Data Min Knowl Disc (2013) 26:435–451
DOI 10.1007/s10618-012-0265-y

Solving non-negative matrix factorization by alternating least squares with a modified strategy

Hongwei Liu · Xiangli Li · Xiuyun Zheng

Received: 1 October 2010 / Accepted: 30 March 2012 / Published online: 18 April 2012

Abstract Non-negative matrix factorization (NMF) is a method to obtain a representation of data using non-negativity constraints. A popular approach is alternating non-negative least squares (ANLS). As is well known, if the sequence generated by ANLS has at least one limit point, then the limit point is a stationary point of NMF. However, no evdience has shown that the sequence generated by ANLS has at least one limit point. In order to overcome this shortcoming, we propose a modified strategy for ANLS in this paper. The modified strategy can ensure the sequence generated by ANLS has at least one limit point, and this limit point is a stationary point of NMF. The results of numerical experiments are reported to show the effectiveness of the proposed algorithm.

Keywords Non-negative matrix factorization · Alternating non-negative least squares

1 Introduction

Non-negative matrix factorization (NMF) is an important and unifying topic in signal processing and linear algebra, which has found numerous applications in many other areas, such as text mining Paul Pauca et al. (2004), subsystem identification Kim

Responsible editor: Chih-Jen Lin.

H. Liu · X. Li (✉) · X. Zheng
Department of Mathematics, Xidian University, Xi'an 710071, People's Republic of China
e-mail: lixiangli213@gmail.com

X. Li
College of Mathematics and Computing Science, Guilin University of Electronic Technology,
Guilin 541004, People's Republic of China

Philip and Tidor (2003), cancer class discovery Gao and Church (2005); Kim and Park (2007), astronomical images Richardson (1972), music transcription Cho and Choi (2005), neurobiology (gene separation) Brunet et al. (2004); Rao and Shepherd (2004), and data analysis(e.g.,pattern recognition, segmentation, clustering, dimensionality reduction) (Guillamet et al. 2003, 2002; Guillamet and Vitria 2002; Ahn et al. 2004; Lee et al. 2002; Li et al. 2005; Pascual-Montano et al. 2006; Okun and Priisalu 2009; Wang et al. 2005; Liu and Zheng 2004; Spratling 2006).

Given a data matrix $A \in R^{m \times n}$ with non-negative entries, find a factorization where

$$A \approx WH, \tag{1}$$

W and H are nonnegative matrices of dimensions $m \times r$ and $r \times n$, respectively. The term r is usually chosen such that $r \ll \frac{mn}{m+n}$.

The usual method of solving this is to reformulate (1) as the following optimization problem:

$$\min_{W,H} F(W, H) \equiv \frac{1}{2}\|A - WH\|_F^2, \quad s.t. \quad W, H \geq 0, \tag{2}$$

where $W, H \geq 0$ means that all elements of W and H are non-negative and $\|\cdot\|_F$ is the Frobenius norm. A point (W, H) is a stationary point of (2) if and only if

$$\begin{aligned} W \geq 0, \qquad H \geq 0, \\ \nabla_W F(W, H) = (WH - A)H^T \geq 0, \\ \nabla_H F(W, H) = W^T(WH - A) \geq 0, \\ \langle W, \nabla_W F(W, H)\rangle = 0, \\ \langle H, \nabla_H F(W, H)\rangle = 0. \end{aligned} \tag{3}$$

Since NMF was first proposed by Paatero and Tapper (1994), many methods have been proposed for NMF, such as multiplicative update algorithms Lee and Seung (1999, 2001), gradient descent methods Chu et al. (2004) and alternating non-negative least squares Paatero and Tapper (1994). Among the existing methods, a popular approach is alternating non-negative least squares(ANLS) Paatero and Tapper (1994).

Alternating non-negative least squares (ANLS) Paatero and Tapper (1994):

Iterate the following until a stopping criterion is satisfied:

$$W^{k+1} = \arg\min_{W \geq 0} F\left(W, H^k\right), \tag{4}$$

$$H^{k+1} = \arg\min_{H \geq 0} F\left(W^{k+1}, H\right). \tag{5}$$

ANLS is also known as "block coordinate descent" approach in bound-constrained optimization. ANLS algorithms can be very fast, work well in practice, have fast convergence. Moreover, 0 elements in ANLS algorithms are not be locked. Based on these advantages of ANLS, there are many scholars who were interested in ANLS, such as Zdunek and Cichocki (2006, 2008), Lin (2007), and Paatero (1999).

For a simpler description, we focus on (5) and rewrite it as

$$\min_H f(H) \equiv \frac{1}{2}\|A - WH\|_F^2 \quad (6)$$
$$s.t. \quad H \geq 0,$$

where A and W are constant matrices. H is a stationary point of (6) if and only if

$$H \geq 0, \ \nabla f(H) \geq 0, \ \langle H, \nabla f(H) \rangle = 0, \quad (7)$$

where $\nabla f(H) = W^T(WH - A)$. Moreover, we can also rewrite (4) as a form similar to (6):

$$\min_W f(W) \equiv \frac{1}{2}\|A^T - H^T W^T\|_F^2 \quad (8)$$
$$s.t. \quad W \geq 0,$$

where A^T and H^T are constant matrices. We can write (6) as follows:

$$f(H) = \frac{1}{2}\langle H, W^T WH \rangle - \langle W^T A, H \rangle + \frac{1}{2}\|A\|_F^2$$

$f(H)$ is a quadratic function, and $W^T W$ is an $r \times r$ positive semi-definite matrix. Based on these good properties, many optimization methods can be applied to subproblems (4) and (5). In Lin (2007), author uses a projected gradient method to solve subproblems (4) and (5), and points out any limit point of the sequence $\{W^k, H^k\}$ generated by ANLS is a stationary point of (2). However, the remaining issue is whether the sequence $\{W^k, H^k\}$ has at least one limit point.

In order to ensure that the sequence $\{W^k, H^k\}$ has at least one limit point, we propose a modified strategy for ANLS in this paper. The modified strategy can ensure the sequence generated by ANLS has at least one limit point, and this limit point is a stationary point of NMF. The modified strategy can also be applied to other ANLS-based methods, such as Kim et al. (2007); Kim and Park (2008).

