Pattern Recognition Letters 63 (2015) 71–77

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

New SVD based initialization strategy for non-negative matrix factorization☆

Hanli Qiao*

Department of Mathematics G. Peano, University of Torino, Via Carlo Alberto 10, Turin 10123, Italy

ARTICLE INFO

Article history:
Received 11 November 2014
Available online 27 June 2015

Keywords:
NMF
SVD
SVD–NMF
NNDSVD
Non-negative matrix factorization
Singular value decomposition

ABSTRACT

There are two problems need to be dealt with for Non-negative Matrix Factorization (NMF): choose a suitable rank of the factorization and provide a good initialization method for NMF algorithms. This paper aims to solve these two problems using Singular Value Decomposition (SVD). At first we extract the number of main components as the rank, actually this method is inspired from Turk and Pentland (1991) [15,16]. Second, we use the singular value and its vectors to initialize NMF algorithms. We title this new method as SVD-NMF. Boutsidis and Gollopoulos (2008) [2] provided the method titled NNDSVD to enhance initialization of NMF algorithms. They extracted the positive section and respective singular triplet information of the unit matrices $\{C^{(j)}\}_{j=1}^{k}$ which obtained by singular vector pairs based on SVD. In this strategy, they use the triplet information of SVD twice with low computational cost. The differences between SVD–NMF and NNDSVD are the once utilization of SVD in former method and different approximations of initializations for NMF algorithms. We report numerical experiments on two face databases ORL, YALE (C.U.C. Laboratory [10]; U.C. Version [17]) and one object database COIL-20 with two versions which are from Columbia University Image Library (Nene et al., 1996) [13]. Results show that SVD–NMF has faster convergence and provides an approximation with smaller errors than that of obtained by NNDSVD and random initialization.

1. Introduction

Lee and Seung published a paper with the title Learning the parts of objects by Non-negative Matrix Factorization (NMF) [11]. They analyzed how NMF could learn the parts of objects for facial images and semantic topics. The non-negativity of NMF only permit the additive combination of multiple basis images to present a face. It is compatible with the intuitive notion of combining parts to form a whole. Two different multiple algorithms for NMF were analyzed in [12]. The main idea of NMF is that utilize two low rank matrices to approximate one large dimension data matrix so that reduce its dimension. NMF can also be applied to a lot of other fields: Ding and his colleagues have deeply researched in data clustering and Combinatorial Optimization (see [4–7]). Oja and Yuan proposed Projective Non-negative Matrix Factorization (PNMF) for image compression and feature extraction [23], whereas NMF can be applied to face detection in paper [3].

NMF is an effective algebraic method in extracting features for classification and clustering. Recently some methods such as graph based manifold learning methods, hypergraph learning methods and multiview learning methods have been proposed and becoming popular strategies. In paper [22], authors proposed an adaptive hypergraph learning method for transductive image classification, of which the hyperedges generated by linking images and their nearest neighbors. They also present a simultaneous learning of the labels of unlabeled images and the weights of hyperedges. For utilizing multiview data effectively in image classification, authors proposed a high-order distance based multiview stochastic learning (HD-MSL) method in [21]. HD-MSL uses a unified representation to combine various features and integrates the labeling information. This approach utilizes the distance obtained from the hypergraph to estimate the high-order relationship instead of using pairwise distance in building the probability distribution. HD-MSL also can automatically learn a combination coefficient of different views. In [20], a novel manifold learning algorithm named sparse patch alignment framework (SPAF) is proposed. According to patch alignment framework, local patches of SPAF are conducted by an optimization strategy and after that for building the manifold, the whole alignment strategy is utilized to build the manifold. Numerical results show the effectiveness of SPAF in image classification and clustering.

NMF framework can be described as follows: given an original non-negative matrix $Z \in \mathbb{R}_{+}^{m\times n}$, find $W \in \mathbb{R}_{+}^{m\times p}$ and $H \in \mathbb{R}_{+}^{p\times n}$, such that:

$$Z \approx WH$$

☆ This paper has been recommended for acceptance by J. Yang

* Corresponding author. Tel.: +39 366 1787869.
E-mail address: hanli.qiao@unito.it

http://dx.doi.org/10.1016/j.patrec.2015.05.019
0167-8655/

where W is called basis matrix, H is called coefficient matrix and p is the rank of the factorization. Note that it is an important index to determine the size of these two low rank matrices. To reduce the dimension of Z, we want p is small. On the other hand, for the accuracy of the approximation, the larger p is, the more accurate the approximation will be. Almost all of the researchers set p as different numbers at the beginning of algorithms. Hence we need to find a method to choose p, which should be much smaller than $\min\{m, n\}$. Generally, it should satisfy the basic rule $(m+n)p < mn$ [12]. In Section 2.1, we will introduce a method based on SVD to choose p that is called Choosing Rule. Here, we give a brief introduction of SVD.

Given a matrix $M \in \mathbb{F}^{m\times n}$, which can be a real or complex matrix, there exists a factorization of the form $M = U\Sigma V^*$, where U is an $m \times m$ unitary matrix over $\mathbb{F}$, Σ is an $m \times n$ diagonal matrix with non-negative real numbers on the diagonal, and the $n \times n$ unitary matrix V^* denotes the conjugate transpose of the $n \times n$ unitary matrix V. Such a factorization is called a SVD of M.

