Matrix/Tensor Factorization

Madhav Datt
Indian Institute of Technology, Kharagpur
madhav@iitkgp.ac.in

Abstract

Matrix and tensor factorization is a group of algorithms and techniques in multivariate analysis. We especially explore non-negative matrix and tensor factorization (NMF and NTF) which, even though is a significantly more challenging problem, has greater applicability. The nonnegativity constraints have been shown to enable natural interpretations and allow better solutions in numerous applications including image processing, text analysis, and bioinformatics. However, the computation of NMF and NTF is a NP-hard problem in general, and remains challenging and expensive due the constraints. Numerous algorithmic approaches have been proposed to efficiently compute NMF and NTF. In this extremely brief survey paper, we start by defining the NMF problem, along with discussing classical approaches, such as alternating least squares, for solution estimation, both from an intuitive and a rigorous mathematical perspective. We survey some of the state of the art numerical methods and algorithms for NMF. Further, we compare these methods and examine their advantages and disadvantages in terms of convergence to optimum values, loss minimization, algorithmic efficiency etc. We also look at modifications to enable these algorithms to handle very large-scale data. Finally, we briefly describe some problems and applications in image processing where non-negative matrix and tensor factorization finds very direct use.

1 Introduction

In this brief and introductory survey, we explore Matrix and Tensor Factorization, which is a group of algorithms in multivariate analysis and linear algebra where a matrix $\mathbf{V}$ is, usually, though not always, factorized into two matrices $\mathbf{W}$ and $\mathbf{H}$. In text mining, signal processing, image processing, and computer vision among other areas, imposing non-negativity constraints to the low-rank factors of matrices (ie. $w \geq 0, h \geq 0 \ \forall \ w \in \mathbf{W}, h \in \mathbf{H}$) and tensors has been shown an effective technique providing physically meaningful interpretations of the signal or image data involved, and is inherent to the data. For example, in image processing and computer vision, involved variables and parameters may correspond to pixels, and nonnegative sparse decomposition is related to the extraction of relevant parts from the images as shown in works (Lee, Daniel D., and H. Sebastian Seung, 1999), (Lee, Daniel D., and H. Sebastian Seung, 2001). Since this problem is not exactly solvable in general, it is commonly approximated through various numerical methods and algorithms.

In the rest of this paper, we focus on discussions and analysis state-of-the-art algorithms for non-negative matrix or tensor factorization (NMF) (Paatero, Pentti, and Unto Tapper, 1994), and explore some of its applications to image processing.

1.1 Basic NMF Model

The basic NMF problem can formally be defined and stated as follows: Given a nonnegative data matrix $\mathbf{Y} \in \mathbb{R}_+^{I \times T}$ (with $y_{it} \geq 0$ or equivalently $\mathbf{Y} \geq 0$) and a reduced rank J ($J \leq \min(I, T)$), find two nonnegative matrices $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_J] \in \mathbb{R}_+^{I \times J}$ and $X = B^T = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_J]^T \in \mathbb{R}_+^{J \times T}$ which factorize $\mathbf{Y}$ as well as possible, that is

$$\mathbf{Y} = \mathbf{A}\mathbf{X} + \mathbf{E} = \mathbf{A}\mathbf{B}^T + \mathbf{E}$$

where the matrix $\mathbf{E} \in \mathbb{R}^{I \times T}$ represents approximation error. The factors $\mathbf{A}$ and $\mathbf{X}$ may have different physical meanings in different applications, and are assumed to be non-negative in standard NMF.

The NMF model can also be represented as a special form of the bilinear model, as follows:

$$\mathbf{Y} = \sum_{j=1}^{J} \mathbf{a}_j \circ \mathbf{b}_j + \mathbf{E} = \sum_{j=1}^{J} \mathbf{a}_j \mathbf{b}_j^T + \mathbf{E}$$

where the symbol $\circ$ denotes the outer product of two vectors. Thus, we can build an approximate representation of the nonnegative data matrix $\mathbf{Y}$ as a sum of rank-one nonnegative matrices $\mathbf{a}_j \mathbf{b}_j^T$. If such decomposition is exact (i.e., $\mathbf{E} = 0$) then it is called the Nonnegative Rank Factorization (NRF) (Dong, Bo, Matthew M. Lin, and Moody T. Chu, 2008). Among the many possible series representations of data matrix $\mathbf{Y}$ by nonnegative rank-one matrices, the smallest integer J for which such a nonnegative rank-one series representation is attained is called the nonnegative rank of the nonnegative matrix $\mathbf{Y}$ and it is denoted by $rank_+(\mathbf{Y})$. The nonnegative rank satisfies the following bounds, as shown by (Dong, Bo, Matthew M. Lin, and Moody T. Chu, 2008):

