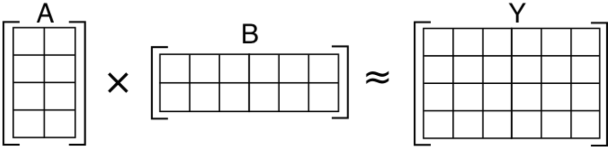
Nonnegative Matrix Factorization (NMF)

Bernard Lampe and Adam Bekit

May 5, 2016

Overview

1. Theory
   - Model
   - Cost Function
   - Uniqueness
   - Algorithms
   - Performance
   - Diversity
2. Application
   - Description
3. Results
4. References



NMF Model

$$\mathbf{Y} = \mathbf{AB} + \mathbf{E},\ \mathbf{Y} \in \mathbb{R}^{M \times N}, \mathbf{A} \in \mathbb{R}^{M \times R}, \mathbf{B} \in \mathbb{R}^{R \times N}$$

$$Y_{ij} \approx \sum_{r=1}^{R} A_{ir} B_{rj}$$

- $\mathbf{Y}$ is a nonnegative data matrix
- $\mathbf{A}$ and $\mathbf{B}$ are nonnegative factor matrices
- $\mathbf{E}$ is an error matrix
- $\{\mathbf{a}_i\}$, columns of $\mathbf{A}$ are the basis vectors
- $\{\mathbf{b}_j\}$, columns of $\mathbf{B}$ are the coordinate vectors

Model Constraints of PCA, VQ, NMF, Dictionary Learning

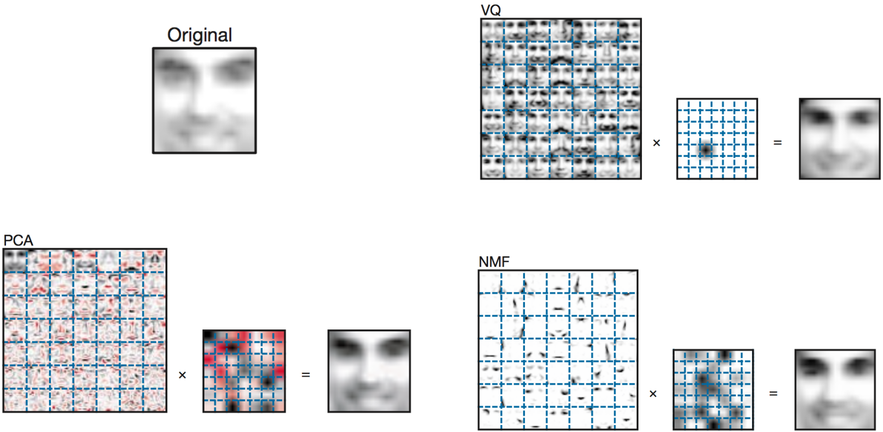
Different algorithms can be viewed as matrix decompositions with different constraints on $\mathbf{A}$ and $\mathbf{B}$.

- VQ require $\{\mathbf{a}_i\}$ to be quantized vectors, and $\{\mathbf{b}_j\}$ to be unitary
  - $\{\mathbf{a}_i\}$ are the centroids of the K-means algorithm
- PCA $\{\mathbf{a}_i\}$ are orthonormal, and $\mathbf{a}_1 = \arg\max_{\|\mathbf{a}_1\|} \mathrm{E}\{\mathbf{a}_1^T \mathbf{y}_i{}^2\}, \quad \forall i$
  - $\{\mathbf{a}_i\}$ are the eigenvectors of the correlation matrix
- NMF requires specification of R, and that $A_{ij} \geq 0$ and $B_{ij} \geq 0$
- Dictionary learning requires specification of R, and $A_{ij} \in \mathbb{R}$ and $B_{ij} \in \mathbb{R}$

