



# Improved SVD-based initialization for nonnegative matrix factorization using low-rank correction

Syed Muhammad Atif<sup>a</sup>, Sameer Qazi<sup>a</sup>, Nicolas Gillis<sup>b,\*</sup>

<sup>a</sup> Graduate School of Science and Engineering, PAF-Karachi Institute of Economics and Technology, Karachi, Pakistan

<sup>b</sup> Department of Mathematics and Operational Research, Faculté polytechnique, Université de Mons, Rue de Houdain 9, Mons 7000, Belgium

## ARTICLE INFO

### Article history:

Received 11 July 2018

Available online 12 February 2019

### MSC:

15A23

68Q17

### Keywords:

Nonnegative matrix factorization

Initialization

Singular value decomposition

Clustering based NMF initialization

CR1-NMF

## ABSTRACT

Due to the iterative nature of the most nonnegative matrix factorization (NMF) algorithms, initialization is a key aspect as it significantly influences both the convergence and the final solution obtained. Many initialization schemes have been proposed for NMF, among which one of the most popular class of methods are based on the singular value decomposition (SVD) and clustering. However, these SVD-based initializations as well as clustering based initializations (if they dense their right factor  $H$ ), do not satisfy a rather natural condition, namely that the error should decrease as the rank of factorization increases. In this paper, we propose a novel SVD-based NMF initialization to specifically address this shortcoming by taking into account the SVD factors that were discarded to obtain a nonnegative initialization. This method, referred to as nonnegative SVD with low-rank correction (NNSVD-LRC), allows us to significantly reduce the initial error at a negligible additional computational cost using the low-rank structure of the discarded SVD factors. NNSVD-LRC has two other advantages compared to other NMF initializations: (1) it provably generates sparse initial factors, and (2) it is faster as it only requires to compute a truncated SVD of rank  $\lfloor \frac{r}{2} + 1 \rfloor$  where  $r$  is the factorization rank of the sought NMF decomposition (as opposed to a rank- $r$  truncated SVD for other methods). We show on several standard dense and sparse data sets that our new method competes favorably with state-of-the-art SVD-based and clustering based initializations for NMF.

© 2019 Elsevier B.V. All rights reserved.

## 1. Introduction

Nonnegative matrix factorization (NMF) is the problem of approximating a input nonnegative matrix  $X$  as the product of two nonnegative matrices: Given  $X \in \mathbb{R}_{\geq 0}^{m \times n}$  and an integer  $r$ , find  $W \in \mathbb{R}_{\geq 0}^{m \times r}$  and  $H \in \mathbb{R}_{\geq 0}^{r \times n}$  such that  $X \approx WH$ . NMF allows to reconstruct data using a purely additive model: each column of  $X$  is a nonnegative linear combination of the columns of  $W$ . For this reason, it is widely employed in research fields like image processing and computer vision [8,21], data mining and document clustering [6], hyperspectral image analysis [19,25], signal processing [31] and computational biology [20]; see also [5,9] and the references therein.

To measure the quality of the NMF approximation, a distance metric should be chosen. In this paper, we focus on the most widely used one, namely the Frobenius norm, leading to the following optimization problem

$$\min_{W \in \mathbb{R}_{\geq 0}^{m \times r}, H \in \mathbb{R}_{\geq 0}^{r \times n}} \|X - WH\|_F^2 \quad \text{such that } W \geq 0 \text{ and } H \geq 0, \quad (1)$$

where  $\|M\|_F = \sqrt{\sum_{i,j} M_{i,j}^2}$  is Frobenius norm of a matrix  $M$ . Most algorithms tackling (1) use standard non-linear optimization schemes such as block coordinate descent methods hence initialization of the factors ( $W, H$ ) is crucial in practice as it will influence

- (i) the number of iterations needed for an algorithm to converge (in fact, if the initial point is closer to a local minimum, it will require less iterations to converge to it), and
- (ii) the final solution to which the algorithm will converge.

Note that, due to the NP-hardness of NMF [27], no polynomial-time algorithm currently exist that can obtain a globally optimal solution in general. Many approaches have been proposed for NMF initialization, for example based on  $k$ -means and spherical  $k$ -means by Wild et al. [29], on fuzzy  $c$ -means by Rezaei et al. [23], on nature inspired heuristic algorithms by Janeczek and Tan [13], on Lanczos bidiagonalization by Wang et al. [28], on subtractive clustering by Casalino et al. [4], on independent component analysis by Kitamura and Ono [14], on the successive projection algorithm by Sauwen et al. [24], and on rank-one approximations by Liu and Tan [18], to name a few; see also Langville et al. [15].

\* Corresponding author.

E-mail address: [nicolas.gillis@umons.ac.be](mailto:nicolas.gillis@umons.ac.be) (N. Gillis).

In this paper, we focus on SVD-based initializations for NMF. Two of the most widely used methods are NNDSVD [2] and SVD-NMF [22] which are described in the next section. These methods suffer from the fact that the approximation error  $\|X - WH\|_F^2$  of the initial factors ( $W, H$ ) increases as the rank increases which is not a desirable property for NMF initializations. Our key contribution is to provide a new SVD-based initialization that does not suffer from this shortcoming while (i) it generates sparse factors which not only provide storage efficiency [10] but also provide better part-based representations [4,7] and resilience to noise [26,30], and (ii) it only requires a truncated SVD of rank  $\lfloor \frac{r}{2} + 1 \rfloor$ , as opposed to a truncated SVD of rank  $r$  for the other SVD-based initializations.

*Outline of the paper* This paper is organized as follows. Section 2 will discuss our proposed solution in details, highlighting the differences with existing SVD-based initializations. In Section 3, we evaluate our proposed solution against other SVD-based initializations on dense and sparse data sets. Section 4 concludes the paper.