$$rank(\mathbf{Y}) \leq rank_+(\mathbf{Y}) \leq \min(I, T)$$

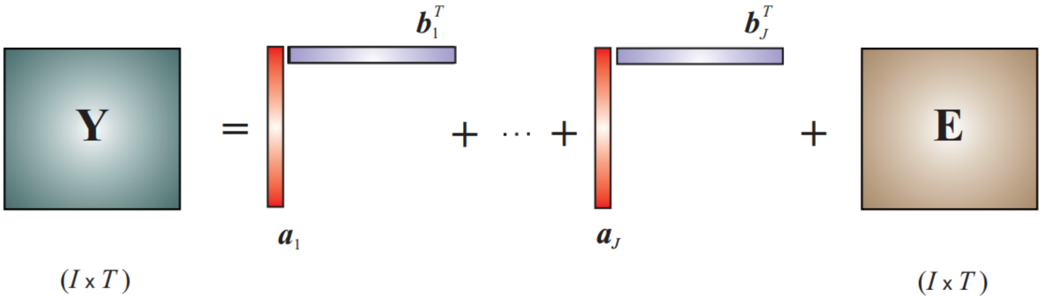

Figure 1: Bilinear NMF model. The nonnegative data matrix $\mathbf{Y} \in \mathbb{R}_+^{I \times T}$ is approximately represented by a sum or linear combination of rank-one nonnegative matrices $\mathbf{Y}^{(j)} = \mathbf{a}_j \circ \mathbf{b}_j = \mathbf{a}_j \mathbf{b}_j^T \in \mathbb{R}_+^{I \times T}$.

An NMF is not necessarily an NRF in the sense that the latter demands the exact factorization whereas the former is usually only an approximation. In many applications we require additional constraints on the elements of matrices $\mathbf{A}$ and/or $\mathbf{X}$, such as smoothness, sparsity, symmetry, and orthogonality, the discussion of which is beyond the scope of this paper.

1.2 Basic Approaches to Parameter Estimation

In order to estimate factor matrices $\mathbf{A}$ and $\mathbf{X}$ in the standard NMF, we need to consider the similarity measure to quantify a difference between the data matrix $\mathbf{Y}$ and the approximative NMF model matrix $\hat{\mathbf{Y}} = \mathbf{AX}$. The choice of the similarity measure (also referred to as distance, divergence or measure of dissimilarity) mostly depends on the probability distribution of the estimated signals or components and on the structure of data or a distribution of noise. The simplest and most often used measure is based on Frobenius norm:

$$D_F(\mathbf{Y}||\hat{\mathbf{Y}}) = \frac{1}{2}||\mathbf{Y} - \mathbf{AX}||_F^2$$

which is also referred to as the squared Euclidean distance. It should be noted that the above cost function is convex with respect to either the elements of the matrix $\mathbf{A}$ or the matrix $\mathbf{X}$, but not both. Another frequently used cost function for NMF is the generalized Kullback-Leibler divergence, as shown in (Lee, Daniel D., and H. Sebastian Seung, 2001), (also called the

Figures 1 and 2 are based on explanations and figures provided in (Cichocki, Andrzej, et al, 2009).

I-divergence):

$$D_{KL}(\mathbf{Y}||\hat{\mathbf{Y}}) = \sum_{it}\left(y_{it}\ln\frac{y_{it}}{[\mathbf{AX}]_{it}} - y_{it} + [\mathbf{AX}]_{it}\right)$$

Alternating minimization of such a cost leads to the ALS (Alternating Least Squares) algorithm which is a fixed point algorithm, where each iteration can be described as follows:

1. Initialize $\mathbf{A}$ randomly or by using a specific deterministic strategy.
2. Estimate $\mathbf{X}$ from the matrix equation $\mathbf{A}^T\mathbf{AX} = \mathbf{A}^T\mathbf{Y}$ by solving

$$\min_X D_F(\mathbf{Y}||\mathbf{AX}) = \frac{1}{2}||\mathbf{Y} - \mathbf{AX}||_F^2 \text{ with fixed } \mathbf{A}.$$

3. Set all the negative elements of $\mathbf{X}$ to zero or some small positive value ϵ.
4. Estimate $\mathbf{A}$ from the matrix equation $\mathbf{XX}^T\mathbf{A}^T = \mathbf{XY}^T$ by solving

$$\min_A D_F(\mathbf{Y}||\mathbf{AX}) = \frac{1}{2}||\mathbf{Y}^T - \mathbf{X}^T\mathbf{A}^T||_F^2 \text{ with fixed } \mathbf{X}.$$

5. Set all the negative elements of $\mathbf{A}$ to zero or some small positive value ϵ.

It is possible to rewrite the above algorithm into a more concise form as the following iterative, fixed point algorithm update statements:

$$\mathbf{X} \leftarrow \max(\epsilon, (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{Y}) = [\mathbf{A}^{\dagger}\mathbf{Y}]_+ \tag{1}$$

$$\mathbf{A} \leftarrow \max(\epsilon, \mathbf{YX}^T(\mathbf{XX}^T)^{-1}) = [\mathbf{YX}^{\dagger}]_+ \tag{2}$$

where, the † symbol denotes the Moore-Penrose inverse of the matrix, and ϵ is a very small constant (typically in the order of 10^{-16}) to enforce positive entries in the resulting matrices.

The above described ALS method is considered as a basic ”workhorse” approach, however it is not guaranteed to converge to a global minimum nor even a stationary point, but only to a solution where the cost functions cease to decrease ((Kolda, Tamara G., and Brett W. Bader, 2009), (Berry, Michael W., M. Browne, A. Langville, P. Pauca, and R. Plemmons, 2007)). Moreover, it is often not sufficiently accurate. The ALS method can be dramatically improved and its computational complexity reduced with algorithms and methods that are discussed in the next section on State of the art Algorithms.

2 State of the art Algorithms

In this section, we discuss some of the most effective state-of-the-art algorithms and numerical methods for matrix and tensor factorization. Further, we examine and compare all the discussed algorithms based on their advantages and disadvantages. Before we go into the discussion on NMF algorithms, we provide a brief and intuitive note on large scale NMF methods, which are particularly relevant to image processing, and how these can be applied in case of the algorithms discussed below.