We sum up here briefly our notations. Let $A = (a_{ij})_{m\times n}$, $\|A\|_F = \sqrt{\sum_i \sum_j a_{ij}^2}$, $Tr(A) = \sum_i a_{ii}$. Let $B \in R^{m\times n}$, $\langle A, B \rangle = Tr(AB^T)$.$A_{\cdot i}$ denotes the ith column of A, $A_{i\cdot}$ denotes the ith row of A. For any given vector y, $y_- = \min\{y, 0\}$. Let $\|x\|$ denote any norm of a vector x, Let $\|x\|_2$ denote 2-norm of a vector x. Let V be a set, $|V|$ denote the the number of elements of V.

The rest of the paper is organized as follows. A modified strategy is discussed in Sect. 2. In Sect. 3, we introduce our algorithm. In Sect. 4, we discuss subproblems. The convergence behavior of the algorithm is investigated in Sect. 5. The results of the numerical experiments are reported in Sect. 6. Finally, we give our conclusions in Sect. 7.

2 A modified strategy

In Lee and Seung (2001), Lee and Seung proposed a multiplicative update algorithm to solve NMF. The algorithm is simple and easy to implement. But Lee and Seung (2001) did not declare that any limit point is stationary. In Gonzalez and Zhang (2005), numerical results indicated that the multiplicative update algorithm may fail to converge to a stationary point. However, Lin (2007) has pointed out that due to possible numerical inaccuracy, a mathematical example is desired before drawing conclusions. In Lin (2007), Lin proposed a modified strategy, in which, if the whole column of W is zero, then it as well as the corresponding row in H are unchanged. This modified strategy can ensure the modified sequence is bounded. Moreover, Lin has proved that any limit point of the modified sequence is a stationary point of (2). Inspired by the works of Lin, we propose a modified strategy and apply it to ANLS.

In NMF, we find that

$$WH = \sum_{j=1}^{r} W_{\cdot j} H_{j\cdot} = \sum_{\|W_{\cdot j}\|\cdot\|H_{j\cdot}\|\neq 0} \lambda_j W_{\cdot j} \cdot \frac{1}{\lambda_j} H_{j\cdot}, \text{ for } \forall \lambda_j > 0.$$

For $\|W_{\cdot j}\| \cdot \|H_{j\cdot}\| \neq 0$, let

$$\bar{W}_{\cdot j} = \lambda_j W_{\cdot j}, \quad \bar{H}_{j\cdot} = \frac{1}{\lambda_j} H_{j\cdot} \text{ and } \|\bar{W}_{\cdot j}\| = \alpha \|\bar{H}_{j\cdot}\|, \ \alpha > 0.$$

By solving the above equation, we can get $\lambda_j = \sqrt{\alpha \frac{\|H_{j\cdot}\|}{\|W_{\cdot j}\|}} > 0$. So we state strategy I as follows.

Strategy I

Let (W^k, H^k) be generated by ANLS.

$$\bar{W}^k = W^k \Lambda_1, \ \bar{H}^k = \Lambda_2 H^k,$$

where $\Lambda_1 = diag(\lambda_j^1), \quad \Lambda_2 = diag(\lambda_j^2), \quad j = 1, 2, \ldots, r,$

$$\lambda_j^1 = \begin{cases} 0\,, & \text{if } \|W_{\cdot j}^k\| \cdot \|H_{j\cdot}^k\| = 0 \\ \sqrt{\alpha \frac{\|H_{j\cdot}^k\|}{\|W_{\cdot j}^k\|}}\,. & \text{otherwise} \end{cases}, \tag{9}$$

$$\lambda_j^2 = \begin{cases} 0\,, & \text{if } \|W_{\cdot j}^k\| \cdot \|H_{j\cdot}^k\| = 0 \\ \sqrt{\frac{\|W_{\cdot j}^k\|}{\alpha \|H_{j\cdot}^k\|}}\,. & \text{otherwise} \end{cases}. \tag{10}$$

Strategy I can ensure the sequence generated by ANLS is bounded, and the convergence of it has been established, see Lemma 2 and Theorem 2. The following properties can be easily obtained.

Property I: $\bar{W}^{k+1} \geq 0$, $\bar{H}^{k+1} \geq 0$, $W^{k+1}H^{k+1} = \bar{W}^{k+1}\bar{H}^{k+1}$, $F(W^{k+1}, H^{k+1}) = F(\bar{W}^{k+1}, \bar{H}^{k+1})$.
Property II: If H^{k+1} is a stationary point of $\min_{H\geq 0} F(W^{k+1}, H)$, then $\bar{H}^{k+1}$ is a stationary point of $\min_{H\geq 0} F(\bar{W}^{k+1}, H)$, i.e.

$$\begin{aligned} &\bar{H}^{k+1} \geq 0, \\ &(\bar{W}^{k+1})^T(\bar{W}^{k+1}\bar{H}^{k+1} - A) \geq 0, \\ &\langle \bar{H}^{k+1}, (\bar{W}^{k+1})^T(\bar{W}^{k+1}\bar{H}^{k+1} - A)\rangle = 0, \end{aligned} \tag{11}$$

Property III: $\|\bar{W}^{k+1}_{\cdot j}\| = \alpha \|\bar{H}^{k+1}_{j\cdot}\|$ for any $j = 1, \cdots, r$.

3 Algorithm

Now, we state our algorithm.

Algorithm 1

(s.0) Given starting point $(\bar{W}^0, \bar{H}^0) \geq 0$, $\alpha > 0$. *Set* $k = 0$.

(s.1) Stop if $(\bar{W}^k, \bar{H}^k)$ *satisfies* (3).
(s.2)

$$W^{k+1} = \arg\min_{W\geq 0} F(W, \bar{H}^k), \tag{12}$$

$$H^{k+1} = \arg\min_{H\geq 0} F(W^{k+1}, H). \tag{13}$$

(s.3) Modify (W^{k+1}, H^{k+1}) *to* $(\bar{W}^{k+1}, \bar{H}^{k+1})$ *using* ***Strategy I***.
(s.4) Let $k = k + 1$. *Go to Step 1.*

Remark 1 Given any random initial $(\bar{W}^0, \bar{H}^0) \geq 0$. Similar to analysis of Lin (2007), we can get

$$F(\bar{W}^k, \bar{H}^k) = F(W^k, H^k) \leq F(W^k, 0) = F(0, 0) \text{ for } k \geq 1,$$

the strict inequality generally holds, i.e. $F(\bar{W}^k, \bar{H}^k) < \frac{1}{2}\|A\|_F^2$.

4 Subproblems

There are many methods to solve subproblems of (12) and (13), such as projected gradient methods Lin (2007), an NMF algorithm which is based on alternating non-negativity constrained least squares (NMF/ANLS) and the active set method Kim and Park (2008), fast Newton-type methods Kim et al. (2007).