## 2. Nonnegative SVD with low-rank correction, a new SVD-based NMF initialization

The truncated SVD is a low-rank matrix approximation technique that approximates a given matrix  $X \in \mathbb{R}^{m \times n}$  as a sum of  $r$  rank-one terms made of singular triplets, where  $1 \leq r \leq \text{rank}(X)$ . Each singular triplet  $(u_i, v_i, \sigma_i)$  ( $1 \leq i \leq r$ ) consists of two column vectors  $u_i$  and  $v_i$  which are the left and the right singular vectors, respectively, associated with the  $i$ th singular value (which we assume are sorted in nonincreasing order). We have

$$X \approx X_r = \sum_{i=1}^r \sigma_i u_i v_i^T = U_r \Sigma_r V_r^T, \quad (2)$$

where  $(\cdot)^T$  is the transpose of given matrix or vector,  $X_r$  is the rank- $r$  approximation of  $X$ , the columns of  $U_r \in \mathbb{R}^{m \times r}$  (resp. of  $V_r \in \mathbb{R}^{n \times r}$ ) are the left (resp. right) singular vectors, and  $\Sigma_r \in \mathbb{R}^{r \times r}$  is the diagonal matrix containing the singular values on its diagonal. According to Eckhart–Young theorem,  $X_r$  provides an optimal rank- $r$  approximation of  $X$  with respect to the Frobenius and spectral norms [12]. To simplify our later derivations, we transform the three factors of the SVD representation into two factors, like in NMF, by multiplying  $U_r$  and  $V_r^T$  by the square root of  $\Sigma_r$  to obtain  $Y_r$  and  $Z_r$ :

$$X \approx X_r = \sum_{i=1}^r y_i z_i = Y_r Z_r, \quad (3)$$

where  $Y_r = U_r \Sigma_r^{1/2}$ ,  $Z_r = \Sigma_r^{1/2} V_r^T$ ,  $y_i = \sqrt{\sigma_i} u_i$  and  $z_i = \sqrt{\sigma_i} v_i^T$  for  $1 \leq i \leq r$ . Matrices  $Y_r$  and  $Z_r$  cannot be used directly for NMF initialization since  $Y_r$  and  $Z_r$  usually contain negative elements (roughly half of them, except for the first factor, by the Perron–Frobenius theorem, see Berman and Plemmons [1]).

Given a vector  $x$ , let us denote  $x^{(\geq 0)} = \max(0, x)$  its nonnegative part and  $x^{(\leq 0)} = \max(0, -x)$  its nonpositive part so that  $x = x^{(\geq 0)} - x^{(\leq 0)}$ .

Using this notation, (3) can be rewritten as:

$$X \approx X_r = \sum_{i=1}^r y_i z_i = \sum_{i=1}^r (y_i^{(\geq 0)} z_i^{(\geq 0)} + y_i^{(\leq 0)} z_i^{(\leq 0)}) - \sum_{i=1}^r (y_i^{(\geq 0)} z_i^{(\leq 0)} + y_i^{(\leq 0)} z_i^{(\geq 0)}) \quad (4)$$

To obtain a feasible initialization for NMF, we have to deal with the second summand which leads to negative elements in the decomposition. Currently, there are mostly two approaches used in practice for this purpose.

The first approach discards the second summand and selects  $r$  product terms from the first summand on the basis of some criterion. In particular, the most widely used method, namely nonnegative double SVD (NNDSVD) by Boutsidis and Gallopoulos [2], selects  $r$  terms as follows: for each  $i$ , it selects  $y_i^{(\geq 0)} z_i^{(\geq 0)}$  if  $\|y_i^{(\geq 0)} z_i^{(\geq 0)}\|_F > \|y_i^{(\leq 0)} z_i^{(\leq 0)}\|_F$ , otherwise it selects  $y_i^{(\leq 0)} z_i^{(\leq 0)}$ . This is equivalent to projecting  $Y_r$  and  $Z_r$  onto the nonnegative orthant but taking advantage of the sign ambiguity of the SVD [3]. The second approach takes the absolute value of the second term, which is equivalent to using  $W = |Y_r|$  and  $H = |Z_r|$  as an initialization for NMF [22]. This method is referred to as SVD-NMF.

Let us denote  $X_r^{\geq 0}$  the solution obtained by one of the two approaches mentioned above. In both cases, we will have

$$X_{r+1}^{\geq 0} \geq X_r^{\geq 0} \quad \text{for all } r \geq 1,$$

since each rank-one factor selected from the SVD is nonnegative. Hence, for  $r$  sufficiently large, the error  $\|X - X_r^{\geq 0}\|_F$  will increase as  $r$  increases since the negative terms are not taken into account; see Fig. 1 for examples on real data sets. Like the unconstrained rank- $r$  approximation  $X_r$  of  $X$ , it would make sense that the approximation quality of  $X_r^{\geq 0}$  increases as  $r$  increases. Another drawback of these approaches is that they either throw away half of the rank-one factors of the first summand and all of the rank-one factors in the second summand (as in NNDSVD) or sum them together so that the sign information is lost (as in SVD-NMF): a lot of information is wasted.

In order to avoid these two important drawbacks, we propose a new method where

- (i) We keep all the terms from the first summand in (4). Hence, we will only need a truncated SVD of rank  $\lfloor \frac{r}{2} + 1 \rfloor$ . In fact, assuming the matrices  $XX^T$  and  $X^T X$  are irreducible<sup>1</sup> (which is the case for all the matrices we have tested in practice), the first rank-one factor  $y_1 z_1$  of the SVD is positive, by the Perron–Frobenius theorem [1]. This implies that  $y_i^{(\geq 0)} z_i^{(\geq 0)} \neq 0$  and  $y_i^{(\leq 0)} z_i^{(\leq 0)} \neq 0$  for all  $i \geq 2$  because the singular triplets are orthogonal to one another [12], that is,  $y_i^T y_1 = z_i z_1^T = 0$  for all  $i \geq 2$ , which implies that  $y_i$  and  $z_i$  contain at least one positive and one negative entry.
- (ii) Although we also discard the second summand as in NNDSVD, we will use this information to improve the terms in the first summand. This can be done computationally very efficiently using the low-rank structure of the second summand; see the details below.

