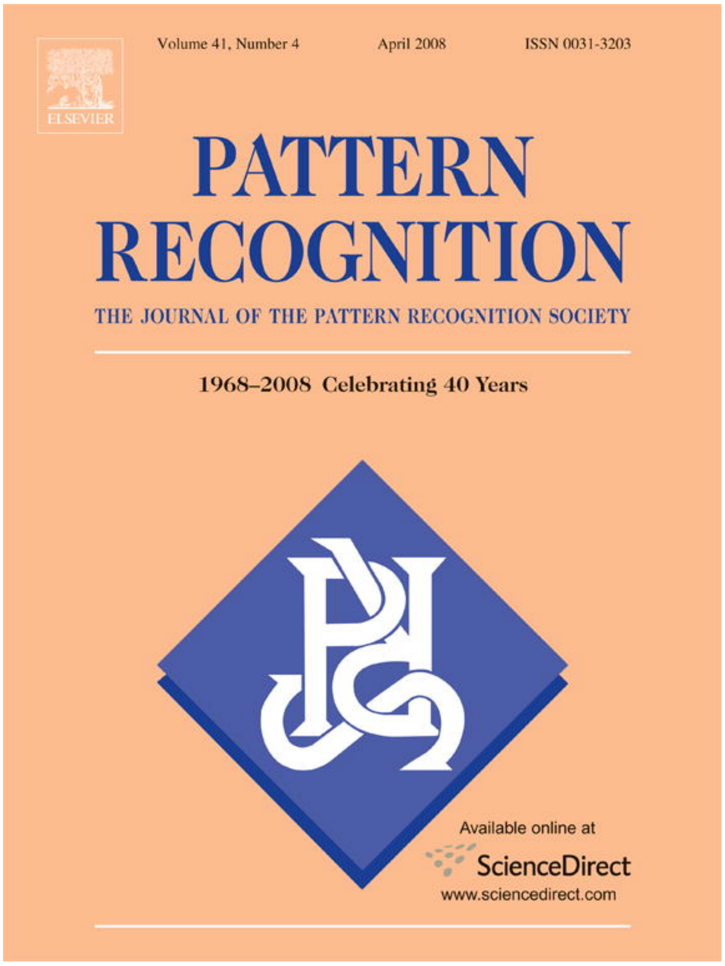
Volume 41, Number 4
April 2008
ISSN 0031-3203
ELSEVIER
PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
1968–2008 Celebrating 40 Years
Available online at
ScienceDirect
www.sciencedirect.com

Available online at www.sciencedirect.com
ScienceDirect

Pattern Recognition 41 (2008) 1350–1362

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/pr

SVD based initialization: A head start for nonnegative matrix factorization

C. Boutsidis[a,*], E. Gallopoulos[b]

[a]*Computer Science Department, Rensselaer Polytechnic Institute, Troy, NY 12180, USA*
[b]*Computer Engineering and Informatics Department, University of Patras, GR-26500 Patras, Greece*

Received 9 February 2007; received in revised form 3 August 2007; accepted 20 September 2007

Abstract

We describe Nonnegative Double Singular Value Decomposition (NNDSVD), a new method designed to enhance the initialization stage of nonnegative matrix factorization (NMF). NNDSVD can readily be combined with existing NMF algorithms. The basic algorithm contains no randomization and is based on two SVD processes, one approximating the data matrix, the other approximating positive sections of the resulting partial SVD factors utilizing an algebraic property of unit rank matrices. Simple practical variants for NMF with dense factors are described. NNDSVD is also well suited to initialize NMF algorithms with sparse factors. Many numerical examples suggest that NNDSVD leads to rapid reduction of the approximation error of many NMF algorithms.
© 2007 Elsevier Ltd. All rights reserved.

Keywords: NMF; Sparse NMF; SVD; Nonnegative matrix factorization; Singular value decomposition; Perron–Frobenius; Low rank; Structured initialization; Sparse factorization

1. Introduction

Nonnegative matrix factorization (NMF), that is the approximation[1] of a (usually nonnegative) matrix, $A \in \mathbb{R}^{m\times n}$, as a product of nonnegative factors, say $W \in \mathbb{R}_+^{m\times k}$ and $H \in \mathbb{R}_+^{k\times n}$, for some selected k, has become a useful tool in a large variety of applications, and the scientific literature and software tools on the subject and variants thereof are rapidly expanding; see e.g. Refs. [1–17] and description of specific software packages in Refs. [18,19]. As usual, we denote by $A \geqslant B$ the componentwise inequality $\alpha_{i,j} \geqslant \beta_{i,j}$ for all elements of (equisized matrices) A, B. For convenience, following Ref. [20] we denote by $\mathbb{R}_+^{m\times n}$ the set of all $m \times n$ nonnegative matrices. The motivation behind NMF is that besides the dimensionality reduction sought in many applications, the underlying data ensemble is nonnegative and can be better modeled and interpreted by means of nonnegative factors. For instance, image collections are frequently stored as matrices of nonnegative elements, where each column encodes one image of the collection. The application of NMF, then, would produce a dictionary of k basis images as columns of factor W, and the nonnegative coefficients for the linear combination of these images that reconstructs an approximation of the originals in factor H. In text mining under the vector space model, document collections are stored as term-document matrices of nonnegative elements, each matrix column encoding one document. Each column of W corresponds to a basic document, and the resulting k documents can be additively combined using coefficients from H to reconstruct an approximation of the document collection as well as for other applications profiting from dimensional reduction such as clustering [8,15,16]. In the above cases, NMF provides a framework for learning parts of images and semantic features of text. Another area of application is space situational alertness, in which data collections consist of spectral reflectance data of a space object containing essential information regarding the materials composing it. Each column of the matrix represents one such spectral measurement. After the NMF, factor W contains spectral signatures that aid in detecting the type of constituent materials for the space object, and H contains coefficients that help in the computation of the

* Corresponding author. Tel.: +1 518 276 3080; fax: +1 518 276 4033.
E-mail addresses: boutsc@cs.rpi.edu (C. Boutsidis), stratis@ceid.upatras.gr (E. Gallopoulos).

[1] We would be referring to NMF for the general approximation problem, even though an acronym such as NMA might be appropriate [1].

0031-3203/$30.00 © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2007.09.010

proportional amounts in which these materials appear in the object [21]. In all cases, the additive nature of the factorization has been proposed as an important aid in interpretation.

Our target NMF problem is as follows:

> Given $A \in \mathbb{R}_+^{m\times n}$, and natural number $k < \min(m, n)$, compute $W \in \mathbb{R}_+^{m\times k}$ and $H \in \mathbb{R}_+^{k\times n}$ that solve $\min_{W\geqslant 0, H\geqslant 0}\phi(A, WH)$, where $\phi(A, WH) : \mathbb{R}^{m\times n} \times \mathbb{R}^{m\times n} \to \mathbb{R}_+$ is some suitable distance metric.

In the NMF algorithms combined with the initialization proposed in this paper, the distance metrics will be the Frobenius norm $\|A - WH\|_F$, or modifications thereof, or the generalized Kullback–Leibler divergence $D(A\|WH)$. All are used extensively in the literature, and are directly related to more general metrics [1].

As defined, the NMF problem is a more general instance of the case where we demand factors whose product exactly equals the matrix (ignoring roundoff, as we will do throughout this paper). Such nonnegative decompositions were a central topic already in early treatments of nonnegative matrices (see e.g. Ref. [20]). In general, there is no guarantee that an exact nonnegative factorization exists for arbitrary k. It is known, however, that if $A\geqslant 0$, then there exists a natural number (called nonnegative rank) and nonnegative W and H having that number as rank so that $A = WH$ holds exactly (cf. Ref. [22].) Furthermore, NMF is a nonconvex optimization problem with inequality constraints and iterative methods become necessary for its solution (see e.g. Refs. [23,24]). Unfortunately, current NMF algorithms typically converge slowly and then at local minima.

There exist many algorithms for approximating matrices using nonnegative factors (Ref. [11] plays pivotal role, cf. the survey [21]). Popular and used by many researchers as basis for further developments are two algorithms proposed in Ref. [25]. These rely on an iterative multiplicative or additive correction of initial guesses for the pair of factors (W, H). The algorithms were proven to converge monotonically and can be interpreted as diagonally rescaled gradient descent methods; cf. Refs. [2,26]. Another important issue addressed by researchers is the incorporation of further constraints appropriate for the problem; see e.g. [6,7,13,15]. One generic constraint is sparsity: It is frequently important to minimize the number of features used in reconstruction, e.g. in sparse linear representation, in the case of signal processing [6,7,27–29]. Ref. [7], for example, uses a sparsity metric and describes an algorithm to satisfy it in the factors.

Due to the iterative nature of all NMF algorithms, the initialization of the pair of factors (W, H) is cited in the literature as an important component in the design of successful NMF methods. In this paper we focus on this issue. With few exceptions (see Refs. [24,30,31]) most NMF algorithms in the literature use random nonnegative initialization for (W, H). Iterates converge to a local minimum, so it becomes necessary to run several instances of the algorithm using different random initializations and then select the best solution. Because NMF can be viewed as a bound optimization problem, it is also likely to suffer from slow convergence [24]. Therefore, the overall process can become quite expensive. As a step toward the development of overall faster algorithms for NMF, we propose a novel initialization strategy that is based on singular value decomposition (SVD) and has the following features: (i) it can be readily combined with all available NMF algorithms; (ii) in its basic form, contains no randomization and therefore converges to the same solution for any given algorithm; (iii) it rapidly provides an approximation with error almost as good as that obtained via the deployment of alternative initialization schemes when these run to convergence. We call the proposed initialization strategy NNDSVD (Nonnegative Double Singular Value Decomposition) to underline the fact that it is based on two SVD processes: One to create the rank-k approximation, followed by a "small" SVD on each of the positive sections of each of the factors. Because of property (iii) it is expected to be especially useful whenever the application constrains the maximum time interval for result delivery. NNDSVD depends on an interesting property (Lemma 1 and Theorem 2) concerning the behavior of unit rank matrices; to the best of our knowledge, this fact remained unnoticed until now, and could have interesting applications in other domains. We show one family of nonnegative matrices for which NNDSVD returns an exact NMF (Proposition 7). The basic algorithm also admits an interesting modification, we name 2-step NNDSVD, that has the potential to provide initialization at an even lower cost.

