Advance Access publication on 19 July 2021
https://doi.org/https://doi.org/10.1093/comjnl/bxab103

Non-negative Matrix Factorization: A Survey

Jiangzhang Gan[1], Tong Liu[1], Li Li[2] and Jilian Zhang[1,*]

[1] *School of Natural and Computational Sciences, Massey University Albany Campus, Auckland 0632, New Zealand*
[2] *College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China*
[3] *College of Cyber Security, Jinan University, Guangzhou 510632, China*
**Corresponding author: zhangjilian@jnu.edu.cn*

Non-negative matrix factorization (NMF) is a powerful tool for data science researchers, and it has been successfully applied to data mining and machine learning community, due to its advantages such as simple form, good interpretability and less storage space. In this paper, we give a detailed survey on existing NMF methods, including a comprehensive analysis of their design principles, characteristics and drawbacks. In addition, we also discuss various variants of NMF methods and analyse properties and applications of these variants. Finally, we evaluate the performance of nine NMF methods through numerical experiments, and the results show that NMF methods perform well in clustering tasks.

Keywords: non-negative matrix factorization; data mining; dimensionality reduction; clustering

Received 1 December 2020; Revised 18 May 2021;
Guest Editor: Xiaofeng Zhu

1. INTRODUCTION

In recent years, technologies related to data analysis have attracted much attention, and many real-world applications are in urgent need for faster and better data processing techniques [1, 2]. In data processing field, matrix is usually an indispensable form for organizing and describing data [3, 4]. First, structured data is normally stored in the form of matrix, whereas unstructured data is usually converted to structured data before processing. Second, research on matrix has a long history and has accumulated tremendous research outputs, which can be directly applied to many data analysis problems [5, 6]. Therefore, as one of the popular organization forms for analysing and processing data, matrix plays a critical role in data science. Matrix factorization is an important method to study properties of matrix, and it can help us understand the nature of matrix. At present, matrix factorization has been widely used in many real applications, such as data compression and recovery, information retrieval, recommender systems and feature extraction [7, 8]. There are many analysis methods using matrix factorization to solve practical problems, such as principal component analysis and independent component analysis [9, 10]. However, these methods have one thing in common, i.e. the obtained matrix contains both positive and negative values, which limits application scenarios of these methods [11, 12]. For example, in real applications such as image processing or text data analysis, the data is non-negative, so the matrix after decomposition should not contain negative values [13–15]. Therefore, it is of practical importance to ensure non-negativity of the elements in the decomposed low-rank matrix.

In 1999, Lee *et al.* proposed a new matrix factorization method, named non-negative matrix factorization (NMF), which has aroused widespread attention among researchers around the world [16]. NMF is a new method for matrix factorization, which can deal with large-scale data. On the other hand, because of the introduction of non-negativity constraints, the decomposition results of NMF have a wide range of practical applications and better interpretability [17, 18].

Current research on NMF mainly focuses on several aspects, i.e. design of regularization terms, design of data items, application and optimization methods. The design of data item is to make the NMF better deal with data with various noises. Although NMF method has been applied to many practical tasks, it still faces some problems on dealing with noisy data sets. This is because traditional NMF methods use mean square error to measure the reconstruction error of matrix factorization. However, in order to reduce the value of objective function, the mean square error loss function forcibly fits outliers, which reduces the accuracy of the original data representation. To deal with this issue, Huang *et al.* employed $l_{2,1}$-norm to measure the reconstruction error of matrix factorization [19]. Du *et al.* proposed an NMF method based on the correntropy-induced metric, which assumes that noise obeys a non-Gaussian distribution [20]. Yang *et al.* utilizes Lasso regularization and Laplacian regularization to deal with noisy data. Here, Lasso regularization is designed to avoid overfitting problems and select sparse subsets of features, whereas Laplacian regularization preservers local structure of data [21].

The design of regularization term is to make NMF algorithms to have a better ability to represent data under various assumptions. For example, sparse representation-based regularization terms can be used to learn meaningful features, graph-based regularization terms can preservers the local structure of data. By designing different regularization terms, NMF can be applied to more practical problems. Hoyer *et al.* proposed an improved NMF model, that is, NMF with sparseness constraints, which can easily control sparseness degree of the basis vector [22]. Yuan *et al.* proposed projective NMF for improving linear mapping ability of the feature space solution of non-negative matrix [23]. Ding *et al.* proposed an orthogonal non-negative matrix factorization (ONMF) [24]. In addition, Ding *et al.* relaxed the restriction on non-negativity of NMF, allowing the input matrix and the decomposed base matrix have negative values, and they proposed semi-NMF and convex-NMF, thereby expanding the scope of application of NMF [25].

Due to the non-negativity constraint, the NMF optimization method is different from tradition matrix factorization method. Therefore, the NMF optimization method is also a hot topic in NMF research. The existing NMF algorithms are mainly based on improvement of the three major NMF optimization methods [26]. Multiplicative update (MU) method is a classic NMF method, which alternately updates the non-negative matrix through multiplication [27, 28]. The method not only reduces reconstruction error gradually, but also guarantees non-negativity of the resulting low-rank matrix. The MU method is easy to implement and can often produce better decomposition results. However, there is no guarantee of convergence, and there are some defects in numerical calculations. Alternating least squares method (ALS) is a simple and intuitive method. It first solves the unconstrained problem, and then projects the obtain matrix to a non-negative space [29, 30]. This method has large errors and there is no convergence guarantee, but it is a better initialization method. Alternating non-negative least squares method (ANLS) is a method for solving optimization problems with boundary constraints [31, 32], which decomposes an NMF problem into two sub-optimization problems with non-negative constraints. Among them, the projected gradient method proposed by Lin *et al.* is a common method to solve the optimal solution of the sub-problem, but disadvantages of this method include excessive amount of calculations and high computational complexity [33]. The hierarchical alternating least squares (HALS) method further decomposes the sub-problems, where each time a certain row or a certain column of the decomposition matrix is solved [34].

As a data analysis method, NMF has been proven to be useful in many real applications. In clustering tasks, NMF-based clustering methods have shown good performance [20, 35]. In image processing, NMF is an effective method for image data dimensionality reduction and feature extraction, which is usually used to extract image features to facilitate fast and automatic recognition [36, 37]. In addition, NMF is also used in text analysis, such as identifying semantic relevance between documents, information extraction and indexing [38]. In recommender systems, the problem of incomplete data can be solved by processing user's historical data through NMF [39, 40]. In the field of biomedicine research, NMF can be used to analyse molecular sequence of DNA, and it also can be used to select drug components for new drug discovery [38, 41].

With deep learning methods receiving more and more attention in data processing, many researchers pay attention to the combination of deep learning methods and NMF for data representation. For example, Trigeorgis *et al.* proposed deep semi-NMF method, where the method gives a reasonable explanation for each layer of networks and can express the hidden layer features of complex data. Flenner *et al.* introduced a deep NMF network capable of producing interpretable hierarchical classification of many types of data [42]. Nie *et al.* proposed a jointly combinatorial scheme to concentrate the strengths of both deep neural networks (DNN) and NMF for speech separation, in which NMF is used to learn the basis spectra that are integrated into a DNN to directly reconstruct the magnitude spectrograms of speech and noise [43, 44]. Chen *et al.* proposed an end-to-end model, named Attention-based Multi-NMF DNN, which combines clinical data and gene expression data extracted by multiple NMF algorithms for prognostic prediction of breast cancer [45]. Wisdom *et al.* proposed a novel recurrent neural network architecture for speech separation, which can solve the optimization problem for sparse NMF [46, 47].

In this paper, we review the NMF problem from four aspects, i.e. data item, regularization item, application and NMF optimization method. Specifically, we present a detailed summary to NMF, including the basic concepts, optimization of NMF and some variants of NMF. The contributions of the paper are 2-fold: (1) we summarize nine classic NMF optimization methods, discussing advantages and disadvantages of these methods and (2) we verify the efficiency and convergence of these NMF methods through numerical experiments. The organization of the paper is as follows. We introduce nine classical NMF algorithms in detail in Section 2. Then, we present five variant models of NMF in Section 3. In Section 4, we verify the performance of nine NMF methods with respect to clustering task.