Our initialization is described in Algorithm 1. It works as follows: Let  $p = \lfloor \frac{r}{2} + 1 \rfloor$ . Then,

1. Compute the rank- $p$  truncated SVD of  $X$ , with  $X_p = \sum_{i=1}^p y_i z_i$ ; see (3).
2. The first rank-one factor of the SVD is used to initialize  $W(:, 1)$  and  $H(1, :)$ , that is,  $W(:, 1) = |y_1|$  and  $H(1, :) = |z_1|$ . Note that the absolute value is used because the SVD has a sign ambiguity (hence could generate  $y_1$  and  $z_1$  with negative entries). In any case,  $|y_1| |z_1|$  is an optimal rank-one approximation since  $X$  is nonnegative [1].
3. The other  $r - 1$  rank-one factors are given by the next  $\lfloor \frac{r}{2} \rfloor$  factors of the truncated SVD as follows:  $W(:, i) = y_i^{(\geq 0)}$ ,  $W(:, i + 1) = y_i^{(\leq 0)}$ ,  $H(i, :) = z_i^{(\geq 0)}$  and  $H(i + 1, :) = z_i^{(\leq 0)}$ , where  $i = 2, 4, \dots$ , in order to obtain a nonnegative NMF initialization  $(W, H)$  with  $r$  factors. Note that, by this construction, the average sparsity of these factors is at least 50%.

<sup>1</sup> A symmetric matrix is irreducible if and only if its associated graph is connected.

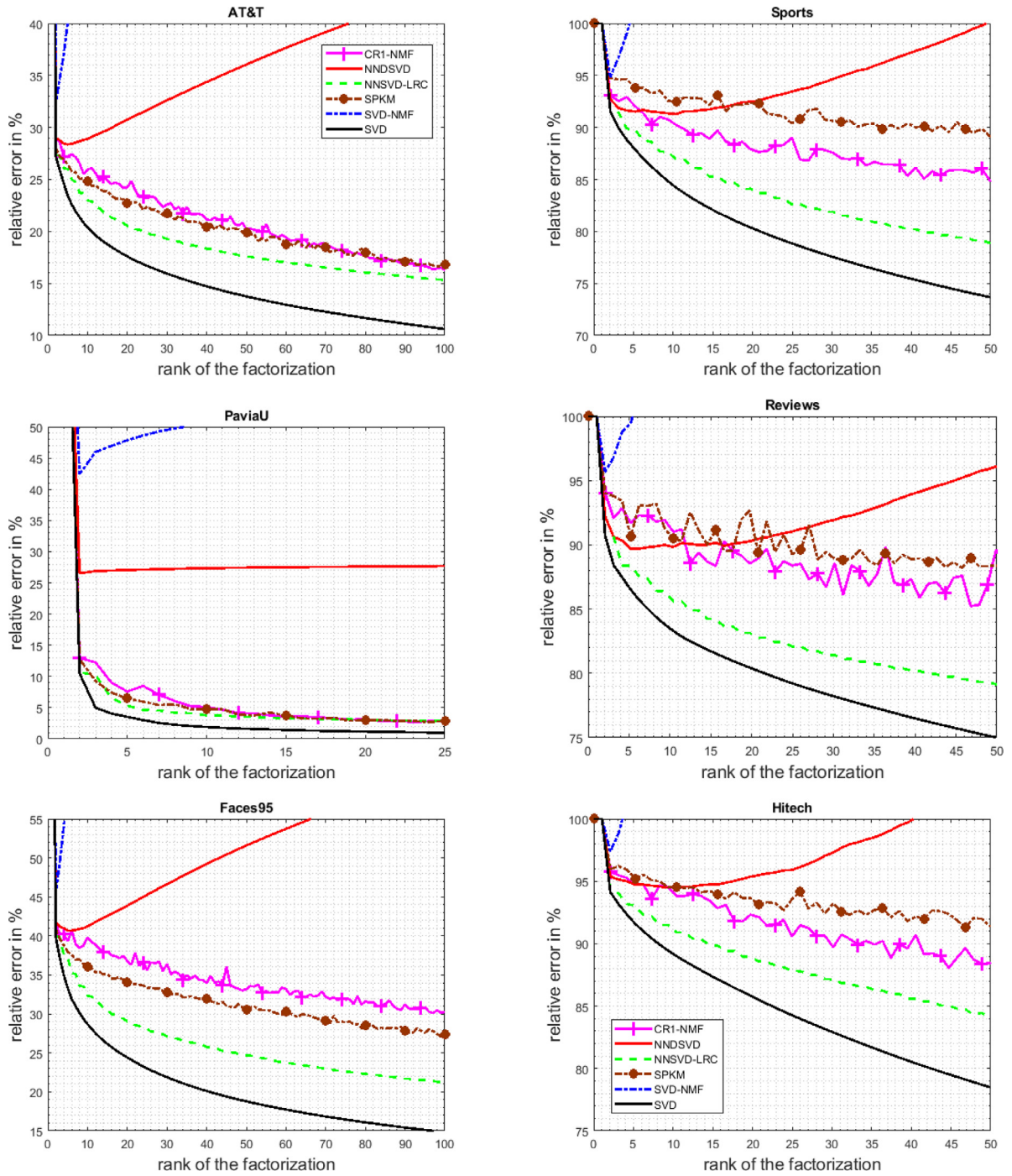


Fig. 1. Relative error of the NMF initializations for different values of the rank  $r$ .

(In practice, SVD factors usually do not contain zero entries hence average sparsity is exactly 50%, ignoring the first rank-one factor.)

4. In order to improve the current solution  $(W, H)$  built using the first  $p$  singular triplets, we propose to update them using the low-rank approximation  $X_p$  by performing a few iteration of an NMF algorithm on the problem

$$\min_{W \geq 0, H \geq 0} \|X_p - WH\|_F^2, \quad \text{where } X_p = Y_p Z_p.$$

The reason for this choice is that, for most NMF algorithms, performing such iterations is significantly cheaper than performing a standard NMF iteration on the input matrix  $X$ . In

fact, the most expensive steps of most NMF algorithms is to compute  $XH^T$ ,  $W^T X$ ,  $HH^T$  and  $W^T W$  which relates to computing the gradient of the objective function; see, e.g., [11]. When  $X = X_p$  has a low-rank representation  $X_p = Y_p Z_p$ , the cost of one NMF iteration reduces from  $O(mnr)$  operations to  $O((m+n)r^2)$  operations.