2.1 Large-Scale NMF

In many image processing applications, especially in dimension reduction applications the data matrix $\mathbf{Y} \in \mathbb{R}^{I\times T}$ can be very large (with millions of entries), but it can be approximately factorized using a rather smaller number of nonnegative components (J), that is, $J << I$ and $J << T$. Then the problem $\mathbf{Y} \approx \mathbf{AX}$ becomes highly redundant and we do not need to use information about all entries of $\mathbf{Y}$ in order to estimate precisely the factor matrices $\mathbf{A} \in \mathbb{R}^{I\times J}$ and $\mathbf{X} \in \mathbb{R}^{J\times T}$. In other words, to solve the large-scale NMF problem we do not need to know the whole data matrix but only a small random part of it. In this approach, instead of performing large-scale factorization

$$\mathbf{Y} = \mathbf{AX} + \mathbf{E}$$

we can consider a two set of linked factorizations using much smaller matrices, given by

$$\begin{aligned} \mathbf{Y}_r &= \mathbf{A}_r \mathbf{X} + \mathbf{E}_r, &\quad \text{for fixed and known } \mathbf{A}_r \\ \mathbf{Y}_c &= \mathbf{A} \mathbf{X}_c + \mathbf{E}_c, &\quad \text{for fixed and known } \mathbf{X}_c \end{aligned}$$

where $\mathbf{Y}_r \in \mathbb{R}_+^{R \times T}$ and $\mathbf{Y}_c \in \mathbb{R}_+^{I \times C}$ are the matrices constructed from the selected rows and columns of the matrix $\mathbf{Y}$, respectively. Analogously, we can construct the reduced matrices: $\mathbf{A}_r \in \mathbb{R}^{R \times J}$ and $\mathbf{X}_c \in \mathbb{R}^{J \times C}$ by using the same indices for the columns and rows as those used for the construction of the data sub-matrices $\mathbf{Y}_c$ and $\mathbf{Y}_r$. (Hasan, Milos, Fabio Pellacini, and Kavita Bala, 2007), (Boutsidis, Christos, Michael W. Mahoney, and Petros Drineas, 2009) discuss several strategies to choose the columns and rows of the input data matrix, where the simplest scenario is to choose the first R rows and the first C columns of the data matrix as shown in Figure 2.

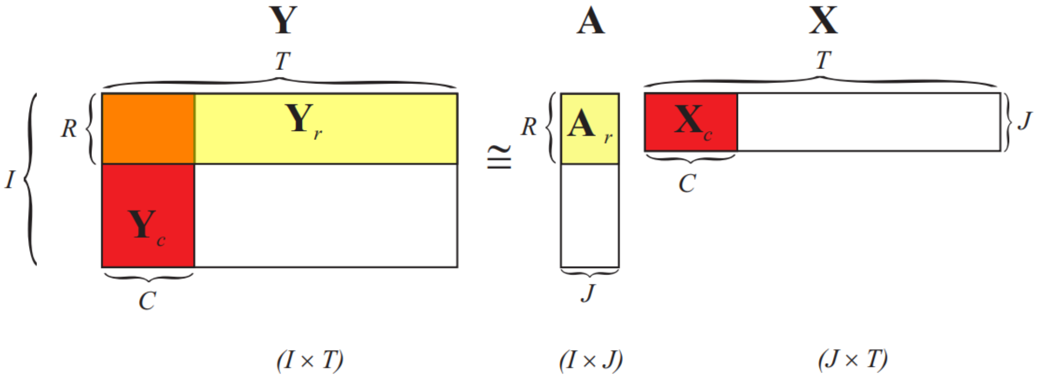

Figure 2: Conceptual illustration of block-wise data processing for large-scale NMF. Instead of processing the whole matrix $\mathbf{Y} \in \mathbb{R}^{I \times T}$, we can process much smaller block matrices $\mathbf{Y}_r \in \mathbb{R}^{R \times T}$ and $\mathbf{Y}_c \in \mathbb{R}^{I \times C}$ and corresponding factor matrices $\mathbf{A}_r \in \mathbb{R}^{R \times J}$ and $\mathbf{X}_c \in \mathbb{R}^{J \times C}$ with $C << T$ and $R << I$.

2.2 Two-Block Coordinate Descent Framework

Almost all NMF algorithms designed, including the "workhorse" alternating least squares algorithm discussed in Section 1.2 use a two-block coordinate descent scheme, that is, they optimize alternatively over one of the two factors, **W** or **H**, while keeping the other fixed. The reason is that the subproblem in one factor is convex. More precisely, it is a nonnegative least squares problem (NNLS): for example, for **H** fixed, we have to solve

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

Note that this problem has a particular structure as it can be decomposed into p independent NNLS in r variables since

$$||\mathbf{X} - \mathbf{WH}||_F^2 = \sum_{i=1}^{p} ||\mathbf{X}_{i:} - \mathbf{W}_{i:}\mathbf{H}||_2^2 = \sum_{i=1}^{p} \mathbf{W}_{i:} \left(\mathbf{H}\mathbf{H}^T\right) \mathbf{W}_{i:}^T - 2\mathbf{W}_{i:} \left(\mathbf{H}\mathbf{X}_{i:}^T\right) + ||\mathbf{X}_{i:}||_2^2. \quad (3)$$

Many algorithms exist to solve the NNLS problem, and NMF algorithms based on two-block coordinate descent differ by which NNLS algorithm is used. A comparative study based on the advantages and disadvantages of these NNLS algorithms is provided in section 2.7. The general framework for these NMF algorithms to update **W** and **H** is given by the following algorithms.

