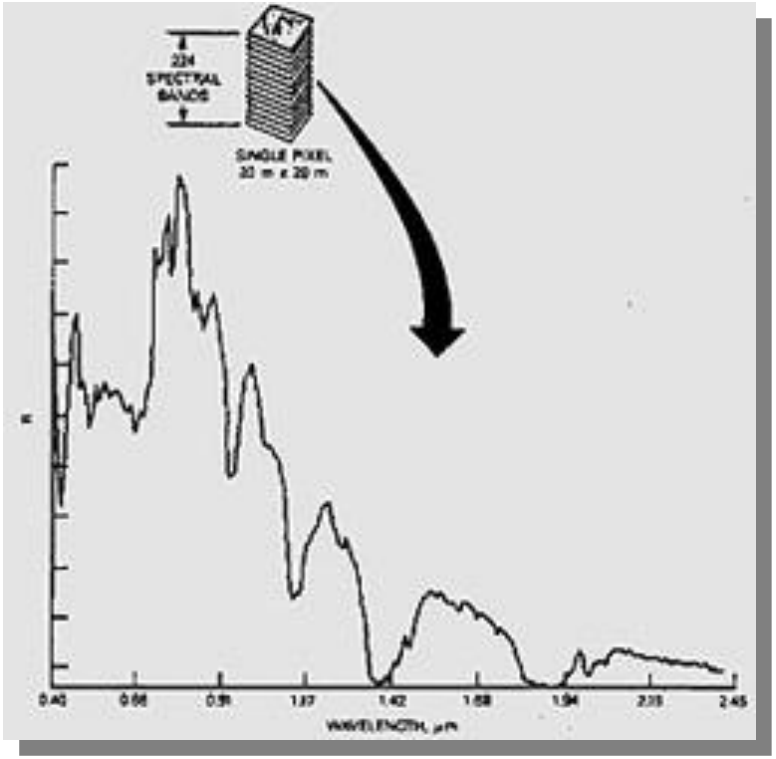
Decomposition / segmentation results in term of Dice's coefficient for slices 50 to 70.

Sparseness constrained hyperspectral image decomposition

MATLAB code: http://www.lair.irb.hr/ikopriva/prezentacije-i-izvjetaji.html

Unsupervised decomposition of hyperspectral images





- SPOT- 4 bands, LANDSAT -7 bands, AVIRIS-224 bands (0.38μ-2.4μ);

Unsupervised decomposition of hyperspectral images

Hyperspectral/multispectral image and static linear mixture model. For image consisting of I_3 bands and J materials linear data model is assumed:

$$\mathbf{X} = \mathbf{AS} = \sum_{j=1}^{J} \mathbf{a}_j \underline{\mathbf{s}}_j$$

$$\mathbf{a}_1 \quad \mathbf{a}_2 \quad \ldots \quad \mathbf{a}_J \quad \equiv \quad \mathbf{A}$$

$$\underline{\mathbf{s}}_1 \quad \underline{\mathbf{s}}_2 \quad \ldots \quad \underline{\mathbf{s}}_J^{\,T} \quad \equiv \quad \mathbf{S}$$

X - measured data intensity matrix, $\mathbf{X} \in \mathbb{R}_{0+}^{I_3 \times I_1 I_2}$

S - unknown abundances matrix, $\mathbf{S} \in \mathbb{R}_{0+}^{J \times I_1 I_2}$

A – unknown matrix of endmember spectral signatures, $\mathbf{A} \in \mathbb{R}_{0+}^{I_3 \times J}$

Unsupervised decomposition of hyperspectral images

Very often in hyperspectral image analysis abundance sum-to-one constraint (ASC) is imposed on abundances coefficients at each pixel,

$$\sum_{j=1}^{J} s_{j(i_1 i_2)} = 1$$

That leads to fully constrained least square (FCLS) problem/algorithm (assuming endmembers matrix is known), ref.a:

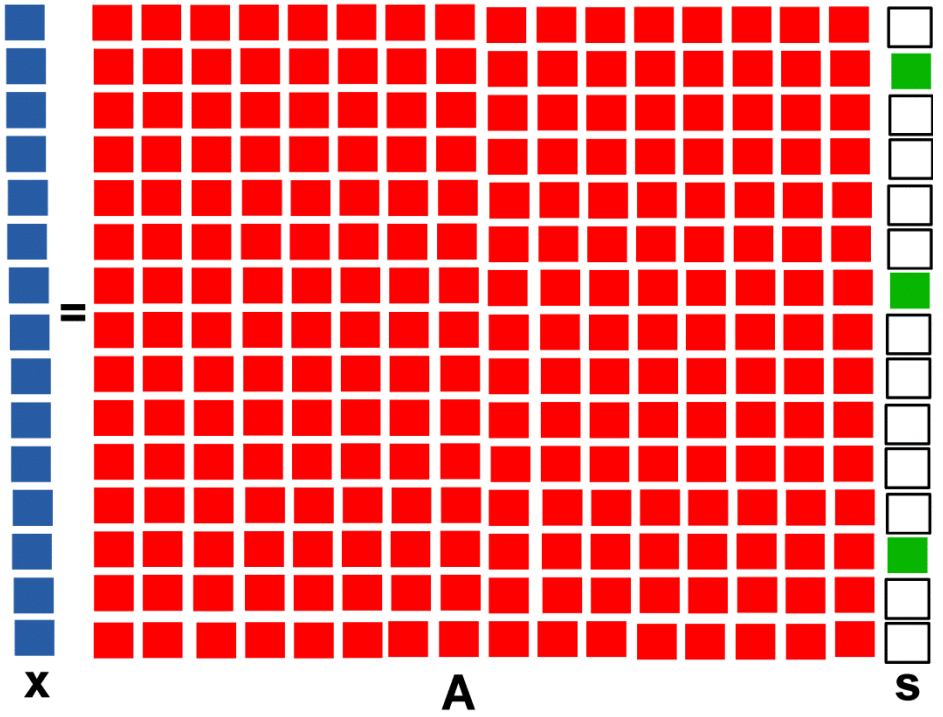
$$P: \arg\min_{\mathbf{s}} \frac{1}{2} \mathbf{x} - \mathbf{A}\mathbf{s} \quad \mathbf{x} - \mathbf{A}\mathbf{s}^{T} - \lambda\left(\sum_{j=1}^{J} s_j - 1\right)$$

