nsNMF
NONSMOOTH NONNEGATIVE MATRIX FACTORIZATION

Alberto Pascual-Montano, Member, IEEE, J.M. Carazo, Senior Member, IEEE, Kieko Kochi, Dietrich Lehmann, and Roberto D. Pascual-Marqui

Jayitha .C - 20171401
Mehthab Shaik - 20161205

Motivation

- All the previous methods achieve sparseness on NMF through addition of constraints / penalties
- The constraints are applied to basis vectors ,encoding vectors or both
- As basis vectors are multiplied with the encoding vectors sparseness in one forces smoothness in the other to keep close to original data if constraints on either.
- If constraints on both - accuracy lowered.
- Additive parts-based sparse representation

Non-negative Matrix Factorization

- Learn part based representation.
- NMF is distinguished from other methods like PCA and vector quantization by its use of non-negativity constraints.
- These constraints lead to a part-based representation as they allow only additive not subtractive combinations.
- We take

$$\mathbf{V} \approx \mathbf{W}\mathbf{H},$$

- We take the divergence function derived from Poisson likelihood as the objective function and minimize it.
- $$D(\mathbf{V}, \mathbf{WH}) = \sum_{i=1}^{p}\sum_{j=1}^{n}\left(V_{ij}\ln\frac{V_{ij}}{(\mathbf{WH})_{ij}} - V_{ij} + (\mathbf{WH})_{ij}\right),$$
- Simplifying it -

$$D(\mathbf{V}, \mathbf{WH}) = \sum_{i=1}^{p}\sum_{j=1}^{n}\left(\sum_{k=1}^{q} W_{ik}H_{kj} - V_{ij}\ln\sum_{k=1}^{q} W_{ik}H_{kj}\right).$$

$$\frac{\partial}{\partial H_{ab}} D(\mathbf{V}, \mathbf{WH}) = \sum_{i=1}^{p} W_{ia} - \sum_{i=1}^{p} \frac{V_{ib} W_{ia}}{\sum_{k=1}^{q} W_{ik} H_{kb}} .$$

$$H_{ab} \leftarrow H_{ab} - \eta_{ab} \frac{\partial}{\partial H_{ab}} D(\mathbf{V}, \mathbf{WH}),$$

$$H_{ab} \leftarrow H_{ab} + \eta_{ab} \left[\sum_{i=1}^{p} \frac{V_{ib} W_{ia}}{\sum_{k=1}^{q} W_{ik} H_{kb}} - \sum_{i=1}^{p} W_{ia} \right],$$

$$\eta_{ab} = \frac{H_{ab}}{\sum_{i=1}^{p} W_{ia}}$$

$$H_{ab} \leftarrow H_{ab} \frac{\sum_{i=1}^{p} (W_{ia} V_{ib}) / \sum_{k=1}^{q} W_{ik} H_{kb}}{\sum_{i=1}^{p} W_{ia}} .$$

$$\frac{\partial}{\partial W_{cd}} D(\mathbf{V}, \mathbf{WH}) = \sum_{j=1}^{n} H_{dj} - \sum_{j=1}^{n} \frac{V_{cj} H_{dj}}{\sum_{k=1}^{q} W_{ck} H_{kj}}.$$

$$W_{cd} \leftarrow W_{cd} - \nu_{cd} \frac{\partial}{\partial W_{cd}} D(\mathbf{V}, \mathbf{WH}),$$

$$W_{cd} \leftarrow W_{cd} + \nu_{cd} \left[\sum_{j=1}^{n} V_{cj} \frac{H_{dj}}{\sum_{k=1}^{q} W_{ck} H_{kj}} - \sum_{j=1}^{n} H_{dj} \right].$$

$$\nu_{cd} = \frac{W_{cd}}{\sum_{j=1}^{n} H_{dj}}$$

$$W_{cd} \leftarrow W_{cd} \frac{\sum_{j=1}^{n} (H_{dj} V_{cj}) / \sum_{k=1}^{q} W_{ck} H_{kj}}{\sum_{j=1}^{n} H_{dj}}.$$

Repeat until convergence:
For a = 1...q do begin
For b = 1...n do

$$H_{ab} \leftarrow H_{ab} \frac{\sum_{i=1}^{p} (W_{ia} V_{ib}) / \sum_{k=1}^{q} W_{ik} H_{kb}}{\sum_{i=1}^{p} W_{ia}}.$$

For c=1...p do begin

$$W_{ca} \leftarrow W_{ca} \frac{\sum_{j=1}^{n} (H_{aj} V_{cj}) / \sum_{k=1}^{q} W_{ck} H_{kj}}{\sum_{j=1}^{n} H_{aj}}.$$

$$W_{ca} \leftarrow \frac{W_{ca}}{\sum_{j=1}^{n} W_{ja}}.$$

End

End

Non-negative Sparse Coding - NNSC

- Linear Sparse coding of data find representations of data such that the hidden components are "sparse".
- This basically means any given input vector can be well represented using only a few significantly non-zero hidden coefficients.
- The objective function combining reconstruction and sparseness is of the form:

$$C(\mathbf{A}, \mathbf{S}) = \frac{1}{2}\|\mathbf{X} - \mathbf{A}\mathbf{S}\|^2 + \lambda \sum_{ij} f(S_{ij}),$$

- The trade-off between sparse and accurate reconstruction is controlled by lambda.