Model Expressiveness

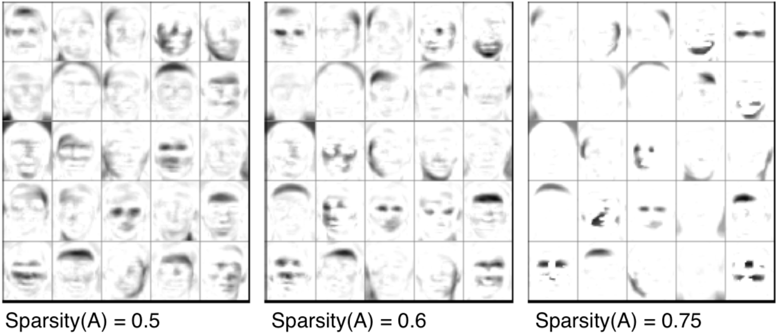
- PCA and ICA can add and subtract basis matrix vectors
- NMF can only add basis matrix vectors leading to "parts based" models
- NN-KSVD and non-negative dictionary learning are specific NMF factorizations

PCA, VQ and NMF Examples



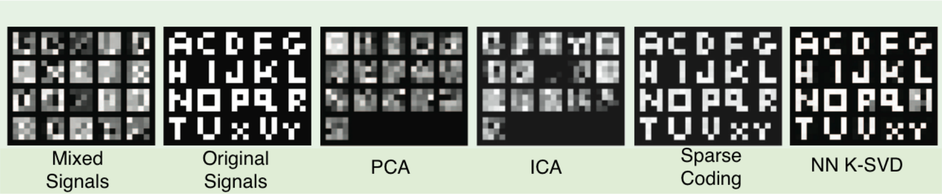
Basis matricies were trained with 2429, 19x19 pixel face images. [Lee and Seung '99]

NMF With Sparsity Example



[Hoyer '04]

Model Expressiveness Examples



[Ivana Tosic and Pascal Frossard '11]

- VQ, PCA and ICA have linearly independent vectors in the basis matrix
- PCA and ICA require $R \leq \min\{M, N\}$
- NMF can have linearly dependent vectors in the basis matrix

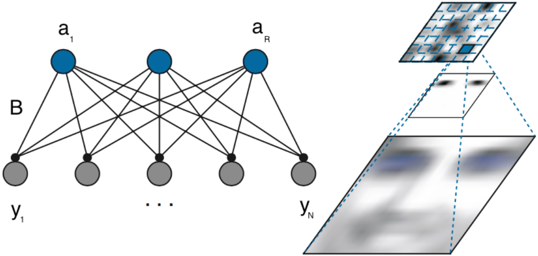
NMF as K-Means Clustering

$$J_{\text{K-means}} = \sum_{i=1}^{n} \min_{1 \le r \le R} \|\mathbf{y}_i - \mathbf{c}_r\|_F^2 = \sum_{r=1}^{R} \sum_{i \in C_r} \|\mathbf{y}_i - \mathbf{c}_r\|_F^2$$
$$= \sum_{i=1}^{n} \sum_{r=1}^{R} h_{ir} \|\mathbf{y}_i - \mathbf{c}_r\|_F^2 = \|\mathbf{Y} - \mathbf{AB}\|_F^2 = J_{nmf}$$

$$\mathbf{c}_r = \frac{1}{|C_r|} \sum_{i \in C_r} \mathbf{y}_i, \quad \mathbf{y}_i \in C_r, \quad h_{ir} = \{0, 1\}$$

- NMF can be considered a clustering of the data into R clusters with $\{\mathbf{a}_i\}$ centroids and $\{\mathbf{b}_j\}$ being unitary [C. Ding, et al '05]
- The centroids do not necessarily have to be positive, therefore this is refered to as "semi-NMF" or "relaxed-NMF"

NMF as Probabilistic Latent Variables



- Visible units $\{\mathbf{y}_i\}$ are connected to hidden latent variables $\{\mathbf{a}_j\}$ through weights in $\mathbf{B}$
- If $\{\mathbf{b}_j\}$ are normal, then $\{\mathbf{b}_j\}$ can be viewed as probability distributions
- If $J_{nmf} = D_{KL}(\mathbf{Y}\|\mathbf{AB})$, then the problem is exactly probabilistic latent semantic analysis [C.Ding '08]