In (13), W^k is a constant matrix, we may write W^k with W. Then, $\min_{H\geq 0} F(W^k, H)$ can be written as (6). The following theorem tells us (6) has a global minimizer on $R_+^{n\times r}$.

Theorem 1 *(**Frank-Wolfe Theorem**) If a quadratic function is bounded from below on a nonempty polyhedron, it attains a minimum there.*

Let $\mathcal{F}(H^0) = \{H \mid f(H) \leq f(H^0), \quad H \geq 0\}$. If the whole column of W is zero, then the level set $\mathcal{F}(H^0)$ is not bounded.

$$W^T W = \begin{pmatrix} \langle W_{\cdot 1} W_{\cdot 1} \rangle & \langle W_{\cdot 1}, W_{\cdot 2} \rangle & \dots & \langle W_{\cdot 1}, W_{\cdot r} \rangle \\ \langle W_{\cdot 2}, W_{\cdot 1} \rangle & \langle W_{\cdot 2}, W_{\cdot 2} \rangle & \dots & \langle W_{\cdot 2}, W_{\cdot r} \rangle \\ \vdots & \vdots & \vdots & \vdots \\ \langle W_{\cdot r}, W_{\cdot 1} \rangle & \langle W_{\cdot r}, W_{\cdot 2} \rangle & \dots & \langle W_{\cdot r}, W_{\cdot r} \rangle \end{pmatrix}. \tag{14}$$

Let

$$M = \begin{pmatrix} m_{11} & m_{12} & \dots & m_{1r} \\ m_{21} & m_{22} & \dots & m_{2r} \\ \vdots & \vdots & \vdots & \vdots \\ m_{r1} & m_{r2} & \dots & m_{rr} \end{pmatrix} = W^T W. \tag{15}$$

If $m_{ll} = 0, \quad l \in \{1, \cdots, r\}$, then $W_{\cdot l} = 0, \quad M_{\cdot l} = 0, \quad M_{l \cdot} = 0$. Let

$$L = \{l \mid m_{ll} = 0, \ \forall l\}, \quad V \in \{1, 2, \cdots, r\} \setminus \{L\},$$

then

$$\begin{aligned} f(H) &= \frac{1}{2}\langle H, W^T W H\rangle - \langle W^T A, H\rangle + \frac{1}{2}\|A\|_F^2 \\ &= \frac{1}{2}\langle H_{V\cdot}, M_{VV} H_{V\cdot}\rangle - \langle (W^T A)_{V\cdot}, H_{V\cdot}\rangle + \frac{1}{2}\|A\|_F^2, \end{aligned}$$

where

$$M_{VV} = \begin{pmatrix} m_{i_1 i_1} & \dots & m_{i_d i_d} \\ \vdots & \vdots & \vdots \\ m_{i_d i_1} & \dots & m_{i_d i_d} \end{pmatrix},$$

$d = |V|, \ i_1, i_2, \cdots, i_d \in V$.

Let

$$g(H_{V\cdot}) = \frac{1}{2}\langle H_{V\cdot}, M_{VV} H_{V\cdot}\rangle - \langle (W^T A)_{V\cdot}, H_{V\cdot}\rangle + \frac{1}{2}\|A\|_F^2.$$

Based on the above analysis, we know that solving (6) is equivalent to solving $\min_{H_{V\cdot} \geq 0} g(H_{V\cdot})$, $H_{L\cdot}$ keeps invariant.

Let

$$\mathcal{L}(H^0_{V\cdot}) = \{H_{V\cdot} \mid g(H_{V\cdot}) \leq g(H^0_{V\cdot}), \quad H_{V\cdot} \geq 0\}.$$

We can show that the level set $\mathcal{L}(H^0_{V\cdot})$ is bounded.

Lemma 1 *The level set $\mathcal{L}(H^0_{V\cdot})$ is bounded.*

Proof Suppose that $H_{V\cdot} \in \mathcal{L}(H^0_{V\cdot})$ and $\|H_{V\cdot}\|_F \to \infty$. Let

$$M'_{VV} = diag\,(m_{tt})\,,\ t \in \{i_1, \cdots, i_d\}.$$

Since $H_{V\cdot} \geq 0, \quad M_{VV} \geq 0$, we have

$$\begin{aligned} g(H_{V\cdot}) &\geq \frac{1}{2}\langle H_{V\cdot}, M'_{VV} H_{V\cdot}\rangle - \langle (W^T A)_{V\cdot}, H_{V\cdot}\rangle + \frac{1}{2}\|A\|_F^2 \\ &\geq \frac{1}{2} m_{min}\|H_{V\cdot}\|_F^2 - \langle (W^T A)_{V\cdot}, H_{V\cdot}\rangle + \frac{1}{2}\|A\|_F^2 \\ &\geq \frac{1}{2} m_{min}\|H_{V\cdot}\|_F^2 - \|(W^T A)_{V\cdot}\|_F \|H_{V\cdot}\|_F + \frac{1}{2}\|A\|_F^2 \\ &\quad \to \infty, \end{aligned} \tag{16}$$

where $m_{min} = \min\{m_{i_1 i_1}, \ldots, , m_{i_d i_d}\}$. This is impossible. So our assumption is incorrect. □

Corollary 1 *If $L = \emptyset$, then $\mathcal{F}(H^0)$ is bounded.*

5 Convergence analysis

In this section, we analyze the global behavior of the algorithm given in the previous section. First, we discuss the boundedness of the sequence generated by Algorithm 1.

Lemma 2 *Let $\{\bar{W}^k, \bar{H}^k\}$ be generated by Algorithm 1. Then $\{\bar{W}^k, \bar{H}^k\}$ is bounded.*

Proof Since

$$H^k = \arg\min_{H \geq 0} F(W^k, H), \tag{17}$$

by the first-order optimality conditions, H^k is a stationary point of $\min_{H\geq 0} F(W^k, H)$, so

$$\langle (W^k)^T (W^k H^k - A), H^k\rangle = 0,$$

$$\begin{aligned} \|A - W^k H^k\|_F^2 &= \|A\|_F^2 - 2\langle A, W^k H^k\rangle + \|W^k H^k\|_F^2 \\ &= \|A\|_F^2 - 2\langle A - W^k H^k, W^k H^k\rangle - \|W^k H^k\|_F^2 \\ &= \|A\|_F^2 - \|W^k H^k\|_F^2 \\ &\geq 0. \end{aligned} \tag{18}$$