- f defines how sparseness is measured.
- To achieve a sparse code f needs to be chosen correctly
- f is typically strictly increasing in absolute value of his argument.
- Ex:-

$$f(s) = |s|$$

$$\|\mathbf{X}-\mathbf{AS}\|^2 = \sum_{ij}[\mathbf{X}_{ij}-(\mathbf{AS})_{ij}]^2$$

- Sparseness aside non-negativity constraints with this objective gave good results:
-

$$C(\mathbf{A},\mathbf{S}) = \frac{1}{2}\|\mathbf{X}-\mathbf{AS}\|^2$$

$$A_{ij} \geq 0, \; S_{ij} \geq 0.$$

- Combining the two final objective becomes :

$$C(\mathbf{A}, \mathbf{S}) = \frac{1}{2}\|\mathbf{X} - \mathbf{AS}\|^2 + \lambda \sum_{ij} S_{ij}$$

subject to the constraints $\forall ij: \; A_{ij} \geq 0, \; S_{ij} \geq 0$ *and* $\forall i: \|\mathbf{a}_i\| = 1$, *where* $\mathbf{a}$ *denotes the* i*:th column of* $\mathbf{A}$. *It is also assumed that the constant* $\lambda \geq 0$.

- The Algorithm :
- 1.Initialize W and H to random strictly positive matrices of appropriate dimensions,normalize each column of W. Let $\mu > 0$ denote the step size.
- Until convergence do :

a. Calculate new **W** as

$$\mathbf{W} \leftarrow \mathbf{W} - \mu(\mathbf{W}\mathbf{H} - \mathbf{V})\mathbf{H}^t.$$

b. Any negative values in **W** are set to zero
c. Normalize each column of **W**.
d. Calculate new **H** as

$$H_{i,j} \leftarrow H_{i,j} \frac{(\mathbf{W}^t\mathbf{V})_{ij}}{(\mathbf{W}^t\mathbf{W}\mathbf{H})_{ij} + \lambda)}.$$

Sparse Non-negative Matrix Factorization

- additional sparseness constraint is imposed on encoding coefficient matrix explicitly.
- This allows it to learn much sparser features.
- The objective function -
-

$$D(\mathbf{V}, \mathbf{WH}) = \sum_{i=1}^{p}\sum_{j=1}^{n}\left(V_{ij}\ln\frac{V_{ij}}{(\mathbf{WH})_{ij}} - V_{ij} + (\mathbf{WH})_{ij}\right) + \alpha\sum_{ij} H_{ij}$$

$\alpha \geq 0.$

- The sparseness is forced via minimizing the sum of all H_{ij}.
- W,H >= 0
- The Algorithm :

Repeat until convergence:
For a = 1...q do begin
For b = 1...n do

$$H_{ab} \leftarrow H_{ab} \frac{\sum_{i=1}^{p}(W_{ia}V_{ib}) / \sum_{k=1}^{q} W_{ik}H_{kb}}{1+\alpha},$$

For c=1...p do begin

$$W_{ca} \leftarrow W_{ca} \frac{\sum_{j=1}^{n}(H_{aj}V_{cj}) / \sum_{k=1}^{q} W_{ck}H_{kj}}{\sum_{j=1}^{n} H_{aj}}.$$

$$W_{ca} \leftarrow \frac{W_{ca}}{\sum_{j=1}^{n} W_{ja}}.$$

End

End

2.2 Local Nonnegative Matrix Factorization (LNMF)

T. Feng, S.Z. Li, H.-Y. Shum, and H. Zhang, "Local Nonnegative Matrix Factorization as a Visual Representation," Proc. Second Int'l Conf. Development and Learning (ICDL '02), 2002

- ➔ Intended for learning spatially localized, parts-based representation of visual patterns

- ➔ Part-based representation of objects achieved by imposing
  - ◆ sparseness constraints on the encoding vectors (matrix H) and
  - ◆ locality constraints to the basis components (matrix W)

Define where $\mathbf{A} = [a_{ij}] = \mathbf{W}^t\mathbf{W}$ and $\mathbf{B} = [b_{ij}] = \mathbf{H}\mathbf{H}^t$ where $\mathbf{A},\mathbf{B} \in \mathbb{R}^{q \times q}$

The LNMF algorithm is based on the following three additional constraints:

- ❑ Maximum Sparseness in **H**
- ❑ Maximum expressiveness of **W**
- ❑ Maximum orthogonality of **W**

2.2 Local Nonnegative Matrix Factorization (LNMF)

- *Maximum Sparseness in **H***
  - H should contain as many zero components as possible
  - Basis component should not be further decomposed into more components
  - For all j, given $||b_j||_1 = 1$, we need to minimize

    $\sum_{i=1}^{n} b_{ij}^2$

    Therefore we need to minimize a_{ii} for all i

- *Maximum expressiveness of **W***
  - The amount of information about example x_j carried by component $\mathbf{b_i}$ is measured by the "*activity*" of the example on the component defined as h_{ij}^2
  - The total activity of all examples on the component $\mathbf{b_i}$ is --------> $\sum_{j=1}^{N_T} h_{ij}^2$
  - The total activity on all the learned components is ----> $\sum_{i=1}^{m} \sum_{j=1}^{N_T} h_{ij}^2$

    Therefore we need to maximise $\sum_{i=1}^{q} b_{ii}$ notation here changed to **B**

2.2 Local Nonnegative Matrix Factorization (LNMF)

- *Maximum orthogonality of **W***
  - Different bases should be as orthogonal as possible, so as to minimize redundancy between different bases
  - This is forced by minimizing $\sum_{\forall i,j,i\neq j} a_{ij}$

Combining constraints from both constraints (1) and (3), we have to minimize $\sum_{\forall i,j} a_{ij}$.