In this paper, we use the state-of-the-art NMF algorithm referred to as accelerated hierarchical alternating least squares (A-HALS) by Gillis and Glineur [11] to perform this step. A proper implementation requires  $O((m+n)r^2)$  operations per iteration instead of  $O(mnr)$  if we would apply A-HALS on the input matrix  $X$ , as explained above. We run A-HALS as long



**Algorithm 1** Nonnegative singular value decomposition with low-rank correction (NNSVD-LRC).

**Input:** A nonnegative matrix  $X$  and factorization rank- $r$ .

**Output:** Nonnegative factors  $W \in \mathbb{R}_+^{m \times r}$  and  $H \in \mathbb{R}_+^{r \times n}$  such that  $X \approx WH$

```

1:  $p = \lfloor \frac{r}{2} + 1 \rfloor$ ;
2:  $[U, \Sigma, V] = \text{truncated-SVD}(X, p)$ ;
3:  $Y_p = U\Sigma^{1/2}$ ;  $Z_p = \Sigma^{1/2}V^T$ ;
4: % Populating  $W$  and  $H$  using  $Y_p$  and  $Z_p$ 
5:  $W(:, 1) = |Y_p(:, 1)|$ ;  $H(1, :) = |Z_p(1, :)|$ ;
6:  $i = 2$ ;  $j = 2$ ;
7: while  $i \leq r$  do
8:   if  $i$  is even then
9:      $W(:, i) = \max(Y_p(:, j), 0)$ ;
10:     $H(i, :) = \max(Z_p(j, :), 0)$ ;
11:   else
12:     $W(:, i) = \max(-Y_p(:, j), 0)$ ;
13:     $H(i, :) = \max(-Z_p(j, :), 0)$ ;
14:     $j = j + 1$ ;
15:   end if
16:    $i = i + 1$ ;
17: end while
18:  $e_0 = \|X_p - WH\|_F$ ;  $k = 0$ ;
19: % Improve  $W$  and  $H$  by applying A-HALS on the low-rank matrix
    $X_p = Y_p Z_p$ 
20: while  $k = 0$  or  $e_k - e_{k-1} \geq \delta e_0$  do
21:   Perform one iteration of A-HALS on  $X_p = Y_p Z_p$  starting from
   ( $W, H$ ) to obtain an improved solution ( $W, H$ ).
22:    $e_{k+1} = \|X_p - WH\|_F$ ;
23:    $k = k + 1$ ;
24: end while

```

as the relative error decreases the initial error by a proportion of  $\delta$ . We used  $\delta = 5\%$  which leads in all tested cases to less than 10 iterations, which are negligible compared to computing the truncated SVD that requires  $\Omega(pmn)$  operations, and to the subsequent NMF iterations, that require  $O(mnr)$  operations.

The idea of using a low-rank approximation of  $X$  to speed up NMF computations was proposed by Zhou et al. [33], but not in combination with A-HALS nor as an initialization procedure.

For these reasons, we will refer to our method as nonnegative SVD with low-rank correction (NNSVD-LRC) as it consist of (i) a selection of nonnegative factors from the SVD followed by (ii) NMF iterations that uses the low-rank approximation  $X_p$  of  $X$ , for a negligible additional computational cost of  $O((m+n)r^2)$  operations.

**Remark 1** (Computation of the error). In Algorithm 1, the error  $\|X_p - WH\|_F$  has to be computed: this can be done in  $O((m+n)r^2)$  operations observing that

$$\begin{aligned} \|X_p - WH\|_F^2 &= \langle X_p, X_p \rangle - 2\langle X_p, WH \rangle + \langle WH, WH \rangle \\ &= \langle Y_p Z_p, Y_p Z_p \rangle - 2\langle Y_p Z_p, WH \rangle + \langle W^T W, H H^T \rangle \\ &= \langle Y_p^T Y_p, Z_p Z_p^T \rangle - 2\langle (W^T Y_p) Z_p, H \rangle + \langle W^T W, H H^T \rangle, \end{aligned}$$

where  $\langle A, B \rangle = \sum_{i,j} A_{i,j} B_{i,j}$  is the inner product associated with the Frobenius norm.

### 3. Numerical experiments

In this section, we compare NNSVD-LRC with four NMF initializations. The first two are the SVD based NMF initializations

**Table 1**  
Dense data sets.

Data set	Image size ( $h \times w$ )	$m = h \times w$	$n$
AT&T Faces <sup>a</sup>	112 × 92	10,304	400
PaviaU <sup>b</sup>	610 × 340	207,400	103
Faces95 <sup>c</sup>	180 × 200	36,000	1440

<sup>a</sup> <https://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>

<sup>b</sup> [http://www.ehu.es/ccwintco/index.php?title=Hyperspectral\\_Remote\\_Sensing\\_Scenes](http://www.ehu.es/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes)

<sup>c</sup> <http://cswwww.essex.ac.uk/mv/allfaces/faces95.html>

**Table 2**  
Document data sets [32].

Dataset Name	#nonzeros	Sparsity	$m$	$n$
Sports	1,091,723	99.14	8580	14,870
Reviews	758,635	98.99	4069	18,483
Hitech	331,373	98.57	2301	10,080

presented in Section 2, namely NNDSVD and SVD-NMF. The third one, CR1-NMF, is a recent hybrid method combining clustering and the computation of rank-one SVDs [18]. The fourth one, SPKM, is one of the first proposed initialization for NMF using spherical  $k$ -means [29]. The code for CR1-NMF and SPKM are available from <https://github.com/zhaoliangliu/cr1-nmf>.