In the sequel, given any vector or matrix variable X, its "positive section", $X_+\geqslant 0$, will be defined to be the vector or matrix of same size that contains the same values as X there where X has nonnegative elements and 0 elsewhere. The "negative section" of X will be the matrix $X_- = X_+ - X$, where again $X_-\geqslant 0$. It follows immediately that any vector or matrix can be written as $X = X_+ - X_-$, and if $X\geqslant 0$ then $X_- = 0$. MATLAB-like notation is followed when necessary. The notions "positive" and "negative", of course, are slight misnomers, since we are really referring to nonnegative and absolute value of nonpositive values respectively. We prefer them, however, and let the handling of zero values be made clear by context. When we seek sparse factors we will refer to "sparse NMF". We occasionally refer to "dense NMF" when we need to underline that we do not seek sparsity. The paper is organized as follows: Section 2 discusses initialization in the context of related work and describes the properties of NNDSVD. Section 3 illustrates the use and performance of the method in a variety of cases.

2. SVD-based initializations

Most research papers to date discussing NMF algorithms mention the need to investigate good initialization strategies (see e.g. Ref. [21]) but, in the absence of any additional information about the problem, initialize the elements of the pair (W, H) with nonnegative random values. In some cases, only one of the factors (e.g. W) is initialized (as random) while the other is chosen to satisfy certain constraints, possibly obtained after solving an optimization problem using the initial values for the former. In the sequel, we would be referring to "initialization of (W, H)" to mean initialization of either or both factors, but would be more specific whenever necessary.

Because of the nature of the underlying optimization problem, repeated runs of any of these algorithms with different initializations will be necessary and will lead to different answers. Before proceeding, we need to clarify what we mean by "good initialization strategy". Two possible answers are: (i) one that leads to rapid error reduction and faster convergence; (ii) one that leads to better overall error at convergence. We concentrate on the former objective while noting that a satisfactory answer to the latter remains elusive.

Because NMF is a constrained low rank matrix approximation, we seek the initialization strategy amongst alternative low rank factorization schemes. Indeed, one of the few published algorithms for nonrandom initialization (see Refs. [30,31]), relies on a method (see Ref. [32]) that provides low rank approximation via clustering. Specifically, spherical k-means (`Skmeans`) is used to partition the columns of A into k clusters, selecting the normalized centroid representative vector (named "concept vector") for each cluster and using that vector to initialize the corresponding column of W. Depending on the NMF algorithm used subsequently, H can either be random or be computed as $\arg\min_{H\geqslant 0}\|A-WH\|_F$. It was shown in Refs. [30,31] that only few iterations of this clustering algorithm are sufficient and that the scheme, we call CENTROID in the sequel, leads to faster error reduction than random initialization. Specifically, numerical experiments in Ref. [31] showed that the method, at some overhead for the clustering phase, can save several expensive NMF update steps.

In our quest that eventually led to the framework proposed in this paper, we first explored initializations inspired by the aforementioned original ideas of Refs. [30,31] for structured initialization. Specifically, we deployed an SVD analogue (cf. Ref. [33]) to the low rank approximation methods used in CENTROID. We first clustered the columns of A into k groups and then initialized (W, H) using nonnegative left and right singular vectors corresponding to the maximum singular value of each group. Their existence is guaranteed by Perron–Frobenius theory; see also Ref. [34]. Results were mixed: sometimes the SVD approach would outperform CENTROID, sometimes not. We were thus motivated to consider alternative approaches.

2.1. NNDSVD initialization

We next present a method for initialization that turns out to be quite effective. We start from the basic property of the SVD, by which, every matrix $A\in\mathbb{R}^{m\times n}$ of rank $r\leqslant\min(m,n)$ can be expressed as the sum of r leading singular factors $A=\sum_{j=1}^{r}\sigma_j u_j v_j^\top$, where $\sigma_1\geqslant\cdots\geqslant\sigma_r>0$ are the nonzero singular values of A and $\{u_j, v_j\}_{j=1}^{r}$ the corresponding left and right singular vectors. Then, for every $k\leqslant r$, the optimal rank-k approximation of A with respect to the Frobenius norm, say $A^{(k)}$, is readily available from the sum of the first k factors (cf. Ref. [35, Schmidt and Eckart–Young theory]), that is

$$A^{(k)} := \sum_{j=1}^{k}\sigma_j C^{(j)} = \arg\min_{\mathrm{rank}(G)\leqslant k}\|A-G\|, \tag{1}$$

where $C^{(j)}=u_j v_j^\top$. We assume, from now on, that A is nonnegative. Our approach uses a modification of expansion (1) that will produce a nonnegative approximation of A and provide, in the same time, effective initial values for (W, H). In particular, every unit rank matrix $C^{(j)}$ is approximated by its nonnegative section $C_+^{(j)}$; subsequently, (W, H) are initialized from selected singular triplets of $C_+^{(j)}$. The factors $C_+^{(j)}$ possess special properties that play a key role in our algorithm. As we will show:

- Their rank is at most 2 because of the "set to zero with small rank increment" property (Lemma 1).
- They are the best nonnegative approximations of $C^{(j)}$ in terms of the Frobenius norm (cf. Lemma 5).
- There exist corresponding singular vectors that are nonnegative and are readily available from the singular triplets $\{\sigma_j, u_j, v_j\}$ of A (cf. Theorem 2).

In summary, NNDSVD can be described as follows: (i) Compute k leading singular triplets of A; (ii) form the unit rank matrices $\{C^{(j)}\}_{j=1}^{k}$ obtained from singular vector pairs; (iii) extract their positive section and respective singular triplet information; (iv) use them to initialize (W, H). The two SVD's, in steps (i) and (iii), motivated the naming of NNDSVD. On the other hand, we will show that because of special properties of $C_+^{(j)}$, Steps (ii) and (iii) can be implemented at very low cost. Using the notation introduced thus far, we show the Lemma that is central in our discussion.

Lemma 1. *Consider any matrix $C\in\mathbb{R}^{m\times n}$ such that* $\mathrm{rank}(C)=1$, *and write $C=C_+-C_-$. Then* $\mathrm{rank}(C_+)$, $\mathrm{rank}(C_-)\leqslant 2$.

Proof. From the rank assumption we can write $C=xy^\top=(x_+-x_-)(y_+-y_-)^\top=(x_+y_+^\top+x_-y_-^\top)-(x_+y_-^\top+x_-y_+^\top)$. All factors are nonnegative; moreover, for each x, y, the nonzero values of the positive section are situated at locations that are complementary than the nonzeros of the corresponding negative section. Consequently, each nonzero element of C is obtained from exactly one term from the terms on the right. Therefore, $C_+=x_+y_+^\top+x_-y_-^\top$ and $C_-=x_+y_-^\top+x_-y_+^\top$ and the rank of each is at most 2. It is worth noting, as an alternate algebraic proof, that we can also write $C=XTY$, where (in MATLAB notation) $X:=[x_+, x_-]$, $Y=[y_+^\top; y_-^\top]$ and $T=[1,-1;-1,1]$. Note that T is unit rank. The expression $C=C_+-C_-$ amounts to decomposing $T=I-J$, where $J=[0,1;1,0]$ so that $C=XY-XJY$ and $C_+=XY$, $C_-=XJY$, each of which has rank at most 2. $\square$

The result, albeit simple, is quite remarkable: It tells us that if we zero out all negative values of a unit rank matrix, the resulting matrix will have rank 2 at most. We thus call the above "set to zero with small rank increment" property. It is worth noting and easy to verify that matrices of $\mathrm{rank}(C)>1$ do not share a similar property. For example consider the matrix

$C = XY^\top$, where $X, Y \in \mathbb{R}^{6\times 2}$ are

$$X = \begin{pmatrix} 1 & 2 \\ -3 & -4 \\ 2 & 3 \\ -4 & -5 \\ -5 & 6 \\ 6 & -7 \end{pmatrix} \quad \text{and} \quad Y = \begin{pmatrix} 2 & 1 \\ 3 & -2 \\ 4 & 4 \\ -5 & -5 \\ 6 & 6 \\ -6 & 7 \end{pmatrix}.$$

Although $\operatorname{rank}(C) = 2$, $\operatorname{rank}(C_+) = 5$. Also $\operatorname{rank}(C_-) = 5$.

Because C_+ is nonnegative, its maximum left and right singular vectors will also be nonnegative from Perron–Frobenius theory. The next theorem says that the remaining, trailing, singular vectors are also nonnegative. Furthermore, because of the special structure of C_+, its singular value expansion is readily available.

Theorem 2. *Let $C \in \mathbb{R}^{m\times n}$ have unit rank, so that $C = xy^\top$ for some $x \in \mathbb{R}^m$, $y \in \mathbb{R}^n$. Let also $\hat{x}_\pm := x_\pm/\|x_\pm\|$, $\hat{y}_\pm := y_\pm/\|y_\pm\|$ be the normalized positive and negative sections of x and y, and $\mu_\pm = \|x_\pm\|\|y_\pm\|$ and $\xi_\pm = \|x_\pm\|\|y_\mp\|$. Then the unordered singular value expansions of C_+ and C_- are*

$$C_+ = \mu_+\hat{x}_+\hat{y}_+^\top + \mu_-\hat{x}_-\hat{y}_-^\top \quad \text{and}$$
$$C_- = \xi_+\hat{x}_+\hat{y}_-^\top + \xi_-\hat{x}_-\hat{y}_+^\top. \tag{2}$$

The maximum singular triplet of C_+ is $(\mu_+, \hat{x}_+, \hat{y}_+)$ if $\mu_+ = \max(\|x_+\|\|y_+\|, \|x_-\|\|y_-\|)$, otherwise it is $(\mu_-, \hat{x}_-, \hat{y}_-)$. Similarly, the maximum singular triplet of C_- is $(\xi_+, \hat{x}_+, \hat{y}_-)$ if $\xi_+ = \max(\|x_+\|\|y_-\|, \|x_-\|\|y_+\|)$ else it is $(\xi_-, \hat{x}_-, \hat{y}_+)$.