Cost Functions

NMF is implemented by minimizing a non-convex cost function

- Euclidean Distance Minimization

$$J^*_{\mathrm{nmf}} = \arg\min_{\mathbf{A},\mathbf{B}} \frac{1}{2}||\mathbf{Y} - \mathbf{AB}||_2^2 \quad \text{s.t.} \quad \mathbf{A}, \mathbf{B} \geq 0$$

- Frobenius Distance Minimization

$$J^*_{\mathrm{nmf}} = \arg\min_{\mathbf{A},\mathbf{B}} \frac{1}{2}||\mathbf{Y} - \mathbf{AB}||_F^2 \quad \text{s.t.} \quad \mathbf{A}, \mathbf{B} \geq 0$$

- K-L Distance Minimization

$$J^*_{\mathrm{nmf}} = \arg\min_{\mathbf{A},\mathbf{B}} D_{KL}(\mathbf{Y}\|\mathbf{AB}) = \arg\min_{\mathbf{A},\mathbf{B}} \sum_{ij}(Y_{ij}\log\frac{Y_{ij}}{[\mathbf{AB}]_{ij}} - Y_{ij} + [\mathbf{AB}]_{ij})$$

Uniqueness

Factorization is not unique

$$\mathbf{Y} \approx (\mathbf{AT})(\mathbf{T}^{-1}\mathbf{B}) = \mathbf{A}'\mathbf{B}'$$

We can enforce a unique solution by including regularization (i.e., sparsity)

$$J^*_{\text{nmf}} = \arg\min_{\mathbf{A},\mathbf{B}} \frac{1}{2}\left|\left|\mathbf{Y} - \mathbf{AB}\right|\right|_2^2 + \lambda_0\|A\|_0 + \lambda_1\|B\|_0 \quad \text{s.t.} \quad \mathbf{A},\mathbf{B} \geq 0$$

If the sparsity of the **A** or **B** is low enough, then the solution is unique

$$\|\mathbf{B}\|_0 < \frac{R\ \text{spark}(\mathbf{A})}{2} \qquad \text{[Donoho and Elad '02]}$$

Spark is the smallest number of columns that are linearly dependent

$$\text{spark}(\mathbf{A}) = \min_{\mathbf{d}\neq 0} \|\mathbf{d}\|_0 \quad \text{s.t.} \quad \mathbf{Ad} = \mathbf{0}$$

Algorithms Used in Project

Optimization algorithms for NMF non-convex cost functions

- Alternating Least Squares (ALS)
  - J_{nmf} is minimized by alternating between **A** and **B**
  - J_{nmf} is minimized while holding **A** fixed and minimizing **B** and vise versa
- Multiplicative Update (MU)
  - J_{nmf} is minimized by fixing **A** and **B**
  - **A** and **B** are updated using a multiplicative update rule
- Hierarchical Alternating Least Squares (HALS)
  - J_{nmf} is minimized by fixing **A** and **B**
  - Only one column of **A** or **B** is updated using an update rule
- Multiplicative Update (MU) with Sparsity Constraint
  - J_{nmf} is minimized by fixing **A** and **B**
  - **A** and **B** are updated using a sparse multiplicative update rule

Alternating Least Squares Optimization

$$J_{\text{nmf}} = \arg\min_{\mathbf{A},\mathbf{B}} \frac{1}{2} \left|\left| \mathbf{Y} - \mathbf{AB} \right|\right|_2^2 \quad \text{s.t.} \quad \mathbf{A}, \mathbf{B} \geq 0$$

	Algorithm: Alternating Least Squares (ALS)
1:	Initialize **A** and **B**
2:	**Repeat**
3:	solve: $\arg\min_{\mathbf{B}} \frac{1}{2} \|\|\mathbf{Y} - \mathbf{AB}\|\|_2^2 \quad \text{s.t.} \quad \mathbf{A}, \mathbf{B} \geq 0$
4:	solve: $\arg\min_{\mathbf{A}} \frac{1}{2} \|\|\mathbf{Y} - \mathbf{AB}\|\|_2^2 \quad \text{s.t.} \quad \mathbf{A}, \mathbf{B} \geq 0$
5:	**Stopping Condition**