The diagonal entries of Σ are known as the singular values of M, which are in descending order. In many cases, the first few singular values account for over 90% of all singular values. This means that we can use p, which account for 90% information of all singular values to be as the rank of the factorization. p should satisfy the basic rule

$$(m+n)p < mn \tag{1}$$

In papers [11,12], Lee and Seung gave us two cost functions to describe NMF problem. One of them is $f(W,H) := \|Z - WH\|_F^2$ and it can be viewed as the following bound optimization problem: Given non-negative matrix Z, find W and H which solve

$$min f(W,H), \quad W \geq 0, \quad H \geq 0 \tag{2}$$

They used multiplicative update rules and additive update rules to solve this problem. We call them MU and AD algorithm respectively. But in MU and AD algorithms, rank p is set by researchers arbitrarily at the beginning of the algorithms, and the initialization values $W_0^{m\times p}$ and $H_0^{p\times n}$ are chosen randomly. We call this method RandomNMF in this paper. NMF can suffer from slow convergence, then the whole computational process can become much expensive. Hence we should find good initialization method to make algorithms be more effective. Currently there are some literatures which propose different methods to improve the initializations of NMF algorithms (see [1,2,9,18,19]). Papers [18,19] use Spherical K-Means clustering to produce a structured initialization for NMF. Although this method is effective, it increases the computational complexity. In paper [1], authors compared six initialization procedures on their Alternating Least Squares (ALS) based algorithms, whereas paper [9] applies population based algorithm to NMF. They used five population based algorithms to compute optimal starting points for single rows of W and single columns of H. This kind of method obviously makes the computation at cost more expensive. NNDSVD is a method that derived from paper [2], of which large experiments indicate that this method is better than centroid initialization because of faster convergence [19]. Based on the properties of unit rank matrices, NNDSVD utilizes them to initialize W and H. At first they form unit rank matrices obtained from singular vector pairs and then extract the positive section and respective singular triplet information of unit matrices to initialize NMF algorithms. The two processes use the singular triplet information of SVD twice even with low cost. In this paper, we propose a new method to determine the rank of factorization and initialize NMF algorithms using SVD only once and the implementation is very easy with much smaller errors and low computational cost. At first, we apply SVD for nonnegative matrix Z and then extract the number of principal components by singular values as the rank of factorization. Second, we use singular triplet information to initialize W and H. There is no randomization in the basic form, therefore for any given algorithm, it converges to the same solution, which is benefit for iterative algorithms. The basic initialize formula based on SVD, therefore we title this method as SVD–NMF. Through the process, we can see that only once utilization of singular triplet information used.

As Boutsidis and Gallopoulos referred [2], good initialization strategy should satisfy the following conditions: (i) one that leads to rapid error reduction and faster convergence; (ii) one that leads to overall error at convergence. We only evaluate the property of SVD–NMF in the first condition while the second one is very difficult to be satisfied for most of algorithms.

We arrange this paper as follows. Section 2 gives two main contents, the first one is the Choosing rule to compute the rank of the factorization and the second one is that introduce the method SVD–NMF. In Section 3, numerical experiments show the effectiveness of SVD–NMF for two kinds of NMF algorithms.

2. SVD–NMF for the initialization of NMF

As we mentioned in Section 1, the rank p of factorization can be calculated by SVD–NMF. For NMF problem, SVD can be expressed as follows. For any matrix $Z \in \mathbb{R}_+^{m\times n}$, there exists a factorization with the following form:

$$Z = \mathbf{U\Sigma V}^T, \tag{3}$$

where the $n \times n$ unitary matrix V^T denotes the transpose of the $n \times n$ unitary matrix V and

$$\Sigma = \begin{pmatrix} \mathbf{\Sigma_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}. \tag{4}$$

Here $\mathbf{\Sigma_1} = diag(\sigma_1, \sigma_2, \ldots, \sigma_r)$, and the diagonal entries are sorted in descending order, i.e. $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$, σ_i, $i = 1, 2, \ldots r$ being the singular values with $r = rank(Z)$. From this property, we can calculate the number of relatively larger singular values. For NMF, we need to find matrices $W \in \mathbb{R}_+^{m\times p}$ and $H \in \mathbb{R}_+^{p\times n}$ and now we will use this behaviur to choose p, which is the rank of factorization for NMF.

2.1. Choosing rule

Turk and Pentland [15,16] used eigenfaces for recognition. Authors utilized the eigenfaces to span an M'-dimensional subspace. The M' significant eigenvectors are chosen as those with largest associated eigenvalues, which inspires us to use the amount of relatively larger singular values to obtain the rank of factorization. Because singular values of Z obtained by SVD are sorted in descending order. The sum of first few singular values accounts for a large proportion of sum of all singular values. We choose the extracting proportion as 0.9 that contains enough information of singular values while avoiding the factorization rank too small to influence the accuracy of factorization. From formula (3), we get the diagonal matrix Σ. At first, we make the sum of all non-zero diagonal entries for Σ, that is $\mathbf{sum_r} = \sigma_1 + \sigma_2 + \cdots + \sigma_r$, and then we choose the number of singular values which accounts for 90% of all non-zero diagonal entries, that is $\mathbf{sum_p} = \sigma_1 + \cdots + \sigma_p$ so we obtain the Choosing rule:

$$\mathbf{sum_p}/\mathbf{sum_r} < 90\% \quad and \quad \mathbf{sum_{p+1}}/\mathbf{sum_r} \geq 90\%, \tag{5}$$

This is meaningful because the non-zero entries of Σ are the square root of non-negative eigenvalues of matrix ZZ^*, then we can get that $r \leq \min\{m, n\}$, where m, n are the number of rows and columns of matrix Z, respectively. After extracting 90% components by the rule (5), we can obtain $p \ll r$. Here we give the MATLAB code of Choosing Rule:

```
function [u s v p]=ChoosingR(Z)
[u,s,v] = svd(Z);
sum1= sum(s);
sum2=sum(sum1);
extract=0;
p = 0;
```

Table 1
Factorization ranks for the first person on ORL database with various extracting proportion.