This ∀i,j, the constrained divergence function described is

$$D(\mathbf{V},\mathbf{WH}) = \sum_{i=1}^{p}\sum_{j=1}^{n}\left(V_{ij}\ln\frac{V_{ij}}{(\mathbf{WH})_{ij}} - V_{ij} + (\mathbf{WH})_{ij}\right) + \alpha\sum_{i,j=1}^{q} a_{ij} - \beta\sum_{i=1}^{q} b_{ii},$$

Where $\alpha,\beta > 0$ represent some constants for expressing the importance of the additional constraints described above.

2.2 Local Nonnegative Matrix Factorization (LNMF)

Repeat until convergence:

For a =1...q do begin

For b = 1...n do

$$H_{ab} \leftarrow \sqrt{H_{ab} \sum_{i=1}^{p} (W_{ia} V_{ib}) / \sum_{k=1}^{q} W_{ik} H_{kb}}.$$

For c = 1...p

$$W_{ca} \leftarrow W_{ca} \frac{\sum_{j=1}^{n} (H_{aj} V_{cj}) / \sum_{k=1}^{q} W_{ck} H_{kj}}{\sum_{j=1}^{n} H_{aj}}.$$

$$W_{ca} \leftarrow \frac{W_{ca}}{\sum_{j=1}^{n} W_{ja}}$$

End

2.5 Nonnegative Matrix Factorization with Sparseness Constraints (NMFSC)

P.O. Hoyer, "Nonnegative Matrix Factorization with Sparseness Constraints," J.MachineLearningResearch,vol. 5,pp. 1457-1469, 2004

- ➔ Another method related to the addition of sparseness constraints to the NMF problem
- ➔ Minimizes $E(\mathbf{V}, \mathbf{WH}) = \|\mathbf{V} - \mathbf{WH}\|^2$ Under the following constraints

$$\text{Sparseness}(\mathbf{W}_i) = S_w, \forall_i, i = 1..q,$$
$$\text{Sparseness}(\mathbf{H}_i) = S_h, \forall_i, i = 1..q,$$

where $\mathbf{W}_i$ is the *i*th column of **W**, $\mathbf{H}_i$ is the *i*th row of **H**, S_w and S_h are the desired sparseness values for **W** and **H**, respectively, and are user-defined parameters.

2.5 Nonnegative Matrix Factorization with Sparseness Constraints (NMFSC)

$$\text{Sparseness}(\mathbf{x}) = \frac{\sqrt{n} - \left(\sum |x_j|\right) / \sqrt{\sum x_j^2}}{\sqrt{n-1}}$$

Where n is the dimensionality of **x**

- This sparseness measure quantifies how much energy of a vector is packed into only a few components
- This function evaluates to 1 if and only if **x** contains only a single nonzero component, and takes a value of 0 if and only if all components are equal, interpolating smoothly between the two extremes

Nonsmooth Nonnegative Matrix Factorization

- All the previous methods achieve sparseness on NMF through addition of constraints
- The constraints are applied to basis vectors, encoding vectors or both
- As the basis vectors are multiplied with encoding vector sparseness in one implies smoothness in the other if constraints are on either
- If constraints on both then accuracy is lowered
- Therefore to tackle these issues the model(V = WH) is itself modified to

$$\mathbf{V} = \mathbf{WSH},$$

- V,W,H are the same as NMF S is the smoothing matrix defined as

$$\mathbf{S} = (1-\theta)\mathbf{I} + \frac{\theta}{q}\mathbf{1}\mathbf{1}^T, \qquad 0 \le \theta \le 1$$

Smoothing Effect of S

- Let Y = SX ,where X is a positive nonzero vector
- When theta is equal to 0 S = I and Y = X , hence no smoothing on X
- When theta = 1 then S is all ones,every element of Y = average of all elements of X ,this is the smoothest possible Y in terms of non-sparseness as now depending on threshold X would either remain X or become zero.
- Thus if S has greater smoothing (higher theta), it forces sparseness on W & H to stay close to original data.
- V = W(SH) - forces sparseness on W
- V = (WS)H - forces sparseness on H
- Thus combined effect is both W & H are sparse.
-

- The final algorithm - in the algorithm for original NMF,
- In the update equation for H , substitute W
- with (WS)
- In the update equation for W , substitute H with (SH).
-

4.1 Experiments on Synthetic Data

Multiplicative nature of the sparse variants of the NMF model will produce a paradoxical effect: Imposing sparseness in one of the factors will almost certainly force smoothness in the other in an attempt to reproduce the data as best as possible.
Additionally, forcing sparseness constraints on both the basis and the encoding vectors will decrease the explained variance of the data by the model

Where Generated a dataset of measuring 5 variables over 20 objects, i.e. $\mathbf{V} \in \mathbb{R}^{5x20}$, this matrix is generated by generating 2 matrices $\mathbf{A} \in \mathbb{R}^{5x3}$ and $\mathbf{B} \in \mathbb{R}^{3x20}$. We set $\mathbf{V} = \mathbf{A} \times \mathbf{B}$

Comparison of Results for Different NMF Models for Different Levels of Sparseness