Thus,

$$\|\bar{W}^k \bar{H}^k\|_F = \|W^k H^k\|_F \leq \|A\|_F.$$

By Property III, $\|\bar{W}^k_{\cdot j}\| = \alpha \|\bar{H}^k_{j\cdot}\|$. According to equivalence of vector norms, there exist $c_1,\ c_2,\ c_3,\ c_4 > 0$, such that

$$c_2\|\bar{W}^k_{\cdot j}\|_2 \leq \|\bar{W}^k_{\cdot j}\| \leq c_1\|\bar{W}^k_{\cdot j}\|_2,$$
$$c_3\|\bar{H}^k_{j\cdot}\|_2 \leq \|\bar{H}^k_{j\cdot}\| \leq c_4\|\bar{H}^k_{j\cdot}\|_2.$$

Thus,

$$\|\bar{W}^k_{\cdot j}\|_2 \geq \frac{\alpha c_3\|\bar{H}^k_{j\cdot}\|_2}{c_1} \text{ and } \|\bar{H}^k_{j\cdot}\|_2 \geq \frac{c_2\|\bar{W}^k_{\cdot j}\|_2}{\alpha c_4}.$$

Since $\bar{W}^k_{\cdot j}\bar{H}^k_{j\cdot} \geq 0,\ \ j = 1, 2, \cdots, r$, we deduce

$$\begin{aligned} \|\bar{W}^k \bar{H}^k\|_F^2 &= \left\langle \sum_{j=1}^r \bar{W}^k_{\cdot j}\bar{H}^k_{j\cdot}, \sum_{j=1}^r \bar{W}^k_{\cdot j}\bar{H}^k_{j\cdot} \right\rangle \\ &\geq \sum_{j=1}^r \|\bar{W}^k_{\cdot j}\bar{H}^k_{j\cdot}\|_F^2 \\ &= \sum_{j=1}^r \|\bar{W}^k_{\cdot j}\|_2^2 \|\bar{H}^k_{j\cdot}\|_2^2 \\ &\geq \frac{\alpha^2 c_3^2}{c_1^2}\|\bar{H}^k_{j\cdot}\|_2^4. \end{aligned} \tag{19}$$

By (18) and (19), we can get

$$\|\bar{H}^k_{j\cdot}\|_2^2 \leq \frac{c_1\|A\|_F}{\alpha c_3}.$$

Thus,

$$\|\bar{H}^k\|_F = \sqrt{\sum_{j=1}^r \|\bar{H}^k_{j\cdot}\|_2^2} \leq \sqrt{\frac{r c_1\|A\|_F}{\alpha c_3}} := b. \tag{20}$$

Similarly, we have

$$\|\bar{W}^k\|_F = \sqrt{\sum_{j=1}^r \|\bar{W}^k_{\cdot j}\|_2^2} \leq \sqrt{\frac{r\alpha c_4\|A\|_F}{c_2}}.$$

This implies that $\{\bar{W}^k, \bar{H}^k\}$ is bounded. □

The above lemma implies that $\{\bar{W}^k, \bar{H}^k\}$ generated by Algorithm 1 has at least one limit point. So, we can obtain the main convergence result of Algorithm 1.

Theorem 2 *Let $\{\bar{W}^k, \bar{H}^k\}$ be generated by Algorithm 1, and (W^*, H^*) be a limit point of $\{\bar{W}^k, \bar{H}^k\}$. Then (W^*, H^*) is a stationary point of problem (2).*

Proof By Lemma 2, there exists $\{\bar{W}^{k_p}, \bar{H}^{k_p}\} \to (W^*, H^*)$. According to Algorithm 1, we have

$$\begin{aligned} H^{k_p} &\geq 0, \\ (W^{k_p})^T(W^{k_p}H^{k_p} - A) &\geq 0, \\ \langle H^{k_p}, (W^{k_p})^T(W^{k_p}H^{k_p} - A)\rangle &= 0, \end{aligned} \tag{21}$$

Moreover, by Property I and Property II we have

$$\begin{aligned} \bar{W}^{k_p} &\geq 0, \\ \bar{H}^{k_p} &\geq 0, \\ (\bar{W}^{k_p})^T(\bar{W}^{k_p}\bar{H}^{k_p} - A) &\geq 0, \\ \langle \bar{H}^{k_p}, (\bar{W}^{k_p})^T(\bar{W}^{k_p}\bar{H}^{k_p} - A)\rangle &= 0. \end{aligned} \tag{22}$$

Taking limits of both sides of (22), we have

$$\begin{aligned} W^* &\geq 0, \\ H^* &\geq 0, \\ (W^*)^T(W^*H^* - A) &\geq 0, \\ \langle H^*, (W^*)^T(W^*H^* - A)\rangle &= 0. \end{aligned} \tag{23}$$

Since

$$\langle (W^*H^* - A)(H^*)^T, W^*\rangle = \langle H^*, (W^*)^T(W^*H^* - A)\rangle = 0,$$

we only need to show that

$$\left(W^*H^* - A\right)\left(H^*\right)^T \geq 0. \tag{24}$$

Suppose

$$\|[(W^*H^* - A)(H^*)^T]_-\|_F = a > 0. \tag{25}$$

By locally sign-preserving property, we have

$$\exists p_0, \ \ \forall p > p_0, \ \ \|[(\bar{W}^{k_p}\bar{H}^{k_p} - A)(\bar{H}^{k_p})^T]_-\|_F \geq \frac{a}{2}.$$

Let $D_{k_p} = -[(\bar{W}^{k_p}\bar{H}^{k_p} - A)(\bar{H}^{k_p})^T]_-$, then

$$\forall\lambda \geq 0,\ \ \bar{W}^{k_p} + \lambda D_{k_p} \geq 0,$$

and

$$F(\bar{W}^{k_p} + \lambda D_{k_p}, \bar{H}^{k_p}) = F(\bar{W}^{k_p}, \bar{H}^{k_p}) - \lambda \| D_{k_p} \|_F^2 + \frac{1}{2}\lambda^2 \langle D_{k_p}\bar{H}^{k_p}, D_{k_p}\bar{H}^{k_p}\rangle. \tag{26}$$