Algorithm BCD Two-Block Coordinate Descent – Framework of Most NMF Algorithms

```
Require: Input nonnegative matrix $\mathbf{X} \in \mathbb{R}_+^{p\times n}$ and factorization rank $r$.
Ensure: $(\mathbf{W}, \mathbf{H}) \geq 0$: A rank-$r$ NMF of $\mathbf{X} \approx \mathbf{WH}$.
Generate some initial matrices $\mathbf{W}^{(0)} \geq 0$ and $\mathbf{H}^{(0)} \geq 0$;
for $t = 1, 2, \ldots$ do
    $\mathbf{W}^{(t)} = \text{update}\big(\mathbf{X}, \mathbf{H}^{(t-1)}, \mathbf{W}^{(t-1)}\big)$.
    ${\mathbf{H}^{(t)}}^T = \text{update}\Big(\mathbf{X}^T, {\mathbf{W}^{(t)}}^T, {\mathbf{H}^{(t-1)}}^T\Big)$.
end for
```

2.3 First-Order Optimality Conditions (Stationary Points)

Given $\mathbf{X}$, let us denote $F(\mathbf{W}, \mathbf{H}) = \frac{1}{2}||\mathbf{X} - \mathbf{WH}||_F^2$. The first-order optimality conditions for equations 1, 2 are

$$\begin{array}{lll} \mathbf{W} \geq 0, & \nabla_{\mathbf{W}} F = \mathbf{WHH}^T - \mathbf{XH}^T \geq 0, & \mathbf{W} \circ \nabla_{\mathbf{W}} F = 0, \\ \mathbf{H} \geq 0, & \nabla_{\mathbf{H}} F = \mathbf{W}^T\mathbf{WH} - \mathbf{W}^T\mathbf{X} \geq 0, & \mathbf{H} \circ \nabla_{\mathbf{H}} F = 0, \end{array} \tag{4}$$

where $\circ$ is the component-wise product of two matrices. Any (W, H) satisfying these conditions is a stationary point of equations 1 and 2 shown in section 1.2.

2.4 Multiplicative Updates

Given $\mathbf{X}$, $\mathbf{W}$ and $\mathbf{H}$, the multiplicative updates (MU) modify $\mathbf{W}$ as follows

$$\mathbf{W} \leftarrow \mathbf{W} \circ \frac{[\mathbf{XH}^T]}{[\mathbf{WHH}^T]} \tag{MU}$$

where $\frac{[\,]}{[\,]}$ denotes the component-wise division between two matrices. The MU were first developed in (Daube-Witherspoon, Margaret E., and Gerd Muehllehner, 1986) for solving NNLS problems, and later rediscovered and used for NMF in (Lee, Daniel D., and H. Sebastian Seung, 2001). The MU are based on the majorization-minimization framework. In fact, (MU) is the global minimizer of a quadratic function majorizing F, that is, a function that is larger than F everywhere and is equal to F at the current iterate (Daube-Witherspoon, Margaret E., and Gerd Muehllehner, 1986) (Lee, Daniel D., and H. Sebastian Seung, 2001). Hence minimizing that function guarantees F to decrease and therefore leads to an algorithm for which F monotonically decreases. The MU can also be interpreted as a rescaled gradient method: in fact,

$$\mathbf{W} \circ \frac{[\mathbf{XH}^T]}{[\mathbf{WHH}^T]} = \mathbf{W} - \frac{[\mathbf{W}]}{[\mathbf{WHH}^T]} \circ \nabla_{\mathbf{W}} F. \tag{5}$$

Another more intuitive interpretation is as follows: we have that

$$\frac{[\mathbf{XH}^T]_{ik}}{[\mathbf{WHH}^T]_{ik}} \geq 1 \quad \iff \quad (\nabla_{\mathbf{W}} F)_{ik} \leq 0.$$

Therefore, in order to satisfy (4), for each entry of $\mathbf{W}$, the MU either (i) increase it if its partial derivative is negative, (ii) decrease it if its partial derivative is positive, or (iii) leave it unchanged if its partial derivative is equal to zero.

If an entry of $\mathbf{W}$ is equal to zero, the MU cannot modify it hence it may occur that an entry of $\mathbf{W}$ is equal to zero while its partial derivative is negative which would not satisfy (4).

2.5 Alternating Nonnegative Least Squares

Alternating nonnegative least squares (ANLS) is a class of methods where the subproblems in $\mathbf{W}$ and $\mathbf{H}$ are solved exactly, that is, the update for $\mathbf{W}$ is given by

$$\mathbf{W} \leftarrow \text{argmin}_{\mathbf{W}\geq 0}||\mathbf{X} - \mathbf{WH}||_F.$$

Many methods can be used to solve the NNLS $\text{argmin}_{\mathbf{W}\geq 0}||\mathbf{X}-\mathbf{WH}||_F$, and dedicated active-set methods have shown to perform very well in practice. Since each iteration of ANLS computes an optimal solution of the NNLS subproblem, each iteration of ANLS decreases the error the most among NMF algorithms following the framework described in Algorithm BCD. However, each iteration is computationally more expensive, and more difficult to implement.