Proof. By construction, each pair of vectors $x_\pm$ and $y_\pm$ have their nonzero values at complementary locations, therefore $x_-^\top x_+ = 0$ and $y_-^\top y_+ = 0$ and each of the matrices $X := [\hat{x}_+, \hat{x}_-]$ and $Y := [\hat{y}_+, \hat{y}_-]$ is orthogonal. Terms $\mu_\pm$ are nonnegative, therefore the result follows by the uniqueness of the singular value expansion. Similarly for the decomposition of C_-. □

The above result establishes nonnegativity for all singular vectors corresponding to nontrivial singular values of $C_\pm$. There is an immediate connection of the decompositions introduced in Theorem 2 with the concept of nonnegative rank, already mentioned in the Section 1.

Definition 3 (*Gregory and Pullman [22]*). The nonnegative rank, $\operatorname{rank}_+(\mathrm{A})$, of $A \in \mathbb{R}_+^{m\times n}$ is the smallest number of nonnegative unit rank matrices into which a matrix can be decomposed additively.

Nonnegative rank is difficult to compute (see e.g. Ref. [36]). It generally holds, however, that $\operatorname{rank}(A) \leqslant \operatorname{rank}_+(\mathrm{A}) \leqslant \min(m, n)$ (cf. Ref. [22]). As shown in Ref. [36], when $\operatorname{rank}(\mathrm{A}) \leqslant 2$, then $\operatorname{rank}_+(A) = \operatorname{rank}(\mathrm{A})$. Combining with our previous results, we can provide precise estimates regarding the nonnegative ranks of $C_\pm$.

Corollary 4. (i) $\operatorname{rank}_+(C_\pm) \leqslant 2$. (ii) $\operatorname{rank}_+(C_\pm) = \operatorname{rank}(C_\pm)$. (iii) *If C contains both positive and negative elements, then* $\operatorname{rank}_+(C_\pm) = 2$. (iv) *If $C \geqslant 0$ (resp. $C \leqslant 0$) then* $\operatorname{rank}_+(C_+) = 1$ (*resp.* $\operatorname{rank}_+(C_-) = 1$).

Table 1
NNDSVD initialization of nonnegative matrix, in MATLAB notation

```
Inputs: Matrix $A \in \mathbb{R}_+^{m\times n}$, integer $k < \min(m, n)$.
Output: Rank-$k$ nonnegative factors $W \in \mathbb{R}_+^{m\times k}$, $H \in \mathbb{R}_+^{k\times n}$.
1. Compute the largest $k$ singular triplets of $A$: $[U, S, V] = \mathtt{psvd}(A, k)$
2. Initialize $W(:,1) = \mathtt{sqrt}(S(1,1)) * U(:,1)$ and $H(1,:) = \mathtt{sqrt}(S(1,1)) * V(:,1)'$
for $j = 2 : k$
    3. $x = U(:, j)$; $y = V(:, j)$;
    4. $xp = \mathtt{pos}(x)$; $xn = \mathtt{neg}(x)$; $yp = \mathtt{pos}(y)$; $yn = \mathtt{neg}(y)$;
    5. $xpnrm = \mathtt{norm}(xp)$; $ypnrm = \mathtt{norm}(yp)$; $mp = xpnrm * ypnrm$;
    6. $xnnrm = \mathtt{norm}(xn)$; $ynnrm = \mathtt{norm}(yn)$; $mn = xnnrm * ynnrm$;
    7. if $mp > mn$, $u = xp/xpnrm$; $v = yp/ypnrm$; $sigma = mp$;
       else $u = xn/xnnrm$; $v = yn/ynnrm$; $sigma = mn$; end
    8. $W(:, j) = \mathtt{sqrt}(S(j, j) * sigma) * u$ and $H(j,:) = \mathtt{sqrt}(S(j, j) * sigma) * v'$;
end
```

The call $\mathtt{psvd}(A, k)$ computes the k leading singular triplets of A, e.g. MATLAB's svds. Functions pos and neg extract the positive and negative sections of their argument: $[A_p] = \mathtt{pos}(A)$ returns $A_p = (A >= 0).*A$; and $[A_n] = \mathtt{neg}(A)$ returns $(A < 0).*(-A)$.

Parts (i), (iii) and (iv) follow directly from the unit rank assumption for C and Theorem 2. Part (ii) follows from the aforementioned result in Ref. [36]. Theorem 2, however, provides an explicit construction for the decomposition and suggests a cheap way to compute it. The next lemma is a straightforward consequence of the definition of the Frobenius norm.

Lemma 5. *Let $C \in \mathbb{R}^{m\times n}$. Then $C_+ = \arg\min_{G\in\mathbb{R}_+^{m\times n}} \|C - G\|_F$.*

Therefore, the best (in terms of the Frobenius norm) nonnegative approximation of each unit rank term $C^{(j)} = u^{(j)}(v^{(j)})^\top$ would be the corresponding $C_+^{(j)}$.

The preceding results constitute the theoretical foundation of NNDSVD. Based on these, the method can be implemented as in Table 1. Note that the method first approximates each of the first k terms in the unit rank singular factor expansion of A by means of their positive sections. These new factors have rank at most 2. Then, each of these factors is approximated by its maximum singular triplet which is then used to initialize (W, H). Note that Step 3 is applied from $j = 2$ onwards since the leading singular triplet is nonnegative and can be readily used to initialize the first column (resp. row) of W (resp. H).

2.2. *NNDSVD approximation*

From the preceding results, it becomes possible bound the error corresponding to the initial factors (W, H) obtained by NNDSVD, specifically, the Frobenius norm of the residual, $R = A - WH$. Denote by $\{\sigma_j\}_{j=1}^r$ the nonzero singular values of A in nonincreasing order and by $\{\sigma_j(C_+), x_j(C_+), y_j(C_+)\}$ the singular triplets of C_+. From Lemma 1, $\operatorname{rank}(C_+) \leqslant 2$, therefore there are only two nontrivial triplets that we index, as usual by $j = 1, 2$. We write, $A = A^{(k)} + E^{(k)}$, where $E^{(k)} := \sum_{j=k+1}^{r} \sigma_j u_j v_j^\top$, therefore

$$A^{(k)} = \sigma_1 C^{(1)} + \sum_{j=2}^{k} \sigma_j C^{(j)} = \sigma_1 C^{(1)} + \sum_{j=2}^{k} \sigma_j C_+^{(j)} - \sum_{j=2}^{k} \sigma_j C_-^{(j)}$$
$$= \sigma_1 C^{(1)} + \sum_{j=2}^{k} \sigma_j \sigma_1(C_+^{(j)}) x_1(C_+^{(j)})(y_1(C_+^{(j)}))^\top + \hat{E},$$

where

$$\hat{E} := \sum_{j=2}^{k} \sigma_j \sigma_2(C_+^{(j)}) x_2(C_+^{(j)})(y_2(C_+^{(j)}))^\top - \sum_{j=2}^{k} \sigma_j C_-^{(j)}.$$

The NNDSVD algorithm (Table 1) selects (W, H) so that

$$WH = \sigma_1 C^{(1)} + \sum_{j=2}^{k} \sigma_j \sigma_1(C_+^{(j)}) x_1(C_+^{(j)})(y_1(C_+^{(j)}))^\top$$
$$= A^{(k)} - \hat{E}.$$

Therefore, $A - WH = E^{(k)} + \hat{E}$ and thus

$$\|E^{(k)}\|_F \leqslant \|R\|_F \leqslant \|E^{(k)}\|_F + \|\hat{E}\|_F, \tag{3}$$

so that $\|\hat{E}\|_F$ measures the deviation from the optimal unconstrained approximation (Eq. (1)). Now, for each j, $\|x_1(C_+^{(j)})(y_1(C_+^{(j)}))^\top\|_F = 1$ since $x_1(C_+^{(j)})$, $y_1(C_+^{(j)})$ are singular vectors and have unit length. Furthermore,

$$\|C_+^{(j)}\|_F^2 + \|C_-^{(j)}\|_F^2 = \|C^{(j)}\|_F^2 = 1,$$

therefore both $\|C_\pm^{(j)}\|_F \leqslant 1$. These lead to the following result:

Proposition 6. *Given $A \in \mathbb{R}_+^{m\times n}$, and the pair (W, H) initialized by* NNDSVD, *then the Frobenius norm of $R = A - WH$ is bounded as follows*:

$$\|E^{(k)}\|_F \leqslant \|R\|_F \leqslant \|E^{(k)}\|_F + \|\hat{E}\|_F, \tag{4}$$

where

$$\|\hat{E}\|_F \leqslant \sum_{j=2}^{k} (\sigma_2(C_+^{(j)}) + 1)\sigma_j \leqslant 2\sum\nolimits_{j=2}^{k} \sigma_j. \tag{5}$$

Even though the upper bound is very loose (e.g. it may become larger than the trivial upper bound $\|A\|_F$ obtained when (W, H) are initialized as all zero) it establishes that the residual is bounded. Of far greater interest is that, in practice, only few iterations are sufficient for NNDSVD to drive the initial residual down to a magnitude that is very close to the one we would have obtained had we applied the underlying NMF algorithm with random initialization but for many more iterations.