It is obvious that $\langle D_{k_p}\bar{H}^{k_p}, D_{k_p}\bar{H}^{k_p}\rangle \geq 0$. If $\langle D_{k_p}\bar{H}^{k_p}, D_{k_p}\bar{H}^{k_p}\rangle = 0$, then $F(\bar{W}^{k_p} + \lambda D_{k_p}, \bar{H}^{k_p}) \to -\infty$, with $\lambda \to \infty$. However,

$$F(\bar{W}^{k_p} + \lambda D_{k_p}, \bar{H}^{k_p}) \geq 0,$$

which is a contradiction. So $\langle D_{k_p}\bar{H}^{k_p}, D_{k_p}\bar{H}^{k_p}\rangle > 0$. Let

$$\lambda_1 = \frac{\| D_{k_p} \|_F^2}{\langle D_{k_p}\bar{H}^{k_p}, D_{k_p}\bar{H}^{k_p}\rangle} = \arg\min_{\lambda \geq 0} F(\bar{W}^{k_p} + \lambda D_{k_p}, \bar{H}^{k_p}).$$

By (20), for any $k_p > 0$, $\|\bar{H}^{k_p}\| \leq b$. Since

$$\frac{\| D_{k_p} \|_F^4}{2\langle D_{k_p}\bar{H}^{k_p}, D_{k_p}\bar{H}^{k_p}\rangle} = \frac{\| D_{k_p} \|_F^4}{2\| D_{k_p}\bar{H}^{k_p} \|_F^2} \geq \frac{\| D_{k_p} \|_F^2}{2\|\bar{H}^{k_p}\|_F^2},$$

we deduce

$$\frac{\| D_{k_p} \|_F^4}{2\langle D_{k_p}\bar{H}^{k_p}, D_{k_p}\bar{H}^{k_p}\rangle} \geq \frac{a^2}{8b^2} = L_1.$$

Therefore, $\forall p > p_0$, we have

$$\begin{aligned} F(\bar{W}^{k_p} + \lambda_1 D_{k_p}, \bar{H}^{k_p}) &= F(\bar{W}^{k_p}, \bar{H}^{k_p}) - \frac{\| D_{k_p} \|_F^4}{2\langle D_{k_p}\bar{H}^{k_p}, D_{k_p}\bar{H}^{k_p}\rangle} \qquad (27) \\ &\leq F(\bar{W}^{k_p}, \bar{H}^{k_p}) - L_1, \\ F(\bar{W}^{k_p+1}, \bar{H}^{k_p+1}) &= F(W^{k_p+1}, H^{k_p+1}) \\ &\leq F(W^{k_p+1}, \bar{H}^{k_p}) \\ &\leq F(\bar{W}^{k_p} + \lambda_1 D_{k_p}, \bar{H}^{k_p}) \qquad (28) \\ &\leq F(\bar{W}^{k_p}, \bar{H}^{k_p}) - L_1. \end{aligned}$$

Thus,

$$F(\bar{W}^{k_{p+1}}, \bar{H}^{k_{p+1}}) \leq \cdots \leq F(\bar{W}^{k_p+1}, \bar{H}^{k_p+1}) \leq F(\bar{W}^{k_p}, \bar{H}^{k_p}) - L_1. \tag{29}$$

Taking limits both sides of (29), we have $L_1 \le 0$, which is a contradiction. This implies that (24) holds. $\square$

6 Numerical experiments

Let

$$r(W, H) = \left\| \begin{matrix} vec(\nabla_W F(W, H))_{I^W \cup J^W} \\ vec(\nabla_H F(W, H))_{I^H \cup J^H} \end{matrix} \right\|_2 ,$$

where $I^W = \{ij : (\nabla_W F(W, H))_{ij} < 0\}$, $J^W = \{ij : W_{ij} > 0\}$, $I^H = \{ij : (\nabla_H F(W, H))_{ij} < 0\}$, $J^H = \{ij : H_{ij} > 0\}$. In this paper, we consider $r(W, H)$ as a residual function and terminate ANLS if one of the following conditions holds:

$$r(\bar{W}^k, \bar{H}^k) \le \min\{\epsilon \|\nabla F(\bar{W}^0, \bar{H}^0)\|_F, \ \ 10^{-2}\}, \ \text{ or } \ k > k_{max} \ \text{ or } \ t > t_{max}, \tag{30}$$

where k is the iteration number of ANLS and t is the running time.

Borrowing an idea from Lin (2007), we use the following stopping conditions for subproblems. The returned matrices H^{k+1} and W^{k+1} from the iterative procedures of solving the subproblems (12) and (13) should respectively satisfy

$$\|vec(\nabla_H F(W^k, H^{k+1}))_{I^H \cup J^H}\|_2 \le \bar{\epsilon_H} \ \text{ or } \ k^H > k^H_{max}$$

and

$$\|vec(\nabla_W F(W^{k+1}, H^{k+1}))_{I^W \cup J^W}\|_2 \le \bar{\epsilon_W} \ \text{ or } \ k^W > k^W_{max}$$

where we set

$$\bar{\epsilon_H} = \bar{\epsilon_W} \equiv \max\{10^{-3}, \epsilon\}\| \cdot \nabla F(W^0, H^0)\|_F$$

in the beginning, ϵ is the tolerance in (30). If methods for solving (13) stop without any iterations, we decrease the stopping tolerance by $\bar{\epsilon_H} = 0.1\bar{\epsilon_H}$. For the subproblem (12), $\bar{\epsilon_W}$ is reduced in a similar way.

The program code was written in MATLAB and run in MATLAB 7.5 environment. The programs were carried out on a PC (CPU 1.86GHz, 1G memory) with the Windows XP operation system. For the numerical experiments, we set following initial parameters:

$$\epsilon = 10^{-6}, \ k_{max} = 10000, \ \ t_{max} = 300, \ \alpha = 1.$$

The test problems are randomly generated by the normal distribution (mean 0 and standard deviation 1), and we use 2-norm in experiments. In order to illustrate the computational effectiveness of the proposed algorithm, for each problem, we present the average number of inner iterations, the number of outer iteration, the final function values, the residual function values and time(in seconds) when the algorithm is terminated.

Our numerical results are summarized in Tables 1, 2, 3, 4, 5 and 6, where we present the following datas:

IItr	the average number of inner iterations (The number of inner iterations divided by the number of outer iterations)
OItr	the number of outer iteration
time(s)	the CPU time in seconds
$F(\cdot)$	the function value
$r(\cdot)$	the residual function value

In order to avoid the initial point affect the numerical results, we use 30 initial points, and every initial point is randomly generated. So IItr, OItr, time(s), $F(\cdot)$ and $r(\cdot)$ listed in Table 1, 2, 3, 4, 5 and 6 denote the average values of IItr, OItr, time(s), $F(\cdot)$ and $r(\cdot)$ at 30 initial points, respectively.

