# Non-negative Matrix Factorization for Dimensionality Reduction

Jbari Olaya
*New Technology Trends (NTT)*
*National School of Applied Sciences*
Tetuan, Morocco
olaya.jbari@etu.uae.ac.ma

Chakkor Otman
*New Technology Trends (NTT)*
*National School of Applied Sciences*
Tetuan, Morocco
otman.chakkor@uae.ac.ma

***Abstract*—What matrix factorization methods do is reduce the dimensionality of the data without losing any important information. In this work, we present the Non-negative Matrix Factorization (NMF) method, focusing on its advantages concerning other methods of matrix factorization. We discuss the main optimization algorithms, used to solve the NMF problem, and their convergence. The paper also contains a comparative study between principal component analysis (PCA), independent component analysis (ICA), and NMF for dimensionality reduction using a face image database.**

***Index Terms*—NMF, PCA, ICA, dimensionality reduction.**

## I. Introduction

Dimensionality reduction is essential for extracting information from high-dimensional data. For that, PCA and ICA are the famous matrix factorization methods uses for this task. However, for many data sets such as images, text,...etc the original data matrices are non-negative. A factorization such as PCA and ICA contains negative values and is difficult to interpret for some applications. In contrast, non-negative matrix factorization restricts the elements in the data matrix to be non-negative.

The philosophy of NMF was firstly introduced, in a paper published, by Paatero and Tapper in 1994, and popularised by Lee and Seung in 1999 [12]. After that, NMF has gradually become an interesting multidimensional data processing tool to many researchers owing to its ability for giving a natural interpretation of the results (significant results) due to the constraint of the non-negativity.

NMF seeks to find two low-rank matrices $(W, H) \in \mathbb{R}_{m\times r} \times \mathbb{R}_{r\times n}$ non-negative, whose product approximates the non-negative data matrix $X \in \mathbb{R}_{m\times n}$ defined as:

$$X \approx WH \tag{1}$$

Where $r$ is the factorization rank $(r << rank(X) << min(m, n))$ which selects how many features will be extracted from the data.

More precisely, each data point represented as a row in X can be approximated by an additive combination of non-negative vectors, which are represented as row in W (see Fig.1).

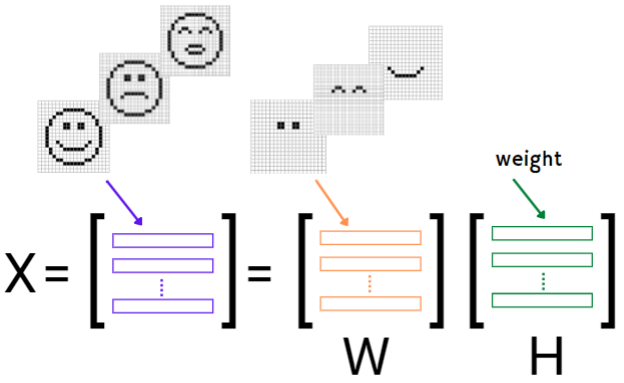


Fig. 1. Non-negative Matrix Factorization: Example of face recognition.

Different to other matrix factorization methods, NMF leads to a part-based representation (i.e, treating objects as a collection of constituent parts), because they allow only additive combinations of the original data. For example, in face recognition, It is interesting to note that the rows of the resulting W are clear parts of human faces (see Fig.1), e.g. nose, ears, and eyes, and these elements will add to each other to recreate the face (original data) [14]. On the other hand, classical factorization methods such as PCA and ICA produces both positive and negative values. Therefore, giving out components that don't offer much interpretability.

Moreover, the remarkable effectiveness of NMF in analyzing nonnegative data has attracted a significant amount of research in many others areas, such as in image processing [6],[19],[4], text mining [3],[21] and source separation problems [15],[7]. Currently, there is ongoing research on NMF to increase its efficiency and robustness.