The above analysis helps us also bound the error in modified versions of NNDSVD that will be described in the next section. These rely on initializing using the pair (W_f, H_f), where $W_f := W + E_W$, $H_f := H + E_H$, (W, H) are as before and E_W, E_H are structured perturbations so that their nonzero elements occur at positions that are complementary to those of W and H, respectively. Also, $\max(\|E_H\|_F, \|E_W\|_F) \leqslant \varepsilon$. Then, because all columns of W and rows of H have unit length,

$$\|A - W_f H_f\|_F = \|A - WH - WE_H - E_W H - E_W E_H\|_F$$
$$\leqslant \|A - WH\|_F + \varepsilon(\|W\|_F + \|H\|_F)$$
$$= \|E\|_F + 2\varepsilon\sqrt{k}. \tag{6}$$

Note that the first term on the right side was bounded in Proposition 6.

We finally show that there are matrices for which NNDSVD is able to return their exact decomposition into nonnegative unit rank factors. To do this, we consider matrices that admit the orthogonal nonnegative factorization described in Refs. [3,37]. In this category belong, for instance, block diagonal matrices where each diagonal block is unit rank and generated by nonnegative vectors.

Proposition 7. *Let $A = W_A D H_A \in \mathbb{R}_+^{m\times n}$, where $W_A \in \mathbb{R}_+^{m\times k}$, $H_A \in \mathbb{R}_+^{k\times n}$, $D \in \mathbb{R}^{k\times k}$ are nonnegative and D diagonal. Let also W_A, H_A be orthogonal, so that $W_A^\top W_A = H_A H_A^\top = I$. Then, if* NNDSVD *is applied to compute rank-$\tilde{k}$ factors, it initializes*: (i) *with the exact values, $W = W_A D^{1/2}$, $H = D^{1/2} H_A$ when $\tilde{k} = k$*; (ii) *with the pair (W, H) that returns the minimum error in Frobenius norm, when $\tilde{k} < k$.*

Proof. By construction, $A = W_A D H_A$ is the (compact) SVD of A. If this is written as sum of k, rank-1 terms, each resulting from the product of the corresponding column of W_A, row of H_A, and diagonal term of D, then NNDSVD will compute the elements exactly since all terms are nonnegative, so their positive sections are identical to the terms themselves. The result for $\tilde{k} < k$ trivially follows by the optimal approximation property (cf.1) of partial SVD with respect to Frobenius norm. □

2.3. Dense variants: NNDSVDa and NNDSVDar

As described, one feature of NNDSVD is that it obtains initial columns and rows for (W, H) from the leading singular vectors of the positive section of each one of the first k singular factors of A. All, except the maximum singular vectors are likely to contain positive as well as negative elements. Therefore, the initial (W, H) are likely to contain a number of zeros commensurate to the latter. In some cases, e.g. when we seek sparse NMF (cf. Ref. [7], discussion in Section 3 and Fig. 9), this is desirable, especially in view of the fact that some NMF algorithms retain the same sparsity in the iterates that was present in the initial (W, H). In the dense case, however, a large number of zeros may become undesirable, as will be illustrated when we compare the performance of all methods (Fig. 10). In particular, it will be seen that in those cases, even though the basic algorithm initially provides rapid error reductions, eventually leads to worse error than RANDOM. It was worth noting that such a behavior was also observed for some algorithms described in Ref. [31]. To address this problem, we deploy two slightly modified variants of the basic algorithm. In these, we perturb the zero values in the original (W, H); in particular,

Table 2
Nonnegative unit rank approximation of arbitrary matrix [27]

```
Input: Matrix $C \in \mathbb{R}^{m\times n}$
Output: Nonnegative $g \in \mathbb{R}^m_+, h \in \mathbb{R}^m_+$ so that $C \approx gh^\top$
1. Compute the largest singular triplet of $C$: $[\sigma, u, v]$
2. Set $g = u_+, h = \sigma v_+$ where $u_+, v_+$ are the nonnegative sections of $u, v$;
for $j = 1, \ldots,$
    3. Compute $g = Ch/h^\top h$ and set $g = g_+$;
    4. Compute $h = g^\top C/g^\top g$ and set $h = h_+$;
end
```

variant NNDSVDa sets all zeros equal to the average of all elements of A; we denote this by $\mathtt{mean}(A)$.[2] Variant NNDSVDar sets each zero element equal to a random value chosen from a uniform distribution in $[0, \mathtt{mean}(A)/100]$. Both variants incur no appreciable overhead on the basic initialization and lead to error bounds such as in Eq. (6). Moreover, the user has control over the number of zero elements that are perturbed and can exercise this judiciously to satisfy $\phi(A, WH) \approx \phi(A, W_f H_f)$.

2.4. Discussion and extensions

We next discuss and evaluate a seemingly similar initialization option that comes to mind naturally and is also SVD-based. This would be to set to zero the positions with negative values in each $\{u_j, v_j\}$ pair of the singular value expansion of A and use multiples of these vectors to initialize (W, H). This straightforward method can be interpreted using our framework: In particular, each $C^{(j)} = u_j v_j^\top$, therefore from Theorem 2, and dropping for simplicity the index j till the end of this paragraph, each one of the two addends on the right of $C_+ = u_+v_+^\top + u_-v_-^\top$ is a scalar multiple of a singular factor of C_+. We remind that $u = (u)_+ - (u)_-$ and $v = (v)_+ - (v)_-$. Therefore, this initialization is equivalent to approximating each C (actually $C_+^{(j)}$) by $u_+v_+^\top$, whereas NNDSVD picks this or $u_-v_-^\top$, depending on the magnitude of the corresponding $\mu_\pm$'s (cf. Theorem 2). We conclude that NNDSVD is preferable since it leads to equal or smaller error contribution from each term.

The aforementioned approach is also closely related to another iterative algorithm, discussed in Ref. [27], for the nonnegative, unit rank approximation of arbitrary matrices. The algorithm, tabulated in Table 2, initializes two vectors with the positive sections of the leading left and right singular vectors of the matrix and then iterates for a certain number of steps. We now show that when the input matrix is any one of the unit rank terms $C = uv^\top$ corresponding to $j > 1$ (actually $C^{(j)}$ and $j > 1$), the algorithm will make no progress, but will return as approximations the positive sections computed in Step 2. Then $g = u_+, h = v_+$ and in Step 3, $g = uv^\top h/h^\top h$, where $h = v_+$, since $\sigma(C) = 1$. It follows that

$$g = uv^\top v_+/v_+^\top v_+ = u(v_+ - v_-)^\top v_+/v_+^\top v_+ = u$$

because $v_-^\top v_+ = 0$.

[2] We deviate slightly from MATLAB notation, where we must use mean(mean(A)).

Therefore, the final value entered in g will be u_+, so there will be no change between steps. Similarly, the new value of h will be the original v_+. Therefore, the approximations returned when the above process is applied to an arbitrary unit rank matrix will be u_+ and v_+.

We finally sketch an extension of NNDSVD, we call *2-step* NNDSVD, that can be especially useful when it becomes difficult or expensive to compute all leading k singular triplets of A. From Theorem 2 we know that not only the maximum but also the trailing singular triplet, $(\sigma_2(C_+^{(j)}), x_2(C_+^{(j)}), y_2(C_+^{(j)}))$, has strictly nonnegative components. Therefore, NNDSVD could be modified as follows: For $j = 2, \ldots$ until all k columns and rows of (W, H) are filled, if the rank of $C_+(j)$ is 1, initialize column j of W and row j of H with scalar multiples of the maximum left and right singular vectors of $C_+^{(j)}$ as is done in the original algorithm. If, however, the rank is 2, then columns and rows $2j, 2j + 1$ of W and H are initialized with scalar multiples of $x_1(C_+^{(j)}), x_2(C_+^{(j)})$ and $y_1(C_+^{(j)})^\top, y_2(C_+^{(j)})^\top$, respectively. If, for example, k is odd and all $C^{(2)}, \ldots, C^{(k+1)/2}$ have rank-2, then these factors are enough to produce a nonnegative initialization for (W, H). Note that using this approach, all singular vectors generating $C_+^{(j)}$ participate in the initialization hence the reconstruction is exact. *2-step* NNDSVD leads to a different upper bound for the residual.

Corollary 8. *Given $A \in \mathbb{R}_+^{m\times n}$, and the pair (W, H) initialized as in 2-step* NNDSVD, *then*

$$\|E^{(k/2)}\|_F \leqslant \|R\|_F \leqslant \|E^{(k/2)}\|_F + \sum_{j=2}^{k/2} \sigma_j. \tag{7}$$

2.5. Beyond initialization

NNDSVD can be readily combined with any existing NMF algorithm. The ones selected for this paper are tabulated in Table 3. The first two (MM and AD) correspond to the additive and multiplicative updates proposed in Ref. [25]. MM uses

$$\begin{aligned} H &\leftarrow H. * ((W^\top A)./(W^\top WH)),\\ W &\leftarrow W. * ((AH^\top)./(WHH^\top)), \end{aligned} \tag{8}$$

where $.*$ and $./$ denote element by element multiplication and division, respectively. These updates do not increase the Frobenius norm of the residual $\|A - WH\|_F$. We refer to the literature for a full description of the design and update formulas for the remaining methods in Table 3.

We finally mention that when the data matrix is symmetric, we might be seeking symmetric (e.g. $A \approx HH^\top$) or weighted symmetric (e.g. $A = HZH^\top$) nonnegative factorizations; see e.g. Ref. [37]. Noting that matrix symmetry is inherited by the positive section, NNDSVD can be adapted to generate a symmetric initialization. Costs become lower because all necessary values are derived from the eigendecomposition rather than the SVD of A and only half the factors need be computed.

Table 3
NMF algorithms used in this paper

Name	Comments	Ref.	Distance metric $\phi(A, WH)$
MM	Multiplicative	[25]	$\phi(A, WH) = \|A - WH\|_F$
AD	Additive	[25]	$\phi(A, WH) = D(A\|\|WH)$
CNMF[a]	Multiplicative	[38]	$\phi(A, WH) = 0.5(\|A - WH\|_F^2 + \alpha\|W\|_F^2 + \beta\|H\|_F^2)$
GD-CLS [b]	Multipl./alternating least sq.	[28]	$\phi(A, WH) = \|A - WH\|_F$
`nmfsc`	Sparse NMF from `nmfpack`	[7]	$\phi(A, WH) = 0.5\|A - WH\|_F^2$

[a]Stands for constrained NMF.
[b]Stands for gradient descent with constrained least squares.