2. NMF ALGORITHMS

Given a non-negative matrix $\mathbf{V} \in \mathbb{R}^{n\times d}$ and a positive integer $r < \min(m, d)$, the problem of NMF is to find non-negative matrices $\mathbf{W} \in \mathbb{R}^{n\times r}$ and $\mathbf{H} \in \mathbb{R}^{r\times d}$, such that the follow is minimized:

$$\begin{aligned} &f(\mathbf{W},\mathbf{H}) = \|\mathbf{V}-\mathbf{W}\mathbf{H}\|_F^2 \\ &s.t., \mathbf{W} \geq 0, \mathbf{H} \geq 0 \end{aligned} \tag{1}$$

From the above NMF problem statement, it is clear that the aim of NMF is to find an approximation of $\mathbf{V}$ using the product of two matrices $\mathbf{W}$ and $\mathbf{H}$. In this section, we introduce nine NMF methods for solving model Equation (1)

2.1. MU

MUs method was originally used to solve the non-negative least squares problem, and Lee *et al.* first applied this method to solve the NMF problem [48–50]. This method updates $\mathbf{W}$ and $\mathbf{H}$ by alternating iterations. The update rules are as follows:

$$\begin{aligned} \mathbf{H}_{bj}^{k+1} &= \mathbf{H}_{bj}^{k} - \frac{\mathbf{H}_{bj}^{k}}{((\mathbf{W}^k)^T\mathbf{W}^k\mathbf{H}^k)_{bj}}\Delta f_{\mathbf{H}} \\ &= \mathbf{H}_{bj}^{k}\frac{((\mathbf{W}^k)^T\mathbf{V})}{((\mathbf{W}^k)^T\mathbf{W}^k\mathbf{W}^k)_{bj}} \end{aligned} \tag{2}$$

and

$$\begin{aligned} \mathbf{W}_{ia}^{k+1} &= \mathbf{W}_{ia}^{k} - \frac{\mathbf{W}_{ia}^{k}}{(\mathbf{W}^k\mathbf{H}^{k+1}(\mathbf{H}^{k+1})^T)_{bj}}\Delta f_{\mathbf{W}} \\ &= \mathbf{W}_{ia}^{k}\frac{(\mathbf{V}(\mathbf{H}^{k+1})^T)}{(\mathbf{W}^k\mathbf{H}^{k+1}(\mathbf{H}^{k+1})^T)_{bj}} \end{aligned} \tag{3}$$

With the above MU method, $\mathbf{W}$ and $\mathbf{H}$ can converge to a stable point and the non-negativity of the matrix is guaranteed during each iteration [51]. MU method has attracted many researchers' attention, because it is easy to implement and it can produce better results. However, there are some issues that need to be solved: (1) since MU is a first-order gradient descent method, its convergence speed is slow; (2) the method does not guarantee convergence to a local minimum, and solution obtained by the method is not necessarily a stable point; and (3) the method may have a zero denominator during iteration [52, 53]. To deal with the above issues, many researchers improved the MU method. For example, Gillis *et al.* modified MU method by adding a small positive lower bound to the result after each update [54]. However, the method may result in a non-sparse resulting matrix. Despite these improvements, the MU method still remains to be a very inefficient numerical method for NMF [55, 56].

2.2. ANLS

ANLS is a popular method to solve NMF [57, 58]. The MU method solves the problem of NMF by alternating updating, which is a special case of ANLS method [59]. ANLS decomposes the corresponding NMF problem into two sub-optimization problems with non-negative constraints, and then solves the optimal solution of the sub-problems. The two sub-problems are given as follows:

$$\mathbf{H}^{t+1} = \arg\min_{\mathbf{H}} \left\|\mathbf{V}-\mathbf{W}^t\mathbf{H}\right\|_F^2 \tag{4}$$

and

$$\mathbf{W}^{t+1} = \arg\min_{\mathbf{W}} \left\|\mathbf{V}-\mathbf{W}\mathbf{H}^{t+1}\right\|_F^2 \tag{5}$$

Compared with the MU method, ANLS can provide better optimization capability and can converge to a stable point. Solving sub-problems with non-negative constraints is the main part of this method. However, since each step of the update is to solve the optimal solution of the sub-problem, the method requires large amount of calculations and has high computational complexity. At present, there are many methods [60, 61] to solve the sub-problems, among which the projected gradient method is a commonly used method to solve the bounded constraint problem.

2.3. ALS

ALS method is easy to implement, and its computational cost is relatively low [62]. The method first solves an unconstrained problem, and then projects the result to a non-negative space, so as to satisfy the non-negative constraint [29, 63], i.e.

$$\mathbf{H}^{t+1} = \max(\arg\min_{\mathbf{H}} \left\|\mathbf{V}-\mathbf{W}^t\mathbf{H}\right\|_F^2, 0) \tag{6}$$

The update method of $\mathbf{W}$ is similar to Equation (6). Experiments show that this method is usually difficult to obtain reasonable results. When dealing with dense matrices, the loss value of the ALS method tends to oscillate during the update process and cannot be used as a stable numerical method for practical calculations. For sparse matrix, in the initial stage of the iteration, the error value decreases faster, but as the iteration progresses, the error value will not continue to decrease [64–66]. Therefore, the ALS method is not suitable for directly solving NMF. In real applications, this method is usually used for preprocessing original data, i.e. initializing the data.

2.4. HALS

HALS method obtains a simpler form by further decomposing the sub-problem, which updates only one column or row of the matrix at a time while keeping the rest of the matrix unchanged [64, 67]. The sub-problem of NMF can be further decomposed into the following form:

$$\begin{aligned}&\|\mathbf{V}-\mathbf{WH}\|_F^2\\&=\|\mathbf{V}-\textstyle\sum_{i\neq j}\mathbf{w}_i\mathbf{h}_i-\mathbf{w}_j\mathbf{h}_j\|_F^2\end{aligned}\tag{7}$$

For Equation (7), HALS method updates a certain column of **W** or **H** each time, and then projects to the non-negative space, i.e.

$$\mathbf{w}_j=\max\{\frac{\mathbf{Vh}_j^T-\sum_{i\neq j}\mathbf{w}_i(\mathbf{h}_j\mathbf{h}_j^T)}{\|\mathbf{h}_j\|^2},0\}\tag{8}$$

or

$$\mathbf{h}_j=\max\{\frac{\mathbf{V}^T\mathbf{w}_j-\sum_{i\neq j}\mathbf{h}_i(\mathbf{w}_j^T\mathbf{w}_j)}{\|\mathbf{w}_j\|^2},0\}\tag{9}$$

Compared with the MU method, HALS method converges faster and guarantees convergence to a local minimum under weaker conditions [68]. In addition, the computational overhead of the HALS method is mainly due to calculation of the gradients. The HALS method is an excellent algorithm that achieves a relative balance between efficiency and accuracy. However, for large-scale matrices, the calculation efficiency of the method is still low.

2.5. Newton-like method

Newton-like method is an effective method for solving convex optimization problems. This method can accelerate the convergence process of objective function by mining second-order effective information of objective function [69, 70]. At iteration t, the method first approximates the objective function around the current iterate $\mathbf{H}^k$ by using the following quadratic model, i.e.

$$\begin{aligned}&\phi^t(\mathbf{H})=f(\mathbf{H}^t)+(\mathbf{H}-\mathbf{H}^t)^T\Delta f(\mathbf{H}^t)+\\&\tfrac{1}{2\alpha}(\mathbf{H}-\mathbf{H}^t)^T\mathbf{D}^t(\mathbf{H}-\mathbf{H}^t)\end{aligned}\tag{10}$$

And then, we can obtain

$$\bar{\mathbf{H}}^t=\arg\min\phi^t(\mathbf{H},\alpha)\tag{11}$$

which is then used to obtain the new iterate by simply setting

$$\mathbf{H}^{t+1}=\mathbf{H}^t+\beta(\bar{\mathbf{H}}_\alpha^t-\mathbf{H}^t)\tag{12}$$

where $\beta\in(0,1]$ is a step size.

2.6. Projected gradient method

Projection gradient method is a popular method for solving bounded constraint problems. The main difference between different projected gradient methods is the choice of step size [71]. NMF is a typical bounded optimization problem, so many researches use projected gradient method to solve NMF. The bounded optimization problem is defined as follows:

$$\begin{aligned}&\min_{x\in R^n}f(x)\\&s,t.l_i\le x_i\le u_i,i=1,\ldots,n\end{aligned}\tag{13}$$

where $f(x)$ is a continuously differentiable function, **l** and **u** are lower and upper bounds, and k is the index of iterations. Projected gradient method updates the current solution $\mathbf{x}^k$ and $\mathbf{x}^{k+1}$ through the following rule:

$$x^{k+1}=P[x^k-\alpha^k\Delta f(x^k)]\tag{14}$$

where

$$P[x_i]=\begin{cases}x_i & \text{if } l_i<x_i<u_i\\u_i & \text{if } x_i\ge u_i\\l_i & \text{if } x_i\le l_i\end{cases}\tag{15}$$

where P maps a point back to the bounded feasible region, and variants of projected gradient methods differ in selecting the step size. Motivated by this fact, Lin *et al.* proposed to employ projected gradient methods for NMF [72], and we list the details in Algorithm 1.