Method	Sparseness constraint	Explained variance (%)	Average sparseness on W	Average sparseness on H
NMF	-	99.99	0.64	0.20
LNMF	-	93.13	0.88	0.05
SNMF	0.1	99.17	0.63	0.20
	0.2	97.22	0.62	0.20
	0.3	94.67	0.67	0.18
	0.4	91.84	0.63	0.20
	0.5	88.89	0.65	0.19
	0.6	85.93	0.61	0.21
	0.7	83.04	0.64	0.19
	0.8	80.25	0.63	0.19
	0.9	77.43	0.59	0.22
NMFSC	0.1	87.11	0.1	0.1
	0.2	93.54	0.2	0.2
	0.3	97.7	0.3	0.3
	0.4	99.30	0.4	0.4
	0.5	96.67	0.5	0.5
	0.6	84.59	0.6	0.6
	0.7	64.75	0.7	0.7
	0.8	38.83	0.8	0.8
	0.9	26.18	0.9	0.9
***ns*NMF**	0.1	99.99	0.66	0.21
	0.2	99.99	0.71	0.22
	0.3	99.99	0.78	0.21
	0.4	99.98	0.84	0.22
	0.5	98.92	0.54	0.45
	0.6	99.30	0.87	0.26
	0.7	98.36	0.87	0.29
	0.8	95.65	0.56	0.52
	0.9	94.24	0.86	0.42

As expected, NMF achieves 100 percent explained variance, with low sparseness values.

The LNMF method, which has no control over the extent of sparseness, explains only 93 percent of the variance while achieving high sparseness for the basis vectors, but extremely low sparseness for the encoding vectors

Despite the fact that the SNMF method is designed to control sparseness, it seems to be incapable of obtaining an actual increase in sparseness, while the explained variance deteriorates tremendously.

The NMFSC method performs as expected, enforcing sparseness, but at the expense of a dramatic loss of faithfulness between the data and the model

The new nsNMF model maintains almost perfect faithfulness to the data (> 99.9 percent explained variance) for a wide range of achieved sparseness, thus outperforming the other methods.

4.2 Experiments on Swimmers Data Set

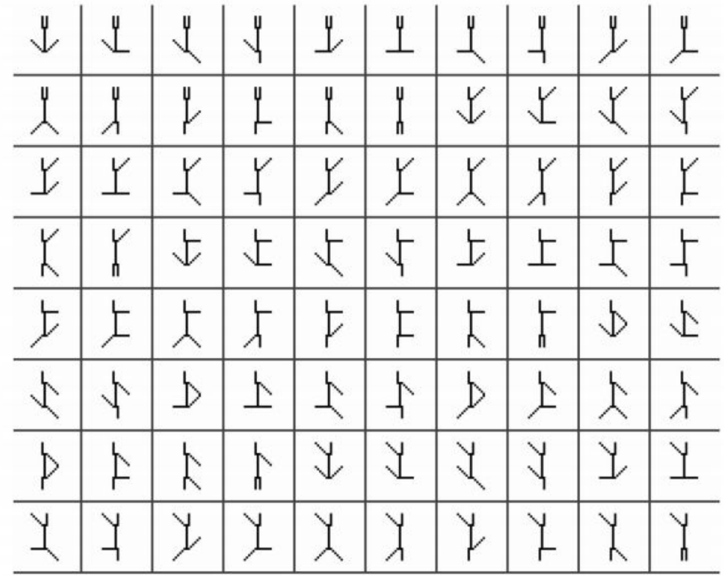


Fig. 1. Sample images from the *swimmer* data set. Each image is composed of a torso and four limbs in different positions.

It consists of a set of black-and-white images with four moving parts (limbs), each able to exhibit four different positions (articulations).

Each individual image contains a “torso” of 12 pixels in the center and four “limbs” of six pixels that can be in one of four different positions.

The swimmer data set was used in to demonstrate the ability of the NMF algorithm to find the parts (limbs)

The factors showed that the 16 different articulated parts were properly resolved (four limbs in four different positions and one common torso)

The torso is not properly resolved since it is explicitly an invariant region that violated the rules used for generating the data.

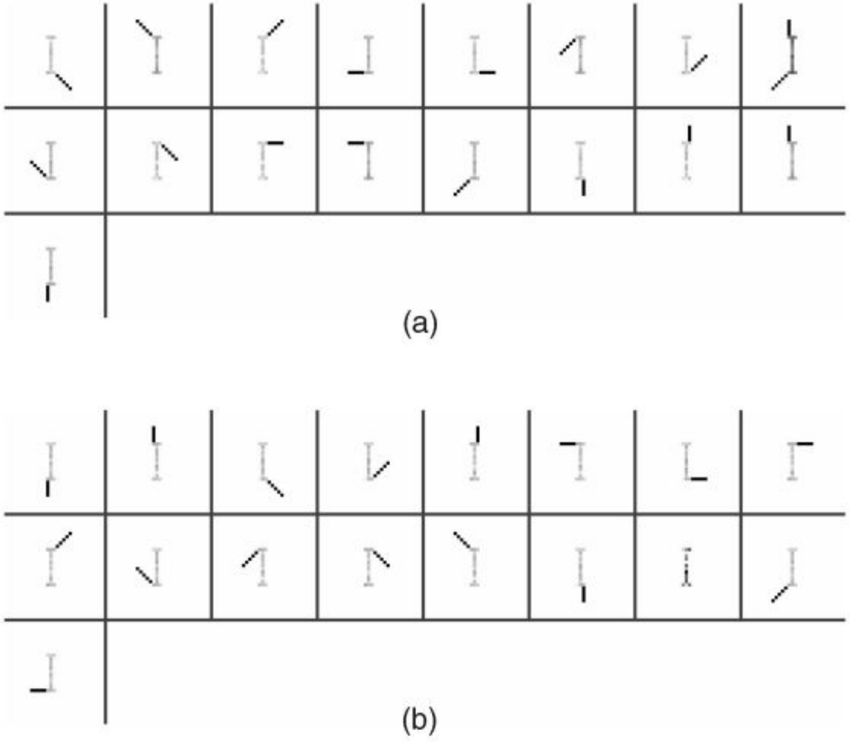


Fig. 2. Results of applying (a) NMF and (b) *ns*NMF algorithms to the *swimmer* data set. In both cases, 17 factors were generated. A value of $\theta = 0.5$ was used for the *ns*NMF model. Note that, in (a), the eighth factor of the first row represents a nonsparse representation of the two limbs and a torso, while, in (b), the seventh factor in the second row contains a stronger torso signal.