Table 4
Datasets for experiments

Name	Matrix size	Comments
CLASSIC3	4299×3891	As specified in Ref. [32] using TMG [44]
IRIS[a,e]	$10\,800 \times 9$	9 human eye iris images of size 90×120
CBCL[e]	361×2429	2429 faces of size 19×19 from Ref. [45]
USGS[b]	256×500	(Hyperspectral) 500 spectra measured at 256 wavelengths
NATURAL IMAGES[c,e]	$262\,144 \times 10$	10 images of natural scenes of size 512×512
SHUTTLE[d,e]	$16\,384 \times 16$	16 shuttle images of size 128×128

[a]Iris images from Ref. [46], also displayed in Fig. 4.
[b]Hyperspectral data collected from the U.S. Geological Survey (USGS) Digital Spectral Library. See also Refs. [38,21] regarding the spectral unmixing application.
[c]The dataset was described and used in Ref. [7].
[d]Images (kindly provided by Prof. R.J. Plemmons and taken at the U.S. Air Force Maui Space Center) are from the space shuttle Columbia on its tragic final orbit, before disintegration upon re-entry in February 2003. Three sample images are depicted in Fig. 1.
[e]Image datasets (SHUTTLE, IRIS, CBCL and NATURAL IMAGES) are vectorized so that each column corresponds to one image.

Table 5
Algorithm–dataset combinations and pointers to figures with results

	CLASSIC3	USGS	SHUTTLE	CBCL	IRIS	NATURAL IMAGES
MM	3	3	3		3	
AD				6		6
CNMF	7	7	7		7	
GD-CLS				8		8
`nmfsc`	9					

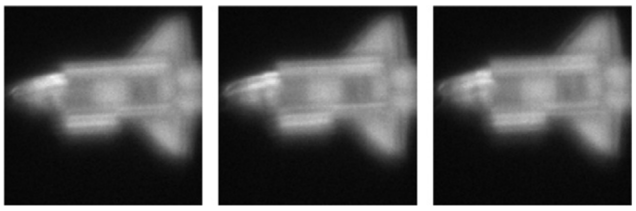
Fig. 1. Sample space shuttle Columbia images.

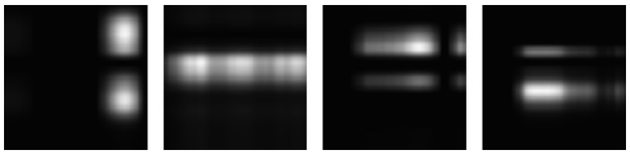
Fig. 2. Basis images for dataset SHUTTLE using Algorithm MM for $k = 4$.

2.6. Computational costs

The two major computational steps of NNDSVD are (i) computing k largest singular triplets, and (ii) computing the maximum singular triplet of the positive section of each singular factor in the singular expansion of A. When appropriate (e.g. for large sparse data), we assume that the (partial) SVD is computed by means of some iterative algorithm; see e.g. Refs. [39–42]. Any improvement in algorithms that compute the above two steps will reduce the runtime of NNDSVD. A rough estimate of the cost of the first step above is $O(kmn)$ for dense A. Hidden, in this notation, is a factor that depends on the number of iterations to convergence and which is unknown a priori. The other step can be performed very effectively, without ever computing explicitly the rank-2 matrices $C_+^{(j)}$: specifically, if $\mathrm{rank}(C_+) = 2$, both singular triplets are readily available

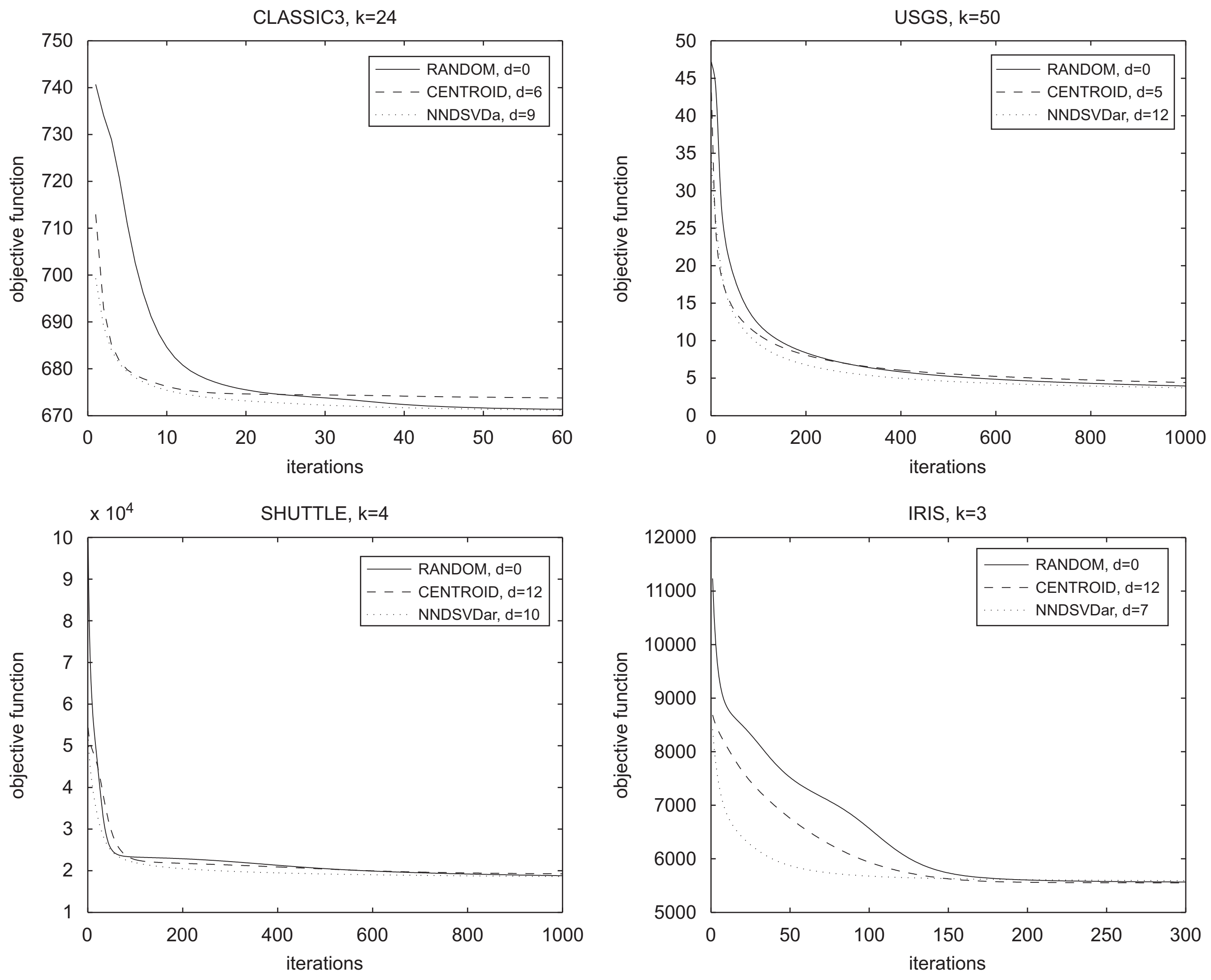

Fig. 3. Algorithm MM for datasets CLASSIC3, USGS, SHUTTLE and IRIS.

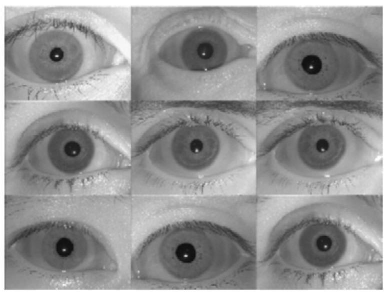

Fig. 4. Dataset of nine human eye iris images.

(Theorem 2) at cost $O(m+n)$. Thus the overall cost for NNDSVD on dense data is $O(kmn)$. The asymptotic cost of structured CENTROID initialization [31], which relies on `Skmeans`, is also $O(kmn)$, though the leading constants in the expressions are typically smaller. Since any initialization method is eventually linked with an NMF algorithm, for fairness we measure and take into account the initialization cost in terms of "iteration equivalents" of the ensuing NMF, that is the number of iterations, say d, in the factorization algorithm, that could have been performed at the time it takes to initialize. We thus set $d=0$ for RANDOM. In all NMF algorithms, update formulas were written so as to enforce a sequencing of operations that was appropriate for the problem dimensions. For example, since all datasets had $k \ll \min(m,n)$, the update formulas computed $W(H^\top H)$ rather than $(WH)H^\top$. It is worth noting that depending on the dimensions and selected sequencing, the runtime differences can be significant. Therefore, the design of a MATLAB implementation, must enforce the sequencing by means of parentheses in the algebraic expression, or else, the default (left-to-right) will be used.

3. Numerical experiments

We ran several experiments with NMF algorithms (Fig. 1) and datasets tabulated in Tables 3 and 4. Table 5 shows the figure number corresponding to results for specific

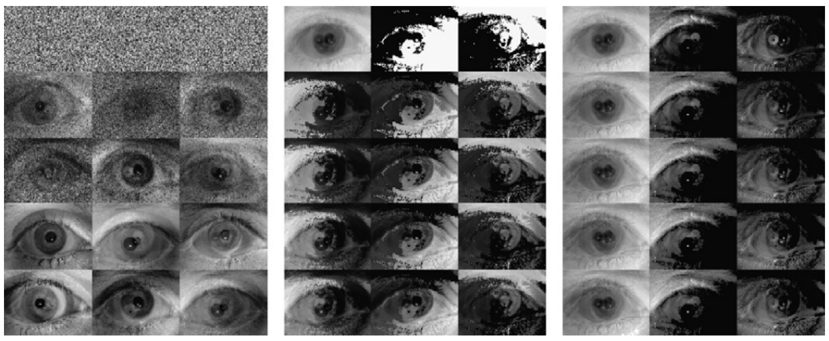

Fig. 5. Progress of approximation on iris dataset using MM: Random (left); NNDSVDar (center); NNDSVD (right). Rows correspond to 0, 20, 40, 60 and 80 iterations using $k = 3$.