[Paatero and Tapper '94]

Multiplicative Update Optimization

$$J_{\text{nmf}} = \arg\min_{\mathbf{A},\mathbf{B}} \frac{1}{2} ||\mathbf{Y} - \mathbf{AB}||_2^2 \quad \text{s.t.} \quad \mathbf{A}, \mathbf{B} \geq 0$$

	Algorithm: Multiplicative Update (MU)
1:	Initialize $\mathbf{A}^0$, $\mathbf{B}^0$, k = 0
2:	**Repeat**
3:	$A_{ir}^{k+1} = A_{ir}^{k} \frac{\left(\mathbf{Y}\mathbf{B}^k\right)_{ir}}{\left(\mathbf{A}^k (\mathbf{B}^k)^T \mathbf{B}^k\right)_{ir}}, \quad 1 \leq i \leq M, 1 \leq r \leq R$
4:	$B_{rj}^{k+1} = B_{rj}^{k} \frac{\left(\mathbf{Y}^T \mathbf{A}^{k+1}\right)_{rj}}{\left(\mathbf{B}^k (\mathbf{A}^{k+1})^T \mathbf{A}^{k+1}\right)_{rj}}, \quad 1 \leq j \leq N, 1 \leq r \leq R$
5:	$k = k + 1$
6:	**Stopping Condition**

[Lee and Seung '99]

	Algorithm: HALS / NN-KSVD
1:	Random Initialization of $\mathbf{A}^0$, J = 1
2:	**Repeat**
3:	Use pursuit algorithm to compute $\{\mathbf{b}_j\}$
	$\arg\min_{\mathbf{B}} \lvert\lvert \mathbf{Y}_i - \mathbf{AB} \rvert\rvert_2^2 \quad \lVert \mathbf{b} \rVert_0 \leq T_0$
4:	Update Stage: For k = 1, 2, . . . , R
	Define group that use $\{\mathbf{a}_k\} : w_k, i \in N, \mathbf{b}_i(k) \neq 0$
	Compute: $\mathbf{E}_k = \mathbf{Y} - (\mathbf{AB} - \mathbf{a}_k(\mathbf{b}^k)^T)$
	Restrict $\mathbf{E}_k$, choose only columns corresponding to w_k, get $\mathbf{E}_k^{w_k}$
	Apply SVD, $\mathbf{E}_k^{w_k} = \mathbf{U}\Delta\mathbf{V}^T, \quad \mathbf{a}_k = \mathbf{u}_1, \quad \mathbf{b}^k = \Delta(1,1)\mathbf{v}_1$
5:	$J = J + 1$
6:	**Stopping Condition**

[Michal Aharon, et al '06]

Multiplicative Update With Sparsity Optimization

$$J_{\text{nmf}} = \arg\min_{\mathbf{A},\mathbf{B}} \frac{1}{2}||\mathbf{Y} - \mathbf{AB}||_2^2 \quad \text{s.t.} \quad \mathbf{A}, \mathbf{B} \geq 0, \; S(\mathbf{a}_i) = S_a, \; S(\mathbf{b}_j) = S_b$$

$$S(\mathbf{x}) = \frac{\sqrt{n} - \|\mathbf{x}\|_1 / \|\mathbf{x}\|_2}{\sqrt{n} - 1} \quad \text{s.t.} \quad n = \dim(\mathbf{x})$$

	Algorithm: Multiplicative Update with Sparsity (MU)
1:	Initialize $\mathbf{A}^0$, $\mathbf{B}^0$, $k = 0$, $\mu_{\mathbf{A}}$, $\mu_{\mathbf{B}}$
2:	**Repeat**
3:	$\mathbf{A} = \mathbf{A} - \mu_{\mathbf{A}}(\mathbf{AB} - \mathbf{Y})\mathbf{B}^T$
4:	project: $\{\mathbf{a}_i\}$ s.t. $\|\mathbf{a}_i\|_1 = S_a$
5:	$\mathbf{B} = \mathbf{B} - \mu_{\mathbf{B}}\mathbf{A}^T(\mathbf{AB} - \mathbf{Y})$
6:	project: $\{\mathbf{b}_j\}$ s.t. $\|\mathbf{b}_j\|_1 = S_b$
7:	**Stopping Condition**