All tests are performed using Matlab R2017b (Student License) on a laptop Intel CORE i5-2540M CPU @2.60GHz 4GB RAM. The code is available from <https://sites.google.com/site/nicolagillis/code>. Due to the space limit, we restrict ourselves to three dense and three sparse widely used data sets; see Tables 1 and 2. We also restrict ourselves to using the multiplicative update algorithm, one of the most widely used one. (On the Matlab code provided online, we provide experiments for two other data sets, namely the CBCL facial images, and the classic document data set, in combination with A-HALS.)

Throughout this section, we will use the following two quantities:

1. the relative error which measures the quality of an NMF solution:

$$\text{relative error}(W, H) = \frac{\|X - WH\|_F}{\|X\|_F},$$

2. the sparsity which measures the proportion of zero entries in a matrix:

$$\text{sparsity}(W) = \frac{\# \text{ of zeros in } W}{\# \text{ of total elements in } W}.$$

#### 3.1. Initial error

Fig. 1 displays the relative errors in percent for different values of  $r$  for each data set. This illustrates the fact that the error of NNDSVD and SVD-NMF increases as  $r$  increases (as soon as  $r$  is sufficiently large); see the discussion in Section 2. In contrast, the error NNSVD-LRC decreases as  $r$  increases, while the error of SPKM and CR1-NMF usually decreases but sometimes increases due to the fact that these methods are not deterministic. However, NNSVD-LRC provides the lowest initial error in all cases. Note that the relative error of SVD-NMF grows much faster than NNDSVD.

One may argue that the above comparison is not totally fair as other NMF initializations did not update their factors  $W$  and  $H$  as opposed to NNSVD-LRC. Therefore, Table 3 displays the relative error in percent of the NMF initializations for different values of the factorization rank  $r$ , after the NNLS update, and also after one iteration of the HALS algorithm. Although the error of other NMF initializations decreases significantly compared to the initial error,

**Table 3**

Comparison of the relative error (in percent) of the NMF initializations when they are aided by one iteration of HALS and the NNLS update of  $H$ . The lowest error is highlighted in bold.

	AT&T			PaviaU			Faces95		
	$r=60$	$r=80$	$r=100$	$r=9$	$r=12$	$r=15$	$r=72$	$r=108$	$r=144$
NNDSVD	37.65	40.60	43.26	27.32	27.43	27.52	56.14	62.33	67.46
NNDSVD + HALS	22.10	21.71	21.35	18.04	15.93	15.91	30.92	30.08	29.52
NNDSVD + NNLS	25.55	25.49	25.46	22.51	22.44	22.44	36.05	36.00	35.98
SVD-NMF	113.50	128.75	141.86	50.17	50.99	51.53	174.88	205.69	229.83
SVD-NMF + HALS	22.14	21.37	20.76	16.20	15.00	14.34	30.86	29.41	28.54
SVD-NMF + NNLS	27.80	27.77	27.76	29.43	29.32	29.31	39.55	39.51	39.48
CR1-NMF	20.54	17.64	16.53	5.25	4.58	3.75	31.84	29.54	28.01
CR1-NMF + HALS	22.28	21.08	20.43	4.89	4.31	3.50	31.01	28.70	27.29
CR1-NMF + NNLS	18.53	16.47	15.37	<b>3.47</b>	<b>3.04</b>	<b>2.69</b>	28.15	25.93	24.27
SPKM	23.18	21.06	20.52	7.46	7.33	7.31	33.02	32.81	30.75
SPKM + HALS	23.03	20.89	20.33	6.31	6.23	6.24	32.31	31.99	29.95
SPKM + NNLS	21.88	19.77	18.85	4.47	4.36	4.33	30.32	29.97	27.69
NNSVD-LRC	<b>17.00</b>	<b>16.04</b>	<b>15.31</b>	4.11	3.54	3.31	<b>22.77</b>	<b>20.85</b>	<b>19.51</b>

	Sports			Reviews			Hitech		
	$r=15$	$r=20$	$r=25$	$r=15$	$r=20$	$r=25$	$r=15$	$r=20$	$r=25$
NNDSVD	91.91	92.58	93.67	90.17	90.46	91.24	94.75	95.50	96.14
NNDSVD + HALS	85.69	84.65	83.90	<b>84.19</b>	83.35	82.78	89.93	89.09	88.29
NNDSVD + NNLS	87.46	86.74	86.29	86.41	85.82	85.52	91.46	90.87	90.29
SVD-NMF	133.33	147.08	161.63	123.63	132.46	143.59	127.74	143.14	157.08
SVD-NMF + HALS	87.04	86.12	85.46	84.83	84.10	83.64	90.72	90.02	89.45
SVD-NMF + NNLS	90.52	90.16	89.99	88.63	88.21	88.05	93.48	93.29	93.07
CR1-NMF	90.56	89.54	87.57	90.29	89.30	87.65	93.05	91.99	92.95
CR1-NMF + HALS	89.09	87.64	85.55	86.41	85.69	84.80	91.64	89.93	91.25
CR1-NMF + NNLS	88.63	87.06	84.73	85.36	84.39	83.77	91.28	89.34	90.79
SPKM	88.38	88.43	87.47	89.54	87.44	86.75	92.30	91.75	90.72
SPKM + HALS	85.45	85.08	84.09	84.98	84.28	83.68	90.42	89.44	87.84
SPKM + NNLS	86.79	86.73	85.37	88.25	86.17	85.08	91.39	90.71	89.38
NNSVD-LRC	<b>85.29</b>	<b>83.69</b>	<b>82.66</b>	84.22	<b>82.76</b>	<b>82.07</b>	<b>89.85</b>	<b>88.59</b>	<b>87.81</b>

it is still higher than NNSVD-LRC on all data sets, except for the combination of CR1-NMF with NNLS update of  $H$  on the PaviaU data set, and the combination of NNDSVD with HALS on the Reviews data set.