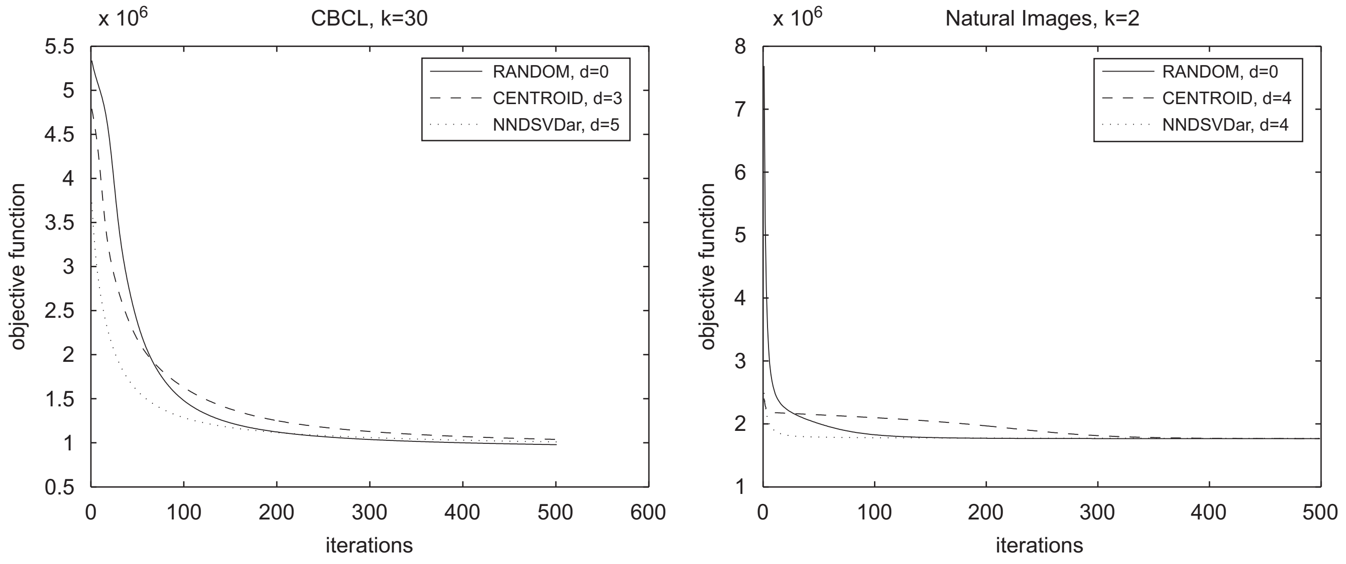

Fig. 6. Algorithm AD for datasets CBCL and NATURAL IMAGES.

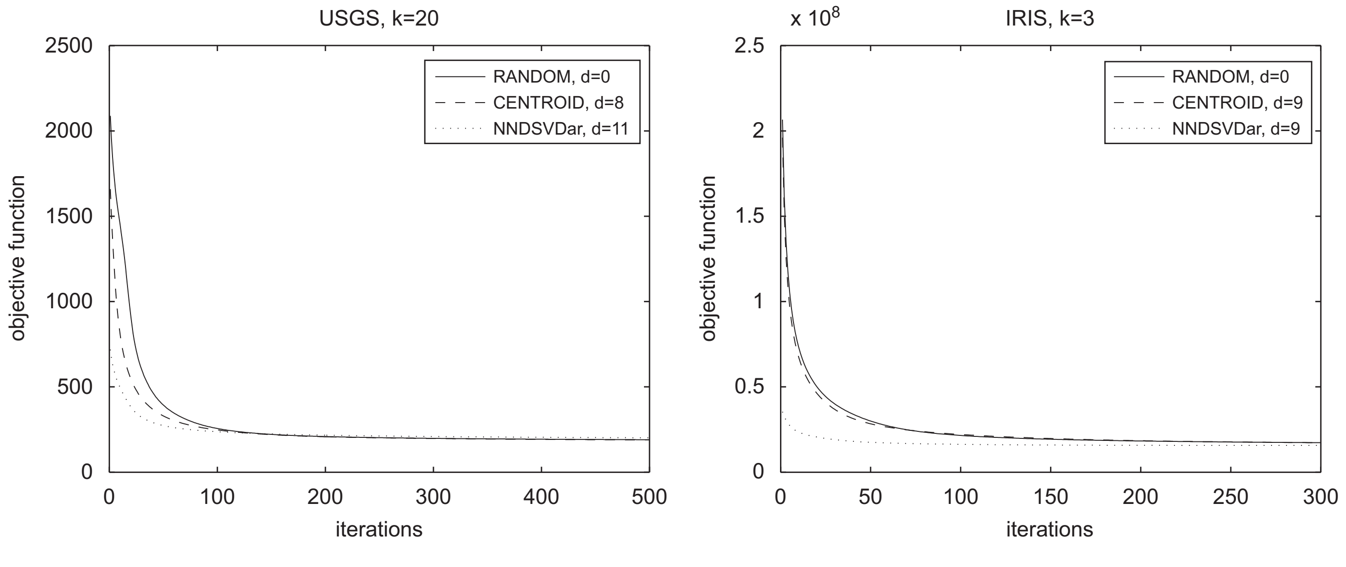

Fig. 7. Algorithm CNMF with $\alpha = \beta = 0.5$ for datasets USGS and IRIS.

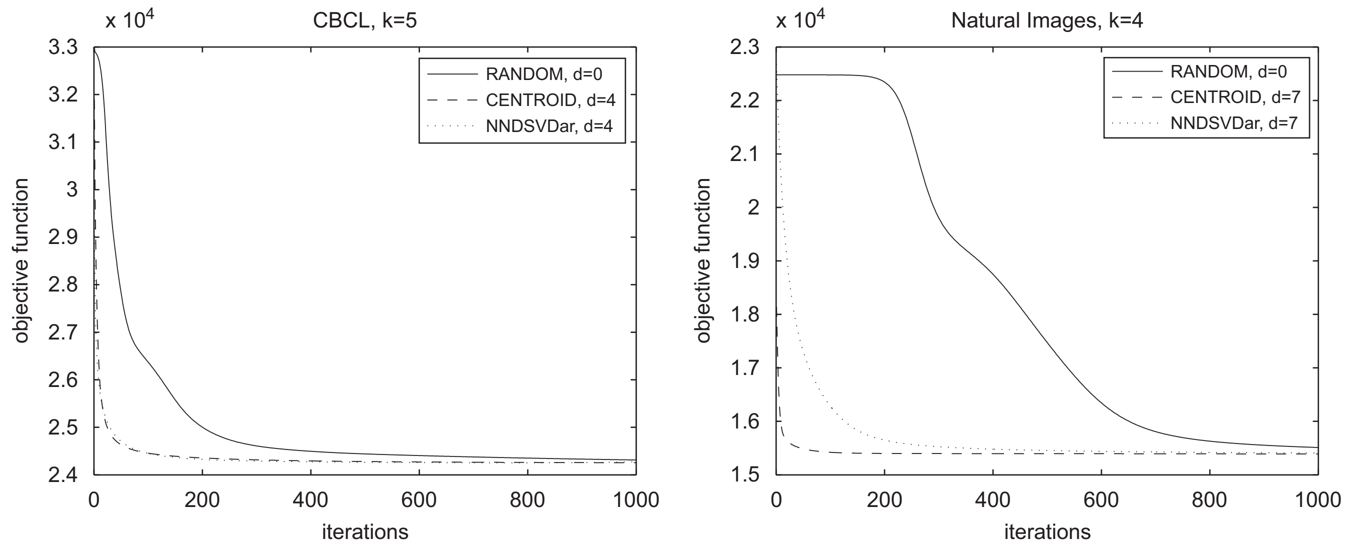

Fig. 8. Algorithm GD-CLS with $\lambda = 0.01$ for datasets CBCL and NATURAL IMAGES.

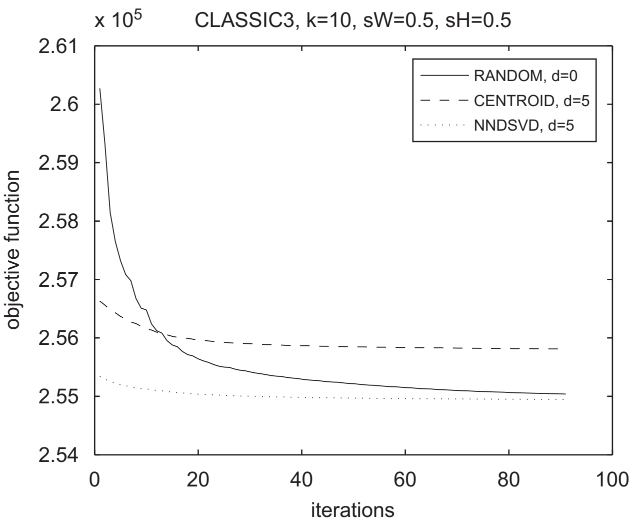

Fig. 9. Sparse NMF algorithm, nmfsc, for dataset CLASSIC3.

algorithm–dataset combinations. Each plot depicts the value of the corresponding objective function (see Table 3 for details) vs. number of iterations. It also displays the value selected for parameter k and the measured value of d that must be taken into account when evaluating the results. When comparing the residual error curves of RANDOM, CENTROID and NNDSVD, we must shift appropriately: In particular, we must compare the errors at iteration j of NNDSVD and iteration $j + d_{\text{NNDSVD}}$ of RANDOM (the subscript distinguishes the d's). Similarly for CENTROID. Finally, when comparing the errors in CENTROID and NNDSVD, we must compare iteration j of the latter with iteration $j + (d_{\text{NNDSVD}} - d_{\text{CENTROID}})$ of the former.