The idea is to check the ability of NMF and nsNMF in finding the limbs and the torso separately, as is expected in a parts-based representation of this data.

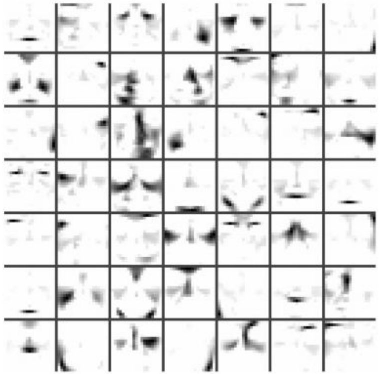
It can be noticed that NMF failed in extracting the 16 limbs and the torso, while nsNMF successfully explained the data using one factor for each independent part.

These results are in total agreement with the nature of both methods: *NMF extract parts of the data in a more holistic manner, while nsNMF sparsely represents the same reality.*

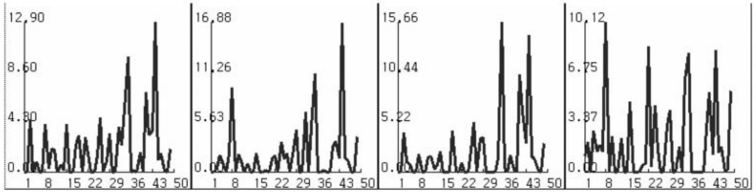
4.3 Experiments on Faces Data Set

In order to test the sparseness ability of nsNMF

The database contains 2,429, 19 x 19 facial low resolution gray-level images.



(a)



(b)

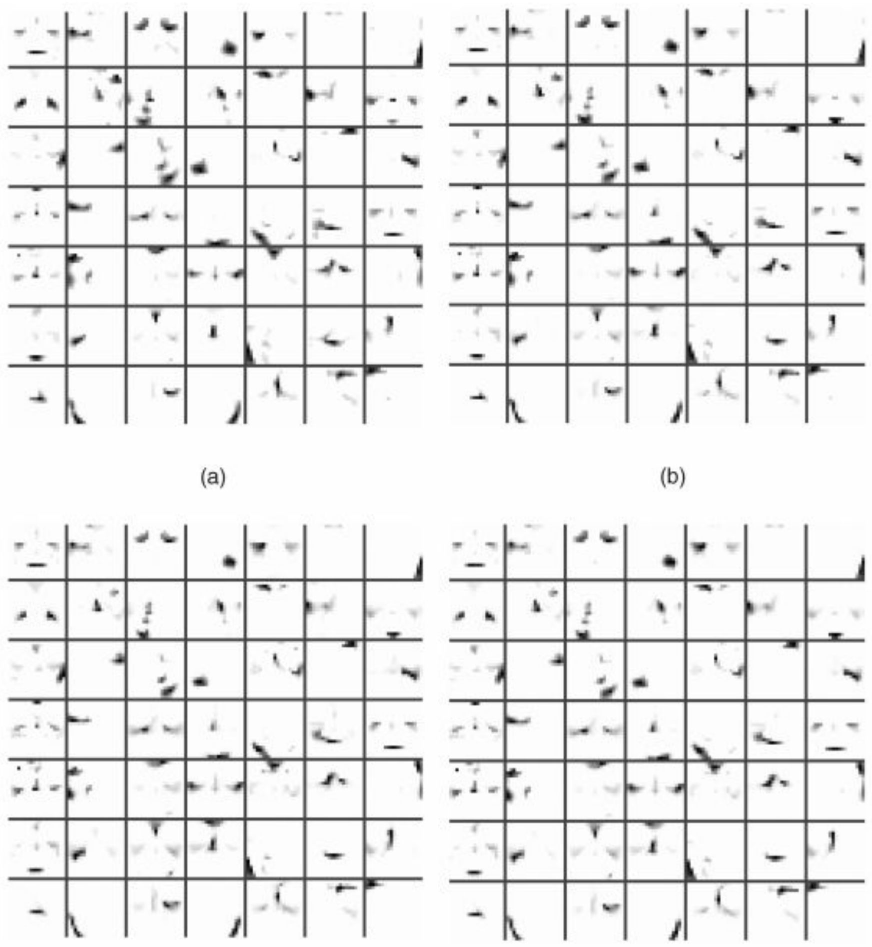
<u>*NMF Results:*</u>

Even if the factors' images give an intuitive notion of a parts-based representation of the original faces, the factorization is not really sparse enough to represent unique parts of an average face.

Allows some undesirable overlapping of parts.

The coefficients are not sparse.

In order to reproduce the input images, the generated model needs to combine almost all of the factors in different overlapped proportions.