a) D.C. Heinz, C. –I Chang, and M.L.G. Althouse, "Fully constrained least squares-based linear unmixing," in *Proc. IEEE Int. Conf. Geosci. Remote Sens*. (*IGARSS),* 1999, vol. 1, pp. 1401-1403.

Unsupervised decomposition of hyperspectral images

In practice ASC i rarely satisfied due to presence of noise and model mismatches (ref. a). Spareness and nonnegativity constraints yield better results.

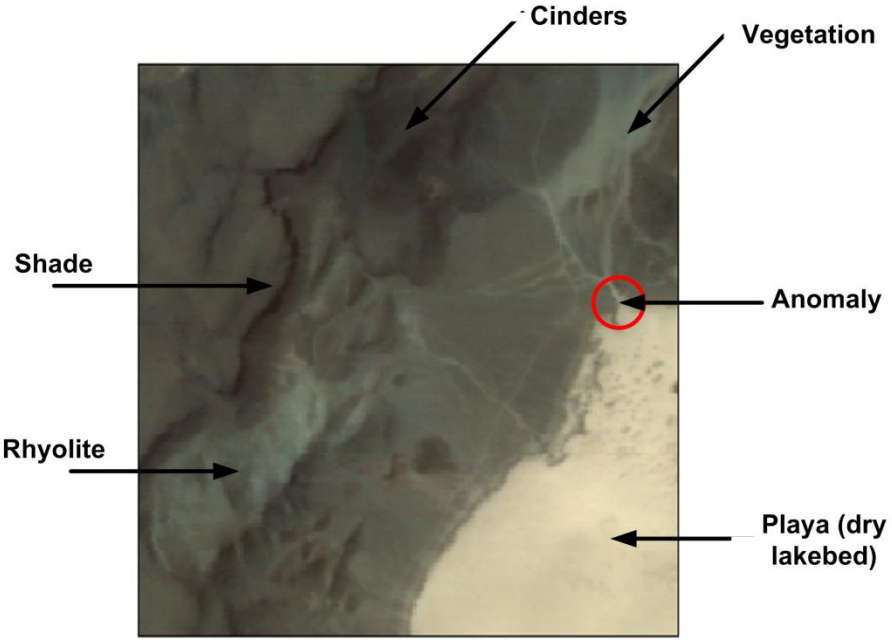
Spareness constraint implies that **only few** out of *J* objects are present at each pixel.



a) J.M. Bioucas-Dias, et. al., "Hyperspectral Unmxing Overview: Geometrical, Statistical, and Sparse Regression-Based Approaches," *IEEE J. Selected Topics in Applied Earth Observations and Remote Sens*ing, ¸2012, vol. 5, pp. 354-379.

Unsupervised decomposition of hyperspectral images

AVIRIS Lunar Crater Vulcanic Field (LCVF) in Norther Nye County, Nevada (ref. a,b, c). There are five signatures of interest, ref. a, b), in the LCVF scene: cinders, rhyolite, playa (dry lakebed), shade, vegetation. In addition, ref c), there is single two-pixel anomaly located at the top edge of the lake. The image is composed of 158 spectral bands of the size 200x200 pixels.



True color image
(40,30,10)

a) J. C. Harsany, C. –I Chang, Hyperspectral image classification and dimensionality reduction: an orothogonal subspace approach. IEEE Transactions on Geoscience and Remote Sensing, Vol.32, no. 4, 779-785, 1994.
b) C. –I. Chang, D. C. Heinz, Constrained Subpixel Target Detection for Reomtely Sensed Imagery. IEEE Transactions on Geoscience and Remote Sensing, Vol.38, no. 3, 1144-1159, 2000.
c) C. –I. Chang, S. -S. Chiang, I. W. Ginsberg, Anomaly detection in hyperspectral imagery, SPIE Conference on Geo-Spatial Image and Data Exploration II, Orlando, Florida, 20-24 April, 2001.