To this end, the remaining sections of this paper are organized as follows. Section1 discusses the major challenges in solving the NMF problem, as well as the input parameters, including the initialization of the matrix W and H and the factorization rank r. In section2, several optimization algorithms are presented. In the last section, we apply the

three methods, PCA, ICA, and NMF, to a face images data.

## II. Non-negative Matrix Factorization

### A. Problem formulation

The non-negative matrix factorization can be mathematically formulated as a constrained optimization problem below:

$$(P)\begin{cases} \textit{Minimize:} & \boldsymbol{D(X|WH)} \\ \textit{Subject to:} & \boldsymbol{W \geqslant 0} \\ & \boldsymbol{H \geqslant 0} \end{cases}$$

where $\boldsymbol{W, H \geqslant 0}$ means that every element of $\boldsymbol{W}$ and $\boldsymbol{H}$ is nonnegative. $\boldsymbol{D(x|y)}$ is a loss function, which is mostly chosen to be Euclidean distance (Euc) $\frac{1}{2}\boldsymbol{(x-y)^2}$, Kullback Leibler divergence (KL) $\boldsymbol{x log(\frac{x}{y}) - x + y}$ or Itakura-Saito (IS) divergence $\boldsymbol{\frac{x}{y} - log(\frac{x}{y}) - 1}$. The choice of the NMF cost function is made according to the type of data to be analyzed. In this article, the Euclidean distance is selected as the objective function:

$$\boldsymbol{D(X|WH) = ||X - WH||_F^2} \tag{2}$$

$\boldsymbol{D}$ is a non-convex function in the two variables $\boldsymbol{W}$ and $\boldsymbol{H}$. It is, therefore, difficult to find the global minima for $\boldsymbol{(P)}$. Another weakness of the NMF problem is that it is ill-posed (i.e, $\boldsymbol{W}$ and $\boldsymbol{H}$ are non-unique) [17]. For example, given a non-singular matrix $\boldsymbol{A}$, so, $\boldsymbol{\hat{W} = WA \geq 0}$ and $\boldsymbol{\hat{H} = A^{-1}H \geq 0}$. Thereby $\boldsymbol{(\hat{W}, \hat{H})}$ is anther solution pair. To overcome this problem usually, the recherches add prior knowledge of $\boldsymbol{W}$ and $\boldsymbol{H}$, such as sparseness or orthogonality.

As a consequence, the optimization problem (P) cannot be solved directly. In literature, Block Coordinate Descent (BCD) is the basic framework for all NMF algorithms, based on the idea that the minimization of the cost function can be achieved by minimizing it in one direction at a time (fixes, for example, W and varies H). In this case, the optimization problem becomes convex. To sum up, we seek the solution of the sub-problems (3)(4), in order to determine the solution of the whole problem.

$$\min_{\boldsymbol{H}} \boldsymbol{||X - WH||_F^2} \quad \text{s.t} \quad \boldsymbol{H \geq 0} \tag{3}$$

$$\min_{\boldsymbol{W}} \boldsymbol{||X - WH||_F^2} \quad \text{s.t} \quad \boldsymbol{W \geq 0} \tag{4}$$

So, all NMF algorithms are solving (P) iteratively, and if they converge, at that time to local minima.

Initialization of W and H is needed, and a good initialization leads to an efficient local solution as illustrated in Fig.2. Due to the sensitivity of this step, researchers have used a variety of initialization methods [22][23][18], and newly, Flavia Esposito [5] has provided a taxonomy of initialization schemes appearing in the literature.

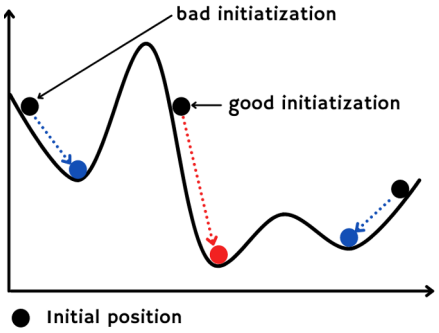


Fig. 2. NMF: bad and good initialisation: case of multiple local optima.