$p_{80\%}$	22	16	22	21	23
$p_{90\%}$	35	26	35	34	37
$p_{100\%}$	61	49	67	65	73

Table 2
Factorization ranks for the first person on YALE database with various extracting proportion.

$p_{80\%}$	27	29	29	26	28
$p_{90\%}$	45	47	46	42	45
$p_{100\%}$	87	93	90	84	89

```
dsum=0;
while(extract/sum2<0.90)
p = p + 1;
dsum=dsum+s(p,p);
extract=dsum;
end
end
```

Table 1 gives us two groups of rank p for different image matrices using Choosing Rule. These ten images derive from ORL [10] and YALE [17] database, respectively. ORL face database has 10 different images of each of 40 distinct persons and the size of each image is 92×112. YALE face database contains 165 gray-scale images of 15 individuals. There are 11 images per person and the pixels of each image are 100×100. We point that each image stored in the computer as different matrix, therefore we get different p of each image after using Choosing Rule. From it, we get that if the extracting proportion is different, the factorization p is different as results showed in Tables 1 and 2. Therefore, foe evaluating appropriateness, we report p with various proportions, 80, 90 and 100% respectively. And we use $p_{.\%}$ denotes the factorization rank obtained by choosing rule with extracting proportion .%. We chose five images of the first person on ORL database and another five images on YALE database. From condition 1, for images of ORL we get $p < \frac{mn}{m+n} = \frac{92\times112}{92+112} \approx 51$ and for YALE, $p < \frac{mn}{m+n} = \frac{100\times100}{200} = 50$, where m, n is the number of row and column of image matrix. Tables 1 and 2 show that all of $p_{80\%}$ and $p_{90\%}$ satisfy condition 1 while $p_{100\%}$ does not. In Section 3.1, we discuss the influences for errors with different factorization ranks among SVD–NMF, NNDSVD and RandomNMF.

2.2. SVD based initialization: SVD–NMF

In this paper, p in numerical experiments are chosen by Choosing Rule (5), which is introduced in Section 2.1. As we mentioned good initialization can make convergence fast and get low cost of computational process. NNDSVD uses singular triplets of SVD twice to initialize matrices $W^{m,p}$ and $H^{p,n}$ for NMF. SVD–NMF only uses SVD once to obtain the singular triplets of $Z^{m,n}$.

For analyzing the Bound Optimization problem (2), we need Eckart Young Theorem [8]

Lemma 1. *Let $V \in \mathbb{R}^{m\times n}$ be a singular value decomposition*

$$V = P\Sigma Q^T, \quad \Sigma = diag(\sigma_1, \sigma_2, \ldots, \sigma_n) \in \mathbb{R}^{m,n},$$

where $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0$ are the singular values of V and $P \in \mathbb{R}^{m,m}$ and $Q \in \mathbb{R}^{n,n}$ are orthogonal matrices. Then for $1 \leq r \leq n$, the matrix

$$B_r = Pdiag(\sigma_1, \sigma_2, \ldots, \sigma_r, \underbrace{0, \ldots, 0}_{n-r})Q^T \tag{6}$$

is a global minimizer of the optimization problem

$$\min\{\ \|V - B\|_F^2 \quad |B \in \mathbb{R}^{m,n}, \quad rank(B) \leq r\} \tag{7}$$

with the corresponding minimum value $\Sigma_{i=r+1}^{n}\sigma^2$. Moreover, if $\sigma_r > \sigma_{r+1}$, then B_r is the unique global minimizer.

From Lemma 1 we can easily use non-negative matrix $Z^{m,n}$ to convert matrix V, then we can get that if there exists a matrix B_r that has the form (6), the bound optimization problem (2) can find the global minimizer B_r. We can compute the SVD of non-negative matrix $Z^{m,n}$ and we obtain the singular triplets of $Z^{m,n}$: $U^{m,m}$, $S^{m,n}$, $V^{n,n}$. If we choose $1 \leq p < r \leq n$, we can find a unique matrix B_p with a fixed p such that: $B_p = Udiag(\sigma_1, \sigma_2, \ldots, \sigma_p, \underbrace{0, \ldots, 0}_{n-p})V^T$, which is the global minimizer of the optimization problem (7). We can easily verify that:

$$B_p = Udiag(\sigma_1, \sigma_2, \ldots, \sigma_p, \underbrace{0, \ldots, 0}_{n-p})V^T = \tilde{U}\tilde{\Sigma}V^T \tag{8}$$

where

$$\tilde{U} = \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1p} \\ u_{21} & u_{22} & \cdots & u_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ u_{m1} & u_{m2} & \cdots & u_{mp} \end{pmatrix} \tag{9}$$

and

$$\tilde{\Sigma} = \begin{pmatrix} \sigma_1 & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_p & \cdots & 0 \end{pmatrix}, \tag{10}$$