Algorithm 1 Projected gradient method for NMF.

```
Input: $\mathbf{V}$, $\mathbf{W^t}$
Initialize $\mathbf{W}^1\ge 0,\mathbf{H}^1\ge 0,k=1,\alpha=0.1,\beta=0.1$
repeat
    Update $\mathbf{H}^{k+1}=P[\mathbf{H}^k-\alpha\Delta f(\mathbf{H}^k)]$
    $\alpha=\alpha\beta$
    $k=k+1$
until Stopping criterion Equation (16) is satisfied
Output: $\mathbf{H}^{t+1}$
```

To ensures the function value in each iteration to decrease, the stopping condition of projected gradient methods is given as follows:

$$f(\mathbf{H}^{k+1})-f(\mathbf{H}^k)\le\gamma\Delta f(\mathbf{H}^k)^T(\mathbf{H}^{k+1}-\mathbf{H}^k)\tag{16}$$

where γ is a parameter.

2.7. Active set method

Since the combination coefficients of matrix are generally sparse, it is a good choice to employ active set algorithms for solving NMF problem [73]. These active set strategies can improve the accuracy of NMF methods, while keeping the computational cost at a low level. For example, Kim *et al.* applied active method to NMF, to improve the efficiency of traditional NMF methods [57].

The constraint $h_{ij}\ge 0$ is said to be active in the optimal solution $\mathbf{H}^*$, if for any feasible solution of NMF, the current iteration **H** is classified into two parts, i.e. active variables and inactive variables. Given a small positive constant ζ, the active set can be estimated as follows:

$$(AS)I(vec(V)) = \{i : vec(V)_i \leq 0\} \tag{17}$$

where $I(vec(V))$ represents the index set that contains estimated indices of the active variables.

We use $I^k = I(vec(\mathbf{H}^t))$ and $F^t = F(vec(\mathbf{H}^t))$ in Equation (18), where F^t denotes the remaining index of $vec(\mathbf{H})$. At iteration t, the variables of $vec(\mathbf{H})$ with indices in I^k are called active variables, whereas the remaining variables are called inactive variables. A non-negative matrix $\mathbf{H}^*$ is said to be a stationary point of ANLS, if for every $i = 1, 2, \ldots, r_n$, we have

$$\begin{cases} vec(\nabla f(\mathbf{W},\mathbf{H}))_i \geq 0, & \forall i \in I^* \\ vec(\nabla f(\mathbf{W},\mathbf{H}))_i = 0, & \forall i \in F^* \end{cases} \tag{18}$$

where we have $I^* = \{i : vec(\mathbf{H}_i) = 0\}$ and $F^* = \{1, 2 \ldots, nr\}$. Strict complementary condition holds $\mathbf{H}^*$, if the strict inequalities hold in the first constraint of Equation (18).

2.8. Alternating direction multiplier method

Alternating direction multiplier method is a tradition optimization method, which is well suited to distributed convex optimization. To facilitate presenting alternating minimization, we first introduce two auxiliary variables **X** and **Y**, and consider the following equivalent model:

$$\begin{aligned} &\min \tfrac{1}{2} \left\| \mathbf{V} - \mathbf{WH} \right\|_F^2 \\ &s.t, \mathbf{W} - \mathbf{X} = 0, \mathbf{H} - \mathbf{Y} = 0, \mathbf{X} \geq 0, \mathbf{Y} \geq 0 \end{aligned} \tag{19}$$

where we have $\mathbf{X} \in \mathbb{R}^{n \times r}$ and $\mathbf{Y} \in \mathbb{R}^{r \times d}$. The augmented Lagrangian function of Equation (19) is defined as follows:

$$\begin{aligned} &\phi(\mathbf{W}, \mathbf{H}, \mathbf{X}, \mathbf{Y}, \mathbf{\Lambda}, \mathbf{\Pi}) = \tfrac{1}{2} \left\| \mathbf{V} - \mathbf{WH} \right\|_F^2 + \\ &\langle \Lambda, \mathbf{W} - \mathbf{X} \rangle + \langle \Pi, \mathbf{H} - \mathbf{Y} \rangle + \tfrac{\alpha}{2} \left\| \mathbf{W} - \mathbf{X} \right\|_F^2 \\ &+ \tfrac{\beta}{2} \left\| \mathbf{H} - \mathbf{Y} \right\|_F^2 \end{aligned} \tag{20}$$

where $\Lambda \in \mathbb{R}^{n \times r}$ and $\Pi \in \mathbb{R}^{r \times d}$ are Lagrangian multipliers, $\alpha, \beta \geq 0$ are penalty parameters for the constraints $\mathbf{W} - \mathbf{X} = 0$ and $\mathbf{H} - \mathbf{Y} = 0$, and $\langle \cdot, \cdot \rangle$ denotes the matrix inner product.

The alternating direction method [74] for Equation (19) is derived by successively minimizing the augmented Lagrangian function $\phi(U)$ with respect to **W**, **H**, **X** and **Y**, one at a time while fixing the others at their most recent values, and then updating the multipliers after each sweep of such alternating minimization. The introduction of the two auxiliary variables **X** and **Y** makes it easy to carry out each of the alternating minimization steps. Specifically, these steps can be written in a closed form as follows

$$\begin{aligned} &\mathbf{W} = \mathbf{\Pi} + \beta(\mathbf{H} - \mathbf{Y}) = \\ &(\mathbf{HH}^T + \alpha\mathbf{H} - \Lambda)(\mathbf{HH}^T + \alpha\mathbf{H})^{-1} \\ &\mathbf{H} = \mathbf{\Pi} + \beta(\mathbf{H} - \mathbf{Y}) = \\ &(\mathbf{W}^T\mathbf{W} + \beta\mathbf{H})^{-1}(\mathbf{W}^T\mathbf{V} + \beta\mathbf{Y} - \mathbf{\Pi}) \\ &\mathbf{X} = \max(\mathbf{W} + \Lambda/\alpha, 0) \\ &\mathbf{Y} = \max(\mathbf{H} + \mathbf{\Pi}/\beta, 0) \\ &\mathbf{\Lambda} = \mathbf{\Lambda} + \alpha(\mathbf{W} - \mathbf{X}) \\ &\mathbf{\Pi} = \mathbf{\Pi} + \beta(\mathbf{H} - \mathbf{Y}) \end{aligned} \tag{21}$$

2.9. NeNMF Method

Traditional NMF solvers often suffer from one or some of the following three problems, i.e. slow convergence, numerical instability and non-convergence [75, 76]. Guan *et al.* presented a new NeNMF solver to simultaneously overcome the aforementioned problems [77]. It applies Nesterov's optimal gradient method (OGM) to alternatively optimize one factor with another fixed. Since Equation (1) is a non-convex minimization problem, it is impractical to obtain the optimal solution. However, the block coordinate descent methods can obtain a local solution to Equation (1) through alternatively solving the following equation until converged

$$\begin{aligned} &\mathbf{H}^{t+1} = \arg\min F(\mathbf{W}^t, \mathbf{H}) = \\ &\tfrac{1}{2} \left\| \mathbf{V} - \mathbf{W}^t\mathbf{H} \right\|_F^2 \end{aligned} \tag{22}$$

Most existing NMF solvers are special implementations under this scheme. NeNMF method employs Nesterov's OGM to solve both problems presented in Equation (22).