There are two ways to generate each problem, one is $A = abs(randn(m, r)) * abs(randn(r, n))$, which implies that A can be broken down into two lower levels of matrix; the other is $A = abs(randn(m, n))$, which suggests A is not necessarily decomposed into two lower levels of matrix. We compare Algorithm 1 with Lin (2007)(PGNMF) , Kim et al. (2007)(FNNMF) and Kim and Park (2008)(ALNMF).

In Tables 1 and 4, we use projected gradient methods to solve subproblems (12) and (13). It can be seen from Table 1 that the number of outer iterations required by PGNMF is less than the corresponding numbers required by Algorithm 1 for most

Table 1 Solving sub-problems using PGNMF $A = abs(randn(m, r)) * abs(randn(r, n))$

(m, n, r)		IItr	OItr	$F(\bar{W}^t, \bar{H}^t)$	$r(\bar{W}^t, \bar{H}^t)$	Time(s)
(100, 50, 5)	PGNMF	52.74	1063.50	7.932e-06	2.533e-03	6.59
	Algorithm 1	47.00	906.50	8.218e-06	2.566e-03	5.12
(50, 50, 10)	PGNMF	93.40	1231.00	1.335e-05	2.562e-03	18.81
	Algorithm 1	61.94	1279.00	1.423e-05	2.577e-03	11.98
(100, 50, 20)	PGNMF	32.83	682.50	3.129e-06	1.543e-03	2.07
	Algorithm 1	10.23	1116.50	5.362e-06	1.592e-03	1.23
(100, 250, 5)	PGNMF	33.04	752.00	2.432e-05	8.832e-03	7.50
	Algorithm 1	17.78	757.00	9.511e-05	8.844e-03	4.11
(200, 200, 8)	PGNMF	20.70	1282.00	1.376e-04	9.333e-03	16.53
	Algorithm 1	11.31	1394.00	1.535e-04	9.998e-03	8.25
(50, 250, 10)	PGNMF	104.62	993.00	5.462e-05	9.987e-03	45.98
	Algorithm 1	24.10	1250.00	1.398e-04	9.961e-03	13.38
(100, 100, 40)	PGNMF	413.88	351.00	2.522e-05	9.807e-03	158.47
	Algorithm 1	322.58	307.00	1.834e-05	9.123e-03	115.50
(200, 100, 50)	PGNMF	236.60	476.00	4.994e-03	2.003e-01	301.41
	Algorithm 1	259.19	463.00	1.162e-03	8.600e-02	300.45

Table 2 Solving sub-problems using FNNMF $A = abs(randn(m, r)) * abs(randn(r, n))$

(m, n, r)		IItr	OItr	$F(\bar{W}^t, \bar{H}^t)$	$r(\bar{W}^t, \bar{H}^t)$	Time(s)
(100, 50, 5)	FNNMF	7.11	675.50	5.795e-06	2.564e-03	2.28
	Algorithm 1	7.02	630.00	6.563e-06	2.564e-03	2.28
(50, 50, 10)	FNNMF	8.48	2101.00	2.906e-05	2.600e-03	7.27
	Algorithm 1	8.56	1768.00	4.204e-05	2.600e-03	6.28
(100, 50, 20)	FNNMF	6.06	557.50	3.049e-06	1.432e-03	1.43
	Algorithm 1	5.60	531.50	3.990e-06	1.410e-03	1.29
(100, 250, 5)	FNNMF	6.95	778.00	1.865e-05	8.844e-03	16.11
	Algorithm 1	6.81	812.00	4.032e-05	7.743e-03	15.66
(200, 200, 8)	FNNMF	7.47	1257.00	3.889e-05	9.959e-03	35.84
	Algorithm 1	7.52	1207.00	4.436e-05	9.940e-03	34.59
(50, 250, 10)	FNNMF	9.12	961.00	5.812e-05	9.959e-03	21.63
	Algorithm 1	9.57	879.00	2.291e-04	9.984e-03	18.52
(100, 100, 40)	FNNMF	36.34	359.00	9.200e-06	9.517e-03	74.30
	Algorithm 1	44.61	304.00	1.822e-05	8.046e-03	69.14
(200, 100, 50)	FNNMF	34.14	688.00	3.134e-05	9.946e-03	266.44
	Algorithm 1	36.19	650.00	4.205e-05	9.321e-03	265.11

Table 3 Solving sub-problems using ALNMF $A = abs(randn(m, r)) * abs(randn(r, n))$

(m, n, r)		IItr	OItr	$F(\bar{W}^t, \bar{H}^t)$	$r(\bar{W}^t, \bar{H}^t)$	Time(s)
(100, 50, 5)	ALNMF	149.82	842.00	8.495e-06	2.260e-03	147.78
	Algorithm 1	149.82	834.00	9.112e-06	2.261e-03	142.67
(50, 50, 10)	ALNMF	99.88	943.00	3.655e-06	3.116e-03	300.16
	Algorithm 1	99.86	816.00	1.618e-05	2.921e-03	233.141
(100, 50, 20)	ALNMF	149.06	149.00	6.670e-02	4.692e-01	300.25
	Algorithm 1	149.04	159.00	5.256e-02	2.867e-01	300.66
(100, 250, 5)	ALNMF	349.49	662.00	2.833e-04	3.440e-02	300.203
	Algorithm 1	342.47	653.00	2.425e-04	2.072e-02	275.22
(200, 200, 8)	ALNMF	398.51	259.00	5.471e-02	3.764e-01	301.14
	Algorithm 1	398.51	268.00	4.864e-02	3.166e-01	301.14
(50, 250, 10)	ALNMF	299.06	313.00	2.345e-01	7.897e-01	300.75
	Algorithm 1	299.06	318.00	2.175e-01	4.113e-01	300.66
(100, 100, 40)	ALNMF	190.00	20.00	3.694e+01	3.708e+01	307.11
	Algorithm 1	190.00	20.00	3.694e+01	2.808e+01	303.75
(200, 100, 50)	ALNMF	262.50	7.00	7.651e+02	3.683e+02	308.50
	Algorithm 1	262.50	8.00	6.168e+02	2.076e+02	319.42