$\tilde{U} \in \mathbb{R}^{m,p}$ and $\tilde{\Sigma} \in \mathbb{R}^{p,n}$. Then matrix B_p is the solution of the bound optimization problem (2). We can instantly obtain that $WH = B_p = \tilde{U}(\tilde{\Sigma}V^T)$ is the solution of (2) that we want to obtain, here $W = \tilde{U}$, $H = \tilde{\Sigma}V^T$. However, the entries of singular vectors of SVD can be negative, we cannot directly use matrices (9) and (10) to initialize W and H. We set the negative elements of matrix $\tilde{U}$ as the absolute values of themselves; then we get $|\tilde{U}|$, where $|.|$ indicates that all entries of $\tilde{U}$ are their absolute values. And we make all the negative entries of $\tilde{\Sigma}V^T$ as their absolute values, thus we getting matrix $|\tilde{\Sigma}V^T|$.

We get the initialization formulas of W, H:

$$W_0 = |\tilde{U}|, \quad H_0 = |\tilde{\Sigma}V^T|$$

then $W_0H_0 \approx B_p$. We use MATLAB to compile SVD–NMF to fulfill the initialization of NMF algorithms. It is the following code:

```
m = size(A,1);
[u s v p] = ChoosingR(A);
W = abs(u(:,1:p));
H = abs(s(1:p,:)*v');
```

From this code we can see that SVD–NMF only computes the singular triplets of original matrix once, hence it is very simple and it can be used the initialization for any NMF algorithms. Moreover we can get more stable results of factorization after applying SVD–NMF. For analyzing SVD–NMF, we combine the initialization with the two algorithms: MU and local non-negative matrix factorization (LNMF) (see [14]). We must note that LNMF is based on another object function KullbackLeibler divergence $D(Z \parallel WH)$ in [11,12]. Here the objective function for LNMF is

$$D(A||B) = \sum_{i,j}\left(A_{ij}\log\frac{A_{ij}}{B_{ij}} - A_{ij} + B_{ij}\right) + \alpha\sum_{i,j}u_{ij} - \beta\sum_{i}v_{ii}, \tag{11}$$

where $B = WH$, $[u_{ij}] = W^TW$, $[v_{ij}] = HH^T$, α, $\beta > 0$ are some constants.

Table 3
The Errors of five image matrices of the first person on ORL face database by MU algorithms, the number of iterations is 100.

$p_{90\%}$	35	26	35	34	37
SVD–NMF	0.1015	0.0931	0.1039	0.1061	0.1041
NNDSVD	0.1149	0.0965	0.1132	0.1084	0.1203
RandomNMF	0.1215	0.1098	0.1207	0.1173	0.1178

Table 4
The Errors of five image matrices of the first person on ORL face database by MU algorithms, the number of iterations is 100.

$p_{80\%}$	22	16	22	21	23
SVD–NMF	0.1245	0.1115	0.1272	0.1300	0.1315
NNDSVD	0.1364	0.1137	0.1318	0.1308	0.1354
RandomNMF	0.1411	0.1223	0.1398	0.1333	0.1425

In paper [12], the updated rules of MU algorithm can be expressed as the following form:

$H \leftarrow H.*((W^TA)./(W^TWH))$

$W \leftarrow W.*((AH^T)./(WHH^T))$

and the updated iterative rules of LNMF in paper [14] has the following form:

$$h_{kl} \leftarrow \sqrt{h_{kl}\sum_i x_{il}\frac{w_{ik}}{\sum_k w_{ik}h_{kl}}}$$

$$w_{kl} \leftarrow \frac{w_{kl}\sum_j x_{kj}\frac{h_{lj}}{\sum_k b_{kl}h_{lj}}}{\sum_j h_{lj}}$$

$$w_{kl} \leftarrow \frac{w_{kl}}{\sum_k w_{kl}}.$$

We also can apply SVD–NMF to other well known NMF algorithms. In this paper we do not do that. In next Section 3, we give numerical results to show errors with increasing iterations and the factorization ranks for SVD–NMF, NNDSVD and RandomNMF. The error we assumed in this paper the following form:

$$error = \frac{\|Z - WH\|_F}{\|Z\|_F} \tag{12}$$

3. Numerical experiments

In this section, we show numerical results on two facial databases: ORL and YALE which are already introduced in Section 2.1 and one object database Coil-20 with two versions which are from Columbia Object Image Library [13]. In the first version named unprocessed consists of images for five of the objects that contain both the object and the background. It is denoted as COIL20-unproc. The second version named processed contains images for 20 objects in which the background has been discarded (and the images consist of the smallest square that contains the object), which is denoted as COIL20-proc. Two natural images from Matlab are considered, one is football.jpg with 256 × 320 pixels and another one is kids.tif with 318 × 400 pixels.

3.1. Numerical results for MU algorithm

Tables 3–5 show the influences of factorization with different rank p, which are calculated by choosing rule with various extracting proportions. They are 80, 90 and 100% respectively. We conclude when extracting proportion is 80%, the factorization ranks are small with much larger errors than those of 90%, which means the factorizations are not good. Whereas much smaller errors can be obtained when extracting proportion is 100%, but with large ranks, which can not reduce dimension well. This situation is not good for high dimensional data reduction. And when extracting proportion is 90% can overcome both of problems under limits. In this case the vast majority p calculated by choosing rule can satisfy condition (1) with relative smaller errors like Tables 3–5 depict.