Recent research has proven that $F(\mathbf{W}^t, \mathbf{H})$ is convex and the gradient $\nabla_H F(\mathbf{W}^t, \mathbf{H})$ is Lipschitz continuous; thus, Nesterov's method can be used to efficiently optimize Equation (22)

$$\begin{aligned} &\mathbf{H}_k = \underset{\mathbf{H} \geq 0}{\arg\min}\, \phi(\mathbf{Y}_k, \mathbf{H}) = F(\mathbf{W}^t, \mathbf{Y}_k) + \\ &\left\langle \nabla_{\mathbf{H}} F(\mathbf{W}^t, \mathbf{Y}_k), \mathbf{H} - \mathbf{Y}_k \right\rangle + \tfrac{\mathbf{L}}{2} \left\| \mathbf{H} - \mathbf{Y}_k \right\|_F^2 \end{aligned} \tag{23}$$

where $\phi(\mathbf{Y}_k, \mathbf{H})$ is the proximal function of $F(\mathbf{W}^t, \mathbf{H})$ on $\mathbf{Y}_k$, $\mathbf{L} = ||\mathbf{W^t}^{\mathbf{T}}\mathbf{W^t}||$ is the Lipschitz constant given in [77], $\langle \cdot, \cdot \rangle$ denotes the matrix inner product, $\mathbf{H}_k$ contains the approximate solution obtained by minimizing the proximal function $\phi(\mathbf{Y}_k, \mathbf{H})$ over H and $\mathbf{Y_k}$ stores the search point that is conducted by linearly combining the latest two approximate solutions, i.e. $\mathbf{H}_k$ and $\mathbf{H}_k - 1$. According to [78], the combination coefficient α_{k+1} is updated in each iteration as follows:

$$\alpha_{k+1} = \frac{1 + \sqrt{4\alpha_k^2 + 1}}{2} \tag{24}$$

NeNMF uses Lagrange multiplier method to solve Equation (23), so we can obtain the Karush–Kuhn–Tucker conditions as follows:

$$\begin{aligned}&\nabla_H \phi(\mathbf{Y}_k, \mathbf{H}) \geq 0\\ &\mathbf{H}_k \geq 0\\ &\nabla_H \phi(\mathbf{Y}_k, H) \otimes \mathbf{H}_k = 0\end{aligned} \tag{25}$$

where $\nabla_{\mathbf{H}} \phi(\mathbf{Y}_k, \mathbf{H}) = \nabla_{\mathbf{H}} F(\mathbf{W}^t, \mathbf{Y}_k) + \mathbf{L}(\mathbf{H}_k - \mathbf{Y}_k)$ is the gradient of $\phi(\mathbf{Y}_k, \mathbf{H})$ with respect to $\mathbf{H}$ at $\mathbf{H}_k$, and $\otimes$ is the Hadamard product. According to Equation (25), we have

$$\mathbf{H}_k = P(\mathbf{Y}_k - \tfrac{1}{L} \nabla_H F(\mathbf{W}^t, \mathbf{Y}_k)) \tag{26}$$

where P($\mathbf{X}$) projects all the the negative entries of $\mathbf{X}$ to zero. By alternatively updating $\mathbf{H}_k, \alpha_{k+1}$ and $\mathbf{Y}_{k+1}$ with Equation (24) and Equation (26) until convergence, the optimal solution can be obtained.

From the above NMF optimization, the majority of traditional NMF optimization algorithms can be unified as alternating minimization or block coordinate descent scheme with different block sizes and various optimization approaches for each block.

3. VARIANTS OF NMF

There are two issues with the standard NMF model. On the one hand, it is impossible to obtain a unique solution by non-negative constraints alone. On the other hand, it is difficult to use a priori knowledge to comprehensively characterize data [22, 61, 79]. To deal with these issues, many researcher design different data item and regularization item, resulting in various variants of NMF. In this section, we introduce five popular variants of NMF.

3.1. Semi-NMF and convex NMF

The basic NMF method constrains each element of the original input matrix to be non-negative, which limits the application of NMF [80, 81]. Ding *et al.* [25] proposed mathematical models of semi-NMF, which are more suitable for general data, thus expanding the application field of original NMF method. Semi-NMF relaxes non-negativity constrains of NMF and allows the data matrix $\mathbf{V}$ and $\mathbf{W}$ to have mixed signs, while restricting only the feature matrix $\mathbf{H}$ to comprise of strictly non-negative components. The objective function is defined as

$$\begin{aligned}&\mathcal{J}_{\text{semi-NMF}} = \tfrac{1}{2} \|\mathbf{V} - \mathbf{WH}\|_F^2\\ &s.t, \mathbf{H} \geq 0\end{aligned} \tag{1}$$

If we regard $\mathbf{W} = [\mathbf{w_1}, \ldots, \mathbf{w_n}]$ as the cluster centroids, then $\mathbf{H} = [\mathbf{h_1}, \ldots, \mathbf{h_n}]$ can be treated as the cluster indicators for each data point. In fact, if we have a matrix $\mathbf{H}$ that is not only non-negative but also orthogonal, then every column vector would have only one positive element, making semi-NMF equivalent to k-means [82].

In semi-NMF, there are no constraints on the basis matrix $\mathbf{W}$. Based on semi-NMF, Ding *et al.* further proposed convex-NMF method. In the convex-NMF, the basis matrix $\mathbf{W}$ is obtained by a linear combination of the samples, i.e. $\mathbf{w}_j = \sum_i u_{ij}\mathbf{v}_i$, where $u_{ij} \geq 0$. The objective function of convex-NMF is defined as

$$\begin{aligned}&\mathcal{J}_{\text{convex-NMF}} = \|\mathbf{V} - \mathbf{VUH}\|_F^2\\ &s.t., \mathbf{U} \geq 0, \mathbf{V} \geq 0\end{aligned} \tag{2}$$

In convex-NMF, for the reason of interpretability, the method restricts basic matrix to convex combinations of the columns of $\mathbf{V}$. This constraint has the advantage that we could interpret the columns $\mathbf{w}_i$ as weighted sums of certain data points. In particular, these columns can capture a notion of centroids. Convex-NMF applies to both non-negative and mixed-sign data matrices [83, 84]. Moreover, convex-NMF has an interesting property, i.e. the factors $\mathbf{U}$ and $\mathbf{H}$ both tend to be sparse.

3.2. ONMF

Although NMF has some excellent properties, there are still rooms for further development. To deal with the issue that the basic NMF cannot achieve a unique solution, Ding *et al.* proposed ONMF. Based on the original NMF, the method imposes an orthogonal constraint on the decomposition factor. The objective function of ONMF is defined as

$$\begin{aligned}&\mathcal{J}_{\text{ONMF}} = \|\mathbf{V} - \mathbf{WH}\|_F^2\\ &s.t., \mathbf{W} \geq 0, \mathbf{H} \geq 0, \mathbf{H}^T\mathbf{H} = \mathbf{I}\end{aligned} \tag{3}$$

or

$$\begin{aligned}&\mathcal{J}_{\text{ONMF}} = \|\mathbf{V} - \mathbf{WH}\|_F^2\\ &s.t., \mathbf{W} \geq 0, \mathbf{H} \geq 0, \mathbf{W}^T\mathbf{W} = \mathbf{I}\end{aligned} \tag{4}$$

The above two models can achieve sparse and unique solutions, and Ding *et al.* proved that the two models are equivalent to k-means clustering model [85, 86]. However, it is worth noting that the substance of the two models is quite different. For example, in terms of clustering tasks, Equation (3) represents clustering based on the columns of input matrix (or samples), whereas Equation (4) represents cluster based on the rows of input matrix (or feature).

3.3. Tri-factorization NMF

In the past decade, researchers have proposed several variants of NMF from different aspects for improving its performance. To improve the uniqueness of the solution and preserve the local property of NMF, the ONMF has been proposed, which imposes the orthogonal condition on the original NMF. Based on the ONMF, Ding *et al.* [24] proposed orthogonal non-negative matrix tri-factorization (NMTF). This method decomposes the data matrix to three factor matrices and preserves double orthogonality conditions, which provide more degrees of freedom than the ONMF. The objective fuction of tri-factorization NMF is presented below

$$\mathcal{J}_{\text{Tri-NMF}} = \|\mathbf{V} - \mathbf{WSH}\|_F^2 \\ s.t, \mathbf{W} \geq 0, \mathbf{H} \geq 0, \mathbf{W}^T\mathbf{W} = \mathbf{I}, \mathbf{HH}^T = \mathbf{I} \tag{5}$$

In this way, columns and rows are clustered simultaneously, and both orthogonality constraints can be satisfied with a good low-rank approximation [87]. During the past decades, NMTF has been successfully used in various applications, such as text data mining, image clustering, recognition and retrieval task, and community detection, etc. [88]. Since NMTF is an important unsupervised learning algorithm, the overwhelming interest of NMTF-based methods is focused on clustering tasks, particularly for image and document clustering problems. Currently, most NMTF-based methods for clustering utilize the square of Euclidean distance as similarity measure to quantify the approximation between the original data matrix and the reconstructed ones.