### 3.2. Sparsity

For the sparsity of the initializations, SVD-NMF generates dense initial factors, with sparsity 0% in all cases (because SVD generated dense factors and SVD-NMF take their absolute values as initial estimates for  $W$  and  $H$ ). For both SPKM and CR1-NMF, sparsity of the left factor  $W$  is 0% on dense data sets (because centroids are the average of several data points). On sparse data sets it goes from 8.63% to 31.92% and from 2.21% to 15.56% for SPKM and CR1-NMF, respectively. For both SPKM and CR1-NMF, sparsity of  $H$  is always  $1 - \frac{1}{r}$ . This was expected as they are clustering based NMF initializations: the matrix  $H$  contains a single non-zero entry per column. Note however that such a highly sparse initialization does not in general lead to good NMF solutions, hence it is recommended to add to each entry of  $H$  a constant before using an NMF algorithm; see the discussion by Liu and Tan [18]. For the numerical experiments in Table 5 where we initialize the NMF multiplicative updates with the different initializations, we use this strategy as done by Liu and Tan [18], otherwise the results of SPKM and CR1-NMF were rather poor. This explains the higher initial error of SPKM and CR1-NMF in Table 5 compared to Table 3.

NNDSVD generates factors with average sparsity 48.58%, with the sparsity of every initialization ( $W$ ,  $H$ ) being between 42.18% and 52.56% for all data sets. NNSVD-LRC generates factors with average sparsity 48.40% (resp. 57.35%), with the sparsity of every initialization ( $W$ ,  $H$ ) being between 25.03% (resp. 51.32%) and 66.25% (resp. 64.60%) for dense (resp. sparse) data sets. This confirms our discussion in Section 2 where the initialization provided by NNSVD-LRC has average sparsity around 50%, similarly as NNDSVD. (Note that

this is not exactly 50% because of the low-rank correction step performed by NNSVD-LRC.)

### 3.3. Computational time

Table 4 reports the computational time for the different initializations on the different data sets, averaged over 100 runs. As expected, NNDSVD and SVD-NMF have roughly the same computational cost, the main cost being the computation of the rank- $r$  truncated SVD. SPKM and CR1-NMF have similar computational cost as NNDSVD and SVD-NMF. Although NNSVD-LRC is expected to be faster than NNDSVD and SVD-NMF (see discussion at the end of Section 2), it is not the case for dense data sets. The reason is that we are using a simple Matlab implementation (which is not as well tuned as the Matlab function `svd`) which requires a loop over the columns of  $W$  and rows of  $H$  and Matlab is not well designed to handle such loops. Using another programming language to make NNSVD-LRC faster is a direction for further research.

### 3.4. Convergence of NMF algorithms

We now compare the five NMF initializations used in combination with one of the most widely used NMF algorithm, namely, the multiplicative updates (MU) by Lee and Seung [16,17]. Table 5 displays the relative error in percent after 1, 10 and 100 iterations of MU.

We observe the following:

- Except for the data sets Reviews with  $r=20$  and Hitech, 1 or 10 iterations of MU are not enough for the four initializations to get back at NNSVD-LRC. This is explained by the fact that the initial error of NNSVD-LRC is much lower, as shown in Fig. 1 and Table 3.

**Table 4**

CPU time (in s.) taken by different NMF initializations for the different data sets. Bold indicates the algorithm that took the least CPU time.

	AT&T			PaviaU			Faces95		
	<i>r</i> =60	<i>r</i> =80	<i>r</i> =100	<i>r</i> =9	<i>r</i> =12	<i>r</i> =15	<i>r</i> =72	<i>r</i> =108	<i>r</i> =144
NNDSVD	0.276	0.328	0.381	0.482	0.552	0.621	6.208	6.821	6.890
SVD-NMF	<b>0.235</b>	<b>0.280</b>	<b>0.315</b>	<b>0.375</b>	<b>0.413</b>	<b>0.438</b>	<b>6.061</b>	<b>6.675</b>	<b>6.737</b>
CR1-NMF	0.594	0.723	0.867	1.219	1.352	1.521	7.518	10.024	10.457
SPKM	0.686	0.752	0.835	2.269	2.507	2.740	8.353	10.618	11.104
NNSVD-LRC	1.105	1.623	2.129	1.805	2.590	3.403	9.362	12.828	13.368
	Sports			Reviews			Hitech		
	<i>r</i> =15	<i>r</i> =20	<i>r</i> =25	<i>r</i> =15	<i>r</i> =20	<i>r</i> =25	<i>r</i> =15	<i>r</i> =20	<i>r</i> =25
NNDSVD	1.922	2.110	2.742	1.470	1.626	2.133	0.881	1.037	1.395
SVD-NMF	1.876	2.089	2.707	1.433	1.586	2.085	0.876	1.019	1.369
CR1-NMF	3.361	3.182	2.991	2.943	2.710	2.555	0.919	0.805	0.772
SPKM	2.347	2.487	2.618	1.452	<b>1.512</b>	<b>1.575</b>	<b>0.522</b>	<b>0.562</b>	<b>0.610</b>
NNSVD-LRC	<b>1.400</b>	<b>1.734</b>	<b>2.178</b>	<b>0.970</b>	1.538	1.810	0.604	0.874	0.992

**Table 5**

Relative error in percent of MU after 1, 10 and 100 iterations when seeded by different NMF initializations on the dense and sparse data sets. The lowest error is highlighted in bold.