Unsupervised decomposition of hyperspectral images

AVIRIS LCVF image has been unmixed by four algorithms:

nonnegative matrix underapproximation (NMU) algorithm;

fast separable NMF algorithm (to identify endmember/mixing matrix) and fast iterative shrinkage thresholding (Fast_IST) algorithm (to identify abundance coefficients) with λ=0.1 and 300 iterations;

fast separable NMF algorithm (to identify endmember/mixing matrix) and fast combinatorial nonnegative least square algorithm (FCNNLS), ref. a;

fast separable NMF algorithm (to identify endmember/mixing matrix) and fully constrained least square (FCLS) algorithm.

FCNNLS algorithm solves the abundance estimation problem by using nonnegaitivity constraints only. That is justified by finding that for matrix **A** with row-span intersecting the positive orthant (which is the case for endmemer spectral profiles) if the problem admits a suffciently sparse solution it is necessary unique (ref. b).

a) M. H. van Benthem and M. R. Keenan, Fast algorithm for the solution of large-scale non-negativity constrained least aquares problem. J. of Chemometrics, Vol.18, pp. 441-450, 2004.
b) A. Bruckstein, M. Elad, and M. Zibulevsky. On the uniqueness of nonnegative sparse solutions to undedetermined systems of equations. IEEE Transactions on Information Theory, Vol.54, no. 11, 4813-4820, 2008.

Unsupervised decomposition of hyperspectral images

All algorithms are implemeted in MATLAB environment under 64-bit Windows operating system, processor Intel Core i7-2600s with a clock speed 3.4 GHz and RAM of size 24 GB.

Computation times in seconds are given bellow:

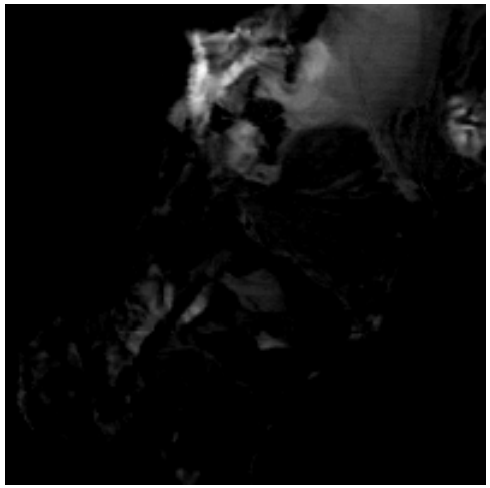
NMU: 28.24s;