3.4. Symmetric NMF

Although NMF has performed better than other data analysis methods in many fields [89, 90]. One of the important reasons is that NMF approximates original data by a linear combination of basis vectors [91, 92]. When the data has non-linear structure or lies on a complicated manifold, then NMF may achieve bad results. Symmetric NMF (SNMF) is an effective approach to cluster data with non-linear structure [93, 94]. It only takes into account symmetric matrix that can be constructed by various similarity metrics and factorizes the matrix into two low-rank matrices ($\mathbf{H}$, $\mathbf{H}^T$). The objective of SNMF is defined as

$$\mathcal{J}_{\text{SNMF}} = \left\|\mathbf{A} - \mathbf{HH}^T\right\|_F^2 \\ s.t, \mathbf{H} \geq 0 \tag{6}$$

where $\mathbf{A} \in \mathbb{R}^{n\times n}$ is the similarity matrix measured by a certain distance metric. Kuang *et al.* showed that SNMF is related to spectral clustering (SC), and both of them share a same loss function with different constraints [70, 95]. Therefore, SNMF can be regarded as a graph clustering method, and it is more effective for non-linearly separable data than NMF. Another merit of SNMF is that it can directly generate the clustering indicator without post-processing, whereas SC needs extra post-processing like k-means to finish the clustering task.

3.5. Kernel NMF

NMF and many of its variants are linear in nature, so it is impossible to distinguish between non-linear structures hidden in the data. In traditional machine learning, kernel function can map low-dimensional space data to high-dimensional feature space or infinite-dimensional space through a non-linear mapping, so as to achieve linear separability of the data [96, 97]. Based on this motivation, Zhou *et al.* applied the kernel method to NMF and proposed Kernel Non-negative Matrix Factorization (KNMF) [98]. The objective function of KNMF is defined as

$$\mathcal{J}_{\text{KNMF}} = \left\|\phi(\mathbf{V}) - \mathbf{W}_\phi\mathbf{H}\right\|_F^2 \\ s.t, \mathbf{H} \geqslant 0 \tag{7}$$

where $\phi(.)$ represents the kernel function, which maps the original matrix $\mathbf{V}$ from low-dimensional space to high-dimensional space, $\mathbf{W}_\phi$ is the base in feature space and $\mathbf{H}$ is its combining coefficients, each column of which denotes the dimension-reduced representation for the corresponding object.

4. EXPERIMENTS

In this section, We experimentally evaluate performance of the nine NMF optimization methods on six public data sets, in terms of efficiency and clustering performance. All experiments are conducted in MATLAB on a Win10 machine with a 2.66GHz Intel Quad-core processor and 8GB memory.

4.1. Experimental setting

To verify performance of the NMF solvers, the experiment uses six public data sets, as shown in Table 1. To ensure fairness in the experiment, all NMF solvers start from the same initial point ($\mathbf{W}^0$, $\mathbf{H}^0$), and set all NMF solver iterations to 100 times. In the first part of the experiment, we compare the time required for all NMF solvers to reach 100 iterations to verify the efficiency of the NMF solver. In the second part of the experiment, k-means is used to cluster the base matrix $\mathbf{W}$, so as to compare performance of each solver in clustering and dimensionality reduction tasks. In the clustering task, we use three evaluation indicators (i.e. ACC, NMI, and Pur) to verify the clustering performance of the nine methods. We listed the details of three evaluation metrics as follows:

$$\text{ACC} = \tfrac{N_{\text{correct}}}{N}, \tag{1}$$

where N represents the total sample number and N_{correct} represents the sample number accurately clustered.

$$\text{NMI} = \tfrac{H(A)+H(B)}{H(A,B)} \tag{2}$$

where A and B are clustering results, and $H(A, B)$ is the joint entropy of A and B.

$$\text{Pur} = \tfrac{1}{N}\textstyle\sum_k \max|w_k \cap c_j| \tag{3}$$

where w_k denotes cluster classes, c_j denotes true classes.

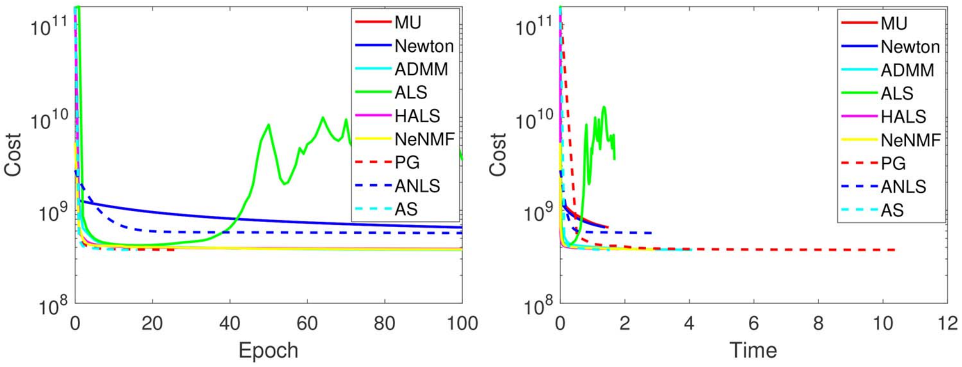

FIGURE 1. Objective value versus iteration epoch and CPU time of nine NMF solvers on Madelon data set.

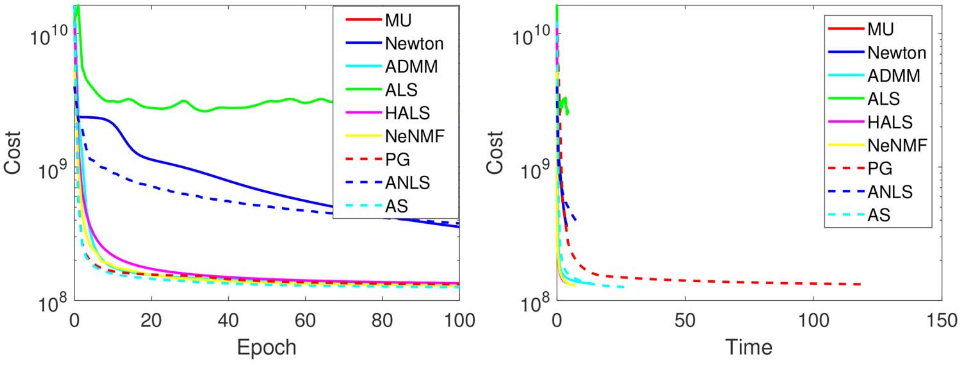

FIGURE 2. Objective value versus iteration epoch and CPU time of nine NMF solvers on PIE_pose data set.

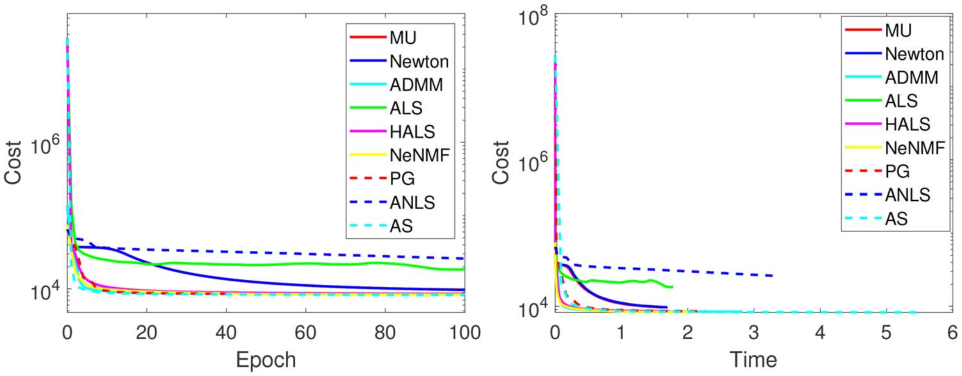

FIGURE 3. Objective value versus iteration epoch and CPU time of nine NMF solvers on Coil20 data set.

4.2. Result analysis

To verify efficiency and convergence of the nine NMF solvers, we report the variation of objective function values for nine NMF solvers during iteration in Figs 1–6. From the above figures, in term of convergence we can see that the ALS method does not guarantee that the value of the objective function is decremented each time during the iteration, other methods can ensure that the objective function is decreasing for each iteration, so that the objective function converges. In terms of efficiency, the ALS method, the NeNMF method, and the HALS method converge to the local optimum value with the least time.