2.6 Hierarchical Alternating Least Squares

Hierarchical alternating least squares (HALS) solves the NNLS subproblem using an exact coordinate descent method, updating one column of W at a time. The optimal solutions of the corresponding subproblems can be written in closed form. In fact, the entries of a column of W do not interact - see Equation (3) - hence the corresponding problem can be decoupled into p quadratic problems with a single nonnegative variable. HALS updates W as follows. For $\ell = 1, 2, \ldots, r$:

$$\begin{aligned}
\mathbf{W}(:,\ell) \quad &\leftarrow \text{argmin}_{\mathbf{W}(:,\ell)\geq 0}\left\|\mathbf{X}-\textstyle\sum_{k\neq\ell}\mathbf{W}(:,k)\mathbf{H}(k,:)-\mathbf{W}(:,\ell)\mathbf{H}(\ell,:)\right\|_F \\
\mathbf{W}(:,\ell) \quad &\leftarrow \max\left(0, \tfrac{\mathbf{X}\mathbf{H}(\ell,:)^T-\sum_{k\neq\ell}\mathbf{W}(:,k)\left(\mathbf{H}(k,:)\mathbf{H}(\ell,:)^T\right)}{||\mathbf{H}(\ell,:)||_2^2}\right)
\end{aligned}$$

In the original HALS, each column of $\mathbf{W}$ is updated only once before updating $\mathbf{H}$. However, as for the MU, it can be sped up by updating $\mathbf{W}$ several times before updating $\mathbf{H}$, or selecting the entries of $\mathbf{W}$ to update following a Gauss-Southwell-type rule. HALS can also be generalized to other cost functions using Taylor expansion (Li, Liangda, Guy Lebanon, and Haesun Park, 2012).

2.7 Comparative Study: Advantages and Disadvantages

In this section, we compare some of the underlying key differences between various algorithms, and more importantly, broad categories of numerical methods used. A statistical analysis of the difference of performances of the above algorithms, and their expected behaviors can be found in section 4 (Analysis and Conclusion). A main difference between the projected iterative methods and the active-set-like methods for the NLS problems (and thus the NMF problem, since the NLS algorithms are used as subroutines for non-negative matrix, and tensor factorization) lies in their convergence or termination. In projected iterative methods, a sequence of tentative solutions is generated so that an optimal solution is approached in the limit. In practice, one has to somehow stop iterations and return the current estimate, which might be only an approximation of the solution. In the active-set and active-set-like methods, in contrast, there is no concept of a limit point. Tentative solutions are generated with a goal of finding the optimal active and passive set partitioning, which is guaranteed to be found in a finite number of iterations since there are only a finite number of possible active and passive set partitionings. Once the optimal active and passive sets are found, the methods terminate. There are trade-offs of these behavior. While the projected iterative methods may return an approximate solution after a few number of iterations, the active-set and active-set-like methods only return a solution after they terminate. After the termination, however, the solution from the active-set-like methods is an exact solution only subject to numerical rounding errors while the solution from the projected iterative methods might be an approximate one.

Other approaches for solving the NLS problems can be considered as a subroutine for the NMF computation. (Bellavia, Stefania, Maria Macconi, and Benedetta Morini, 2006) have studied an interior point Newton-like method, and (Franc, Vojtech, Vaclav Hlavac, and Mirko Navara, 2005) presented a sequential coordinate-wise method. Some observations about the NMF computation based on these methods as well as other methods are offered in (Cichocki, Andrzej, et al, 2009). (Chu, Moody T., and Matthew M. Lin, 2008) proposed an algorithm based on low-dimensional polytope approximation: Their algorithm is motivated by a geometrical interpretation of NMF that data points are approximated by a simplicial cone (Donoho, David, and Victoria Stodden, 2004).

More specifically, the MU became extremely popular mainly because (i) they are simple to implement[1], (ii) they scale well and are applicable to sparse matrices[2], and (iii) they were proposed in the paper of Lee and Seung (Lee, Daniel D., and H. Sebastian Seung, 1999) which launched the research on NMF. However, the MU converge relatively slowly. HALS has been observed to converge much faster than the MU while having almost the same computational cost. On the other hand, HALS is, under some mild assumptions, guaranteed to converge to a stationary point. For the ANLS algorithm, because usually the initial guess $\mathbf{WH}$ is a poor approximation of $\mathbf{X}$, it does not make much sense to solve the NNLS subproblems exactly at the first steps of Algorithm BCD, and therefore it might be profitable to use ANLS rather in a refinement step of a cheaper NMF algorithm (such as the MU or ALS).

3 Applications in Image Processing

The reason why NMF has become so popular is because of its ability to automatically extract sparse and easily interpretable factors. In this section, we illustrate this property of NMF through its applications in image processing.

3.1 Facial Feature Extraction

Let each column of the data matrix $\mathbf{X} \in \mathbb{R}_+^{I \times T}$ be a vectorized gray-level image of a face, with the (i, j)th entry of matrix $\mathbf{X}$ being the intensity of the ith pixel in the jth face. NMF generates two factors $(\mathbf{W}, \mathbf{H})$ so that each image $\mathbf{X}(:, j)$ is approximated using a linear combination of the columns of $\mathbf{W}$; see Equation and Figure 3 for an illustration. Since $\mathbf{W}$ is nonnegative, the columns of $\mathbf{W}$ can be interpreted as images (that is, vectors of pixel intensities) which we refer to as the basis images. As the weights in the linear combinations are nonnegative ($\mathbf{H} \geq 0$), these basis images can only be summed up to reconstruct each original image. Moreover, the large number of images in the data set must be reconstructed approximately with only a few basis images (in fact, r is in general much smaller than n), hence the latter should be localized features (hence sparse) found simultaneously in several images.