Furthermore, the factorization rank r is another important issue that needs to be fixed. A small value of r may lead to loss of features, and a large value of r, may be responsible for modeling noise. So, the rank r should both reduce the noise in the data and effectively model the key features. The choice of r is generally based on experiments or experience. However, several rank selection techniques have been recently proposed such as [9], [20],[16], [24].

### B. Algorithms

*1) Projected Gradient Descent:* Gradient descent (GD) is a popular optimization algorithm used for solving NMF problem. GD find the minimum of a convex function more quickly by descending, at the current point, in the opposite direction of the gradient of the function.

In our case, the GD is used to solve each sub-problem. We consider the sub-problem (3), the steps of the GD algorithm are as the following:

1) Pick an initial point $\boldsymbol{H_0}$
2) Loop until stopping condition is met:
   a) Descent direction: pick the descent direction as $\boldsymbol{\nabla_H D(X, WH)}$
   b) Step size: pick a step size $\boldsymbol{\eta_H}$
   c) Update: $\boldsymbol{H \leftarrow H - \eta_H \circ \nabla_H D(X, WH)}$

With $\boldsymbol{\nabla D}$ is the gradient of the cost function $\boldsymbol{D}$. The GD update of W is similar to that of H.

The idea of the Projected Gradient Descent (PGD) is simple: if the point $\boldsymbol{H - \eta_H \circ \nabla_H D(X, WH)}$ after the gradient update is leaving the nonnegative space, project it back. So, PGD has one more step which is the projection: $\boldsymbol{H \leftarrow arg\min_{H \geq 0} ||X - H||_F^2}$.

*2) Multiplicative Update:* Multiplicative update rules (MUR) is the common NMF algorithm and the most widely used due to their simplicity. Multiplicative methods can be obtained in different ways, either by a heuristic approach, or a Majoration-Minimization (MM) approach. We present them successively below.

- **Heuristic update**

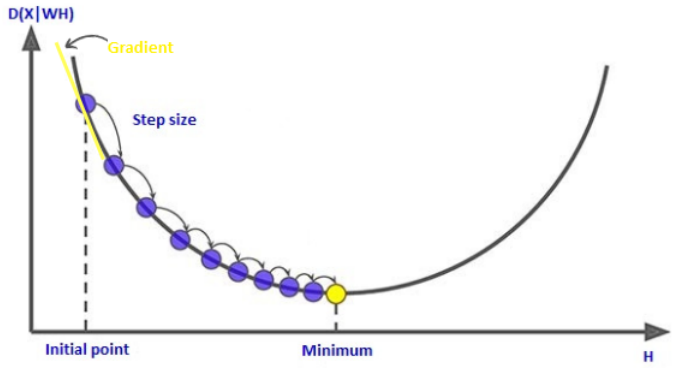


Fig. 3. Gradient Descend Procedure.

Lee and seung (1999) [12] were apparently the first to give the heuristic MU, based on the traditional GD algorithm (presented in the previous section) with an adaptive learning rate (step size). To avoid subtraction in the GD update, in [11] proposed to set the learning rate as:

$$\eta_H = \frac{H}{W^T W H} \qquad \eta_W = \frac{W}{W H H^T}$$

$W$ and $H$ have been derived [1] to minimize the factorization error between $X$ and $WH$. The basic update rules are as follows:

$$H \leftarrow H \circ \frac{W^T X}{W^T W H}$$
$$W \leftarrow W \circ \frac{X H^T}{W H H^T}$$

With $\circ$ (resp. $\frac{\cdot}{\cdot}$ ) denotes elementwise multiplication (resp.division).