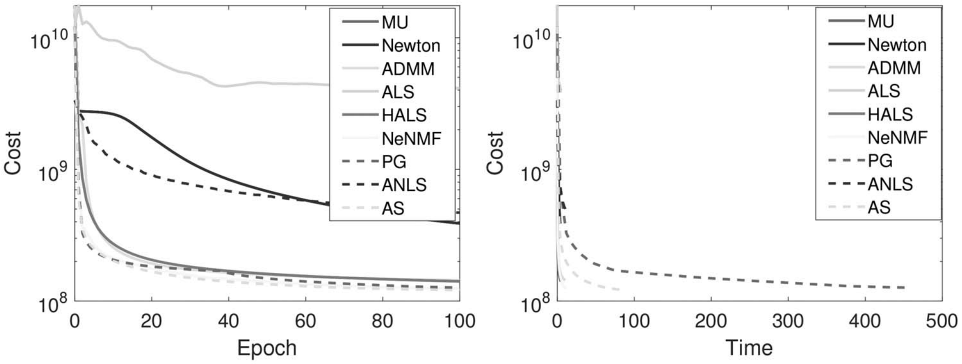

FIGURE 4. Objective value versus iteration epoch and CPU time of nine NMF solvers on AR_face data set.

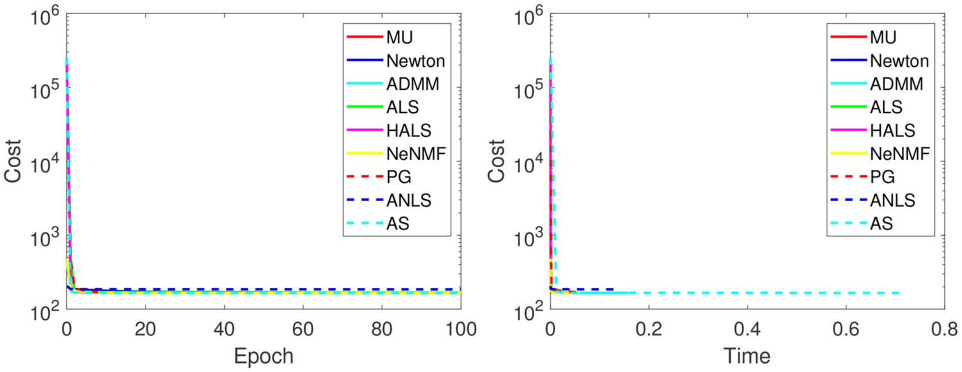

FIGURE 5. Objective value versus iteration epoch and CPU time of nine NMF solvers on Ecoil data set.

TABLE 1. Statistics of the data sets used in the experiment.

Dataset	#samples	#features	#classes
Madelon	2600	500	2
PIE_pose	2856	1024	68
Coil20	1440	1024	20
AR_face	3120	560	120
Ecoil	336	343	8
Jaffe	213	1024	10

To validate performance of nine NMF solvers, we perform experiments using the clustering task and report the experimental results in Tables 2 and 3. From the two tables we can see that the NeNMF method has stable performance and achieves the best results on most of the data sets. The ALS method has achieved the best results on PIE_pose 271 data set, but obtained the worst results on other data sets, which further proves that the ALS method is easy to cause numerical instability. In addition, the other methods outperform *k*-means in most of the cases, which proves the effectiveness of NMF method in clustering task.

5. CONCLUSIONS

NMF decomposes the input non-negative matrix into two low-rank non-negative matrices, which can be effectively applied for many real-world applications. In this paper, we introduced some properties of NMF and summarized several NMF methods in detail. In addition, we also paid attention to variants of NMF and their application scenarios. Although NMF has been successfully applied in many fields, there are still some problems remained to be solved. We proposed several future directions based on the shortcomings of NMF, i.e. (1) although

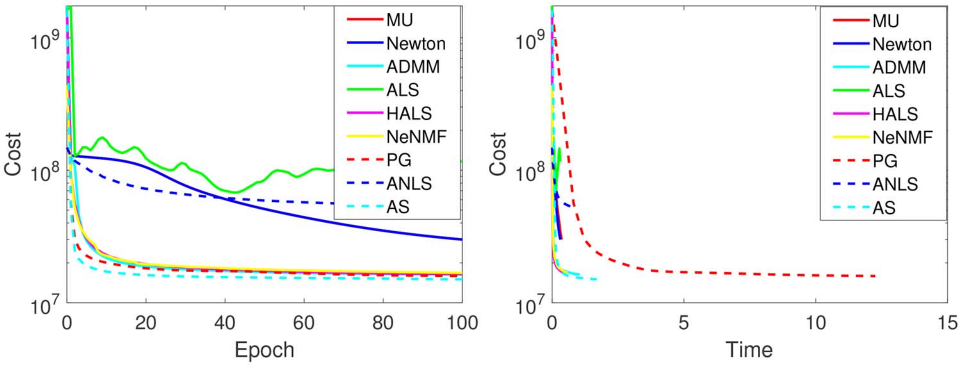

FIGURE 6. Objective value versus iteration epoch and CPU time of nine NMF solvers on Jaffe data set.

TABLE 2. Clustering results on three public data sets, i.e. Madelon, PIE_pose27, and Coil20.

Data sets		Madelon			PIE_pose27l			Coil20	
	ACC	NMI	Pur	ACC	NMI	Pur	ACC	NMI	Pur
k-means	0.5073	0.0002	0.5073	0.2458	0.4933	0.2675	0.5403	0.7254	0.5799
MU	0.5012	0.0012	0.5314	0.4569	0.6763	0.4874	0.6590	0.7525	0.6840
ALS	0.5135	0.0005	0.5355	0.6870	0.8382	0.7227	0.5667	0.6259	0.2020
PG	0.5112	0.0004	0.5023	0.6411	0.7898	0.6646	0.5778	0.7325	0.6312
Newton	0.5234	0.0021	0.4007	0.6008	0.6696	0.8006	0.6010	0.7456	0.7015
ADMM	0.5322	0.0006	0.0.5311	0.5932	0.6494	0.7351	0.6171	0.7351	0.7286
HALS	0.5335	0.0005	0.5698	0.5270	0.7327	0.5532	0.6125	0.6987	0.7012
AS	0.5762	0.0168	0.5489	0.5875	0.7572	0.6201	0.6521	0.7500	0.6778
NeNMF	0.5754	0.0165	0.5456	0.6838	0.7976	0.7027	0.6493	0.7512	0.6819
NALS	0.5788	0.0165	0.5788	0.4405	0.6824	0.4762	0.5840	0.7146	0.6188

TABLE 3. Clustering results on three public data sets, i.e. AR_Face, Ecoil, and Jaffe.

Data sets		AR_Face			Ecoil			Jaffe	
	ACC	NMI	Pur	ACC	NMI	Pur	ACC	NMI	Pur
k-means	0.2554	0.5661	0.2728	0.6190	0.5508	0.6964	0.7887	0.8182	0.8122
MU	0.3468	0.6185	0.3728	0.5476	0.5381	0.7798	0.8357	0.8448	0.8498
ALS	0.3840	0.6511	0.4087	0.7381	0.6049	0.7708	0.6385	0.7226	0.6479
PG	0.3689	0.6595	0.3971	0.4673	0.5066	0.7857	0.8122	0.8742	0.8592
Newton	0.3927	0.7247	0.4007	0.6008	0.6696	0.8006	0.7430	0.8187	0.7015
ADMM	0.2973	0.6326	0.4009	0.4688	0.6494	0.7351	0.6171	0.7351	0.7286
HALS	0.3099	0.5959	0.3369	0.3036	0.1706	0.5714	0.4085	0.4889	0.46951
AS	0.3599	0.6396	0.3856	0.5833	0.5456	0.7917	0.8310	0.8629	0.8451
NeNMF	0.3619	0.6441	0.3875	0.6339	0.5120	0.7768	0.8967	0.8704	0.8967
NALS	0.3003	0.6020	0.3272	0.4107	0.0427	0.4286	0.5305	0.6182	0.5869

there are many NMF algorithms, accuracy and convergence rate still need to be improved, (2) when processing large scale data, efficiency of NMF method needs to be improved, and (3) it is interesting and meaningful to apply NMF methods to many different real-world applications.

DATA AVAILABILITY STATEMENTS

The data underlying this article are available in the UCI Machine Learning Repository at http://archive.ics.uci.edu/ml/index.php and the website of Feature Selection Data sets at https://jundongl.github.io/scikit-feature/.

FUNDING

The National Key Research and Development Plan of China (Grant No. 2018YFB1003701), the National Natural Science Foundation of China (Grant No 62020106013, 61876046, 61972177), Guangdong Provincial Key R&D Plan (No. 2019B1515120010), the Guangxi 'Bagui' Teams for Innovation and Research, the Marsden Fund of New Zealand (MAU1721), and the China Scholarship Council.