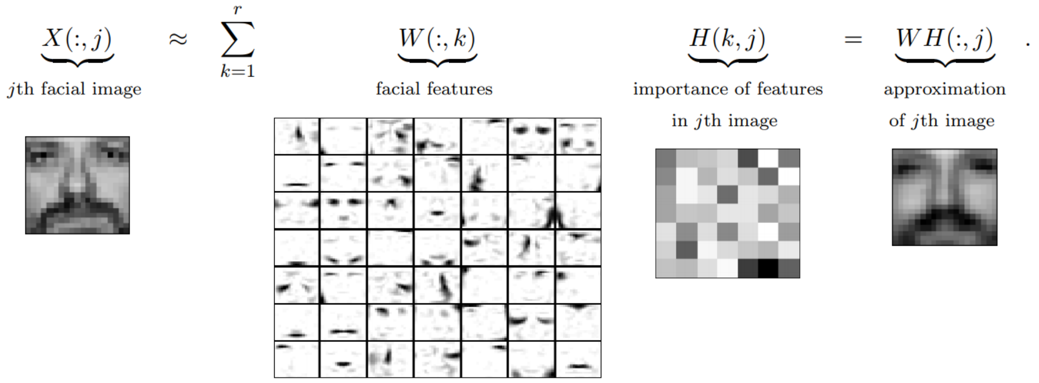
$\underbrace{X(:,j)}_{j\text{th facial image}} \approx \sum_{k=1}^{r} \underbrace{W(:,k)}_{\text{facial features}} \underbrace{H(k,j)}_{\text{importance of features in } j\text{th image}} = \underbrace{WH(:,j)}_{\text{approximation of } j\text{th image}}.$

Figure 3: Decomposition of the CBCL face database, MIT Center For Biological and Computation Learning (2429 gray-level 19-by-19 pixels images) using $r = 49$.

In the case of facial images, the basis images are features such as eyes, noses, mustaches, and lips (see Figure 3) while the columns of $\mathbf{H}$ indicate which feature is present in which image (Lee, Daniel D., and H. Sebastian Seung, 1999). Thus, a major potential application of NMF is in face recognition. It has for example been observed that NMF is more robust to occlusion than PCA (which generates dense factors): in fact, if a new occluded face (e.g., with sun glasses) has to be mapped into the NMF basis, the non-occluded parts (e.g., the mustache or the lips) can still be well approximated.

[1] For example, in Matlab: `W = W.*(X*H').*/(W*(H*H'))`.

[2] When computing the denominator WHH^T in the MU, it is crucial to compute HH^T first in order to have the lowest computational cost.

3.2 Action Recognition via Depth Silhouette Extraction

NMF, is shown by (Masurelle, Aymeric, Slim Essid, and Gal Richard, 2014), to be the most effective and accurate method for action/gesture recognition in images via depth silhouette extraction. NMF outperforms other methods such as PCA by over 9%.

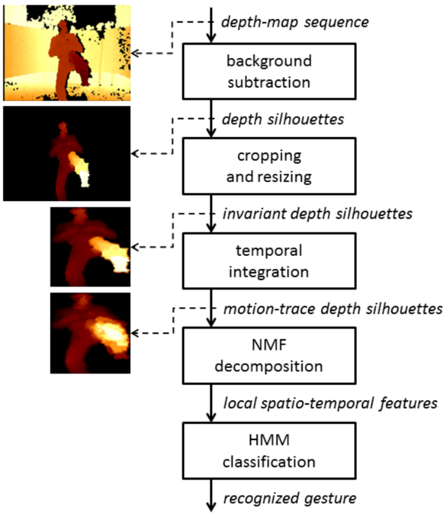

Figure 4: Schematic illustration of gesture recognition approach with NMF.

In the pipeline shown in Figure 4 the motion-trace representation can be seen as a global representation of a short time instant of a full body gesture, and must be decomposed into local spatio-temporal patterns. NMF is known to be appropriate for representing an image as a linear combination of images parts which may be previously learned. Here the observation matrix $\mathbf{X}$ is formed by a concatenation of vectorized version of motion-traces. $\mathbf{W}$ is a dictionary of local spatio-temporal features and $\mathbf{H}$ represents the activations of each local motion feature for each respective motion-traces. $\mathbf{X}$, $\mathbf{W}$ and $\mathbf{H}$ have respectively the dimensions $F \times N$, $F \times K$ and $K \times N$ with F the dimension of the vectorized motion-trace representation, K the number of dictionary components and N the number of motion-trace examples. $D_{SE}(x|y) = \frac{1}{2}(x - y)^2$ is used as divergence.

4 Analysis and Conclusion

In this section we provide an analysis of the performance of the above discussed NMF algorithms based on numerical experiments on the dense CBCL data set (on the left, Figure 5), and the sparse Classic document data set (on the right, Figure 5), performed by (Cichocki, Andrzej, et al, 2009), (Ho, Ngoc-Diep, 2008), (Gillis, Nicolas, 2011), and (Berry, Michael W., M. Browne, A. Langville, P. Pauca, and R. Plemmons, 2007).