The platform used was a 2.0 GHz Pentium IV with 1024 MB RAM running Windows. Codes were written in MATLAB 7.0.1. We have been using a variety of methods to compute the partial SVD. In this paper, we used PROPACK [42]; this MATLAB library is based on Lanczos bidiagonalization with partial reothogonalization and provides a fast alternative to MATLAB's native `svds`. Care is required when using any of these functions so that they are forced to return nonnegative leading singular vectors, since MATLAB can also return entirely nonpositive leading singular vectors. Their product is, of course, nonnegative. This is not sufficient for NNDSVD, because it utilizes the positive section of these vectors. In that case, nonpositive vectors return zero values. Therefore, to be cautious we use the absolute values of the leading singular vectors. A design choice we need to mention is that in CENTROID, H was initialized as random. This choice was dictated by the cost of intrinsic (`lsqnonneg`) and other off-the-shelf MATLAB functions (`nnls` and codes from Ref. [43]) for solving the nonnegative least squares problem necessary to produce H from W. Specifically, their runtime was too high to make them viable as components of an initialization method. Furthermore we used an internally developed implementation of Skmeans clustering. The initializations used in the experiments are listed below. We note that the figures were selected after extensive experimentation with initializations, algorithms and datasets and selection of representative results.

RANDOM	Initialize the pair (W, H) to random using MATLAB's `rand`.
CENTROID	Initialize W as in Ref. [31] and H as random. `Skmeans` ran for 10 iterations.
NNDSVD	As specified in Table 1.
NNDSVDa	Perturbed NNDSVD using $\texttt{mean}(A)$ (cf. Section 2.3).
NNDSVDar	Perturbed NNDSVD using values in $[0, \texttt{mean}(A)/100]$ (cf. Section 2.3).

Fig. 3 shows experiments with MM. Fig. 5 shows the progress of the approximation when using data set IRIS (Fig. 4) after RANDOM, NNDSVD and NNDSVDar initializations. The figure displays the basis images resulting from matricizing the columns of matrix W after the specified number of iterations. A noteworthy result is that different initializations lead to different basis images.

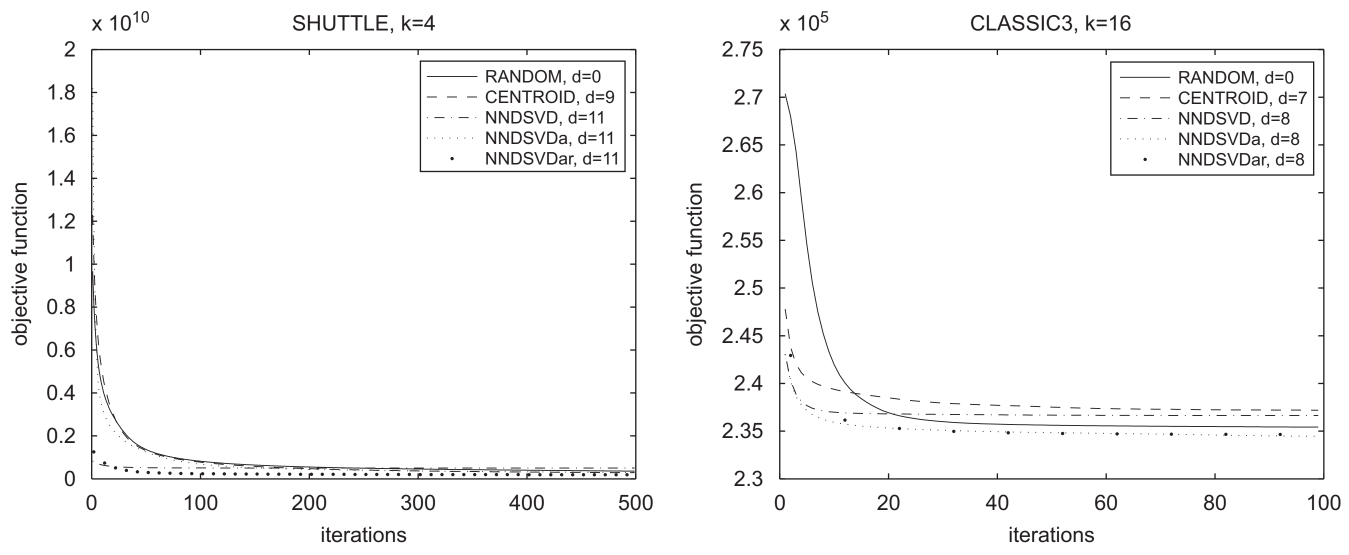

Fig. 10. NNDSVD, NNDSVDa, NNDSVDar and CENTROID on CNMF ($\alpha = \beta = 0.5$) for datasets SHUTTLE and CLASSIC3.

Figs. 6–8 show results for algorithms AD, CNMF and GD-CLS, respectively. Fig. 9 depicts results with the sparse algorithm. In this case we used basic NNDSVD and no variants, as justified in Section 2.3. The plots also show the selected values of parameters (sW, sH) that determine the desired sparsity level for (W, H).

As alluded in Section 1, in the image datasets of Table 4, the data matrix A is constructed by stacking in columns the vectorized images in the collection, as is common practice in the NMF literature. The columns of W correspond to "basic images". Each image of the collection (column of A) is then approximated as a linear combination of the basis images using as reconstruction coefficients the elements in the corresponding column of H. Figs. 2 and 5, for example, show the resulting basis images for the SHUTTLE and IRIS datasets. Dataset CLASSIC3 is a term-document matrix used as benchmark in text mining, hence each column of W encodes a basis document. Finally, in dataset USGS, each column encodes a "spectral signature" that becomes useful in spectral unmixing; cf. Refs. [9,21,30] for more information regarding these and other similar datasets in the context of NMF.

Overall, even taking into account the aforementioned d-shifts in iterations, the plots confirm that NNDSVD-type initializations are very fast. They also outperform RANDOM in all test cases, and generally appear to be a better choice (except when combined with GD-CLS) than CENTROID. They also need less time to reduce the NMF objective function, the improvements being most dramatic after only few iterations of the NMF algorithm as is evidenced by the pronounced "knee" shape of the corresponding error curves depicted in the figures, as well as in the visualization for dataset IRIS in Fig. 5. Besides the gains in computational efficiency, sometimes algorithms based on NNDSVD appear to lead to smaller error than RANDOM and CENTROID, even at convergence; see e.g. Fig. 9. One more feature that differentiates NNDSVD from RANDOM and CENTROID is that the basic method is deterministic. Finally, NNDSVD appears to be the first initialization scheme that comes with provable theoretical guarantees for the approximation error (Proposition 6).

Assuming that the reader proceeds with caution, since our evidence is entirely experimental and the heuristics are based solely on the sparsity structure of the dataset, the above experiments provide some guidance toward the selection of the most suitable NNDSVD variant. Specifically, if we seek some sparsity constraints, the basic NNDSVD algorithm appears well suited, since it tends to generate sparse factors. Otherwise, it is probably better to use one of NNDSVDa and NNDSVDar. In our experiments, NNDSVDar tends to return somewhat superior results than NNDSVDa. Nevertheless, after very extensive experiments, no algorithm emerged as an overall winner, so the final decision rests upon the user's experience with the variants' performance on the data under study.[3] .

Fig. 10 illustrates the performance of all initialization methods used in this paper combined with algorithm CNMF. With dataset SHUTTLE (left), all methods reach about the same final error, though NNDSVDar has better performance and a pronounced knee behavior. With dataset CLASSIC3 (right), however, the error in RANDOM eventually catches up. As mentioned earlier, this behavior was the motivation behind the design of the NNDSVD variants; indeed, not only NNDSVDar has the most pronounced knee behavior but also results in the smallest final error.

We conclude that NNDSVD provides initial values that enable the followup NMF algorithms significant reduction of the initial residual after very few iterations and at low overall cost, at levels that are comparable to the residual obtained after running the algorithm to convergence with any of the three basic initializations. We also mention some further issues that have

[3] After submission, we became aware of very recent work in Ref. [47], including extensive experiments with some of the variants of NNDSVD described herein as well as others. We expect these to be useful to the potential user of our double SVD approach for initializing NMF.

arisen in the course of this investigation, such as the study and generalization of the property in Lemma 1, the application of clustering with NNDSVD in the spirit of CENTROID and the performance of the method in the context of distance metrics such as those in Ref. [1], that can be better tailored to prior knowledge about the data. Finally, it is worth reminding that even though NNDSVD frequently leads to errors that are comparable or superior to those achieved by random initialization, we did not provide any guarantees regarding the quality of the local minimum reached by the subsequent NMF algorithm. If the computed solution is unsatisfactory, one might conclude that NNDSVD hinders progress toward a better local minimum because its basic form is deterministic and does not allow for multiple runs. To the extent that the effectiveness of such a strategy to escape from local minima is justified, one could opt to use multiple runs of NNDSVDar, the randomized variant of NNDSVD, or simply repeatedly run the NMF algorithm with random initialization keeping track and finally comparing all results, including those obtained with NNDSVD. Clearly, the field is open for research contributions coupling deterministic initialization strategies with NMFs leading to even smaller approximation errors!

Acknowledgments

Discussions with and original experiments of our colleague, Dimitris Zeimpekis, were very helpful. We also thank him for providing us with MATLAB code for `Skmeans`. We thank Professors Robert J. Plemmons for his invaluable advice and encouragement, Daniel Szyld for his comments on an early version of this manuscript and Petros Drineas for many helpful observations. We are also grateful to Stefan Wild for providing us useful details about and codes from his original papers [30,31]. Finally, we thank the reviewers of the paper for their insightful comments. Research was supported in part by the University of Patras K. Karatheodori Grant no. B120.