REFERENCES

[1] Liu, D. and Ye, X. (2020) A matrix factorization based dynamic granularity recommendation with three-way decisions. *Knowl. Based Syst.*, 191, 105243.

[2] Yi, B., Shen, X., Liu, H., Zhang, Z., Zhang, W., Liu, S. and Xiong, N. (2019) Deep matrix factorization with implicit feedback embedding for recommendation system. *IEEE Trans. Industr. Inform.*, 15, 4591–4601.

[3] Zhu, X.*et al.* (2021) Joint prediction and time estimation of COVID-19 developing severe symptoms using chest CT scan. *Med. Image Anal.*, 67, 101824.

[4] Zhang, L., Zhang, L., Du, B., You, J. and Tao, D. (2019) Hyperspectral image unsupervised classification by robust manifold matrix factorization. *Inform. Sci.*, 485, 154–169.

[5] Guo, Y., Wu, Z. and Shen, D. (2019) Learning longitudinal classification-regression model for infant hippocampus segmentation. *Neurocomputing*. 10.1016/j.neucom.2019.01.108.

[6] Jin, D., Yu, Z., Jiao, P., Pan, S., Yu, P.S. and Zhang, W. (2021) A survey of community detection approaches: From statistical modeling to deep learning. *arXiv*, preprint arXiv:2101.01669.

[7] Shen, H.T., Zhu, X., Zhang, Z., Wang, S.-H., Chen, Y., Xu, X. and Shao, J. (2021) Heterogeneous data fusion for predicting mild cognitive impairment conversion. *Inf. Fusion*, 66, 54–63.

[8] Zhu, X., Shichao, Z., Yonggang, L., Jilian, Z., Lifeng, Y. and Yue, F. (2018) Low-rank sparse subspace for spectral clustering. *IEEE Trans. Knowl. Data Eng.*, 1–1.

[9] Wagner, F., Dvorak, G., Nemec, S., Pietschmann, P., Figl, M. and Seemann, R. (2017) A principal components analysis: how pneumatization and edentulism contribute to maxillary atrophy. *Oral Dis.*, 23, 55–61.

[10] Zhu, X., Li, X., Zhang, S., Xu, Z., Yu, L. and Wang, C. (2017) Graph PCA hashing for similarity search. *IEEE Trans Multimedia*, 19, 2033–2044.

[11] Gao, F., Liu, X., Dong, J., Zhong, G. and Jian, M. (2017) Change detection in SAR images based on deep semi-NMF and SVD networks. *Remote Sens. (Basel)*, 9, 435.

[12] Zheng, W., Zhu, X., Wen, G., Zhu, Y., Yu, H. and Gan, J. (2020) Unsupervised feature selection by self-paced learning regularization. *Pattern Recognit. Lett.*, 132, 4–11.

[13] Zhao, L., Zhuang, G. and Xu, X. (2008) Facial expression recognition based on PCA and NMF. In *2008 7th World Congress on Intelligent Control and Automation*, 6826–6829.

[14] M'sik, B. and Casablanca, B.M. (2020) Topic modeling coherence: A comparative study between LDA and NMF models using COVID'19 corpus. 9.

[15] Lee, D. and Seung, H.S. (2000) Algorithms for non-negative matrix factorization. *Adv. Neural Inf. Process. Syst.*, 13, 556–562.

[16] Zhang, Z., Lin, H., Zhao, X., Ji, R. and Gao, Y. (2018) Inductive multi-hypergraph learning and its application on view-based 3D object classification. *IEEE Trans. Image Process.*, 27, 5957–5968.

[17] Ren, B., Pueyo, L., Zhu, G.B., Debes, J. and Duchêne, G. (2018) Non-negative matrix factorization: robust extraction of extended structures. *Astrophys. J.*, 852, 104.

[18] Zhu, X., Zhang, S., Hu, R., He, W., Lei, C. and Zhu, P. (2018) One-step multi-view spectral clustering. *IEEE Trans. Knowl. Data Eng.* 10.1109/TKDE.2018.2873378.

[19] Kong, D., Ding, C. and Huang, H. (2011) Robust nonnegative matrix factorization using l21-norm. In *Proc. 20th ACM Int. Conf. Information and Knowledge Management*, 673–682.

[20] Yang, Z., Liang, N., Yan, W., Li, Z. and Xie, S. (2020) Uniform distribution non-negative matrix factorization for multiview clustering. *IEEE Trans. Cybern.*

[21] Hedjam, R., Abdesselam, A. and Melgani, F. (2021) NMF with feature relationship preservation penalty term for clustering problems. *Pattern Recognit.*, 112, 107814.

[22] Hoyer, P.O. (2004) Non-negative matrix factorization with sparseness constraints. *J. Mach. Learn. Res.*, 5, 1457–1469.

[23] Yuan, Z. and Oja, E. (2005) Projective nonnegative matrix factorization for image compression and feature extraction. In *Scandinavian Conference on Image Analysis*, 333–342.

[24] Ding, C., Li, T., Peng, W. and Park, H. (2006) Orthogonal nonnegative matrix t-factorizations for clustering. *KDD*, 126–135.

[25] Ding, C.H., Li, T. and Jordan, M.I. (2008) Convex and semi-nonnegative matrix factorizations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32, 45–55.

[26] Zhu, X., Yang, J., Zhang, C. and Zhang, S. (2019) Efficient utilization of missing data in cost-sensitive learning. *IEEE Trans. Knowl. Data Eng.* 10.1109/TKDE.2019.2956530.

[27] Kwon, K., Shin, J.W. and Kim, N.S. (2014) NMF-based speech enhancement using bases update. *IEEE Signal Process. Lett.*, 22, 450–454.

[28] Le Roux, J., Hershey, J.R. and Weninger, F. (2015) Deep NMF for speech separation. In *2015 IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 66–70.

[29] Cichocki, A. and Zdunek, R. (2007) Regularized alternating least squares algorithms for non-negative matrix/tensor factorization. In *Int. Symposium on Neural Networks*, 793–802.

[30] Chu, D., Shi, W., Eswar, S. and Park, H. (2020) An alternating rank-k nonnegative least squares framework (ARkNLS) for non-negative matrix factorization. *arXiv*, preprint arXiv:2007.06118.

[31] Lin, X. and Boutros, P.C. (2020) Optimization and expansion of non-negative matrix factorization. *BMC Bioinformatics*, 21, 1–10.

[32] Gu, R., Du, Q. and Billinge, S.J. (2021) A fast two-stage algorithm for non-negative matrix factorization in streaming data. arXiv preprint arXiv:2101.08431.

[33] Li, X., Zhang, W. and Dong, X. (2017) A class of modified FR conjugate gradient method and applications to non-negative matrix factorization. *Comput. Math. Appl.*, 73, 270–276.

[34] Mei, J., De Castro, Y., Goude, Y., Azaïs, J.-M. and Hébrail, G. (2018) Nonnegative matrix factorization with side information for time series recovery and prediction. *IEEE Trans. Knowl. Data Eng.*, 31, 493–506.

[35] Li, X., Cui, G. and Dong, Y. (2018) Discriminative and orthogonal subspace constraints-based nonnegative matrix factorization. *ACM Trans. Intell. Syst. Technol.*, 9, 1–24.

[36] Dai, X., Su, X., Zhang, W., Xue, F. and Li, H. (2020) Robust Manhattan non-negative matrix factorization for image recovery and representation. *Inform. Sci.*, 527, 70–87.

[37] Meng, Y., Shang, R., Jiao, L., Zhang, W., Yuan, Y. and Yang, S. (2018) Feature selection based dual-graph sparse non-negative matrix factorization for local discriminative clustering. *Neurocomputing*, 290, 87–99.

[38] Braytee, A., Liu, W. and Kennedy, P.J. (2017) Supervised context-aware non-negative matrix factorization to handle high-dimensional high-correlated imbalanced biomedical data. In *2017 Int. Joint Conf. Neural Networks (IJCNN)*, 4512–4519.

[39] Bobadilla, J., Bojorque, R., Esteban, A.H. and Hurtado, R. (2017) Recommender systems clustering using Bayesian non negative matrix factorization. *IEEE Access*, 6, 3549–3564.

[40] Gan, J., Peng, Z., Zhu, X., Hu, R., Ma, J. and Wu, G. (2021) Brain functional connectivity analysis based on multi-graph fusion. *Med. Image Anal.* 10.1016/j.media.2021.102057.