- **Majorization-Minimization update**

Geting the MU by Majorization-Minimization (MM) framework means to replace the optimization problem (3 or 4) with a sequence of simpler optimization problems. In other terms, finding a surrogate upper bound of the cost function at each iteration and minimize.
Let

$$C(\theta) = D(X|WH), \;\; \theta = \{W, H\}$$

thus

$$\min_{H \geqslant 0} D(X|WH) = \min_{\theta} C(\theta) \qquad (5)$$

The MM algorithm solving (5) by two main steps:

– $1^{st}$ **step** : Construct a surrogate [2] function $G_k(\theta)$ of $C(\theta)$ at the current iterate $\theta_k$.

– $2^{nd}$ **step** : Minimize the surrogate to get the next iterate: $\theta_{k+1} = arg\min_\theta G_k(\theta)$.

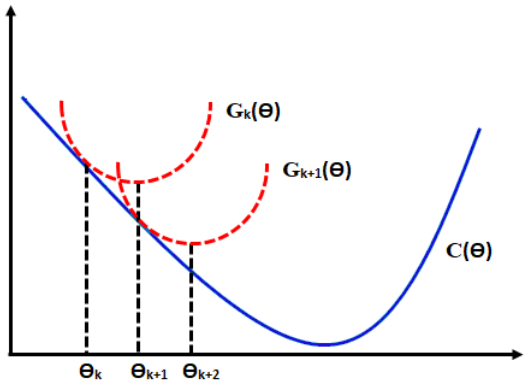


Fig. 4. Majorisation-Minimisation (MM) procedure.

The MM procedure guarantees the cost is non-increasing (monotonic descent property) at each iteration:

$$C(\theta_{k+1}) \leqslant G_k(\theta_{k+1}) \leqslant G_k(\theta_k) \leqslant C(\theta_k) \qquad (6)$$

We can also observe this property from figure 4.

Building a surrogate function is a very important step for the MM algorithm, and it's not easy. However, several inequalities used in literature which helps in finding this function, including Jensen's Inequality, Convexity Inequality and Cauchy-Schwarz Inequality. In our case where,

$$D(X|WH) = ||X - WH||_F^2 = \sum_i \sum_j (x_{ij} - \sum_k w_{ik} h_{kj})^2$$

$w_{ik}, h_{kj} \geq 0$

The construction of the surrogate function is done by the convexity of the function $(x_{ij} - x)^2$

$$\left(x_{ij} - \sum_k w_{ik} h_{kj}\right)^2 \leq \sum_k \frac{a_{ikj}^{(t)}}{b_{ij}^{(t)}} \left(x_{ij} - \frac{b_{ij}^{(t)}}{a_{ikj}^{(t)}} w_{ik} h_{kj}\right)^2$$

Where

$$a_{ikj}^{(t)} = w_{ik}^{(t)} h_{kj}^{(t)}, \quad b_{ij}^{(t)} = \sum_k w_{ik}^{(t)} h_{kj}^{(t)}$$

This suggests the alternating multiplicative updates

$$w_{ik}^{(t+1)} = w_{ik}^{(t)} \frac{\sum_j x_{ij} h_{kj}^{(t)}}{\sum_j b_{ij}^{(t)} h_{kj}^{(t)}}$$

$$b_{ij}^{(t+\frac{1}{2})} = \sum_k w_{ik}^{(t+1)} h_{kj}^{(t)}$$

$$h_{kj}^{(t+1)} = h_{kj}^{(t)} \frac{\sum_i x_{ij} w_{ik}^{(t+1)}}{\sum_i b_{ij}^{(t+\frac{1}{2})} w_{ik}^{(t+1)}}$$

The update seems simpler in the equivalent matrix form,

[1] More detail on the MU derivation is given by [2].
[2] A surrogate is a function that approximates another function, it has to verify $G_k(\theta) \geqslant C(\theta)$ and $G_k(\theta_k) = C(\theta_k)$.

$$H \leftarrow H \circ \frac{W^T X}{W^T W H}$$

$$W \leftarrow W \circ \frac{X H^T}{W H H^T}$$

We notice that the MM update coincides with the heuristic update.