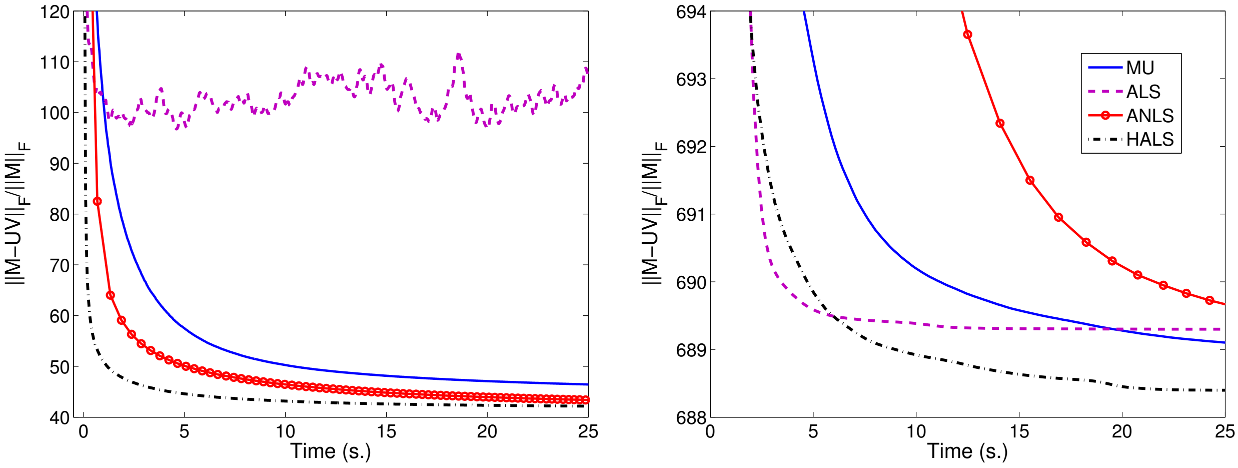

Figure 5: Comparison of MU, ALS, ANLS and HALS. On the left: CBCL facial images with $r = 49$; On the right: Classic document data set with $m = 7094$, $n = 41681$ and $r = 20$. The figure displays the average error using the same ten initial matrices W and H for all algorithms, randomly generated with the `rand` function of Matlab. For MU and HALS, we used the implementation from `https://sites.google.com/site/nicolasgillis/`, for ANLS and ALS from `http://www.cc.gatech.edu/~hpark/nmfsoftware.php`. All tests and numerical evaluations discussed here are based on experiments performed using Matlab by (Gillis, Nicolas, 2011)

As anticipated from the theoretical background and above discussed mathematics of the different algorithms in the previous sections, we see that:

- The MU converge rather slowly.
- ALS oscillates for the dense matrix (CBCL data set) and performs quite poorly while, for the sparse matrix (Classic data set), it converges initially very fast but then stabilizes and cannot compute a solution with small objective function value.
- ANLS performs rather well for the dense matrix and is the second best after HALS. However, it performs rather poorly for the sparse matrix.
- HALS performs the best as it generates the best solutions within an allotted time.

In conclusion, NMF is an easily interpretable linear dimensionality reduction technique for nonnegative data. It is a rather versatile technique with many applications, and brings together a broad range of researchers. In the context of big data and data science, which is continuing to become an increasingly important topic. Further research on NMF includes the design of more efficient algorithms, in particular for regularized problems.

References

[Lee, Daniel D., and H. Sebastian Seung1999] Lee, Daniel D., and H. Sebastian Seung. *Learning of the parts of objects by non-negative matrix factorization.* Nature 401.6755 (1999): 788.

[Lee, Daniel D., and H. Sebastian Seung2001] Lee, Daniel D., and H. Sebastian Seung. *Algorithms for Nonnegative Matrix Factorization.* Advances in neural information processing systems (NIPS). 2001.

[Berry, Michael W., M. Browne, A. Langville, P. Pauca, and R. Plemmons2007] Berry, Michael W., M. Browne, A. Langville, P. Pauca, and R. Plemmons. *Algorithms and applications for approximate nonnegative matrix factorization.* Computational statistics and data analysis 52.1 (2007): 155-173.

[Cichocki, Andrzej, Rafal Zdunek, and Shun-ichi Amari2006] Cichocki, Andrzej, Rafal Zdunek, and Shun-ichi Amari. *Csiszars divergences for non-negative matrix factorization: Family of new algorithms.* International Conference on Independent Component Analysis and Signal Separation. Springer, Berlin, Heidelberg, 2006.

[Sra, Suvrit, and Inderjit S. Dhillon2006] Sra, Suvrit, and Inderjit S. Dhillon. *Generalized nonnegative matrix approximations with Bregman divergences.* Advances in neural information processing systems (NIPS). 2006.

[Ding, Chris, Tao Li, Wei Peng, and Haesun Park2006] Ding, Chris, Tao Li, Wei Peng, and Haesun Park. *Orthogonal nonnegative matrix t-factorizations for clustering.* Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2006.

[Dong, Bo, Matthew M. Lin, and Moody T. Chu2008] Dong, Bo, Matthew M. Lin, and Moody T. Chu. *Nonnegative rank factorization via rank reduction.* http://www4.ncsu.edu/ mtchu/Research/Papers/Readme.html, page (preprint), 2008.

[Kolda, Tamara G., and Brett W. Bader2009] Kolda, Tamara G., and Brett W. Bader. *Tensor decompositions and applications.* SIAM International Conference on Data Mining (SDM) review, 51(3) (2009): 455-500.