FastSepNMF+IST: 114.8 s

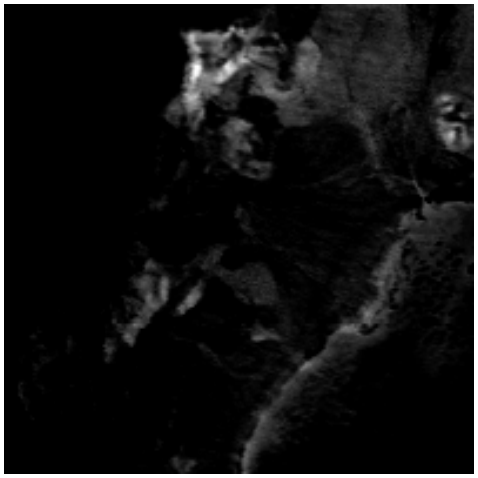
FastSepNMF+FCNNLS: 0.14 s

FastSepNMF+FCLS: 16.42s

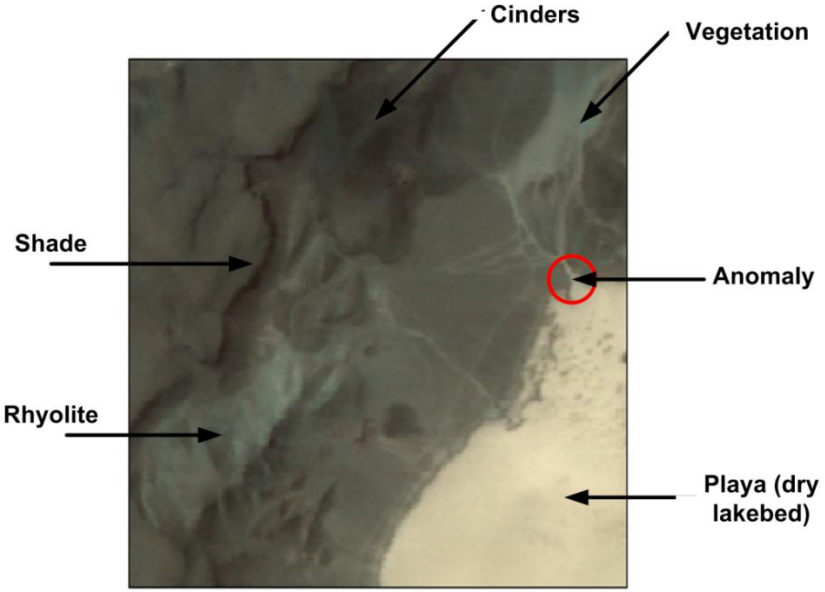
CINDERS



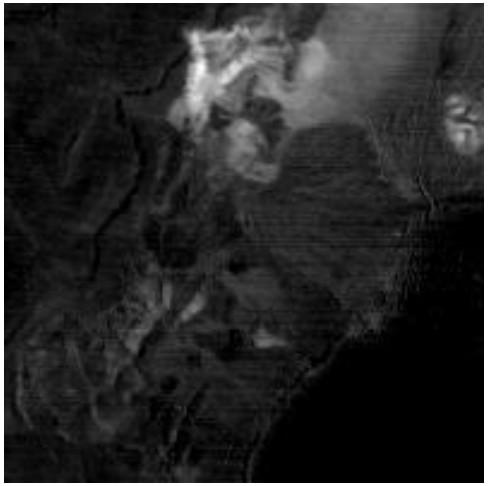
Fast_Sep_NMF + IST



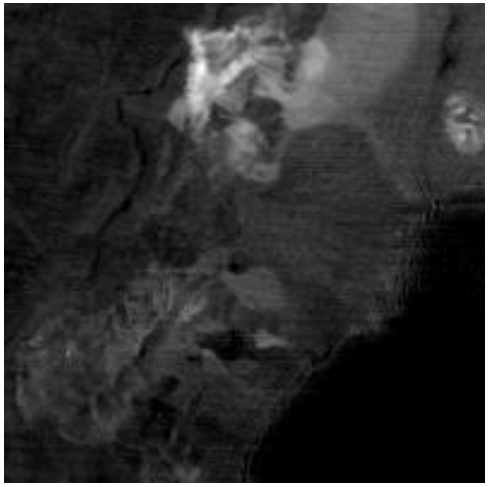
NMU



True color image
(40,30,10)