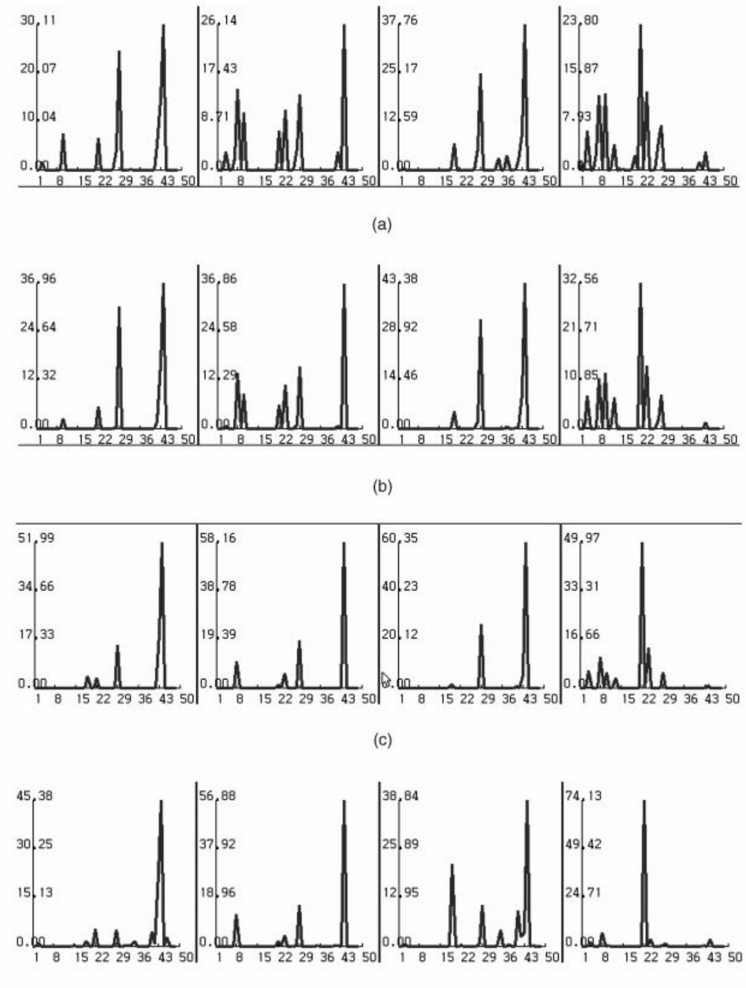
(a) (b) (c) (d)

nsNMF Results:

These results demonstrate the intrinsic nature of the nsNMF algorithm where more localized features appear with increasing values of the sparseness parameters

Less overlapping occurs due to enforcing of sparseness in both the basis vectors and the encoding vectors

nsNMF is not only able to extract localized features from a given data set, but tries to explain each item in the data set by the additive combination of a minimum number of components



(a)

(b)

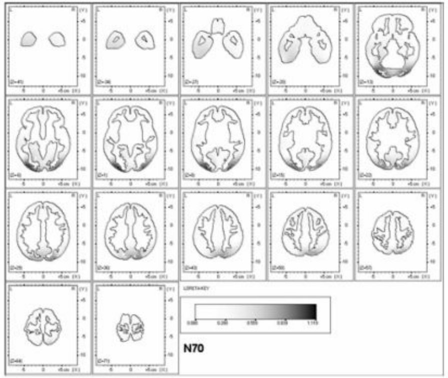
(c)

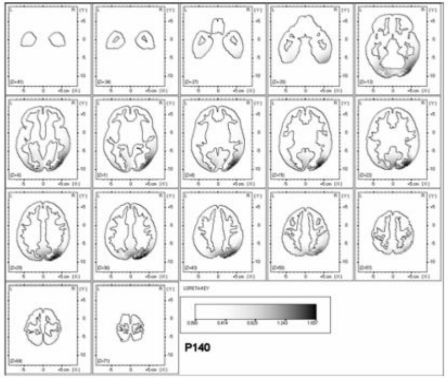
(d)

