JMLR: Workshop and Conference Proceedings 39:129–141, 2014 ACML 2014

A Fast Hierarchical Alternating Least Squares Algorithm for Orthogonal Nonnegative Matrix Factorization

Keigo Kimura KKIMURA@MAIN.IST.HOKUDAI.AC.JP
Yuzuru Tanaka TANAKA@MEME.HOKUDAI.AC.JP
Mineichi Kudo MINE@MAIN.IST.HOKUDAI.AC.JP
Graduate School of Information Science and Technology, Hokkaido University, Sapporo, 060-0814 Japan

Editor: Dinh Phung and Hang Li

Abstract

Nonnegative Matrix Factorization (NMF) is a popular technique in a variety of fields due to its component-based representation with physical interpretablity. NMF finds a nonnegative hidden structures as oblique bases and coefficients. Recently, Orthogonal NMF (ONMF), imposing an orthogonal constraint into NMF, has been gathering a great deal of attention. ONMF is more appropriate for the clustering task because the resultant constrained matrix consisting of the coefficients can be considered as an indicator matrix. All traditional ONMF algorithms are based on multiplicative update rules or project gradient descent method. However, these algorithms are slow in convergence compared with the state-of-the-art algorithms used for regular NMF. This is because they update a matrix in each iteration step. In this paper, therefore, we propose to update the current matrix column-wisely using Hierarchical Alternating Least Squares (HALS) algorithm that is typically used for NMF. The orthogonality and nonnegativity constraints are both utilized efficiently in the column-wise update procedure. Through experiments on six real-life datasets, it was shown that the proposed algorithm converges faster than the other conventional ONMF algorithms due to a smaller number of iterations, although the theoretical complexity is the same. It was also shown that the orthogonality is also attained in an earlier stage.
Keywords: Orthogonal Nonnegative Matrix Factorization, Orthogonal Factorization.

1. Introduction

Orthogonal Nonnegative Matrix Factorization (ONMF), firstly proposed by Ding et al. (2006), factorizes a nonnegative matrix into two nonnegative matrices under the one-sided orthogonal constraint imposed on the first *factor* matrix. That is, ONMF is a minimization problem:

$$\min_{\mathbf{F},\mathbf{G}} \|\mathbf{X} - \mathbf{F}\mathbf{G}^T\|_F^2,$$
$$\text{subject to } \mathbf{F} \geq 0, \mathbf{G} \geq 0,\ \mathbf{F}^T\mathbf{F} = \mathbf{I},$$

where $\mathbf{X} \in \mathbb{R}^{M\times N}$, $\mathbf{F} \in \mathbb{R}^{M\times J}$,$\mathbf{G} \in \mathbb{R}^{N\times J}$ $(J \ll N, M)$ and $\mathbf{I}$ is the identity matrix. Here T denotes the transpose and $\|\cdot\|_F^2$ denotes the squared Frobenius norm (the sum of squared elements). In this formulation, $\mathbf{F}^T\mathbf{F} = \mathbf{I}$ is imposed as a condition, but the strict application of both nonnegativity and orthogonality to bases is too strong. In fact, it yields a part of

normal vectors in the standard basis. Therefore, in a practical sense, the optimization problem should be stated as

$$\min_{\mathbf{F},\mathbf{G}} \|\mathbf{X} - \mathbf{F}\mathbf{G}^T\|_F^2 + \lambda\|\mathbf{F}^T\mathbf{F} - \mathbf{I}\| \tag{1}$$

with a positive coefficient λ. This corresponds to a Lagrangian formulation. In practice, we will see such a formulation in the following section.

As long as the authors' knowledge, conventional algorithms for solving ONMF problems are all based on matrix-wise alternating block coordinate descent. However, it is known that matrix-wise update algorithms cannot effectively utilize the gradient of the objective function and result in slow convergence (Cichocki and Anh-Huy (2009), Kim and Park (2011)). In NMF without orthogonal constraint, some state-of-the-art algorithms update $\mathbf{F}$ and $\mathbf{G}$ column-wisely or element-wisely to gain faster convergence. In ONMF, however, it is difficult to incorporate the orthogonal constraint into column-wise or element-wise coordinate descent updates.

In this paper, we propose a Fast Hierarchical Alternating Least Squares (HALS) algorithm for ONMF. Our algorithm is based on a column-wise update algorithm proposed by Cichocki and Anh-Huy (2009). To enable such a column-wise update, we derive a column-wise orthogonal constraint. We explicitly utilize the nonnegativity in the orthogonal constraint.

The rest of this paper is organized as follows. We will summarize previously proposed NMF algorithms and ONMF algorithms by connecting them to the corresponding optimization criteria in Section 2. Then we will explain the HALS algorithm for standard NMF (Cichocki and Anh-Huy (2009)) in Section 3. The way of utilizing HALS for ONMF will be explained and the algorithm HALS ONMF will be proposed in Section 4. Section 5 will be devoted for evaluation of the proposed algorithm on several real-life datasets. Conclusion will be given in Section 6.

We will use a bold uppercase letter for a matrix, such as $\mathbf{X}$, and an italic lowercase letter for a vector such as $\boldsymbol{x}$. Both $\mathbf{X}_{ij}$ and $\boldsymbol{x}_{ij}$ stand for the (i,j)th element in a matrix $\mathbf{X}$. A vector $\mathbf{1}_J \in \mathbb{R}^J$ shows the vector whose elements are of one's.

2. Related Work

In this section, we provide a brief review of NMF and ONMF algorithms.