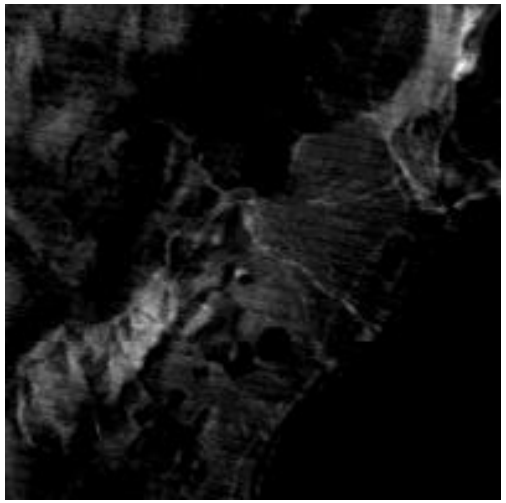


Fast_Sep_NMF + FCNNLS

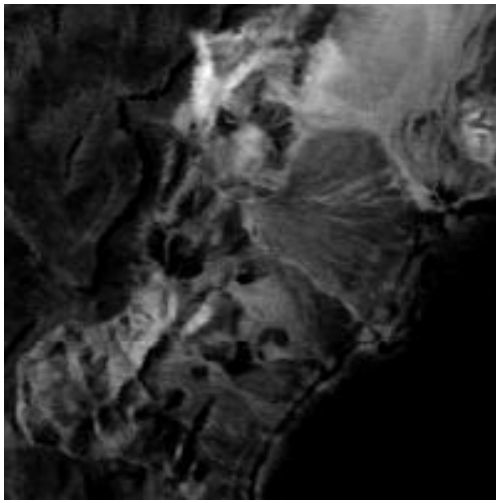


Fast_Sep_NMF + FCLS

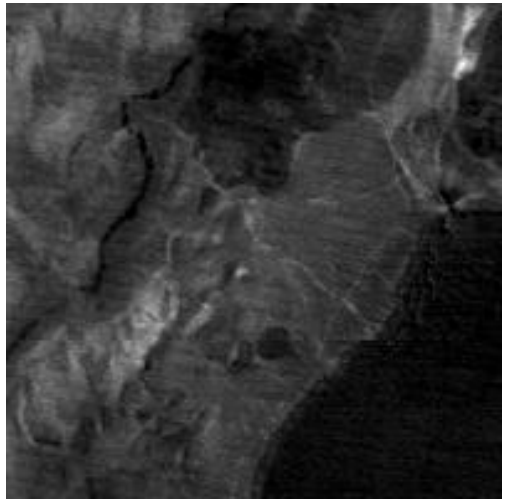
RHYOLITE



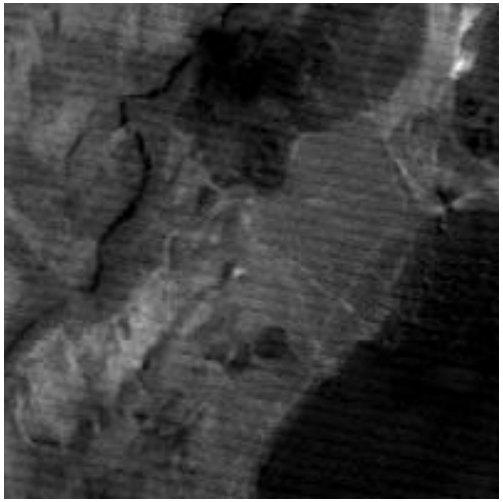
Fast_Sep_NMF + IST



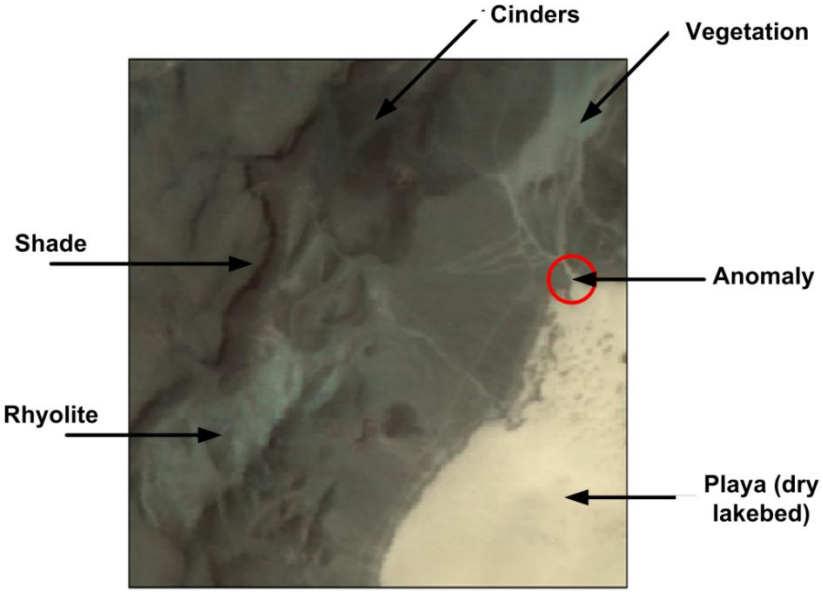
NMU



Fast_Sep_NMF + FCNNLS

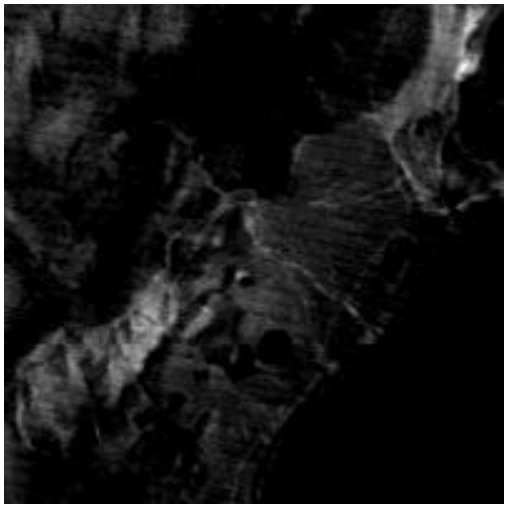


Fast_Sep_NMF + FCLS

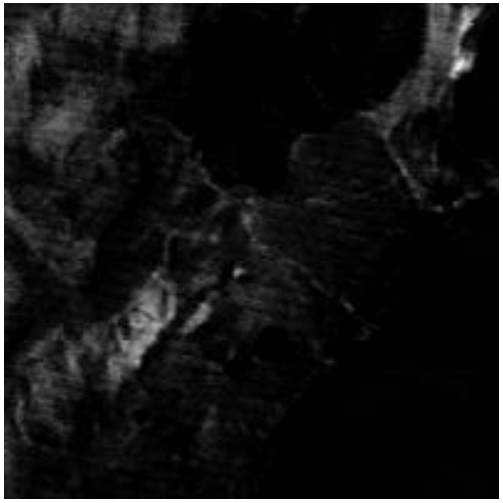


True color image
(40,30,10)