Table 5
The Errors of five image matrices of the first person on ORL face database by MU algorithms, the number of iterations is 100.

$p_{100\%}$	61	49	67	65	73
SVD–NMF	0.0726	0.0690	0.0767	0.0793	0.0755
NNDSVD	0.0985	0.0738	0.0907	0.0861	0.0914
RandomNMF	0.0981	0.0898	0.0948	0.0874	0.0848

In the next numerical experiments, we choose first five images of each of 40 persons to conduct a 200 × 10304 matrix on ORL and on YALE all images conduct one 161 × 10000 matrix. And on COIL20-unproc, each object has 72 images with pixel 448 × 416, hence we get a 360 × 186368 image matrix. The same as COIL20-unproc, on COIL20-proc each object has 72 images with pixel 128 × 128, therefore we obtain a 1440 × 16384 image matrix.

Fig. 1 shows the errors obtained by SVD–NMF, NNDSVD and RandomNMF for MU algorithm on the four databases with the increase of iterations. p in the titles is the rank of factorization obtained by choosing rule when the extracting proportion is 0.9. We can observe that in this case the errors obtained by SVD–NMF much smaller than those using NNDSVD and RandomNMF. Based on formula (12), these factorizations using SVD–NMF much more approximate to original matrix on these four databases. Except for results on YALE, of which the errors using RandomNMF similar with those exploiting SVD-NMF. In the other three cases, SVD-NMF has smaller errors with increasing iterations and smallest error when iteration is 1000. We conclude that in these databases SVD-NMF has rapid error reduction and smaller overall error when iterations are equal with NNDSVD and RandomNMF.

Fig. 2 gives us the reconstruction images using the factorization of SVD–NMF, NNDSVD and RandomNMF for the ten image matrices on ORL and YALE face database, respectively, as we mentioned before. In this case, the iteration number is 100. Because faces on YALE database with more different expressions: happy, sad, normal, sleepy or wink than that of ORL database, therefore the reconstruct results of YALE are a little bit of worse than that of ORL. We deduce that it is more difficult to fulfil face recognition on YALE database than ORL.

In Fig. 3, we use two natural images from Matlab to compare the errors of SVD–NMF, NNDSVD and RandomNMF for MU algorithm with increasing iterations. Rank p of factorization is 78 for image football.jpg using choosing rule. And the second data derives from image kids.tif, of which the factorization rank p is 140. Fig. 3 shows that SVD–NMF has faster convergence than another two methods in MU algorithm for NMF.

3.2. Numerical results for LNMF algorithm

In this section, we will evaluate the properties of SVD–NMF, NNDSVD and RandomNMF for LNMF algorithm. Since the data matrix sizes we evaluated in section 3.1 are very large, we choose each last object of ORL, YALE, COIL20-unproc, and COIL20-proc to evaluate. With increasing the iterations, Fig. 4 depicts the errors obtained by SVD–NMF, NNDSVD and RandomNMF for the four databases, respectively. Results are a little different from that of using MU algorithm. On ORL, SVD-NMF converges faster than NNDSVD when iterations are small, such as 30–200, then they have much similar performances until iterations increase to almost 800. When iterations are larger than 800, NNDSVD has relatively smaller errors than SVD–NMF. In this case, we note that RandomNMF has the smallest errors among the three methods, this because SVD–NMF and NNDSVD is based on problem 2 rather than objective function (11). Results on YALE and