__nsNMF Results:__

Encodings in comparison to NMF are a lot more sparse

5 REAL APPLICATION: LOW-RESOLUTION BRAIN ELECTROMAGNETIC TOMOGRAPHY (LORETA)

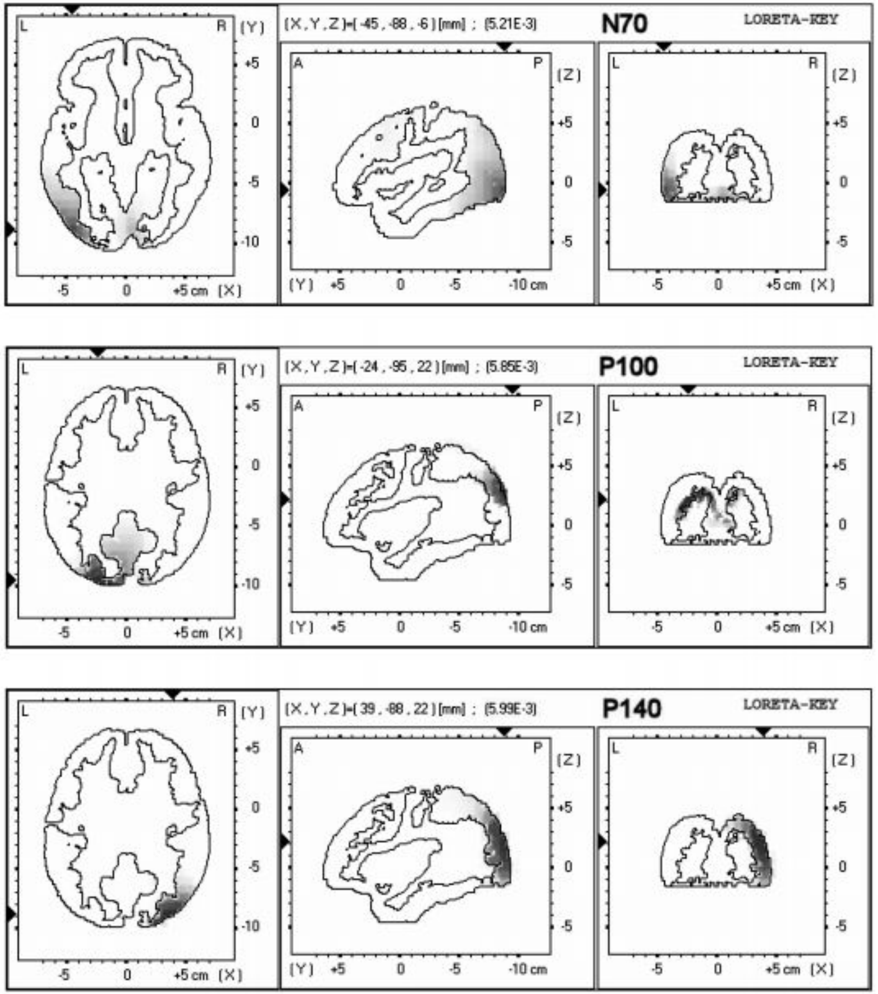


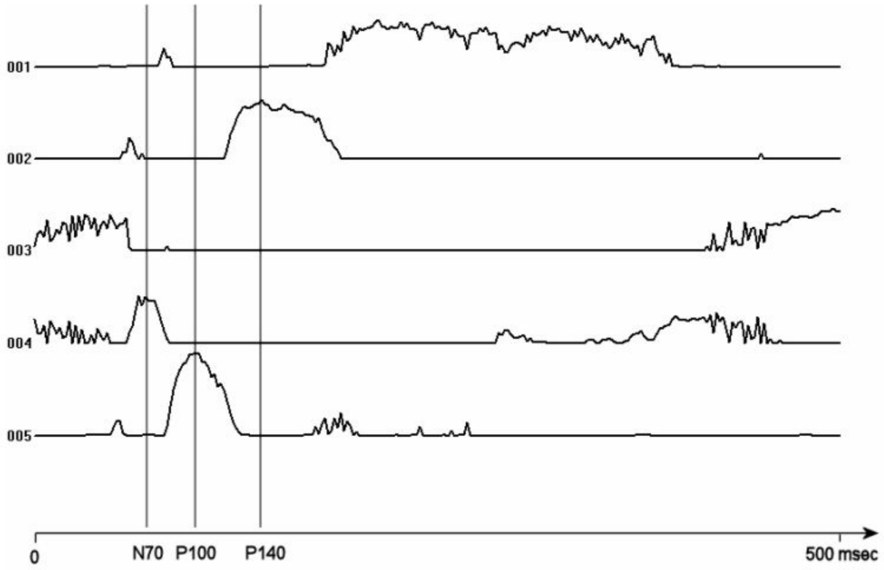


This data set consists of time series of electric neuronal activity (current density) calculated at a large number of voxels located over the human cortex

Model the spatio-temporal current densities in terms of a small number of factor pairs. In this model, the basis vectors would be certain normalized spatial distributions that have maximum activity in brain areas that are specialized in certain cognitive functions and the encoding vectors would be the time course of activation of the corresponding spatial distribution basis vector.

The basis vectors consisting of the normalized spatial distribution of current density should be zero almost everywhere, except for nonzero values at some few brain regions, and the encoding vectors consisting of time course of activation should be zero almost everywhere, except at certain time intervals.





On the left we have the basis vectors and on the right we have encodings, as we can see both are sparse. Time axis is basically being segmented.

Implementation on Synthetic Data

We implemented the following algorithms

- NMF - Non Negative Matrix Factorization
- SNMF - Sparse Non Negative Matrix Factorization
- nsNMF - Non Smooth Non Negative Matrix Factorization

Parameters for Comparison

- Explained Variance - Used sklearn package implementation, compares relative variance obtained relative to given true values
- Sparseness of W
- Sparseness of H

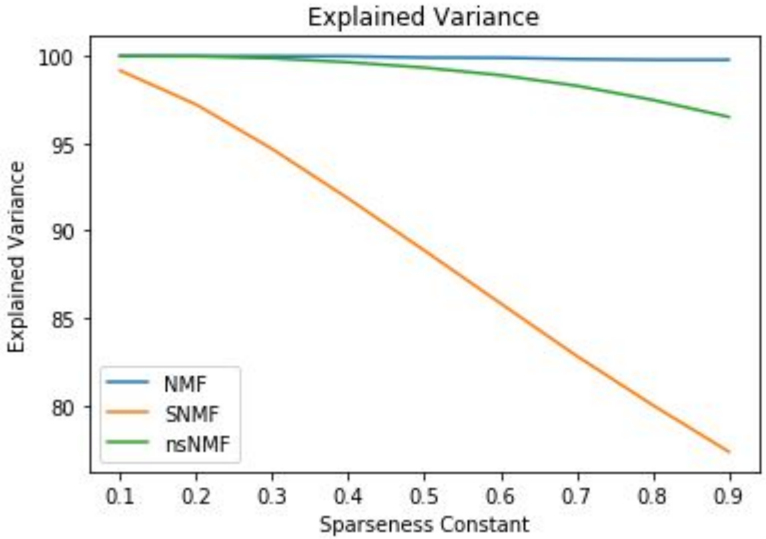
$$\mathrm{Sparseness}(\mathbf{x}) = \frac{\sqrt{n} - \left(\sum |x_j|\right)/\sqrt{\sum x_j^2}}{\sqrt{n-1}}$$