Table 4 Solving sub-problems using PGNMF $A = abs(randn(m, n))$

(m, n, r)		IItr	OItr	$F(\bar{W}^t, \bar{H}^t)$	$r(\bar{W}^t, \bar{H}^t)$	Time(s)
(100, 50, 5)	PGNMF	21.38	516.00	6.947e+02	4.509e-03	1.48
	Algorithm 1	10.72	369.50	6.947e+02	4.563e-03	0.51
(50, 50, 10)	PGNMF	27.12	420.00	2.269e+02	7.739e-03	1.84
	Algorithm 1	15.86	394.00	2.270e+02	7.833e-03	1.02
(100, 50, 20)	PGNMF	23.35	648.00	3.349e+02	5.781e-03	6.83
	Algorithm 1	15.10	566.50	3.350e+02	9.646e-03	3.63
(100, 250, 5)	PGNMF	11.86	882.50	4.027e+03	9.951e-03	2.72
	Algorithm 1	13.61	480.50	4.027e+03	9.881e-03	1.85
(200, 200, 8)	PGNMF	21.16	651.00	6.317e+03	8.709e-03	6.74
	Algorithm 1	23.51	517.00	6.317e+03	9.541e-03	5.29
(50, 250, 10)	PGNMF	41.04	522.00	1.555e+03	4.832e-03	5.84
	Algorithm 1	15.94	426.00	1.555e+03	9.906e-03	2.52
(100, 100, 40)	PGNMF	30.78	2162.00	5.464e+02	2.421e-03	73.40
	Algorithm 1	17.60	1997.50	5.476e+02	9.960e-03	42.81
(200, 100, 50)	PGNMF	35.99	2425.00	1.144e+03	9.736e-03	231.49
	Algorithm 1	15.58	1946.50	1.145e+03	9.930e-03	80.70

Table 5 Solving sub-problems using FNNMF $A = abs(randn(m, n))$

(m, n, r)		IItr	OItr	$F(\bar{W}^t, \bar{H}^t)$	$r(\bar{W}^t, \bar{H}^t)$	Time(s)
(100, 50, 5)	FNNMF	5.82	367.00	6.947e+02	4.610e-03	0.80
	Algorithm 1	5.17	309.50	6.947e+02	4.336e-03	0.63
(50, 50, 10)	FNNMF	6.41	356.50	2.269e+02	7.515e-03	0.88
	Algorithm 1	5.40	277.00	2.270e+02	7.322e-03	0.52
(100, 50, 20)	FNNMF	7.48	581.00	3.34e+02	4.106e-03	4.48
	Algorithm 1	6.20	560.50	3.345e+02	7.689e-03	3.52
(100, 250, 5)	FNNMF	7.14	678.50	4.027e+03	1.937e-03	6.63
	Algorithm 1	5.79	582.50	4.027e+03	9.942e-03	4.46
(200, 200, 8)	FNNMF	8.13	533.00	6.317e+03	5.203e-03	13.27
	Algorithm 1	6.69	544.00	6.317e+03	9.866e-03	9.63
(50, 250, 10)	FNNMF	7.14	449.00	1.555e+03	3.623e-03	3.47
	Algorithm 1	5.09	431.50	1.556e+03	9.93e-03	2.58
(100, 100, 40)	FNNMF	12.38	1258.50	5.467e+02	2.138e-03	47.97
	Algorithm 1	8.26	1177.00	5.475e+02	9.846e-03	35.01
(200, 100, 50)	FNNMF	8.53	1874.00	1.146e+03	9.852e-03	126.40
	Algorithm 1	8.52	1449.50	1.146e+03	9.725e-03	100.27

Table 6 Solving sub-problems using ALNMF $A = abs(randn(m, n))$

(m, n, r)		IItr	OItr	$F(\bar{W}^t, \bar{H}^t)$	$r(\bar{W}^t, \bar{H}^t)$	Time(s)
(100, 50, 5)	ALNMF	149.75	638.00	6.906e+02	4.760e-03	95.45
	Algorithm 1	149.75	606.00	6.906e+02	4.770e-03	89.61
(50, 50, 10)	ALNMF	99.69	360.00	2.423e+02	8.044e-03	68.11
	Algorithm 1	99.69	327.00	2.423e+02	8.033e-03	61.48
(100, 50, 20)	ALNMF	149.52	309.00	3.344e+02	8.899e-02	300.06
	Algorithm 1	149.52	314.00	3.344e+02	2.038e-02	300.61
(100, 250, 5)	ALNMF	349.39	703.00	4.084e+03	1.135e-02	300.09
	Algorithm 1	349.39	576.00	4.084e+03	9.990e-03	232.25
(200, 200, 8)	ALNMF	398.77	309.00	6.346e+03	5.376e-02	300.36
	Algorithm 1	398.77	325.00	6.346e+03	1.635e-02	300.64
(50, 250, 10)	ALNMF	299.25	401.00	1.527e+03	3.543e-01	300.70
	Algorithm 1	299.25	399.00	1.527e+03	8.216e-02	300.53
(100, 100, 40)	ALNMF	196.15	51.00	5.579e+02	9.568e+00	300.80
	Algorithm 1	196.15	52.00	5.575e+02	1.657e+00	303.23
(200, 100, 50)	ALNMF	282.35	17.00	1.316e+03	4.347e+01	304.81
	Algorithm 1	282.35	17.00	1.316e+03	7.375e+00	300.73

of the test problems. From Table 4, we can see that the number of outer iterations required by Algorithm 1 is less than the corresponding numbers required by PGNMF. However, from Tables 1 and 4, we observe that the implementation of Algorithm 1 is superior to that of PGNMF from the average numbers of inner iterations and CPU time.

In Tables 2 and 5, we solve subproblems (12) and (13) with fast Newton-type methods. Table 2 shows that the average numbers of inner iterations and CPU time of FNNMF and Algorithm 1 are very close, Algorithm 1 is able to solve all test problems in a small number of outer iterations. From Table 5, it is obvious that average numbers of inner iterations and the number of outer iterations for Algorithm 1 are smaller than that for FNNMF, CPU time of Algorithm 1 is less than that of FNNMF.

From Tables 3 and 6, we can see that the numerical results obtained by ALNMF and the corresponding results required by Algorithm 1 are very close.

From Tables 1, 2, 3, 4, 5 and 6, we observe that the final function values and residual function values obtained by PGNMF, FNNMF and ALNMF and those of Algorithm 1 are very close.

Now we analyze the space complexity of Algorithm 1. Whatever method is used to solve the subproblems of ANLS, the space complexity of ANLS is greater than mn. According to Strategy I, $(\|W^k_{\cdot 1}\|, \|W^k_{\cdot 2}\|, \ldots, \|W^k_{\cdot r}\|)$, $(\|H^k_{1\cdot}\|, \|H^k_{2\cdot}\|, \ldots, \|H^k_{r\cdot}\|)^T$, $(\lambda^2_1, \lambda^2_2, \ldots, \lambda^2_r)$ and $(\lambda^1_1, \lambda^1_2, \ldots, \lambda^1_r)$ use r memory bits, respectively. So the space complexity of Algorithm 1 is $4r$ higher than that of ANLS. By $r \ll \frac{mn}{m+n}$, we have $\frac{4r}{mn} \ll \frac{4}{m+n}$. For large scale NMF problems, $\frac{4}{m+n}$ is a very small number. So, For large scale NMF problems, the space complexity of Algorithm 1 is almost the same as that of ANLS.