2.1. Nonnegative Matrix Factorization

NMF aims to find a nonnegative matrix $\mathbf{F} = [\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_J] \in \mathbb{R}_+^{N\times J}$ and another nonnegative matrix $\mathbf{G} = [\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_J] \in \mathbb{R}_+^{M\times J}$ whose product approximates a given nonnegative matrix $\mathbf{X} \in \mathbb{R}_+^{N\times M}$:

$$\min_{\mathbf{F},\mathbf{G}} \|\mathbf{X} - \mathbf{F}\mathbf{G}^T\|_F^2,$$
$$\text{subject to } \mathbf{F} \geq 0, \mathbf{G} \geq 0. \tag{2}$$

Since NMF problem is not convex both in $\mathbf{F}$ and $\mathbf{G}$, and thus, various algorithms have been proposed (Lee and Seung (2000), Cichocki et al. (2009), Kim and Park (2011) and

Hsieh and Dhillon (2011)). They are categorized according to the way of updates as follows.

Matrix-wise update algorithms
Lee and Seung (2000) proposed a Multiplicative Update (MU) algorithm. This MU algorithm is one of efficient algorithms for NMF proposed in the early stage, and thus, many extensions followed (*e.g.*,Cai et al. (2011), Cichocki et al. (2009)). However, from the viewpoint of convergence, they were not efficient (Kim et al. (2014)). Lin (2007) proposed a Project Gradient Descent (PGD) algorithm for NMF. This algorithm solves NMF problem by solving a Nonnegative Least Squares (NLS) problem for each matrix alternatively and has relatively faster convergence than MU algorithms. Their difference is that MU algorithm uses a fixed step-size in the gradient descent method, while PGD uses a flexible step-size. Nevertheless, PGD still needs more iteration than necessary.
Vector-wise update algorithms
Cichocki and Anh-Huy (2009) proposed a Hierarchical Alternating Least Squares (HALS) algorithm. HALS algorithm solves a set of column-wise NLS problems for each column and update $\mathbf{F}$ and $\mathbf{G}$ column-wisely. Since each of column-wise NLS problems can be solved at high accuracy and efficiently, HALS converges very fast. Kim and Park (2011) proposed an active-set like algorithm that also decomposes a matrix NLS problem into a set of column-wise sub-problems. The difference between HALS and the active-set like method lies on the way to solve a column-wise sub-problem. The former uses the gradient to solve a sub-problem, while the latter uses active-set method to solve that. The active-set method has two stages to solve that, first they find a feasible point, in standard NMF, it is a nonnegative point. Second they minimize a column-wise NLS problem with keeping feasibility. These algorithms can be thought as the state-of-the-art algorithms, because they converge empirically faster than matrix-wise update algorithm. However, the addition of constraints such as $\mathbf{F}^T\mathbf{F} = \mathbf{I}$ is difficult in such column-wise updates. Especially, the latter active-set like algorithm is difficult to work with equality constraints.
Element-wise update algorithms
Hsieh and Dhillon (2011) proposed an element-wise update algorithm called a Greedy Coordinate Descent (GCD) algorithm. To the authors' knowledge, it is the fastest algorithm for NMF. The GCD algorithm takes a greedy strategy to decrease the value of the objective function. It selects and updates the most contributable variables for minimization. The reason for the lower computational cost is that it does not update unnecessary elements. Unfortunately, GCD algorithm cannot work with such a constraint that affects all elements of one column as the same time, such as the graph regularized constraint that minimizes $\alpha(\mathrm{tr}(\mathbf{F}^T\mathbf{L}\mathbf{F}))$ where $\mathbf{L}$ is a graph Laplacian matrix of $\mathbf{X}^T\mathbf{X}$. The GCD relies on the fact that, with fixed $\mathbf{G}$, updating an element f_{ij} of $\mathbf{F}$ changes only the gradients of elements in the same row $\boldsymbol{f}_{i\cdot}$ because the gradient in $\mathbf{F}$ is given by $(-2\mathbf{X}\mathbf{G} + 2\mathbf{F}\mathbf{G}^T\mathbf{G})$. In more detail, GCD iteratively selects and updates the most contributable variable f_{ij} in the ith row. The GCD is not applicable for ONMF because the orthogonal condition requires an interaction between different rows.

2.2. Orthogonal NMF

An additional orthogonal constraint, $\mathbf{F}^T\mathbf{F} = \mathbf{I}$, is imposed in ONMF. At first, we briefly review the first ONMF algorithm proposed by Ding et al. (2006) and reveal the problem behind ONMF.

The goal of ONMF is to find a nonnegative orthogonal matrix $\mathbf{F}$ and a nonnegative matrix $\mathbf{G}$ minimizing the following objective function with a Lagrangian multiplier λ,

$$L(\mathbf{F}, \mathbf{G}) = \|\mathbf{X} - \mathbf{F}\mathbf{G}^T\|_F^2 + \mathrm{Tr}[\lambda(\mathbf{F}^T\mathbf{F} - \mathbf{D})], \tag{3}$$

where $\mathbf{D}$ is a diagonal matrix and Tr is the trace.[1] The KKT complementary condition gives

$$(-2\mathbf{X}\mathbf{G} + 2\mathbf{F}\mathbf{G}^T\mathbf{G} + 2\mathbf{F}\lambda)_{nj}\mathbf{F}_{nj}^2 = 0, \;\; n = 1, 2, \ldots, N, \;\; j = 1, 2, \ldots J. \tag{4}$$