		AT&T			PaviaU			Faces95		
		<i>r</i> =60	<i>r</i> =80	<i>r</i> =100	<i>r</i> =9	<i>r</i> =12	<i>r</i> =15	<i>r</i> =72	<i>r</i> =108	<i>r</i> =144
NNDSVD	1	24.58	24.51	24.47	22.02	22.02	22.02	35.39	35.44	35.53
SVD-NMF		30.03	30.02	30.02	29.70	29.67	29.63	42.55	42.57	42.58
CR1-NMF		29.45	29.48	29.51	23.47	24.58	25.56	42.13	42.18	42.21
SPKM		29.82	29.78	29.79	30.17	30.86	31.12	42.36	42.30	42.30
NNSVD-LRC		<b>16.91</b>	<b>15.95</b>	<b>15.21</b>	<b>4.00</b>	<b>3.45</b>	<b>3.22</b>	<b>22.60</b>	<b>20.68</b>	<b>19.35</b>
NNDSVD	10	21.71	21.52	21.40	13.52	13.42	13.27	30.05	29.81	29.72
SVD-NMF		27.18	27.15	27.14	15.74	15.32	14.92	35.95	35.90	35.91
CR1-NMF		26.41	26.51	26.58	6.94	7.29	7.04	33.00	32.66	32.46
SPKM		27.75	27.50	27.31	12.30	12.24	12.27	34.56	34.11	33.68
NNSVD-LRC		<b>16.63</b>	<b>15.66</b>	<b>14.93</b>	<b>3.78</b>	<b>3.28</b>	<b>3.06</b>	<b>22.04</b>	<b>20.16</b>	<b>18.88</b>
NNDSVD	100	17.83	17.09	16.52	7.39	7.37	6.45	24.24	22.87	21.99
SVD-NMF		17.06	16.40	15.92	3.81	3.67	3.37	21.73	20.32	19.45
CR1-NMF		<b>15.94</b>	<b>14.72</b>	<b>13.96</b>	<b>3.29</b>	3.00	2.77	21.33	19.75	18.64
SPKM		17.08	16.23	15.62	4.39	4.02	3.91	22.06	20.57	19.41
NNSVD-LRC		15.97	14.98	14.20	3.45	<b>2.93</b>	<b>2.71</b>	<b>21.01</b>	<b>19.17</b>	<b>17.91</b>
		Sports			Reviews			Hitech		
		<i>r</i> =15	<i>r</i> =20	<i>r</i> =25	<i>r</i> =15	<i>r</i> =20	<i>r</i> =30	<i>r</i> =15	<i>r</i> =20	<i>r</i> =25
NNDSVD	1	87.22	86.53	86.07	85.57	84.91	84.58	91.13	90.56	89.99
SVD-NMF		90.90	90.56	90.46	88.89	88.54	88.38	93.60	93.48	93.31
CR1-NMF		90.05	88.92	86.80	88.01	86.75	85.60	92.22	90.42	92.19
SPKM		91.97	91.98	91.31	92.63	92.28	93.08	94.99	95.00	94.61
NNSVD-LRC		<b>84.79</b>	<b>83.38</b>	<b>81.79</b>	<b>83.90</b>	<b>82.56</b>	<b>81.80</b>	<b>89.59</b>	<b>88.23</b>	<b>87.29</b>
NNDSVD	10	84.17	82.70	81.40	82.78	<b>81.66</b>	81.01	<b>88.48</b>	87.34	86.24
SVD-NMF		84.02	82.68	81.34	83.01	81.90	81.03	89.52	<b>86.89</b>	<b>85.59</b>
CR1-NMF		85.92	83.63	81.81	83.02	82.27	81.43	89.34	87.54	86.94
SPKM		84.11	82.86	81.73	83.12	82.60	81.81	88.75	87.21	86.22
NNSVD-LRC		<b>83.68</b>	<b>81.85</b>	<b>80.93</b>	<b>82.55</b>	81.71	<b>80.55</b>	88.54	86.99	85.64
NNDSVD	100	82.93	81.38	79.89	82.30	80.96	79.92	87.88	86.49	85.10
SVD-NMF		82.96	81.09	<b>79.55</b>	82.20	<b>80.76</b>	<b>79.63</b>	<b>87.70</b>	<b>86.14</b>	<b>84.79</b>
CR1-NMF		83.10	81.42	79.96	82.22	81.13	79.94	87.92	86.53	85.20
SPKM		83.03	81.11	79.87	82.21	81.19	80.27	87.94	86.43	85.09
NNSVD-LRC		<b>82.88</b>	<b>81.08</b>	79.81	<b>82.04</b>	81.01	79.92	87.87	86.34	85.06

- After 100 iterations of the MU, some initializations are sometimes able to get back at NNSVD-LRC: there is no clear winner (although on these 6 data sets, NNSVD-LRC finds in most cases the best final solution, in 8 out of the 18 cases –the second best being SVD-NMF with 6 out of 18). The MU have converged (close) to different stationary points and there is no guarantee in general that NNSVD-LRC will lead to better local solutions.

In summary, NNSVD-LRC is able to obtain better (and sparse) initial solutions compared to NNDSVD, SVD-NMF, CRNMF and SPKM, with comparable computational cost. It should therefore always be preferred if one wants to quickly obtain a good solution.

However, due to the complexity of NMF [27], if one wants to obtain a possibly better solution, it is recommended to use multiple initializations and keep the best solution obtained [5].

#### 4. Conclusion

In this paper, we presented a novel SVD-based NMF initialization. Our motivation was to address the shortcomings of previously proposed SVD-based NMF initializations. Our newly proposed method, referred to as nonnegative singular value decomposition with low-rank correction (NNSVD-LRC), has the following advantages

1. the initial error decreases as the factorization  $r$  increases,
2. the sparsity of the initial factors ( $W, H$ ) is close to 50%,
3. it is computationally cheaper as it only requires the computation of a truncated SVD of rank  $p = \lfloor \frac{r}{2} + 1 \rfloor$ , instead of  $r$ , and
4. it takes advantage of the discarded factors using highly efficient NMF iterations based on the low-rank approximation computed by the SVD.

In summary, NNSVD-LRC provides better initial NMF factors (both in terms of error and sparsity) at a low computational cost. This was confirmed on both dense and sparse real data sets. This allows NMF algorithms to converge faster to a stationary point, although there is no guarantee that this stationary point will have lower error than other initializations, as NMF is a difficult non-convex optimization problem [27].

### Acknowledgement

The authors thank the anonymous reviewers for their insightful comments which helped improve the paper. The financial support of HEC Pakistan is highly acknowledged for granting PhD scholarship to the first author. NG acknowledges the support of the European Research Council (ERC starting grant no. 679515).