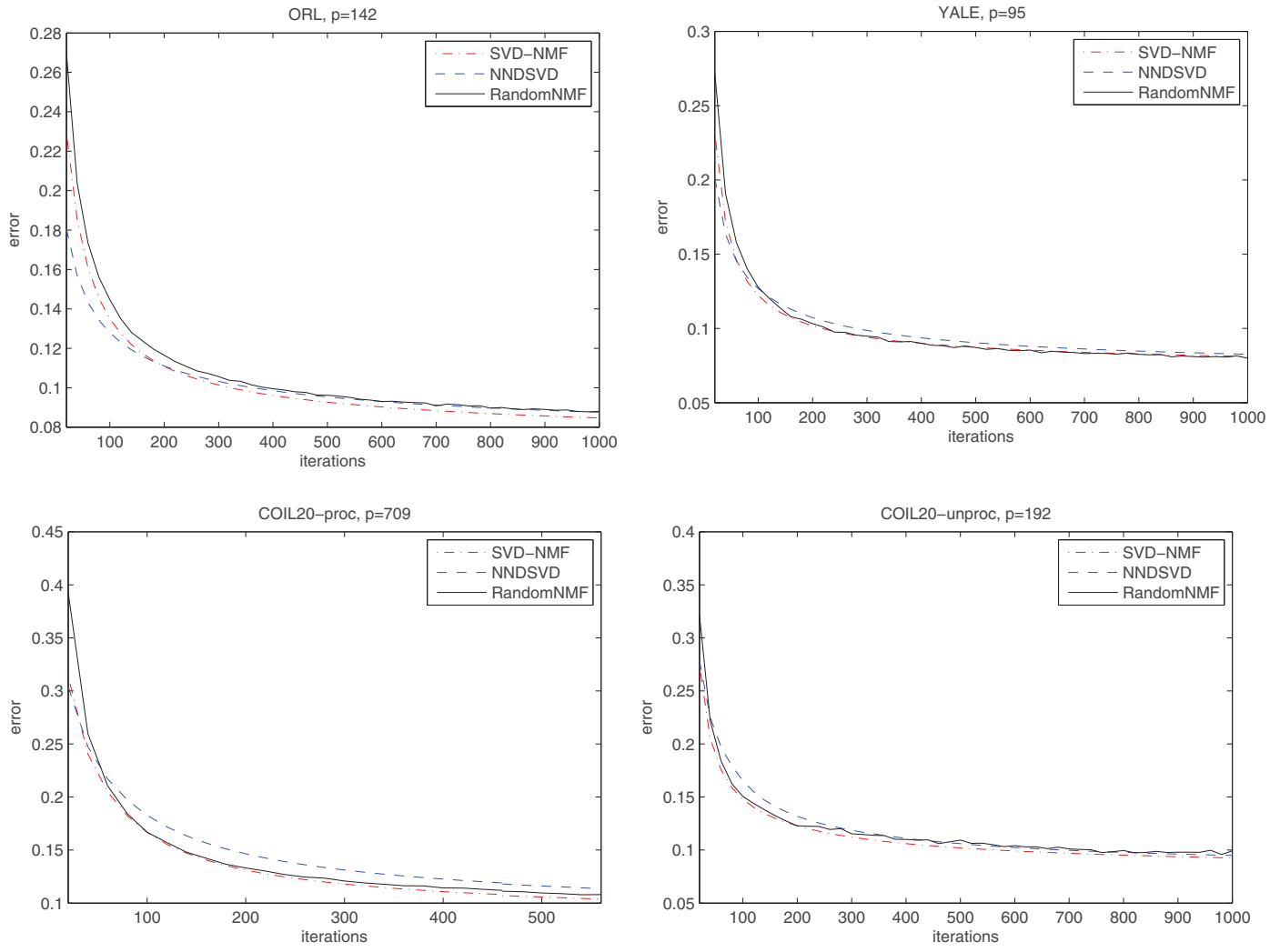


Fig. 1. Errors of comparison for ORL, YALE and two versions of COIL20 using MU algorithm.

(a) ORL (b) YALE

Fig. 2. The reconstruction of two face databases. The first row using SVD–NMF, second row using NNDSVD and the last row using RandomNMF for MU algorithm.

COIL20-unproc are converse, which SVD–NMF has much smaller errors than NNDSVD on YALE, but opposite results hold on COIL20-unproc occur. On COIL20-proc, SVD–NMF has a little bit smaller errors than NNDSVD with increasing iterations to 700 , when iterations larger than it, they have almost same errors.

Fig. 5 is the reconstruction of the second person on ORL and YALE face database. Each subject has five images with different expressions, light or other factors. These reconstruct images in Fig. 5 have more details such as clearer eyebrows, nose, lips than those of in Fig. 2. This holds because LNMF can impose the local information of the whole face. Images have significant contrasting in (b) of Fig. 3 because the first image is affected by lightness whereas others do not have it.

3.3. Numerical results on Coil-20 database

In this Section, we apply SVD–NMF, NNDSVD and RandomNMF to each single image of all objects on COIL20-unproc and COIL20-proc. And we show the number of images using SVD–NMF of which the errors smaller than that of using NNDSVD and RandomNMF. We use

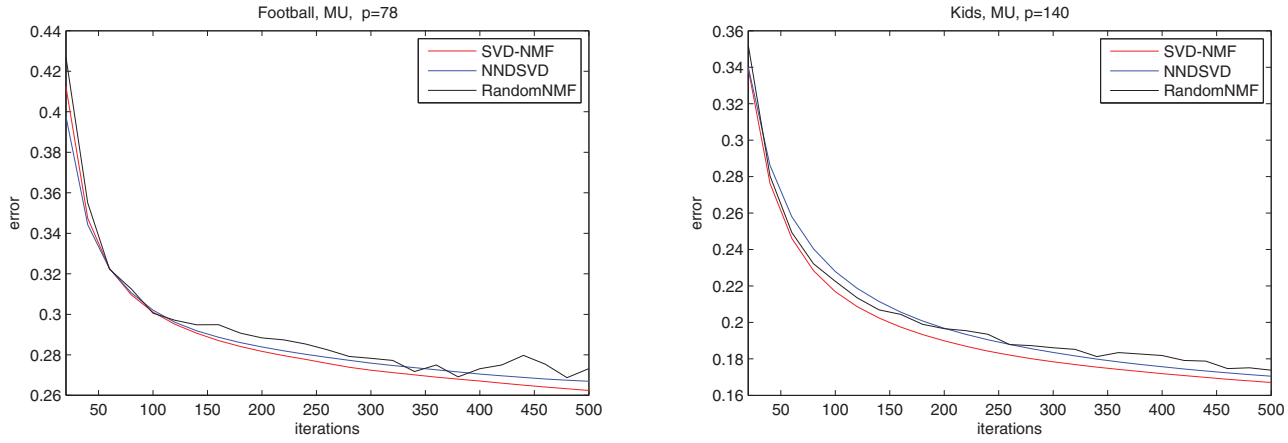


Fig. 3. Errors of comparison of two natural images from Matlab.