[Hoyer '04]

Performance

Big-O Complexity of NMF Algorithms

$$\mathbf{Y} = \mathbf{AB} + \mathbf{E},\ \mathbf{Y} \in \mathbb{R}^{M \times N}, \mathbf{A} \in \mathbb{R}^{M \times R}, \mathbf{B} \in \mathbb{R}^{R \times N}$$

- For R = 1: "Easy", [Perron-Frobenius and Eckart-Young theorems '11]
- Rank($\mathbf{Y}$) = 2: Rank$_+$($\mathbf{Y}$) = 2, "Easy"
- Rank($\mathbf{Y}$) = R: $\mathcal{O}((M \times N)^{R^2})$, [Arora et al. '12, Moitra '13]

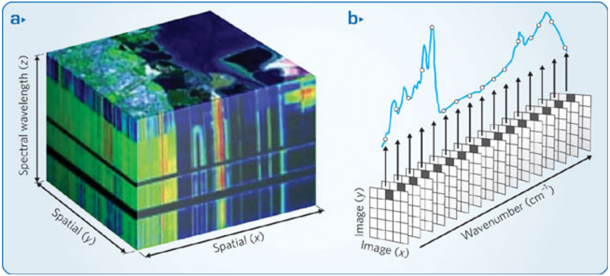
Algorithm	Complexity per iteration
ALS:	$\mathcal{O}(M \times N \times R)$
MU:	$\mathcal{O}(M \times N \times R)$
HALS:	$\mathcal{O}(M \times N \times R)$

[Guoxu Zhou, et al '14]

Other Methods

- Inexact Alternating Least Square (IALS)
- Accelerating Proximal Gradient (APG)
- Seperable NMF (Sep-NMF)
- Squential NMF (Seq-NMF)
- Multi-factor NMF (MF-NMF)
- NMF Based on Low Rank Approximations
- Non-negative Sparse Coding
- Many Others...

Problem Description



[David Bannon '09]

- HSI pixels are non-negative
- Pixels are superpositions of a finite number of materials with non-negative reflectance
- Applying NMF to solve the unsupervised HSI unmixing problem

Diversity in Hyperspectral Images (HSI)

- Minimize mutual coherence between $\{\mathbf{a}_i\}$

$$M(\{\mathbf{a}_i\}) = \max_{1 \leq i \neq j \leq R} \frac{|\mathbf{a}_i^T \mathbf{a}_j|}{\|\mathbf{a}_i\|_2 \|\mathbf{a}_j\|_2}$$

- Minimize sparsity of $\{\mathbf{b}_j\}$

$$1 \leq \|\mathbf{b}_j\|_0 \leq R$$

HSI Decomposition

- What are the basis vectors that "best" represent the image?
  - The material types can be represented by the columns of **A**
  - You can group and classify pixels from these types
- What are the coefficients for each pixel in the image?
  - The abundance fractions can be represented by the columns of **B**
- Each pixel in **Y** is then represented by a linear combination of **A** and columns of **B**

$$Y_{ij} \approx \sum_{r=1}^{R} A_{ir} B_{rj}$$

- The error due to approximation is

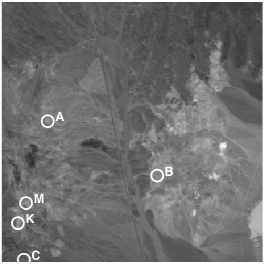
$$\mathbf{E} = \mathbf{Y} - \mathbf{AB}$$