Table 7 Image data

		PGNMF	Algorithm 1
CBCL	IITR	170.99	115.86
(m,n,r)	OITR	470.00	582.00
(361,2429,49)	Times(s)	300.17	300.42
	$F(W, H)$	3.100e+05	3.011e+05
	$r(W, H)$	2.464e+02	2.183e+02
ORL	IITR	106.01	72.78
(m,n,r)	OITR	196.00	113.00
(10304,400,25)	Times(s)	301.05	307.25
	$F(W, H)$	9.496e+08	9.570e+08
	$r(W, H)$	3.762e+04	6.131e+03

We consider two image problems used in Hoyer (2004):

1. CBCL face image database.
   http://cbcl.mit.edu/cbcl/software-datasets/FaceData2.html.
2. ORL face image database.
   http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.

Table 7 presents average results of using 30 random initial points. For all two problems, the numerical results obtained by PGNMF and those of Algorithm 1 are very close.

7 Conclusions

Matrix factorization is an effective tool for large-scale data processing and analysis. NMF method, which decomposes the nonnegative matrix into two nonnegative factor matrices, provides a new way for matrix factorization. In this paper we propose a modified strategy for ANLS. Global convergence of the method is established. The numerical results indicate that our method is a promising tool for NMF.

Acknowledgements The project was supported by National Natural Science Foundation of China (Grant No. 61072144, 61179040).

References

Ahn J-H, Kim S-K, Oh J-H, Choi S (2004) Multiple nonnegativematrix factorization of dynamic PET images. In ACCV

Brunet J-P, Tamayo P, Golub T R, Mesirov J P (2004) Metagenes and molecular pattern discovery using matrix factorization. Proc Nat Acad Sci USA 101:4164–4169

Cho Y-C, Choi S (2005) Nonnegative features of spectro-temporal sounds for classification. Pattern Recognit Lett 26:1327–1336

Chu M, Diele F, Plemmons R, Ragni S (2004) Optimality, computation, and interpretations of nonnegative matrix factorizations. SIAM J Matrix Anal 4–8030

Gao Y, Church G (2005) Improving molecular cancer class discovery through sparse non-negative matrix factorization. Bioinformatics 21:3970–3975

Gonzalez EF, Zhang Y (2005) Accelerating the Lee-Seung algorithm for nonnegative matrix factorization, Technical report, Rise University

Guillamet D, Vitria J (2002) Classifying faces with nonnegative matrix factorization. In Proceedings of the Fifth Catalan Conference for Artificial Intelligence

Guillamet D, Schiele B, Vitria J (2002) Analyzing non-negative matrix factorization for image classification. 16th International Conference on Pattern Recognition, vol 2, pp 116–119

Guillamet D, Vitria J, Schiele B (2003) Introducing a weighted non-negative matrix factorization for image classification. Pattern Recognit Lett 24:2447–2454

Kim H, Park H (2007) Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. Bioinformatics 23:1495–1502

Kim H, Park H (2008) Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method. SIAM J Matrix Anal Appl 30:713–730

Kim PM, Tidor B (2003) Subsystem identification through dimensionality reduction of large-scale gene expression data. Genome Res 13:1706–1718

Kim D, Sra S, Dhillon IS (2007) Fast newton-type methods for the least squares nonnegative matrix approximation problem. In: Proceedings of SIAM Conference on Data Mining

Lee DD, Seung HS (1999) Learning the parts of objects by non-negative matrix factorization. Nature 401:788–791

Lee DD, Seung HS (2001) Algorithms for non-negative matrix factorization. In: Proceedings of Neural Information Processing Systems, vol 13, pp 556–562

Lee JS, Lee DD, Choi S, Lee DS (2002) Application of nonnegative matrix factorization to dynamic positron emission tomography. In 3rd International Conference on Independent Component Analysis and Blind Signal Separation, pp 556–562

Li H, Adali T, Wang W, Emge D (2005) Non-negative matrix factorization with orthogonality constraints for chemical agent detection in raman spectra. Machine Learning for Signal Processing, pp 253–258

Lin C-J (2007) Projected gradient methods for nonnegative matrix factorization. Neural Comput 19:2756–2779

Lin C-J (2007) On the convergence of multiplicative update algorithms for non-negative matrix factorization. IEEE Transac Neural Netw 18:1589–1596

Liu W, Zheng N (2004) Non-negative matrix factorization based methods for object recognition. Pattern Recognit Lett 25:893–897

Okun O, Priisalu H (2009) Fast nonnegative matrix factorization and its application for protein fold recognition. EURASIP J Appl Signal Process 2006:62

Paatero P (1999) The multilinear enginea table-driven least squares program for solving multilinear problems, including the n-way parallel factor analysis model. J Comput Graph Stat 8:1–35

Paatero P, Tapper U (1994) Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values. Environmetrics 5:111–126

Pascual-Montano A, Carazo JM, Kochi K, Lehmann D, Pascual-Marqui RD (2006) Nonsmooth nonnegative matrix factorization (nsNMF). IEEE Transac Pattern Anal Mach Intell 28:403–415

Paul Pauca V, Shahnaz F, Berry Michael W, Plemmons Robert J (2004) Text mining using non-negative matrix factorization. In: Proceedings of the Fourth SIAM International Conference on Data Mining 2004

Rao N, Shepherd SJ (2004) Extracting characteristic patterns from genomewide expression data by non-negative matrix factorization. In Computational Systems Bioinformatics Conference

Richardson WH (1972) Bayesian-based iterative method of image restoration. J Opt Soc Am 62:55–59

Spratling MW (2006) Learning image components for object recognition. J Mach Learn Res 7:793–815

Wang Y, Jia Y, Hu C, Matthewturk (2005) Non-negative matrix factorization framework for face recognition. Int J Pattern Recognit Artif Intell 19:495–511

Zdunek R, Cichocki A (2006) Non-negative matrix factorization with quasi-newton optimization. International Conference on Artificial Intelligence and Soft Computing (ICAISC 2006), pp 870–879

Zdunek R, Cichocki A (2008) Fast nonnegativematrix factorization algorithms using projected gradient approaches for large-scale problems. Hindawi Publishing Corporation Computational Intelligence and Neuroscience