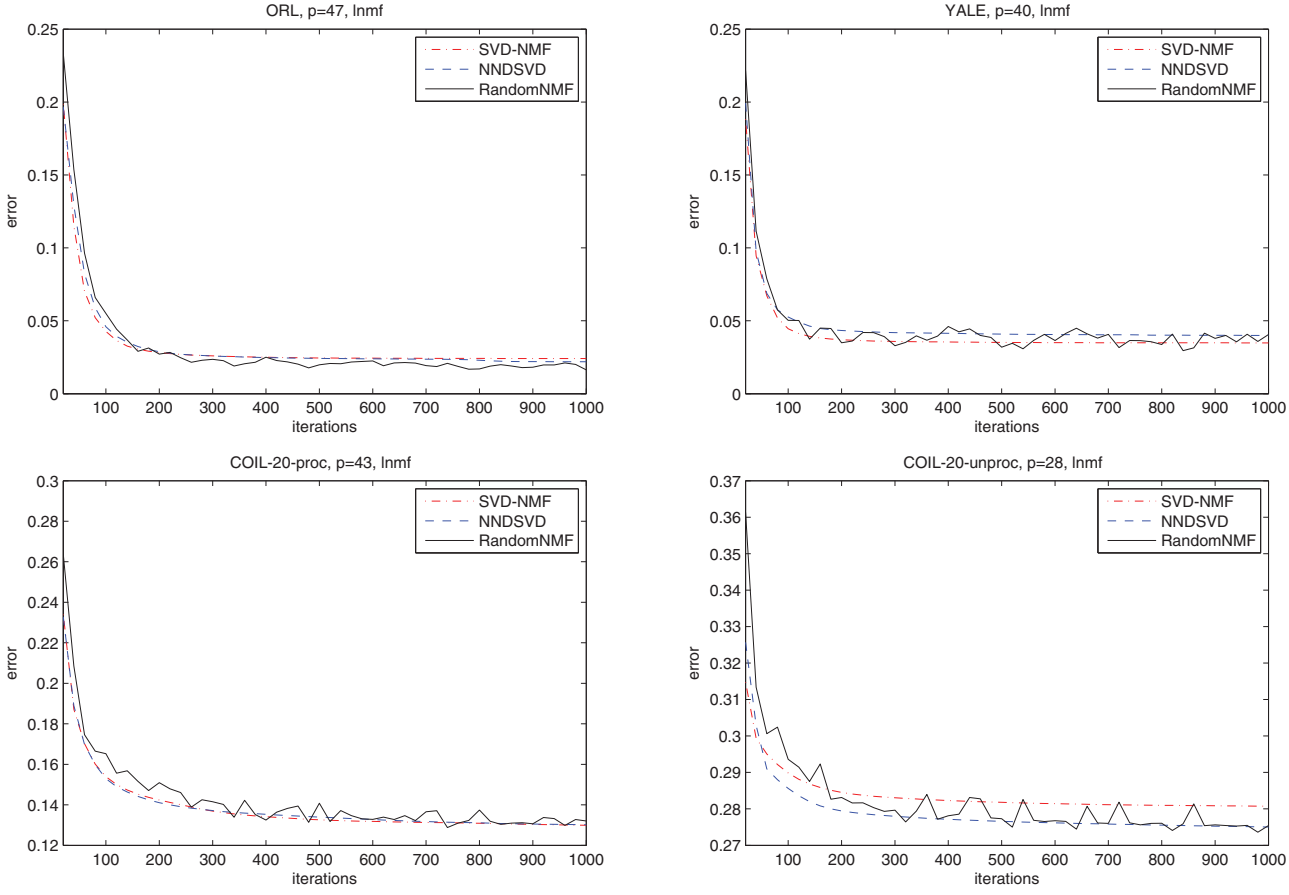


Fig. 4. Errors of comparison for ORL, YALE and two versions of COIL20 using lnmf with $\alpha = 1$, $\beta = 0.1$.

Table 6
The number of images using SVD–NMF of which the errors smaller than that of using NNDSVD and RandomNMF on COIL20-unproc.

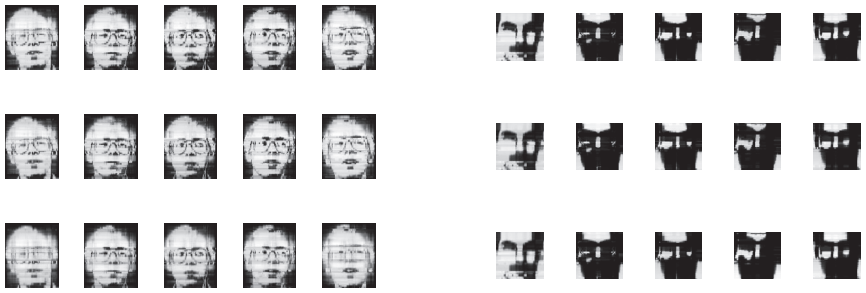
Iteration	100	300	500	700	1000
$Num_{E(SVD--NMF)<E(NNDSVD)}$	333	311	301	291	283
$Num_{E(SVD--NMF)<E(RandomNMF)}$	341	322	308	304	300

Table 7
The number of images using SVD–NMF that the errors smaller than that of using NNDSVD and RandomNMF on COIL20-proc.