Then the update rule of the constrained matrix $\mathbf{F}$ is derived as

$$\mathbf{F}_{nj} \leftarrow \mathbf{F}_{nj}\sqrt{\frac{(\mathbf{X}\mathbf{G})_{nj}}{[\mathbf{F}(\mathbf{G}^T\mathbf{G} + \lambda)]_{nj}}}. \tag{5}$$

The point is in the way to determine the value of Lagrange multiplier λ. Since it is not easy to solve this problem for every value of λ, Ding et al. (2006) ignored the nonnegativity and relied only on $\mathbf{F}^T\mathbf{F} = \mathbf{I}$ to have a unique value of λ. By multiplying $\mathbf{F}^T$ from the left in (5), we have

$$\lambda = \mathbf{F}^T\mathbf{X}\mathbf{G} - \mathbf{G}^T\mathbf{G}.$$

Thus, we have the final update form of (5) as

$$\mathbf{F}_{nj} \leftarrow \mathbf{F}_{nj}\sqrt{\frac{(\mathbf{X}\mathbf{G})_{nj}}{(\mathbf{F}\mathbf{F}^T\mathbf{X}\mathbf{G})_{nj}}}.$$

Note that their formulation with the specific value of λ does not strictly satisfy the orthogonality. Rather it is advantageous in avoiding the *zero-lock* problem appearing both in ONMF and NMF: Once an element becomes zero in the middle of iterations, the element will not be recasted in the following steps (see the multiplicative update rule (5)). Besides, when the orthogonality constraint is strictly posed with nonnegativity, each row vector of $\mathbf{F}$ must have only one non-zero value. That is, any algorithm using a multiplicative update rule falls easily into the *zero-lock* problem. Therefore, ONMF algorithms put the first priority on the approximation while loosening the degree of the orthogonality.

An orthogonal NMF algorithm can be seen as an algorithm that balances the trade-off between the orthogonality and the approximation with a weighting parameter as seen in (1). We dare not categorize ONMF algorithms by the type of updates because all conventional ONMF algorithms are based only on matrix-wise updates. Rather, those algorithm should be categorized according to if it employs a weighting parameter or not. If an algorithm minimizes an objective function with a weighting parameter α and the value of α is not appropriately chosen, then the algorithm would fail in either approximation or orthogonality. Such a failure are often reported in past experimental results Li et al. (2010), Mirzal (2014)

1. Hereafter, we will not state the nonnegative constraint explicitly.

Table 1: A summary of categorization of ONMF algorithms

Author(Year)	Updates		Weighting Parameter
	Matrix	Vector	(YES/NO)
Ding et al. (2006)	MU		**NO**
Yoo and Choi (2008)	MU		**NO**
Li et al. (2010)	MU		YES
Pompili et al. (2012)	PGD		YES
Mirzal (2014)	MU		YES
This paper		HALS	**NO**

and Pompili et al. (2012).

Without weighting parameter

The first ONMF algorithm was based on MU algorithm (Ding et al. (2006)). This algorithm does not need a weighting parameter. They solves approximately the Lagrangian (1) instead as we reviewed. Yoo and Choi (2008) also proposed another MU based algorithm. They used the gradient on the Stiefel manifold that is a set of all orthogonal matrices. The gradient on the Stiefel manifold is compatible with MU algorithm because the manifold constrains every matrix to be orthogonal and the employed MU algorithm guarantees nonnegative values.[2]

With weighting parameter

Mirzal (2014) proposed a convergent algorithm that is also based on MU algorithm in practice. He proposed two algorithms, one of which is the same as the one by Li et al. (2010). The first algorithm introduces a weighting parameter α instead of the Lagrangian multiplier λ in (1) (Li et al. (2010)). The second algorithm is a convergent algorithm. The convergence of the algorithm is proved, but this algorithm needs high computational cost. In this algorithm, the zero-lock problem was forcibly avoided by replacing zero values with a small positive value ϵ. There are algorithms that put the first priority on nonngetivity than orthogonality. Pompili et al. (2012) tackled directly the zero-lock problem. They use Augmented Lagrangian method. In more detail, they used the gradient on the Stiefel manifold that is the set of orthogonal matrices and explicitly introduced a Lagrangian multiplier ψ for nonnegativity. The initial value of Lagrangian was approximated to avoid the zero-lock problem. They increase the value of ψ gradually to strengthen the nonnegativity. As a result, the nonnegativity was not strictly guaranteed in the algorithm. More worsely, it has three parameters to be set appropriately for orthogonality, nonnegativity and the step size.

In total, there are mainly two problems on these ONMF algorithms. One problem is difficulty to introduce orthogonal constraint $\mathbf{F}^T\mathbf{F} = \mathbf{I}$ in the corresponding NMF algorithms. This prevents to extend the state-of-the-art NMF algorithms to the corresponding ONMF ones. The other is the zero-lock problem. This problem prevents us from using Lagrangian and alternatively forces us to take a balance between orthogonality and nonnegativity appropriately.

2. In general, the resultant constrained matrix by Yoo and Choi (2008) also will not satisfy the strict orthogonality because MU algorithm is the gradient descent with the fixed step size, and thus, it may undershoot or overshoot.