[Paatero, Pentti, and Unto Tapper1994] Paatero, Pentti, and Unto Tapper. *Positive matrix factorization: A nonnegative factor model with optimal utilization of error estimates of data values.* Environmetrics 5.2 (1994): 111-126.

[Hasan, Milos, Fabio Pellacini, and Kavita Bala2007] Hasan, Milos, Fabio Pellacini, and Kavita Bala. *Matrix row-column sampling for the many-light problem.* ACM Transactions on Graphics (TOG). Vol. 26. No. 3. ACM, 2007.

[Boutsidis, Christos, Michael W. Mahoney, and Petros Drineas2009] Boutsidis, Christos, Michael W. Mahoney, and Petros Drineas. *An improved approximation algorithm for the column subset selection problem.* Proceedings of the 20th annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics, 2009.

[Kim, Jingu, Yunlong He, and Haesun Park2014] Kim, Jingu, Yunlong He, and Haesun Park. *Algorithms for nonnegative matrix and tensor factorizations: A unified view based on block coordinate descent framework.* Journal of Global Optimization 58.2 (2014): 285-319.

[Bellavia, Stefania, Maria Macconi, and Benedetta Morini2006] Bellavia, Stefania, Maria Macconi, and Benedetta Morini. *An interior point Newtonlike method for nonnegative leastsquares problems with degenerate solution.* Numerical Linear Algebra with Applications 13.10 (2006): 825-846.

[Chu, Moody T., and Matthew M. Lin2008] Chu, Moody T., and Matthew M. Lin. *Low-dimensional polytope approximation and its applications to nonnegative matrix factorization.* SIAM Journal on Scientific Computing 30.3 (2008): 1131-1155.

[Cichocki, Andrzej, et al2009] Cichocki, Andrzej, et al. *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation.* John Wiley and Sons, 2009.

[Gillis, Nicolas2011] Gillis, Nicolas. *Nonnegative matrix factorization: Complexity, algorithms and applications.* Doctoral dissertation, Universite catholique de Louvain. Louvain-La-Neuve: CORE (2011).

[Kim, Hyunsoo, and Haesun Park2008] Kim, Hyunsoo, 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 (2008): 713-730.

[Ho, Ngoc-Diep2008] Ho, Ngoc-Diep. *Nonnegative matrix factorization algorithms and applications.* PhD thesis, Universite catholique de Louvain, 2008.

[Li, Liangda, Guy Lebanon, and Haesun Park2012] Li, Liangda, Guy Lebanon, and Haesun Park. *Fast Bregman divergence NMF using Taylor expansion and coordinate descent.* Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2012.

[Franc, Vojtech, Vaclav Hlavac, and Mirko Navara2005] Franc, Vojtech, Vaclav Hlavac, and Mirko Navara. *Sequential coordinate-wise algorithm for the non-negative least squares problem.* CAIP. Vol. 3691. 2005.

[Moitra, Ankur2013] Moitra, Ankur. *An almost optimal algorithm for computing nonnegative rank.* SIAM Journal on Computing 45.1 (2016): 156-173.

[Liu, Tongliang, Mingming Gong, and Dacheng Tao2017] Liu, Tongliang, Mingming Gong, and Dacheng Tao. *Large-cone nonnegative matrix factorization.* IEEE transactions on neural networks and learning systems (2017).

[Vandaele, Arnaud, Nicolas Gillis, Franois Glineur, and Daniel Tuyttens2016] Vandaele, Arnaud, Nicolas Gillis, Franois Glineur, and Daniel Tuyttens. *Heuristics for exact nonnegative matrix factorization.* SJournal of Global Optimization 65.2 (2016): 369-400.

[Masurelle, Aymeric, Slim Essid, and Gal Richard2014] Masurelle, Aymeric, Slim Essid, and Gal Richard. *Gesture recognition using a NMF-based representation of motion-traces extracted from depth silhouettes.* Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on. IEEE, 2014.

[Gillis, Nicolas2014] Gillis, Nicolas. *The why and how of nonnegative matrix factorization.* Regularization, Optimization, Kernels, and Support Vector Machines 12.257 (2014).

[Daube-Witherspoon, Margaret E., and Gerd Muehllehner1986] Daube-Witherspoon, Margaret E., and Gerd Muehllehner. *An iterative image space reconstruction algorthm suitable for volume ECT.* IEEE transactions on medical imaging 5.2 (1986): 61-66.

[Yang, Jaewon, and Jure Leskovec2013] Daube-Witherspoon, Margaret E., and Gerd Muehllehner. *Overlapping community detection at scale: a nonnegative matrix factorization approach.* Proceedings of the sixth ACM international conference on Web search and data mining. ACM, 2013.

[Yang, Zhirong, and Erkki Oja2010] Yang, Zhirong, and Erkki Oja. *Linear and nonlinear projective nonnegative matrix factorization.* IEEE Transactions on Neural Networks 21.5 (2010): 734-749.

[Donoho, David, and Victoria Stodden2004] Donoho, David, and Victoria Stodden. *When does nonnegative matrix factorization give a correct decomposition into parts?.* Advances in neural information processing systems (NIPS). 2004.

[Sandler, Roman, and Michael Lindenbaum2009] Sandler, Roman, and Michael Lindenbaum. *Nonnegative matrix factorization with earth mover's distance metric.* Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009.