Iteration	100	300	500	700	1000
$Num_{E(SVD--NMF)<E(NNDSVD)}$	1259	1202	1148	1103	1065
$Num_{E(SVD--NMF)<E(RandomNMF)}$	1384	1296	1240	1204	1180

$Num_{E(.)<E(.)}$ to stand for this quantity in Tables 6 and 7. Table 6 shows that when iteration equal to 100, the errors of more than 92.5% of images using SVD–NMF smaller than those of using NNDSVD and RandomNMF. Whereas iteration equal to 1000, the proportion more than 78.6%. The results mean that on COIL20-unproc, MU algorithm using SVD–NMF have faster convergences and smaller errors. The similar results show in Table 7, which on COIL20-proc, SVD–NMF has better property than NNDSVD and RandomNMF.

As we introduced in Section 1, NMF has a lot of applications. Good initialization will bring good factorization, then NMF can be applied



(a) ORL (b) YALE

Fig. 5. The reconstruction of two face databases and the first row using SVD–NMF, second row using NNDSVD and the last row using RandomNMF for LNMF algorithm.

well for many fields. SVD–NMF as a new method to impose initialization of NMF is proposed in this paper. It has some goodness: (i) it can be easily combined with other NMF algorithms; (ii) the computational cost is cheap because we only compute the singular triplets once; (iii) it can reach fast convergence; (iv) it is simple and can be implemented easily. But we still need some other strategy to deal with the negative entries of singular triplets. When we made experiences, we found that if we change the negative elements we can get better or worse results. Hence, how to implement the negative entries rather than replace them by the absolute values is the future work.

Acknowledgments

Prof. Alessandra De Rossi and Dr. Roberto Cavoretto who are my tutors, gave me a lot of favors to complete this paper. They gave me a lot of corrections and suggestions. The most important is that they helped me to familiar with the environment of University of Turin so that I can work well in our University. I thank them for their advises and favors.

We would like to thank the reviewer for their careful work and for the effort in thoroughly reviewing our paper and the constructive comments.

References

[1] N. L. Amy, D. M. Carl, A. Russell, C. James, D. David, Algorithms, initializations, and convergence for the nonnegative matrix factorization, SAS Technical Report, 2014 arXiv:1407.7299.
[2] C. Boutsidis, E. Gallopoulos, Svd based initialization: a head start for nonnegative matrix factorization, Pattern Recognit. 41 (4) (2008) 1350–1362.
[3] X. Chen, L. Gu, S. Li, H. Zhang, Learning representative local features for face detection, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1, 2001, pp. 1126–1131.
[4] C. Ding, X. He, H. Simon, On the equivalence of non-negative matrix factorization and spectral clustering, in: Proceedings of the SIAM Conference on Data Mining, 2005.
[5] C. Ding, T. Li, M. Jordan, Non-negative matrix factorization for combinatorial optimization: spectral clustering, graph matching, and clique finding, in: Proceedings of the Eighth IEEE International Conference on Data Mining, 2008, pp. 183–192.
[6] C. Ding, T. Li, W. Peng, Non-negative matrix factorization and probabilistic latent semantic indexing: equivalence, chi-square statistic, and a hybrid method, in: Proceedings of the National Conference on Artificial Intelligence, 2006.
[7] C. Ding, T. Li, W. Peng, On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing, Comput. Stat. Data Anal. 52 (2008) 3913–3927.
[8] N. Higham, Matrix nearest problems and applications, in: M. Gover, S. Barnett (Eds.), Applications of Matrix Theory, Oxford University Press, 1989, pp. 1–27.
[9] A. Janecek, Y. Tan, Using population based algorithms for initializing nonnegative matrix factorization, in: ICSI 2011, Part II, LNCS 6729, 2011, pp. 307–316.
[10] C. U. C. Laboratory, The orl database of faces. http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.
[11] D.D. Lee, H.S. Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (1999) 788–791.
[12] D.D. Lee, H.S. Seung, Algorithms for non-negative matrix factorization, In Adv. Neural Inf. Process. 13 (2000) 556–562.
[13] S. Nene, S.K. Nayar, H. Murase, Columbia object image library (coil-20), Technical Report CUCS-005-96, 1996.
[14] S.Z. Li, X.W. Hou, H.J. Zhang, Q.S. Cheng, Learning spatially localized, parts-based representation, in: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, 2001, pp. 207–212.
[15] M. Turk, A. Pentland, Eigenfaces for recognition, Cogn. Neurosci. Sci. 3 (1) (1991) 71–86.
[16] M. Turk, A. Pentland, Face recognition using eigenfaces, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1991, pp. 586–591.
[17] U. C. Version, Yale university face database. http://vision.ucsd.edu/content/yale-face-database.
[18] S. Wild, Seeding non-negative matrix factorization with the spherical k-means clustering, Master's thesis, University of Colorado, 2002.
[19] S. Wild, J. Curry, A. Dougherty, Improving non-negative matrix factorizations through structured initialization, Pattern Recognit. 37 (2004) 2217–2232.
[20] J. Yu, R. Hong, M. Wang, J. You, Image clustering based on sparse patch alignment framework, Pattern Recognit. 47 (2014) 3512–3519.
[21] J. Yu, Y. Rui, Y.Y. Tang, D. Tao, High-order distance based multiview stochastic learning in image classification, IEEE Trans. Cybern. 44 (12) (2014) 2431–2442.
[22] J. Yu, D. Tao, M. Wang, Adaptive hypergraph learning and its application in image classification, IEEE Trans. Image Process. 7 (2012) 3262–3272.
[23] Z. Yuan, E. Oja, Projective non-negative matrix factorization for image compression and feature extraction, in: Proceedings of the 14th Scandinavian Conference on Image Analysis, 2005, pp. 333–342.