3. Hierarchical Alternating Least Squares Algorithm for NMF

The key idea of HALS is an efficient decomposition of residual. Suppose that all the elements of matrices $\mathbf{F}$ and $\mathbf{G}$ are fixed except for the jth columns $\boldsymbol{f}_j$ and $\boldsymbol{g}_j$. Since $\mathbf{F}\mathbf{G}^T = \sum_{j=1}^{J} \boldsymbol{f}_j \boldsymbol{g}_j^T$, the objective function (2) can be minimized by finding more appropriate $\boldsymbol{f}_j$ and $\boldsymbol{g}_j$ such as

$$\min_{\boldsymbol{f}_j, \boldsymbol{g}_j} J_j = \|\mathbf{X}^{(j)} - \boldsymbol{f}_j \boldsymbol{g}_j^T\|_F^2 \tag{6}$$

where $\mathbf{X}^{(j)} = \mathbf{X} - \sum_{k \neq j} \boldsymbol{f}_k \boldsymbol{g}_k^T$ is a residue. Since $\boldsymbol{f}_j$ affects only $\boldsymbol{g}_j$, HALS alternatively minimizes (6) for $j = 1, 2, \ldots, J, 1, 2, \ldots$, keeping the nonnegative constraints, $\boldsymbol{f}_j \geq 0$ and $\boldsymbol{g}_j \geq 0$. This objective function (6) with nonnegative constraints can be considered as an NLS problem. HALS solves the set of such NLS problems.

In order to find a stationary point, the gradients of (6) in $\boldsymbol{f}_j$ and $\boldsymbol{g}_j$ are calculated:

$$\mathbf{0} = \frac{\partial J_j}{\partial \boldsymbol{f}_j} = \boldsymbol{f}_j \boldsymbol{g}_j^T \boldsymbol{g}_j - \mathbf{X}^{(j)} \boldsymbol{g}_j, \text{ and} \tag{7}$$

$$\mathbf{0} = \frac{\partial J_j}{\partial \boldsymbol{g}_j} = \boldsymbol{g}_j \boldsymbol{f}_j^T \boldsymbol{f}_j - \mathbf{X}^{(j)T} \boldsymbol{f}_j. \tag{8}$$

Hence, we have the following update rules:

$$\boldsymbol{f_j} \leftarrow \frac{1}{\boldsymbol{g_j}^T \boldsymbol{g}_j} [\mathbf{X}^{(j)} \boldsymbol{g}_j]_+, \tag{9}$$

$$\boldsymbol{g_j} \leftarrow \frac{1}{\boldsymbol{f_j}^T \boldsymbol{f}_j} [\mathbf{X}^{(j)T} \boldsymbol{f}_j]_+, \tag{10}$$

where $[x]_+ = \max(\epsilon, x)$ (ϵ is a sufficiently small and positive value).

In addition, we may normalize so as to $\|\boldsymbol{f}_j\|_2^2 = 1$ after updating. Assuming this normalization we may remove $\boldsymbol{g}_j^T \boldsymbol{g}_j$ and $\boldsymbol{f}_j^T \boldsymbol{f}_j$ from (9) and (10), respectively. Now the update rules (9) and (10) becomes simpler:

$$\begin{aligned} \boldsymbol{f_j} &\leftarrow [\mathbf{X}^{(j)} \boldsymbol{g}_j]_+, \text{ and} \\ \boldsymbol{g_j} &\leftarrow [\mathbf{X}^{(j)T} \boldsymbol{f}_j]_+. \end{aligned}$$

Since $\mathbf{X}^{(j)} = \mathbf{X} - \sum_{k \neq j} \boldsymbol{f}_k \boldsymbol{g}_k^T = \mathbf{X} - \mathbf{F}\mathbf{G}^T + \boldsymbol{f}_j \boldsymbol{g}_j^T$, we finally obtain the following column-wise update rules:

$$\begin{aligned} \boldsymbol{f_j} &\leftarrow [(\mathbf{X}\mathbf{G})_j - \mathbf{F}(\mathbf{G}^T\mathbf{G})_j + \boldsymbol{f}_j \boldsymbol{g}_j^T \boldsymbol{g}_j]_+, \text{ and} \\ \boldsymbol{g_j} &\leftarrow [(\mathbf{X}^T\mathbf{F})_j - \mathbf{G}(\mathbf{F}^T\mathbf{F})_j + \boldsymbol{g}_j \boldsymbol{f}_j^T \boldsymbol{f}_j]_+. \end{aligned}$$

Note that $\mathbf{X}\mathbf{G}$ and $\mathbf{G}^T\mathbf{G}$ do not change their values while updating vectors $\boldsymbol{f}_j$ ($j = 1, \ldots, J$). Therefore HALS computes $\mathbf{X}\mathbf{G}$ and $\mathbf{G}^T\mathbf{G}$ before updating those vectors. Similarly, we pre-calculate $\mathbf{X}^T\mathbf{F}$ and $\mathbf{F}^T\mathbf{F}$ before updating $\boldsymbol{g}_j$ ($j = 1, \ldots, J$).[3] This is the HALS algorithm usable for regular NMF.

3. Sometimes it is called Fast HALS

4. Hierarchical ALS algorithm for ONMF

Since $\boldsymbol{f}_j$ affects the other columns in $\mathbf{F}^T\mathbf{F}$, the orthogonal constraint cannot be directly introduced in the HALS algorithm above. In this paper, we exploit a simple fact that if the sum of nonnegative values is zero then all the values are zero. Since the orthogonal condition $\mathbf{F}^T\mathbf{F} = \mathbf{I}$ means $\boldsymbol{f}_k^T\boldsymbol{f}_j = 0$ for every $k \neq j$, a single condition $\sum_{k\neq j} \boldsymbol{f}_k^T\boldsymbol{f}_j = 0$ for fixed j is equivalent to the former $J-1$ conditions. That is, one matrix condition $\mathbf{F}^T\mathbf{F} = \mathbf{I}$ is equivalently replaced with $2J$ column-wise constraints of $\boldsymbol{f}_j^T\boldsymbol{f}_j = 1$ and $\sum_{k\neq j} \boldsymbol{f}_k^T\boldsymbol{f}_j = 0$ for every j. As will be shown, the newly derived column-wise constraints can be updated with $O(M)$ for each column (M is the the number of rows of $\mathbf{X}$ to be factorized).