### References

- [1] A. Berman, R.J. Plemmons, Nonnegative matrices in the mathematical sciences, SIAM, 9, 1994.
- [2] C. Boutsidis, E. Gallopoulos, SVD based initialization: a head start for nonnegative matrix factorization, Pattern Recognit. 41 (2008) 1350–1362.
- [3] R. Bro, E. Acar, T.G. Kolda, Resolving the sign ambiguity in the singular value decomposition, J. Chemom. 22 (2008) 135–140.
- [4] G. Casalino, N. Del Buono, C. Mencar, Subtractive clustering for seeding non-negative matrix factorizations, Inf. Sci. 257 (2014) 369–387.
- [5] A. Cichocki, R. Zdunek, A.H. Phan, S.I. Amari, Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-Way Data Analysis and Blind Source Separation, John Wiley & Sons, 2009.
- [6] R. Du, B. Drake, H. Park, Hybrid clustering based on content and connection structure using joint nonnegative matrix factorization, J. Global Optim. (2017).
- [7] M. Elad, M.A.T. Figueiredo, Y. Ma, On the role of sparse and redundant representations in image processing, Proc. IEEE 98 (2010) 972–982.
- [8] T. Ensari, Character recognition analysis with nonnegative matrix factorization, Int. J. Comput. 1 (2016) 219–222.
- [9] N. Gillis, The why and how of nonnegative matrix factorization, Regularization, Optimization, Kernels, and Support Vector Machines (2014) 257–291.
- [10] N. Gillis, F. Glineur, Using underapproximations for sparse nonnegative matrix factorization, Pattern Recognit. 43 (2010) 1676–1687.
- [11] N. Gillis, F. Glineur, Accelerated multiplicative updates and hierarchical ALS algorithms for nonnegative matrix factorization, Neural Comput. 24 (2012) 1085–1105.
- [12] G.H. Golub, C.F. Van Loan, Matrix computations, Johns Hopkins Studies in the Mathematical Sciences, 4th, The Johns Hopkins University Press, Baltimore, 2013. OCLC: ocn824733531
- [13] A. Janecek, Y. Tan, Using population based algorithms for initializing nonnegative matrix factorization, in: Y. Tan, Y. Shi, Y. Chai, G. Wang (Eds.), Advances in Swarm Intelligence, volume 6729, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 307–316.
- [14] D. Kitamura, N. Ono, Efficient initialization for nonnegative matrix factorization based on nonnegative independent component analysis, Acoustic Signal Enhancement (IWAENC), 2016 IEEE International Workshop on, IEEE, 2016, pp. 1–5.
- [15] A.N. Langville, C.D. Meyer, R. Albright, J. Cox, D. Duling, Initializations for the nonnegative matrix factorization, in: Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Citeseer, 2006, pp. 23–26.
- [16] D.D. Lee, H.S. Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (1999) 788.
- [17] D.D. Lee, H.S. Seung, Algorithms for non-negative matrix factorization, in: T.K. Leen, T.G. Dietterich, V. Tresp (Eds.), Advances in Neural Information Processing Systems 13, MIT Press, 2001, pp. 556–562.
- [18] Z. Liu, V.Y. Tan, Rank-one nmf-based initialization for nmf and relative error bounds under a geometric assumption, IEEE Trans. Signal Process. 65 (2017) 4717–4731.
- [19] R. Luce, P. Hildebrandt, U. Kuhlmann, J. Liesen, Using separable nonnegative matrix factorization techniques for the analysis of time-resolved raman spectra, Appl. Spectrosc. 70 (2016) 1464–1475.
- [20] R. Maruyama, K. Maeda, H. Moroda, I. Kato, M. Inoue, H. Miyakawa, T. Aonishi, Detecting cells using non-negative matrix factorization on calcium imaging data, Neural Netw. 55 (2014) 11–19.
- [21] S.J. Prajapati, K.R. Jadhav, Brain tumor detection by various image segmentation techniques with introduction to non negative matrix factorization, Brain 4 (2015) 600–603.
- [22] H. Qiao, New SVD based initialization strategy for non-negative matrix factorization, Pattern Recognit. Lett. 63 (2015) 71–77.
- [23] M. Rezaei, R. Boostani, M. Rezaei, An efficient initialization method for non-negative matrix factorization, J. Appl. Sci. 11 (2011) 354–359.
- [24] N. Sauwen, M. Acou, H.N. Bharath, D.M. Sima, J. Veraart, F. Maes, U. Himmelreich, E. Achten, S. Van Huffel, The successive projection algorithm as an initialization method for brain tumor segmentation using non-negative matrix factorization, PLoS ONE 12 (2017) e0180268.
- [25] M. Shiga, K. Tatsumi, S. Muto, K. Tsuda, Y. Yamamoto, T. Mori, T. Tanji, Sparse modeling of EELS and EDX spectral imaging data by nonnegative matrix factorization, Ultramicroscopy 170 (2016) 43–59.
- [26] F. Sun, M. Xu, X. Hu, X. Jiang, Graph regularized and sparse nonnegative matrix factorization with hard constraints for data representation, Neurocomputing 173 (2016) 233–244.
- [27] S.A. Vavasis, On the complexity of nonnegative matrix factorization, SIAM J. Optim. 20 (2010) 1364–1377.
- [28] X. Wang, X. Xie, L. Lu, An effective initialization for orthogonal nonnegative matrix factorization, J. Comput. Math. 30 (2012) 34–46.
- [29] S. Wild, J. Curry, A. Dougherty, Improving non-negative matrix factorizations through structured initialization, Pattern Recognit. 37 (2004) 2217–2232.
- [30] M. Ye, Y. Qian, J. Zhou, Multitask sparse nonnegative matrix factorization for joint spectral-spatial hyperspectral imagery denoising, IEEE Trans. Geosci. Remote Sens. 53 (2015) 2621–2639.
- [31] K. Yoshii, K. Itoyama, M. Goto, Student's T Nonnegative Matrix Factorization and Positive Semidefinite Tensor Factorization for Single-Channel Audio Source Separation, IEEE, 2016. Pp. 51–55
- [32] S. Zhong, J. Ghosh, Generative model-based document clustering: a comparative study, Knowl. Inf. Syst. 8 (3) (2005) 374–384.
- [33] G. Zhou, A. Cichocki, S. Xie, Fast nonnegative matrix/tensor factorization based on low-rank approximation, IEEE Trans. Signal Process. 60 (2012) 2928–2940.