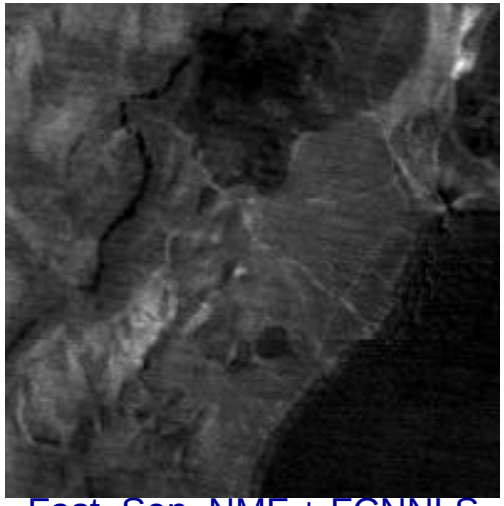
VEGETATION



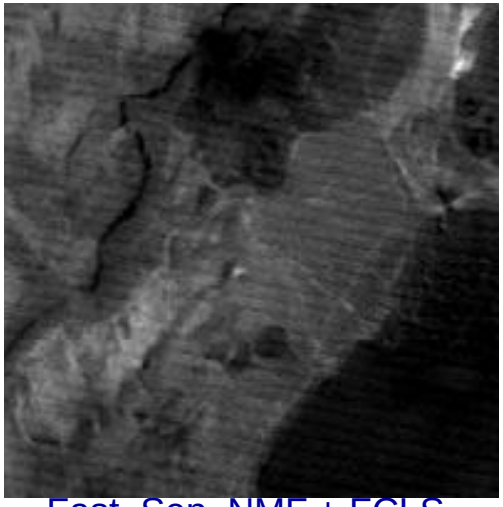
Fast_Sep_NMF + IST



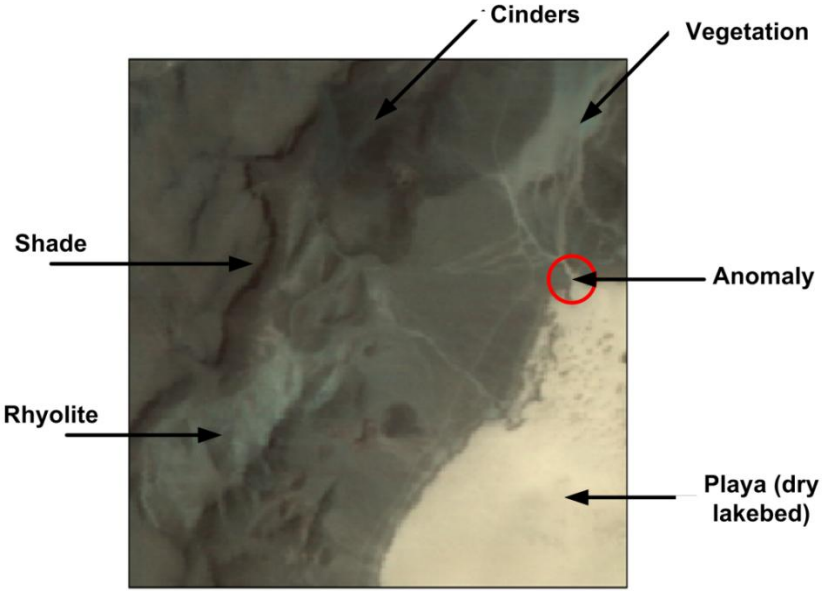
NMU



Fast_Sep_NMF + FCNNLS



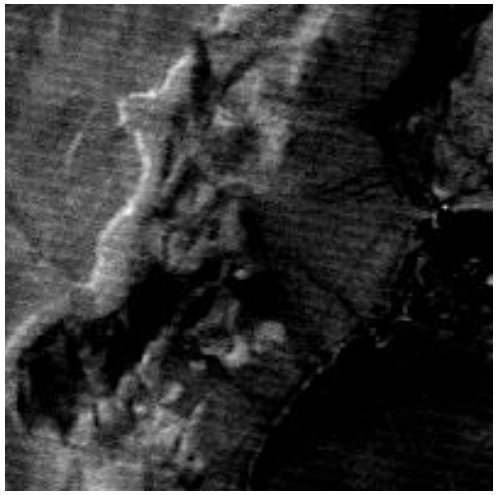
Fast_Sep_NMF + FCLS



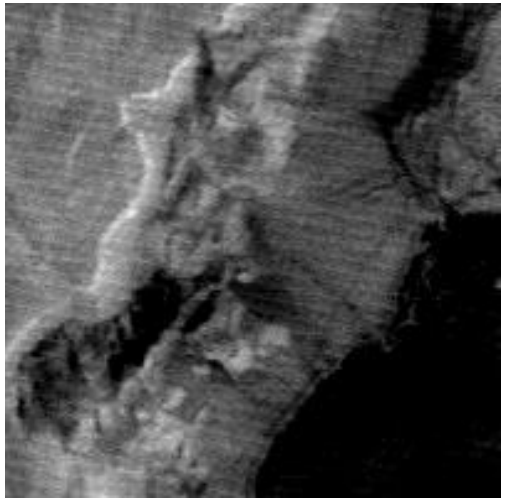
True color image
(40,30,10)