4.1. Column-wise Orthogonal Constraint

Now it suffices to impose the conditions

$$\mathbf{F}^{(j)T}\boldsymbol{f}_j = \sum_{k\neq j} \boldsymbol{f}_k^T\boldsymbol{f}_j = 0, \quad j = 1, 2, \ldots, J, \tag{11}$$

in addition to the normalization to $\|\boldsymbol{f}_j\|^2 = \boldsymbol{f}_j^T\boldsymbol{f}_j = 1$. Thus we introduce constraint $\mathbf{F}^{(j)T}\boldsymbol{f}_j = 0$ $(j = 1, 2, \ldots, J)$ into (3) as the column-wise orthogonal constraint. The positivity of the elements is preserved with the ϵ-truncate function $[\ ]_+$.

4.2. Derivation of Hierarchical ALS Algorithm for Orthogonal NMF

With the derived column-wise constraint (11), the localized objective function is formulated as a Lagrangian:

$$\begin{aligned} L(\boldsymbol{f}_j, \boldsymbol{g}_j, \lambda_j) &= \|\mathbf{X}^{(j)} - \boldsymbol{f}_j\boldsymbol{g}_j^T\|_F^2 + \lambda_j(\mathbf{F}^{(j)T}\boldsymbol{f}_j), \quad \text{where} \\ &\quad \mathbf{X}^{(j)} = \mathbf{X} - \sum_{k\neq j} \boldsymbol{f}_k\boldsymbol{g}_k^T, \\ &\quad \mathbf{F}^{(j)} = \sum_{k\neq j} \boldsymbol{f}_k, \quad \lambda_j \geq 0. \end{aligned}$$

The gradient is given as

$$\frac{\partial L}{\partial \boldsymbol{f}_j} = -2\mathbf{X}^{(j)}\boldsymbol{g}_j + 2\boldsymbol{f}_j\boldsymbol{g}_j^T\boldsymbol{g}_j + \lambda_j\mathbf{F}^{(j)}. \tag{12}$$

By solving $\partial L/\partial \boldsymbol{f}_j = 0$ and forcibly keeping the nonnegativity, we obtain the update rule under assumption of normalization of $\boldsymbol{f}_j^T\boldsymbol{f}_j$=1 as post-processing,

$$\boldsymbol{f}_j \leftarrow [\mathbf{X}^{(j)}\boldsymbol{g}_j - \frac{\lambda_j}{2}\mathbf{F}^{(j)}]_+. \tag{13}$$

Unfortunately, the setting of the value of λ still remains as a problem. In this study, we take the same way as Ding et al. (2006) did. By multiplying $\mathbf{F}^{(j)}$ from the left in (12) and using $\mathbf{F}^{(j)T}\boldsymbol{f}_j = 0$, we obtain

$$\lambda_j = \frac{2\mathbf{F}^{(j)T}\mathbf{X}^{(j)}\boldsymbol{g}_j}{\mathbf{F}^{(j)T}\mathbf{F}^{(j)}}.$$

Table 2: Arithmetic operations necessary for each updating.

Method (year)	#Operation addition/subtraction	multiplication	division	sqrt	Complexity overall
Ding et al. (2006)	$MNJ + 2NJ^2$	$MNJ + 2NJ^2 + NJ$	NJ	NJ	$O(MNJ)$
Yoo and Choi (2008)	$MNJ + 2NJ^2$	$MNJ + 2NJ^2 + NJ$	NJ	–	$O(MNJ)$
Li et al. (2010)	$MNJ + (M+3N)J^2 + 3NJ$	$MNJ + (M+3N)J^2 + 2NJ$	NJ	–	$O(MNJ)$
Mirzal (2014)	$MNJ + (M+3N)J^2 + 5NJ$	$MNJ + (M+3N)J^2 + 3NJ$	NJ	–	$O(MNJ)$
This paper	$MNJ + (M+N)J^2 + 4NJ$	$MNJ + (M+N)J^2 + 5NJ$	J	–	$O(MNJ)$

Hence (13) becomes

$$\boldsymbol{f}_j \leftarrow [\mathbf{X}^{(j)}\boldsymbol{g}_j - \frac{\mathbf{F}^{(j)T}\mathbf{X}^{(j)}\boldsymbol{g}_j}{\mathbf{F}^{(j)T}\mathbf{F}^{(j)}}\mathbf{F}^{(j)}]_+.$$

Since the orthogonal constraint $\mathbf{F}^{(j)T}\boldsymbol{f}_j = 0$ does not affect $\boldsymbol{g}_j$, we can use the same update rule of HALS-NMF, that is, with (10),

$$\begin{aligned} \boldsymbol{f}_j &\leftarrow [\mathbf{X}^{(j)}\boldsymbol{g}_j - \frac{\mathbf{F}^{(j)T}\mathbf{X}^{(j)}\boldsymbol{g}_j}{\mathbf{F}^{(j)T}\mathbf{F}^{(j)}}\mathbf{F}^{(j)}]_+, \text{ and} \\ \boldsymbol{g_j} &\leftarrow [\mathbf{X}^{(j)T}\boldsymbol{f}_j]_+. \end{aligned}$$