Linear Unmixing

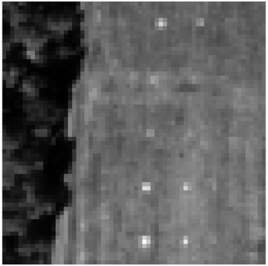
- What are the basis vectors that “best” represent the image?
  - The material types can be represented by the columns of **A**
  - You can group and classify pixels from these types
- What are the coefficients for each pixel in the image?
  - The abundance fractions can be represented by the columns of **B**
- Each pixel in **Y** is then represented by a linear combination of the columns of **A**

$$Y_{ij} \approx \sum_{r=1}^{R} A_{ir} B_{rj}$$

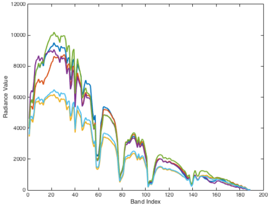
Collected and Synthetic Data



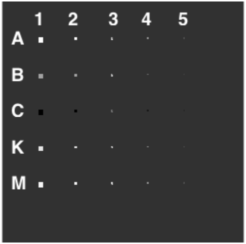
Cuprite Dimension 350x350x189



Hydice Dimension 64x64x169

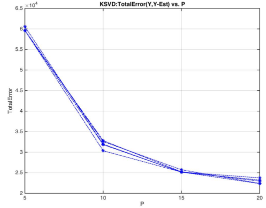


Plots of marked endmembers

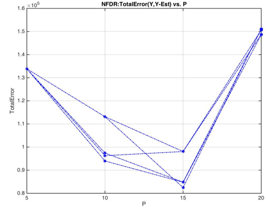


Synthetic 200x200x189

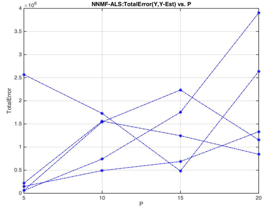
Numerical Error & Mutual Coherence



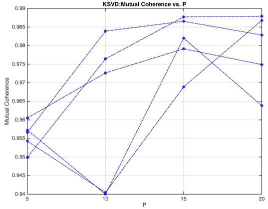
KSVD Error vs. P



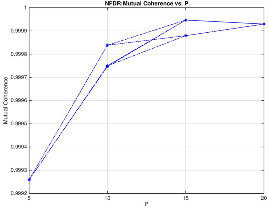
NFDR Error vs. P



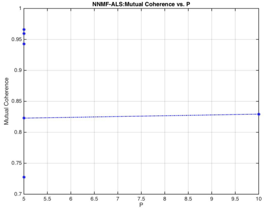
ALS Error vs. P



KSVD MC vs. P

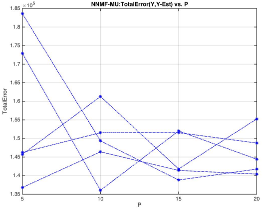


NFDR MC vs. P

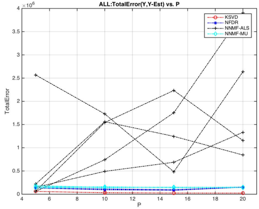


ALS MC vs. P

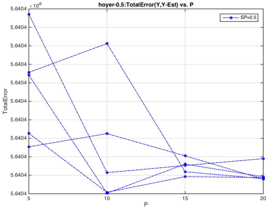
Numerical Error & Mutual Coherence



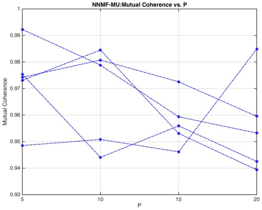