SHADE

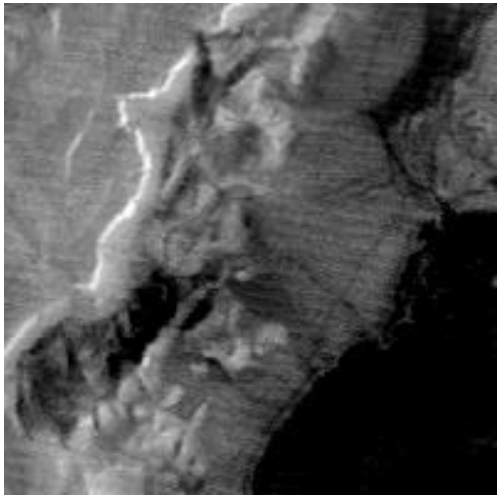
Fast_Sep_NMF + IST



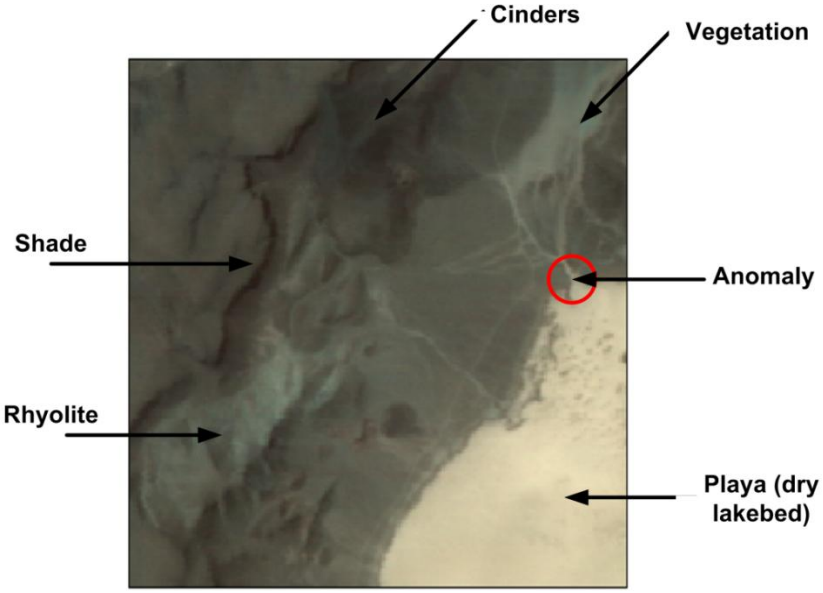
NMU



Fast_Sep_NMF + FCNNLS

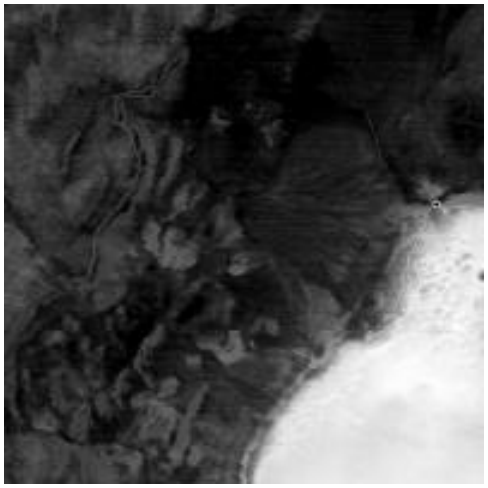


Fast_Sep_NMF + FCLS

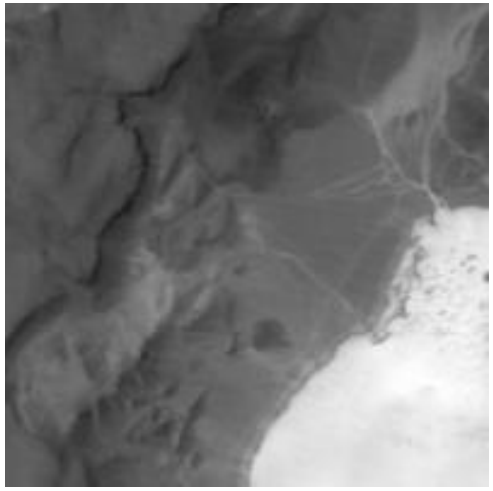


True color image
(40,30,10)

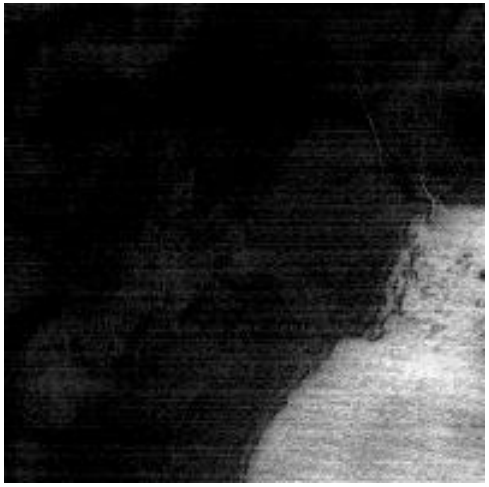
PLAYA (dry lakebed)