Using $\mathbf{X}^{(j)} = \mathbf{X} - \sum_{p\neq l}\boldsymbol{f}_p\boldsymbol{g}_p^T = \mathbf{X} - \mathbf{F}\mathbf{G}^T + \boldsymbol{f}_j\boldsymbol{g}_j^T$, we have the final form of updating rules:

$$\begin{aligned} \boldsymbol{f}_j &\leftarrow [\boldsymbol{h} - \frac{\mathbf{F}^{(j)T}\boldsymbol{h}}{\mathbf{F}^{(j)T}\mathbf{F}^{(j)}}\mathbf{F}^{(j)}]_+, \text{ and} \\ \boldsymbol{g_j} &\leftarrow [(\mathbf{X}^T\mathbf{F})_j - \mathbf{G}(\mathbf{F}^T\mathbf{F})_j + \boldsymbol{g}_j\boldsymbol{f}_j^T\boldsymbol{f}_j]_+, \quad \text{where} \\ \boldsymbol{h} &= (\mathbf{X}\mathbf{G})_j - \mathbf{F}(\mathbf{G}^T\mathbf{G})_j + \boldsymbol{f}_j\boldsymbol{g}_j^T\boldsymbol{g}_j. \end{aligned}$$

The zero-lock problem is resolved by $[\,]_+$ operation as Mirzal (2014) does. The proposed HALS ONMF algorithm is shown in Algorithm 1.

4.3. Computational Complexity

We compared the computational complexities of ONMF algorithms including ours. The asymptotic worst-case computational complexities are all the same, that is, $O(MNJ)$, where M and N are the number of rows and columns of $\mathbf{X}$, respectively, and J is the number of components, equivalently the number of columns of $\mathbf{F}$. They are somewhat different in the number of operations as seen in Table 2.[4] The proposed HALS ONMF is beneficial only if $M \ll N$ and J is large enough. However, the actual speed of these algorithms strongly depends on the number of iterations until convergence. As will be shown, the number of iterations is much smaller in the proposed algorithm than those of the other algorithms.

4. We omit Pompili's PGD based ONMF in Table 2. Because the PGD scheme needs linear search with a fixed step-size and, thus, the speed depends on the other type of iterations. They reported their ONMF needs a higher computational cost than traditional ONMF algorithms.

Algorithm 1 Fast HALS-Orthogonal NMF

Input: Nonnegative matrix $\mathbf{X}$, Number of components J
Output: Decomposing nonnegative matrices $\mathbf{F}$ and $\mathbf{G}$ such that $\mathbf{X} \simeq \mathbf{F}\mathbf{G}^T$ and $\mathbf{F}^T\mathbf{F} \cong \mathbf{I}$.

Initialize $\mathbf{F}$ and $\mathbf{G}$ arbitrary.
$\mathbf{U} = \mathbf{F}\mathbf{1}_J$
repeat
  $\mathbf{A} = \mathbf{X}\mathbf{G}$
  $\mathbf{B} = \mathbf{G}^T\mathbf{G}$
  for $j = 1$ **to** J **do**
    $\mathbf{F}^{(j)} = \mathbf{U} - \boldsymbol{f}_j$
    $\boldsymbol{h} = \mathbf{A}_j - \mathbf{F}\mathbf{B}_j + \mathbf{B}_{jj}\boldsymbol{f}_j$
    $\boldsymbol{f}_j = [\boldsymbol{h} - \frac{\mathbf{F}^{(j)}\boldsymbol{h}}{\mathbf{F}^{(j)T}\mathbf{F}^{(j)}}\mathbf{F}^{(j)}]_+$
    $\boldsymbol{f}_j = \boldsymbol{f}_j / \parallel \boldsymbol{f}_j \parallel^2$
    $\mathbf{U} = \mathbf{F}^{(j)} + \boldsymbol{f}_j$
  end for
  $\mathbf{C} = \mathbf{X}^T\mathbf{F}$
  $\mathbf{D} = \mathbf{F}^T\mathbf{F}$
  for $j = 1$ **to** J **do**
    $\boldsymbol{g}_j \leftarrow [\mathbf{C}_j - \mathbf{G}\mathbf{D}_j + \mathbf{D}_{jj}\boldsymbol{g}_j]_+$
  end for
until Convergence criterion is satisfied.

Table 3: Datasets used in the experiments. Here #nnz is the number of non-zero values.

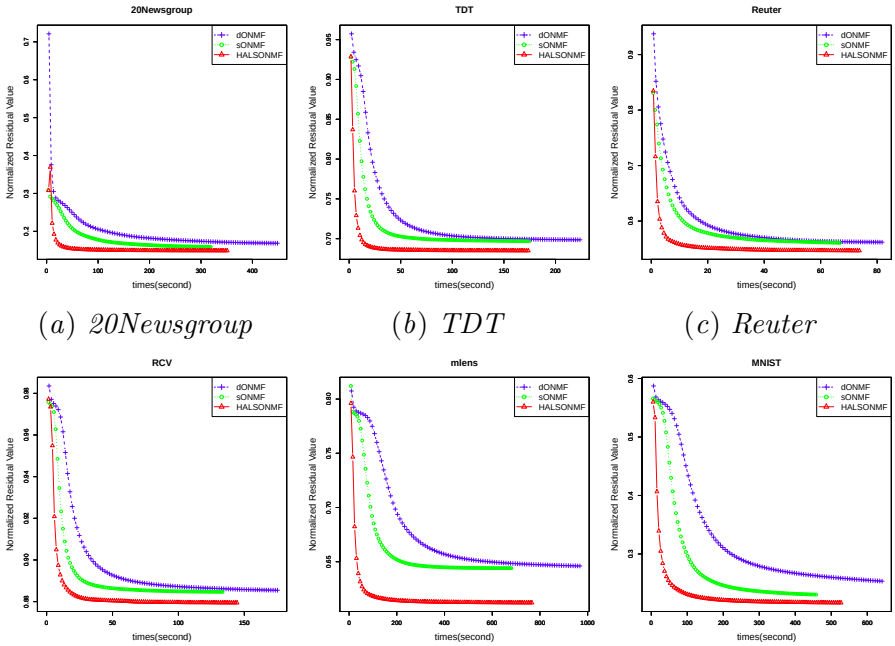
Dataset	Size	#nnz	type
20Newsgroup	61188×18774	2435219	Document
TDT	36771×9394	1224135	Document
RCV	29992×9625	730879	Document
Reuters21678	18993×8293	389455	Document
MNIST	784×70000	10505375	Image
Mlens	71567×65133	10000054	Rating