We generated 2 matrices, **A** (5x3) and **B** (3x20) and their product is the matrix we want to factorise,

V = A x B

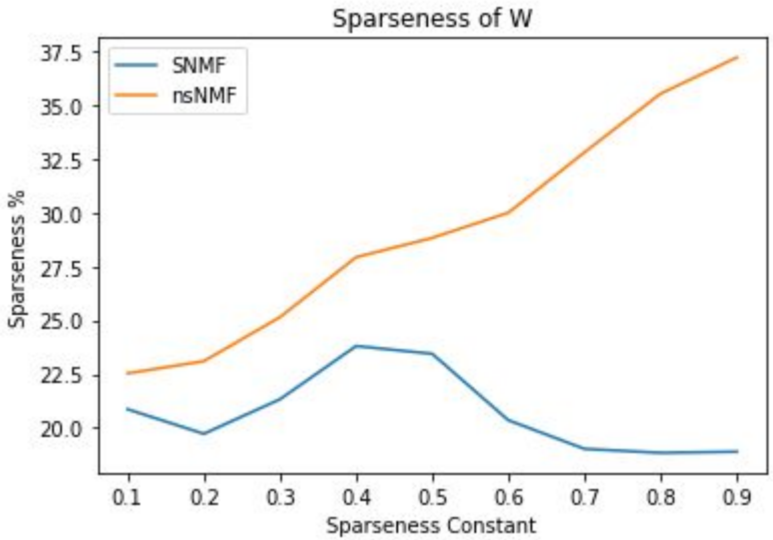
Therefore the number of factors (p) is 5 and the number of objects (n) is 20

Implementation Results



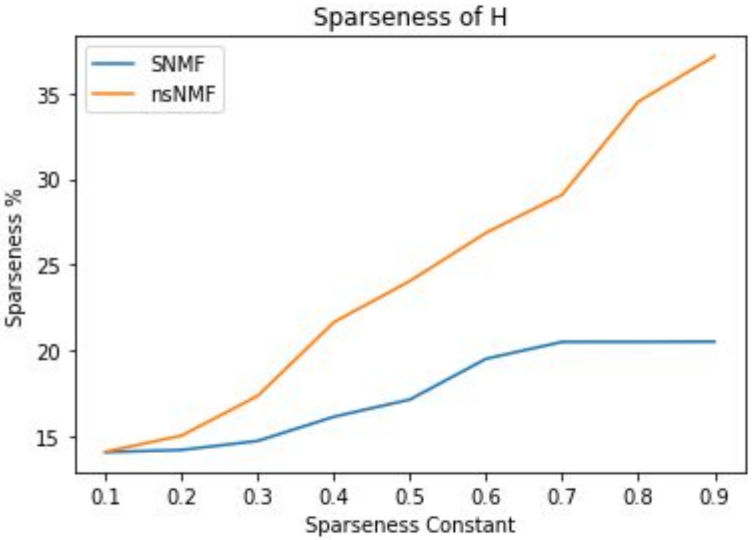
Computed Explained Variance over 3 algorithms (NMF, SNMF and nsNMF) , the results we obtained are as expected. As the sparsity parameter is increased, we see that the explained variance of the NMF results stays the same as it doesn’t take sparsity into account, it also tries to minimize the error, hence it obtains maximum variance, SNMF takes into account sparsity (of H particularly) and the variance it obtains is sacrificed in exchange for more sparsity. In comparison to both, as the sparsity is increased, the explained variance obtained by the nsNMF algorithm isn’t affected as badly, making nsNMF more ideal and resistant to the battle between sparsity and variance conservation

Implementation Results



Here we compared the sparseness of H obtained from both SNMF and nsNMF (we have omitted NMF as it doesn't take sparsity into account), We notice that as we increase the sparseness constant (or parameter) in the case of SNMF the sparsity of H does increase but the sparsity of the corresponding W decreased, whereas in the case of nsNMF both the sparsity of W and H increase, the algorithm claims to improve the sparsity of both on which it does deliver whereas the SNMF algorithm can increase the sparsity of H but while sacrificing the sparsity of W making nsNMF the more desirable of the two

Implementation Results



Here we compared the sparseness of H obtained from both SNMF and nsNMF (we have omitted NMF as it doesn't take sparsity into account), We notice that as we increase the sparseness constant (or parameter) in the case of SNMF the sparsity of H does increase but the sparsity of the corresponding W decreased, whereas in the case of nsNMF both the sparsity of W and H increase, the algorithm claims to improve the sparsity of both on which it does deliver whereas the SNMF algorithm can increase the sparsity of H but while sacrificing the sparsity of W making nsNMF the more desirable of the two

Conclusion

- ★ NMF has effective ability in extracting human intelligible features.
- ★ The approach presented is an attempt to improve the ability of the classical NMF algorithm in this process by producing truly sparse components of the data structure and, at the same time, to identify which of these components are better represented by each individual item.
- ★ The experimental results on both synthetic data and real data sets have shown that the nsNMF algorithm outperformed the existing sparse NMF variants in performing parts-based representation of the data while maintaining the goodness of fit.
- ★ This capability is very useful in real data mining applications where dimensionality reduction can be achieved while the interpretation of the data becomes easier.