MU Error vs. P



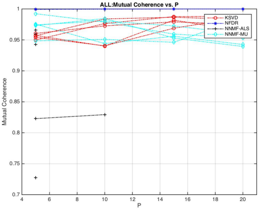
ALL Error vs. P



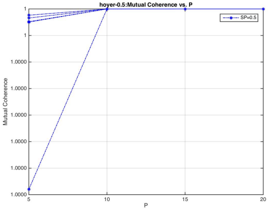
Hoyer 0.5 Error vs. P



MU MC vs. P

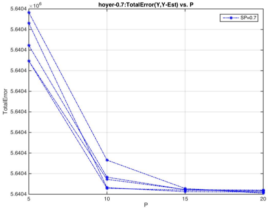


ALL MC vs. P

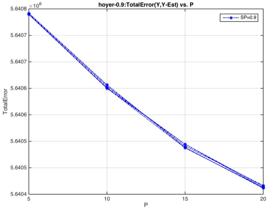


Hoyer 0.5 MC vs. P

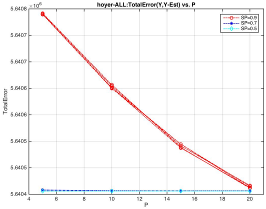
Numerical Error & Mutual Coherence



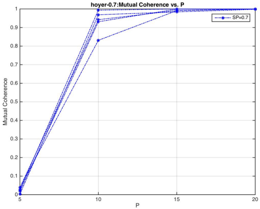
Hoyer 0.7 Error vs. P



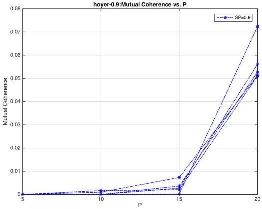
Hoyer 0.9 Error vs. P



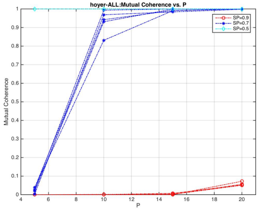
Hoyer All Error vs. P



Hoyer 0.7 MC vs. P

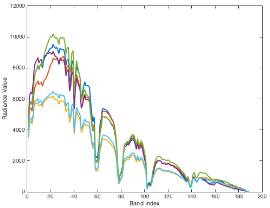


Hoyer 0.9 MC vs. P

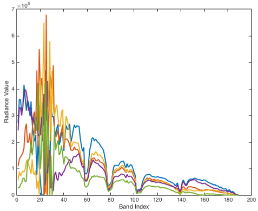


Hoyer All MC vs. P

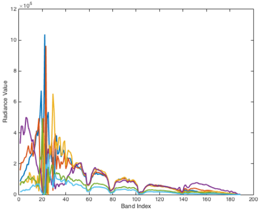
Reconstruction ALS and MU



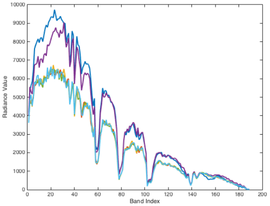
Original Radiance Data



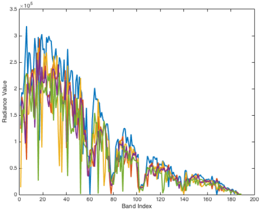
ALS Result



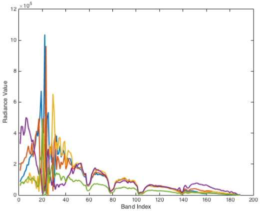
ALS/Mult Result



NFINDR Result

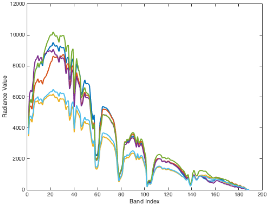


Multiplicative Result

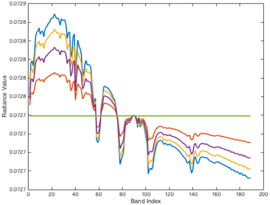


Mult/ALS Result

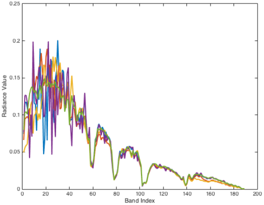
Reconstruction Hoyer and NN-KSVD



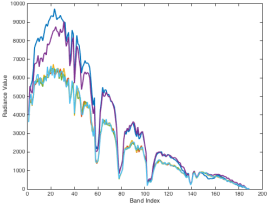
Original Radiance Data



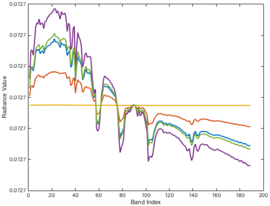
Hoyer (Sparsity $= 0.7$)