5. Experiments

5.1. Datasets

We compared the performance of those algorithms on six datasets ranging from document datasets to rating datasets. The summary of datasets is shown in Table 3.[5]

5. These datasets are downloadable datasets from http://www.cad.zju.edu.cn/home/dengcai/Data/TextData.html (Cai et al. (2009)).



(a) *20Newsgroup* (b) *TDT* (c) *Reuter*

(d) *RCV* (e) *mlens* (f) *MNIST*

Figure 1: Comparison of ONMF algorithms. The proposed HALS algorithm converges faster than the other two conventional algorithms.

5.2. Performance Evaluation

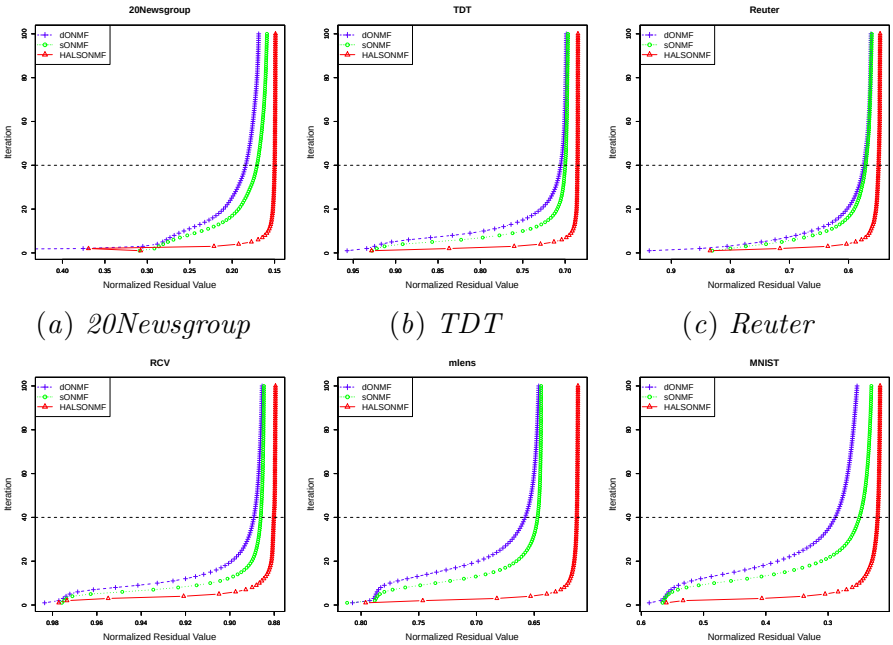
We evaluated the degree of approximation and the degree of orthogonality by two indices:

$$\text{Normalized Residual Value:} \quad \frac{\| \mathbf{X} - \mathbf{F}\mathbf{G}^T \|_F^2}{\| \mathbf{X} \|_F^2}, \text{ and} \tag{14}$$

$$\text{Orthogonality:} \quad \|\mathbf{F}^T\mathbf{F} - \mathbf{I}\|_F^2. \tag{15}$$

The smaller value of the measure, the better the algorithm is. We compared our algorithm with conventional two ONMF algorithms without a weighting parameter. We ignored any ONMF algorithm requiring a weighting parameter because of an additional high cost necessary for determining the value through trial and error. We compared the proposed ONMF algorithms (**HALS**) with **dONMF** algorithm (Ding et al. (2006)) and **sONMF** algorithm (Yoo and Choi (2008)). We employed the same evaluation setting as in Li et al. (2012). The average measure value over 10 trials with different initial values is reported here. We fixed the number of iterations to 100 for all algorithms. We evaluated the computation time (seconds), the normalized residual value (14), and the degree of orthogonality (15).