The main difference between the MU and PGD algorithms is that, the learning rate of PGD is flexible and that of MU is fixed. Accordingly, MU is slower in convergence than PGD.

*3) Alternating Least Squares:* The Alternating Least Squares (ALS) algorithm solves the problem (P) iteratively without non-negativity constraint and then projects the solution into a non-negative space. So The update rules are as follows:

$$H \leftarrow max(arg\min_{H} ||X - WH||_F^2, 0)$$

$$W \leftarrow max(arg\min_{W} ||X - WH||_F^2, 0)$$

Generally the ALS algorithm suffers from lack of convergence.

*4) Alternating Nonnegative Least Squares:* Unlike ALS, the Alternating Nonnegative Least Squares (ALNS) impose the nonnegativity constraints in the sub-problem (3),(4), and find their optimal solution .i.e:

$$H \leftarrow arg\min_{H \geq 0} ||X - WH||_F^2$$

$$W \leftarrow arg\min_{W \geq 0} ||X - WH||_F^2$$

Each step of the update is dedicated to finding the optimal solution to the sub-problem, there are many methods used to solve this bounded constraint problem including PGD [13], Active set [10], Projected quasi-Newton [1], and Projected Barzilai–Borwein [8]. Compared with the MU algorithm, ANLS has relatively faster convergence than MU algorithms.

## III. Method

Comparisons between the three matrix factorization methods (i.e. PCA, ICA, and NMF) were made using a face image database. Before applying such a method, the data were converted into greyscale and centered (Fig.5). Also, the law rank (number of components) in the three methods is fixed as r=3. *randomized SVD*, *FastICA*, and *MU* are the used algorithms for solving PCA, ICA, and NMF respectively. Also, we use nndsvd (nonnegative double singular value decomposition) as initialization method for NMF. The source code is available online[3].

[3]https://scikit-learn.org/stable/auto_examples/decomposition/plot_faces_decomposition.html

Fig. 5. First three centered FEI faces.

Fig. 6. The first three components given by PCA.

Fig. 7. The first three components given by ICA.

Fig. 8. The first three components given by NMF.

In this experiment, we have used images of a FEI face database maintained by the Department of Electrical Engineering of FEI[4] , São Paulo, Brazil. The dataset contains 200 face images (100 men and 100 women), where the size of each image is 360 $\times$ 260 pixels.

## IV. Result and discussion

The first three components given by PCA, ICA, and NMF are shown in the Fig.6, Fig.7 and Fig.8 respectively. From these figures, we see that PCA was unable to reconstruct the faces because of the noisy components. ICA lost some important features in the faces and the images seem unclear. However, the images extracted from NMF represent more caractéristiques and details of the face. So, the faces are better reconstructed than PCA and ICA.

Moreover, PCA needs 2.018s for training to give the extracted components and ICA 4.032s. In contrast, NMF takes more time which is 10.609s.

## V. Conclusion

In this work, we describe the NMF problem and its famous algorithms such as MU, PGD, and ANLS. We also compared

[4]This database is available on https://fei.edu.br/∼cet/facedatabase.html.

NMF with PCA and ICA methods using a face image database. PCA and ICA show a significant loss of face information. On the other hand, NMF was able to extract face features and retain more information after reducing the dimensionality.

## References

[1] Pierre Ablin, Dylan Fagot, Herwig Wendt, Alexandre Gramfort, and Cédric Févotte. A quasi-newton algorithm on the orthogonal manifold for nmf with transform learning. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 700–704. IEEE, 2019.

[2] Juan José Burred. Detailed derivation of multiplicative update rules for nmf. *Paris, France*, 2014.

[3] Yong Chen, Hui Zhang, Rui Liu, Zhiwen Ye, and Jianying Lin. Experimental explorations on short text topic mining between lda and nmf based schemes. *Knowledge-Based Systems*, 163:1–13, 2019.