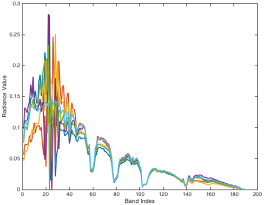
KSVD 10 Iterations



NFINDR Result



Hoyer (Sparsity $= 0.99$)



KSVD 250 Iterations

Algorithm Rankings

Criteria	**Best**				**Worst**
Abs Error	KSVD	NFINDR	MU	ALS	Hoyer
Repeatability	Hoyer	KSVD	NFINDR	MU	ALS
Increasing P	KSVD	Hoyer	NFINDR	ALS	MU
Shape	Hoyer	NFINDR	KSVD	ALS	MU
Run time	ALS	MU	Hoyer	NFINDR	KSVD

Conclusions

- NMF can be used to do linear unmixing of HSI data into constituent materials
- Scale and permutation ambiguity need application data to resolve
- Sparsity helps considerably with the uniqueness and repeatability
  - However, sparsity needs to be known a-priori
- Sparsity did resolve the shape better
  - However, Hoyer frequently converged to a uniform answer
- The run time of KSVD was much larger than the other algorithms

References

Chein-I Chang (2013)
Hyperspectral Data Processing: Algorithms Design and Analysis
Wiley, Appendix A.4.2

Chris Ding, He Xiaofeng, Horst Simon (2005)
On the Equivalence of Nonnegative Matrix Factorization and Spectral Clustering
Proceedings of SIAM Data Mining Conference 4, 606 – 610.

Chris Ding, Li Tao, Peng Wei (April 2008)
On the Equivalence Between Non-negative Matrix Factorization and Probabilistic Latent Semantic Indexing
Computational Statistics and Data Analysis 52.8, 3913 – 3927.

Daniel D. Lee, and H. Sebastian Seung (October 1999)
Learning the parts of objects by non-negative matrix factorization
Nature 401, 788 – 791.

References

Daniel D. Lee, and H. Sebastian Seung
Algorithms for Non-negative Matrix Factorization
Advances in Neural Information Processing Systems 556 – 562.

David Bannon (2009)
Hyperspectral Imaging: Cubes and Slices
Nature Photonic 3, 627 - 629.

David L. Donoho (March 2003)
Optimally Sparse Representation in General Dictionaries via L1 Minimization
Proceeds of the National Academy of Science 100, 2197 – 2202.

Guoxu Zhou, Andrzej Cichocki, Qibin Zhao, and Shengli Xie (May 2014)
Nonnegative Matrix and Tensor Factorizations
IEEE Signal Processing 54(14).

References

Ivana Tosic, Pascal Frossard (March 2011)
Dictionary Learning, What is the right representation for my signal?
IEEE Signal Processing Magazine March 2011, 27 – 38.

Michal Aharon, Michael Elad, Alfred Bruckstein (November 2006)
K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation
IEEE Trans. on Signal Processing Vol. 54, No. 11, 4311 – 4322.

Michal Aharon and Michael Elad (2006)
https://github.com/jbhuang0604/SelfSimSR/tree/master/Lib/KSVD

Miles Lopes (2013)
Estimating unknown sparsity in compressed sensing
International Conf. on Machine Learning (ICML) arXiv:1204.4227.

References

Miles Lopes (2015)
Compressed Sensing without Sparsity Assumption
http://arxiv.org/abs/1507.07094 arXiv:1507.07094.

Ondrej Mandula (2011)
https://github.com/aludnam/MATLAB/tree/master/nmfpack

Patrik Hoyer (February 2002)
Non-negative Sparse Coding
Neural Networks for Signal Processing 12, 557 – 565.

Patrik Hoyer (November 2004)
Non-negative Matrix Factorization with Sparseness Constraints
Machine Learning Research 5, 1457 – 1469.

Pentti Paatero, Unto Tapper (June 1994)
Positive Matrix Factorization: A non-negative factor model with optimal utilization of error estimates of data values
Environmetrics 5(2), 111 – 126.