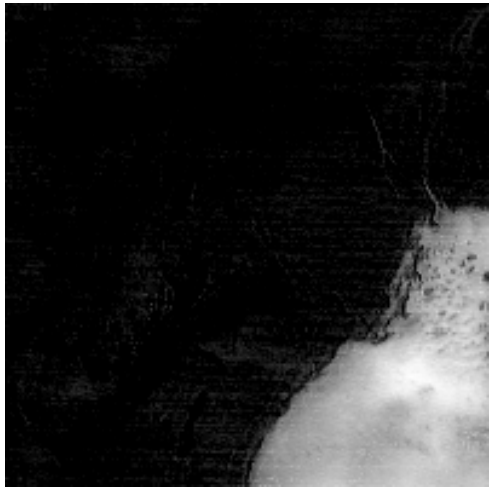
Fast_Sep_NMF + IST



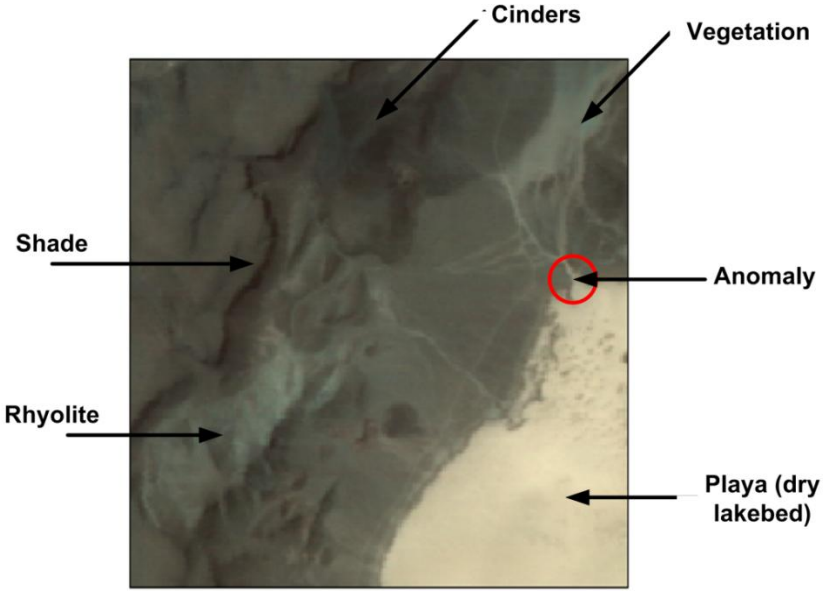
NMU



Fast_Sep_NMF + FCNNLS



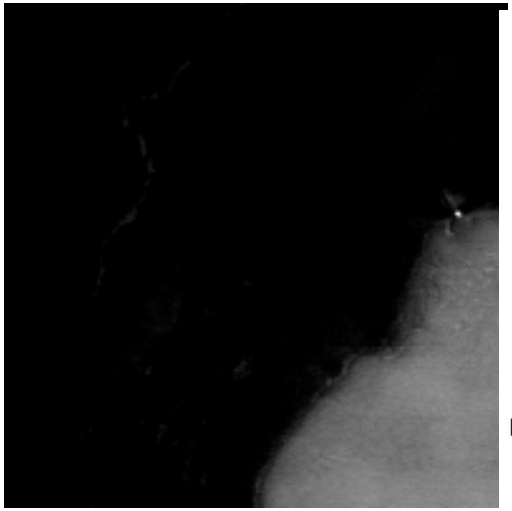
Fast_Sep_NMF + FCLS



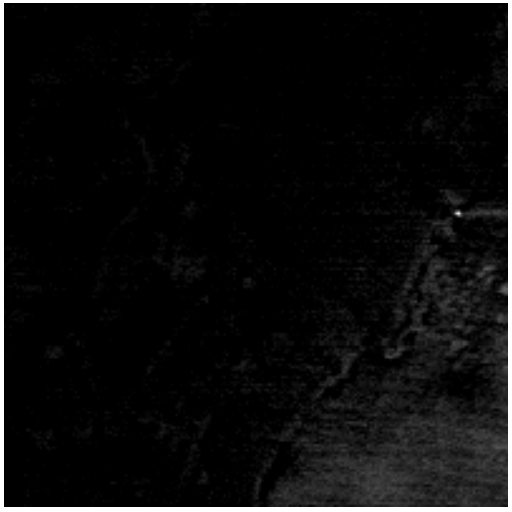
True color image
(40,30,10)

ANOMALY

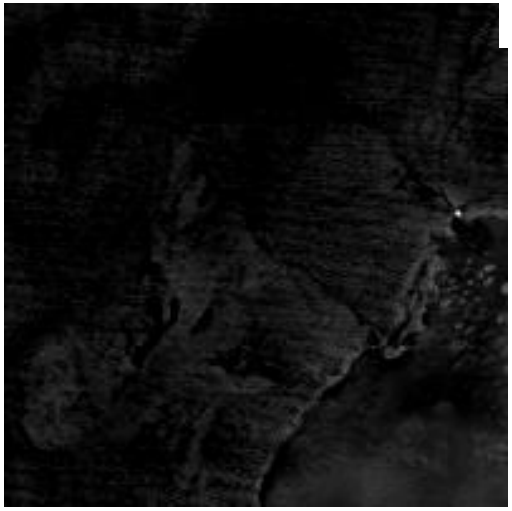
Fast_Sep_NMF + IST



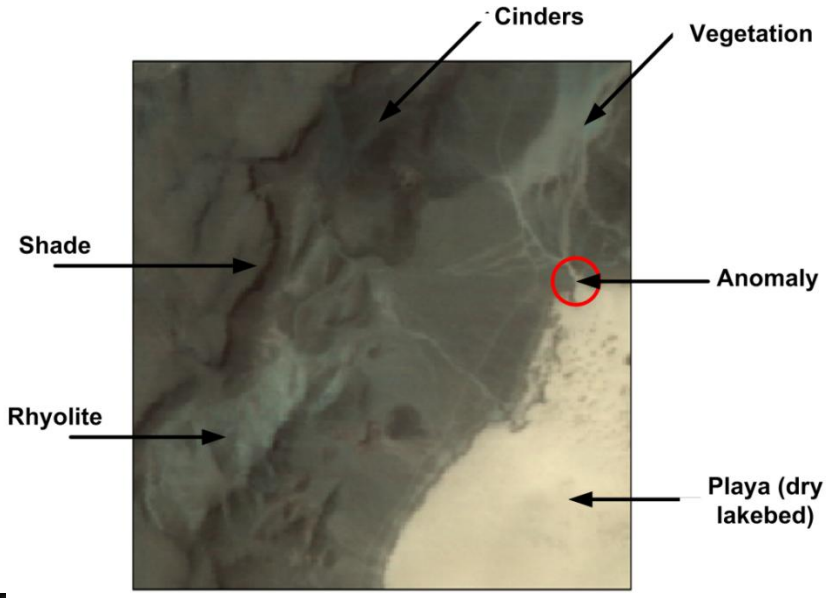
NMU



Fast_Sep_NMF + FCNNLS



Fast_Sep_NMF + FCLS



True color image
(40,30,10)

THANK YOU !!!!!!!