Figure 1 shows the values of normalized residual for $J = 30$ (the number of components). The proposed HALS converges faster than the other two algorithms. The HALS converges before 200 seconds on all datasets due to the smaller number of iterations. This is because of the efficiency of minimization is guaranteed in each NLS problem. The vector-wise update algorithms, for each vector, find the best solution by one updating ((7) and (8)) for each



(a) *20Newsgroup* (b) *TDT* (c) *Reuter*

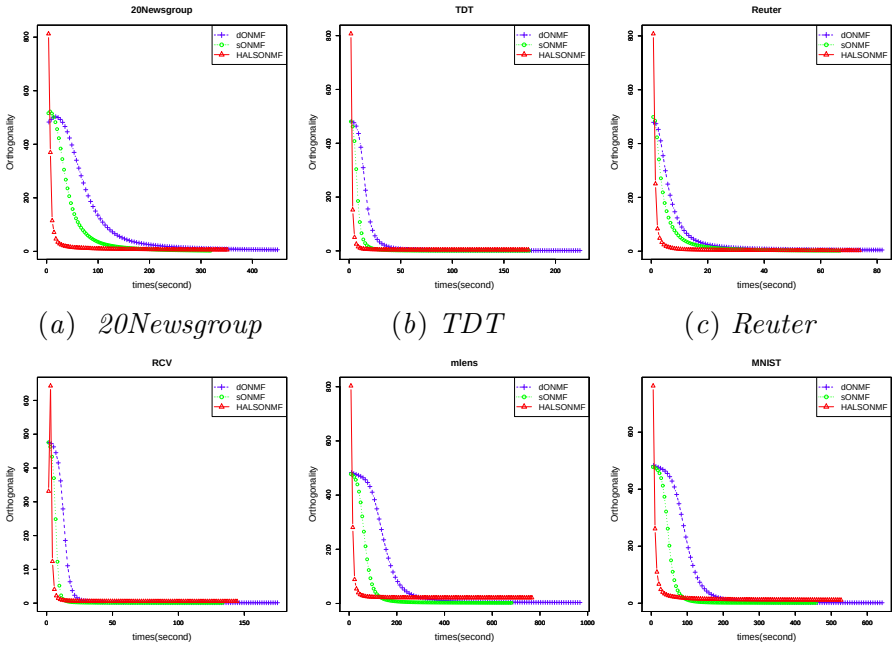
(d) *RCV* (e) *mlens* (f) *MNIST*

Figure 2: Iteration number until convergence in three algorithms. The proposed HALS algorithm converges around 40 iterations.

of column-wise NLS problems. Figure 2 shows the number of iterations consumed until reaching a pre-determined accuracy. It is difficult to specify a necessary accuracy, but HALS converges around 40 iterations that is much smaller than the others. Figure 3 shows the degree of orthogonality for $J = 30$. The HALS archived a high degree of orthogonality earlier than those of the other two, though the final degree of orthogonality is a little less than those of the other two algorithms.

6. Conclusion

In this paper we have proposed a fast algorithm for solving the one-sided orthogonal nonnegative matrix factorization problems. Orthogonal NMF algorithms proposed so far were slow in convergence because they are based on multiplicative update algorithm or project gradient descent, both of which require matrix-wise updates. Therefore, we have proposed a column-wise update algorithm. To incorporate the orthogonality condition and the nonnegativity condition in the column-wise updating rule, we have derived another but equivalent set of conditions. Experiments on six real-life datasets showed that the proposed algorithm is faster in convergence while keeping a satisfactory level of orthogonality.



(a) *20Newsgroup* (b) *TDT* (c) *Reuter*

(d) *RCV* (e) *mlens* (f) *MNIST*

Figure 3: Orthogonality attained by three ONMF algorithms. The proposed HALS algorithm converges faster than the other two conventional algorithms, but the final state is worse than the other two.

References

Deng Cai, Xuanhui Wang, and Xiaofei He. Probabilistic dyadic data analysis with local and global consistency. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 105–112. ACM, 2009.

Deng Cai, Xiaofei He, Jiawei Han, and Thomas S Huang. Graph regularized nonnegative matrix factorization for data representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1548–1560, 2011.

Andrzej Cichocki and PHAN Anh-Huy. Fast local algorithms for large scale nonnegative matrix and tensor factorizations. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 92(3):708–721, 2009.

Andrzej Cichocki, Rafal Zdunek, Anh Huy Phan, and Shun-ichi Amari. *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009.

Chris Ding, Tao Li, Wei Peng, and Haesun Park. Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pages 126–135. ACM, 2006.

Cho-Jui Hsieh and Inderjit S Dhillon. Fast coordinate descent methods with variable selection for non-negative matrix factorization. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pages 1064–1072. ACM, 2011.

Jingu Kim and Haesun Park. Fast nonnegative matrix factorization: An active-set-like method and comparisons. *SIAM Journal on Scientific Computing*, 33(6):3261–3281, 2011.

Jingu Kim, Yunlong He, and Haesun Park. Algorithms for nonnegative matrix and tensor factorizations: A unified view based on block coordinate descent framework. *Journal of Global Optimization*, 58(2):285–319, 2014.

Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems 13*, pages 556–562, 2000.

Liangda Li, Guy Lebanon, and Haesun Park. Fast bregman divergence nmf using taylor expansion and coordinate descent. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pages 307–315. ACM, 2012.

Zhao Li, Xindong Wu, and Hong Peng. Nonnegative matrix factorization on orthogonal subspace. *Pattern Recognition Letters*, 31(9):905–911, 2010.

Chih-Jen Lin. Projected gradient methods for nonnegative matrix factorization. *Neural Computation*, 19(10):2756–2779, 2007.

Andri Mirzal. A convergent algorithm for orthogonal nonnegative matrix factorization. *Journal of Computational and Applied Mathematics*, 260:149–166, 2014.

Filippo Pompili, Nicolas Gillis, P-A Absil, and François Glineur. Two algorithms for orthogonal nonnegative matrix factorization with application to clustering. *arXiv preprint arXiv:1201.0901*, 2012.

Jiho Yoo and Seungjin Choi. Orthogonal nonnegative matrix factorization: Multiplicative updates on stiefel manifolds. In *Intelligent Data Engineering and Automated Learning*, pages 140–147. Springer, 2008.