[4] Chen Cui, Xujun Wu, Jun Yang, and Juyan Li. A novel dibr 3d image hashing scheme based on pixel grouping and nmf. *Wireless Communications and Mobile Computing*, 2020, 2020.

[5] Flavia Esposito. A review on initialization methods for nonnegative matrix factorization: Towards omics data experiments. *Mathematics*, 9(9):1006, 2021.

[6] Yixian Fang, Huaxiang Zhang, and Yuwei Ren. Graph regularised sparse nmf factorisation for imagery denoising. *IET Computer Vision*, 12(4):466–475, 2018.

[7] Cédric Févotte, Emmanuel Vincent, and Alexey Ozerov. Single-channel audio source separation with nmf: divergences, constraints and algorithms. *Audio Source Separation*, pages 1–24, 2018.

[8] Yakui Huang, Hongwei Liu, and Shuisheng Zhou. Quadratic regularization projected barzilai–borwein method for nonnegative matrix factorization. *Data mining and knowledge discovery*, 29(6):1665–1684, 2015.

[9] Yu Ito, Shin-ichi Oeda, and Kenji Yamanishi. Rank selection for non-negative matrix factorization with normalized maximum likelihood coding. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, pages 720–728. SIAM, 2016.

[10] Hyunsoo Kim and Haesun Park. Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method. *SIAM journal on matrix analysis and applications*, 30(2):713–730, 2008.

[11] Daniel Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, 13, 2000.

[12] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[13] Chih-Jen Lin. Projected gradient methods for nonnegative matrix factorization. *Neural computation*, 19(10):2756–2779, 2007.

[14] Xianzhong Long, Hongtao Lu, Yong Peng, and Wenbin Li. Graph regularized discriminative non-negative matrix factorization for face recognition. *Multimedia tools and applications*, 72(3):2679–2699, 2014.

[15] Andri Mirzal. Nmf versus ica for blind source separation. *Advances in Data Analysis and Classification*, 11(1):25–48, 2017.

[16] Laura Muzzarelli, Susanne Weis, Simon B Eickhoff, and Kaustubh R Patil. Rank selection in non-negative matrix factorization: systematic comparison and a new mad metric. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

[17] Weiwei Pan and Finale Doshi-Velez. A characterization of the non-uniqueness of nonnegative matrix factorizations. *arXiv preprint arXiv:1604.00653*, 2016.

[18] Hanli Qiao. New svd based initialization strategy for non-negative matrix factorization. *Pattern Recognition Letters*, 63:71–77, 2015.

[19] Yaser Esmaeili Salehani, Ehsan Arabnejad, Abderrahmane Rahiche, Athmane Bakhta, and Mohamed Cheriet. Msdb-nmf: Multispectral document image binarization framework via non-negative matrix factorization approach. *IEEE Transactions on Image Processing*, 29:9099–9112, 2020.

[20] Steven Squires, Adam Prügel-Bennett, and Mahesan Niranjan. Rank selection in nonnegative matrix factorization using minimum description length. *Neural computation*, 29(8):2164–2176, 2017.

[21] Ding Tu, Ling Chen, Mingqi Lv, Hongyu Shi, and Gencai Chen. Hierarchical online nmf for detecting and tracking topic hierarchies in a text stream. *Pattern Recognition*, 76:203–214, 2018.

[22] Stefan Wild, James Curry, and Anne Dougherty. Improving non-negative matrix factorizations through structured initialization. *Pattern recognition*, 37(11):2217–2232, 2004.

[23] Yun Xue, Chong Sze Tong, Ying Chen, and Wen-Sheng Chen. Clustering-based initialization for non-negative matrix factorization. *Applied Mathematics and Computation*, 205(2):525–536, 2008.

[24] Xiaohui Yang, Wenming Wu, Xin Xin, Limin Su, and Liugen Xue. Adaptive factorization rank selection-based nmf and its application in tumor recognition. *International Journal of Machine Learning and Cybernetics*, pages 1–19, 2021.