References

[1] I. Dhillon, S. Sra, Generalized nonnegative matrix approximations with Bregman divergences, in: Proceedings of the NIPS, Vancouver, 2005, pp. 283–290.
[2] M. Chu, F. Diele, R. Plemmons, S. Ragni, Optimality, computation, and interpretation of nonnegative matrix factorization, unpublished preprint, October 2004.
[3] C. Ding, T. Li, W. Peng, H. Park, Orthogonal nonnegative matrix tri-factorizations for clustering, in: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press, New York, 2006, pp. 126–135.
[4] D. Donoho, V. Stodden, When does non-negative matrix factorization give a correct decomposition into parts?, Adv. Neural Inf. Process. Syst. 17 (2004).
[5] L. Eldén, Matrix Methods in Data Mining and Pattern Recognition, SIAM, Philadelphia, PA, USA, 2007.
[6] P. Hoyer, Non-negative sparse coding, in: Proceedings of the IEEE Workshop on Neural Networks for Signal Processing, Martigny, Switzerland, 2002, pp. 557–565.
[7] P. Hoyer, Non-negative matrix factorization with sparseness constraints, J. Mach. Learn. Res. 5 (2004) 1457–1469.
[8] D.D. Lee, H.S. Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (1999) 788–791.
[9] D. Kim, S. Sra, I.S. Dhillon, Fast Newton-type methods for the least squares nonnegative matrix approximation problem, in: Proceedings of the 2007 SIAM Conference on Data Mining, SIAM, Philadelphia, 2007, pp. 343–354.
[10] S. Marinetti, L. Finesso, E. Marsilio, Matrix factorization methods: application to thermal NDT/E, NDT&E Int. 39 (2006) 611–616.
[11] P. Paatero, U. Tapper, Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values, Environmetrics 5 (1994) 111–126.
[12] H. Park, H. Kim, One-sided non-negative matrix factorization and non-negative centroid dimension reduction for text classification, in: M. Berry, M. Castellanos (Eds.), Proceedings of the Text Mining 2006 Workshop held with 6th SIAM International Conference on Data Mining (SDM 2006), SIAM, Philadelphia, 2006.
[13] P. Pauca, F. Shahnaz, M. Berry, R. Plemmons, Text mining using non-negative matrix factorizations, in: 4th SIAM Conference on Data Mining, Orlando, FL, SIAM, Philadelphia, April 2004, pp. 452–456.
[14] P. Sajda, S. Du, T. Brown, R. Stoyanova, D. Shungu, X. Mao, L. Parra, Nonnegative matrix factorization for rapid recovery of constituent spectra in magnetic resonance chemical shift imaging of the brain, IEEE Trans. Med. Imaging 23 (12) (2004) 1453–1465.
[15] F. Shahnaz, M. Berry, V. Pauca, R. Plemmons, Document clustering using nonnegative matrix factorization, Inf. Process. Manage. 42 (2) (2006) 373–386.
[16] W. Xu, X. Liu, Y. Gong, Document clustering based on non-negative matrix factorization, in: Proceedings of the 26th ACM SIGIR, ACM Press, New York, 2003, pp. 267–273.
[17] D. Zhang, S. Chen, Z.-H. Zhou, Two-dimensional non-negative matrix factorization for face representation and recognition, in: Proceedings of the ICCV'05 Workshop on Analysis and Modeling of Faces and Gestures (AMFG'05), Lecture Notes in Computer Science, vol. 3723, Springer, Berlin, 2005, pp. 350–363.
[18] A. Cichocki, R. Zdunek, NMFLAB MATLAB toolbox for non-negative matrix factorization. URL: ⟨www.bsp.brain.riken.jp/ICALAB/nmflab.html/⟩.
[19] A. Pascual-Montano, P. Carmona-Saez, M. Chagoyen, F. Tirado, J. Carazo, R. Pascual-Marqui, bioNMF: a versatile tool for non-negative matrix factorization in biology, BMC Bioinformatics 7 (2006) 366.
[20] A. Berman, R. Plemmons, Nonnegative Matrices in the Mathematical Sciences, SIAM, Philadelphia, 1994.
[21] M.W. Berry, M. Browne, A.N. Langville, V.P. Pauca, R.J. Plemmons, Algorithms and applications for approximate nonnegative matrix factorization, Comput. Stat. Data Anal. 52 (1) (2007) 155–173.
[22] D. Gregory, N. Pullman, Semiring rank: Boolean rank and nonnegative rank factorization, J. Combin. Inf. System Sci. 3 (1983) 223–233.
[23] D. Bertsekas, Nonlinear programming, Athena Scientific, Belmont, MA, 1999.
[24] R. Salakhutdinov, S. Roweis, Z. Ghahramani, On the convergence of bound optimization algorithms, in: Proceedings of the 19th Conference in Uncertainty in Artificial Intelligence (UAI '03), Morgan Kaufmann, Los Altos, CA, 2003, pp. 509–516.
[25] D.D. Lee, H.S. Seung, Algorithms for non-negative matrix factorizations, Adv. Neural Inf. Process. Syst. 13 (2001) 556–562.
[26] W. Liu, J. Yi, Existing and new algorithms for nonnegative matrix factorization, Technical Report, University Texas at Austin, 2003.
[27] M. Aharon, M. Elad, A. Bruckstein, K-SVD and its non-negative variant for dictionary design, in: M. Papadakis, A. Laine, M. Unser (Eds.), Proceedings of the SPIE Conference, Wavelets XI, vol. 5914, 2005, pp. 327–339
[28] V. Pauca, R. Plemmons, M. Giffin, K. Hamada, Unmixing spectral data for sparse low-rank non-negative matrix factorization, in: Proceedings of the Amos Technical Conference Maui, 2004.
[29] W. Liu, N. Zheng, Learning sparse features for classification by mixture models, Pattern Recognition Lett. 25 (2004) 155–161.
[30] S. Wild, Seeding non-negative matrix factorizations with the spherical *k*-means clustering, Master's Thesis, University of Colorado, Department of Applied Mathematics, 2003.

[31] S. Wild, J. Curry, A. Dougherty, Improving non-negative matrix factorizations through structured intitialization, Pattern Recognition 37 (2004) 2217–2232.

[32] I.S. Dhillon, D.S. Modha, Concept decompositions for large sparse text data using clustering, Mach. Learn. 42 (1) (2001) 143–175.

[33] D. Zeimpekis, E. Gallopoulos, CLSI: a flexible approximation scheme from clustered term-document matrices, in: H. Kargupta et al. (Eds.), Proceedings of the 5th SIAM International Conference on Data Mining, SIAM, Philadelphia, 2005, pp. 631–635.

[34] M. Catral, L. Han, M. Neumann, R. Plemmons, On reduced rank nonnegative matrix factorization for symmetric nonnegative matrices, Linear Algebra Appl. 393 (1) (2004) 107–126.

[35] G. Stewart, J.g. Sun, Matrix Perturbation Theory, Academic Press, Boston, 1990.

[36] J. Cohen, U. Rothblum, Nonnegative ranks, decompositions, and factorizations of nonnegative matrices, Linear Algebra Appl. 190 (1993) 149–168.

[37] C. Ding, X. He, H. Simon, On the equivalence of nonnegative matrix factorization and spectral clustering, in: Proceedings of the 5th SIAM International Conference on Data Mining, Philadelphia, 2005, pp. 606–610.

[38] V. Pauca, J. Piper, R. Plemmons, Nonnegative matrix factorization for spectral data analysis, Linear Algebra Appl. 416 (1) (2006) 29–47.

[39] J. Baglama, D. Calvetti, L. Reichel, IRBL: an implicitly restarted block Lanczos method for large-scale Hermitian eigenproblems, SIAM J. Sci. Comput. 24 (5) (2003) 1650–1677.

[40] M. Berry, Large scale singular value decomposition, Int. J. Supercomput. Appl. 6 (1992) 13–49.

[41] G. Golub, C.V. Loan, Matrix Computations, third ed., The Johns Hopkins University Press, Baltimore, 1996.

[42] R. Larsen, PROPACK: a software package for the symmetric eigenvalue problem and singular value problems on Lanczos and Lanczos bidiagonalization with partial reorthogonalization. URL: ⟨http://soi.stanford.edu/∼rmunk/PROPACK/⟩.

[43] Mathworks MATLAB file exchange, ⟨http://www.mathworks.com/matlabcentral/fileexchange⟩, (accessed 3.1.07).

[44] TMG: A MATLAB Toolbox for generating term-document matrices from text collections, ⟨http://scgroup.hpclab.ceid.upatras.gr/scgroup/Projects/TMG/⟩, (accessed 3.1.07).

[45] CBCL face database # 1 ⟨http://cbcl.mit.edu/cbcl/software-datasets/FaceData2.html⟩, (accessed 3.1.07).

[46] University of Bath Iris image database ⟨http://www.bath.ac.uk/elec-eng/research/sipg/irisweb/sampleiris.htm⟩, (accessed 3.1.07).

[47] K. Heinrich, Automated gene classification using nonnegative matrix factorization on biomedical literature, Ph.D. Thesis, The University of Tennessee, Knoxville, May 2007.

About the Author—CHRISTOS BOUTSIDIS was born in Serres, Greece, on August 1983. He received a bachelor degree in computer science and engineering from the University of Patras in July 2006. As of September 2006, he is a Graduate Ph.D. student in Computer Science Department at Rensselaer Polytechnic Institute.

About the Author—EFSTRATIOS GALLOPOULOS is a Professor at the University of Patras. Between 1985 and 1995 he held positions at UIUC and UCSB. His Ph.D. (computer science) is from the University of Illinois at Urbana-Champaign and his B.Sc. (1st class) from Imperial College. His research is on scientific computing, problem solving environments and parallel computing.