[41] Zhu, X., Li, X., Zhang, S., Ju, C. and Wu, X. (2017) Robust joint graph sparse coding for unsupervised spectral feature selection. *IEEE Trans. Neural Netw. Learn. Syst.*, 28, 1263–1275.

[42] Flenner, J. and Hunter, B. (2017) A deep non-negative matrix factorization neural network. *Semantic Scholar*.

[43] Nie, S., Liang, S., Liu, W., Zhang, X. and Tao, J. (2018) Deep learning based speech separation via NMF-style reconstructions. *IEEE/ACM Trans. Audio Speech Lang. Process.*, 26, 2043–2055.

[44] Shi, X., Guo, Z., Nie, F., Yang, L., You, J. and Tao, D. (2015) Two-dimensional whitening reconstruction for enhancing robustness of principal component analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38, 2130–2136.

[45] Chen, H., Gao, M., Zhang, Y., Liang, W. and Zou, X. (2019) Attention-based multi-NMF deep neural network with multimodality data for breast cancer prognosis model. *Biomed. Res. Int.*, 2019.

[46] Wisdom, S., Powers, T., Pitton, J. and Atlas, L. (2017) Deep recurrent NMF for speech separation by unfolding iterative thresholding. In *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 254–258.

[47] Shi, X., Sapkota, M., Xing, F., Liu, F., Cui, L. and Yang, L. (2018) Pairwise based deep ranking hashing for histopathology image classification and retrieval. *Pattern Recognit.*, 81, 14–22.

[48] Pompili, F., Gillis, N., Absil, P.-A. and Glineur, F. (2014) Two algorithms for orthogonal nonnegative matrix factorization with application to clustering. *Neurocomputing*, 141, 15–25.

[49] Lin, C.-J. (2007) On the convergence of multiplicative update algorithms for nonnegative matrix factorization. *IEEE Trans. Neural Netw.*, 18, 1589–1596.

[50] Zhu, X., Li, X. and Zhang, S. Block-row sparse multiview multilabel learning for image classification. *IEEE Trans. Cybern.*, 46, 450–461.

[51] Cuitino, A. and Ortiz, M. (1992) A material-independent method for extending stress update algorithms from small-strain plasticity to finite plasticity with multiplicative kinematics. *Eng. Comput.*, 9, 437–437.

[52] Eggert, J. and Korner, E. (2004) Sparse coding and NMF. In *IEEE Cat. No. 04CH37541*, 2529–2533.

[53] Ge, Z., Demyanov, S., Chakravorty, R., Bowling, A. and Garnavi, R. (2017) Skin disease recognition using deep saliency features and multimodal learning of dermoscopy and clinical images. In *MICCAI*, 250–258.

[54] Gillis, N. and Glineur, F. (2008) Nonnegative factorization and the maximum edge biclique problem. *arXiv*, preprint arXiv:0810.4225.

[55] Leuschner, J., Schmidt, M., Fernsel, P., Lachmund, D., Boskamp, T. and Maass, P. (2019) Supervised non-negative matrix factorization methods for MALDI imaging applications. *Bioinformatics*, 35, 1940–1947.

[56] Shi, X., Xing, F., Zhang, Z., Sapkota, M., Guo, Z. and Yang, L. (2020) A scalable optimization mechanism for pairwise based discrete hashing. *IEEE Trans. Image Process.*, 30, 1130–1142.

[57] Kim, H. and 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.

[58] Ang, A.M.S. and Gillis, N. (2019) Accelerating nonnegative matrix factorization algorithms using extrapolation. *Neural Comput.*, 31, 417–439.

[59] Hugelier, S., Piqueras, S., Bedia, C., De Juan, A. and Ruckebusch, C. (2018) Application of a sparseness constraint in multivariate curve resolution–alternating least squares. *Anal. Chim. Acta*, 1000, 100–108.

[60] Rajabi, R. and Ghassemian, H. (2014) Spectral unmixing of hyperspectral imagery using multilayer NMF. *IEEE Geosci. Remote Sens. Lett.*, 12, 38–42.

[61] Trigeorgis, G., Bousmalis, K., Zafeiriou, S. and Schuller, B. (2014) A deep semi-NMF model for learning hidden representations. In *ICML*, 1692–1700.

[62] Takács, G. and Tikk, D. (2012) Alternating least squares for personalized ranking. In *Proc. Sixth ACM Conf. Recommender Systems*, 83–90.

[63] Shi, X., Guo, Z., Xing, F., Liang, Y. and Yang, L. (2020) Anchor-based self-ensembling for semi-supervised deep pairwise hashing. *Int. J. Comput. Vis.*, 1–18.

[64] Kimura, K., Tanaka, Y. and Kudo, M. (2015) A fast hierarchical alternating least squares algorithm for orthogonal nonnegative matrix factorization. In *ACML*, 129–141.

[65] Lee, S. and Pang, H.-S. (2015) Multichannel non-negative matrix factorisation based on alternating least squares for audio source separation system. *Electron. Lett.*, 51, 197–198.

[66] Zhu, X., Zhu, Y. and Zheng, W. (2019) Spectral rotation for deep one-step clustering. *Pattern Recognit.* 10.1016/j.patcog.2019.107175.

[67] Shi, X., Xing, F., Xu, K., Chen, P., Liang, Y., Lu, Z. and Guo, Z. (2020) Loss-based attention for interpreting image-level prediction of convolutional neural networks. *IEEE Trans. Image Process.*

[68] Phan, A.H. and Cichocki, A. (2008) Multi-way nonnegative tensor factorization using fast hierarchical alternating least squares algorithm (HALS). In *Proc. 2008 Int. Symposium on Nonlinear Theory and its Applications*.

[69] Kim, D., Sra, S. and Dhillon, I.S. (2007) Fast newton-type methods for the least squares nonnegative matrix approximation problem. In *SDM*.

[70] Kuang, D., Ding, C. and Park, H. (2012) Symmetric nonnegative matrix factorization for graph clustering. In *SIAM*, 106–117.

[71] Teboulle, M. and Vaisbourd, Y. (2020) Novel proximal gradient methods for nonnegative matrix factorization with sparsity constraints. *SIAM J. Imag. Sci.*, 13, 381–421.

[72] Lin, C.-J. (2007) Projected gradient methods for nonnegative matrix factorization. *Neural Comput.*, 19, 2756–2779.

[73] Kim, J. and Park, H. (2011) Fast nonnegative matrix factorization: an active-set-like method and comparisons. *Siam J. Sci. Comput.*, 33, 3261–3281.

[74] Zhang, S., Huang, D., Lei, X., Chng, E.S. and Dong, M. (2016) Non-negative matrix factorization using stable alternating direction method of multipliers for source separation. In *Signal Information Processing Association Summit Conference*.

[75] Li, T. and Ding, C.H. (2013) Nonnegative matrix factorizations for clustering: a survey.

[76] Huang, Z., Zhou, A. and Zhang, G. (2012) Non-negative matrix factorization: a short survey on methods and applications. In *Int. Symposium on Intelligence Computation and Applications*, 331–340.

[77] Guan, N., Tao, D., Luo, Z. and Yuan, B. (2012) NeNMF: an optimal gradient method for nonnegative matrix factorization. *IEEE Trans. Signal Process.*, 60, 2882–2898.

[78] Nesterov, B.Y. (2007) A method of solving a convex programming problem with convergence rate.

[79] Yoshii, K., Tomioka, R., Mochihashi, D. and Goto, M. (2013) Beyond NMF: time-domain audio source separation without phase reconstruction. In *ISMIR*, 369–374.

[80] Tropp, J.A. (2003) Literature survey: nonnegative matrix factorization. University of Texas at Asutin. 26.

[81] Devarajan, K. (2021) A statistical framework for non-negative matrix factorization based on generalized dual divergence. *Neural Netw.*

[82] Shu, Z.-Q., Wu, X.-J., Hu, C., You, C.-Z. and Fan, H.-H. (2021) Deep semi-nonnegative matrix factorization with elastic preserving for data representation. *Multimed. Tools Appl.*, 80, 1707–1724.

[83] Wang, J. and Mu, R. (2021) A regularized convex nonnegative matrix factorization model for signed network analysis. *Soc. Netw. Anal. Min.*, 11, 1–12.

[84] Cui, G., Li, X. and Dong, Y. (2018) Subspace clustering guided convex nonnegative matrix factorization. *Neurocomputing*, 292, 38–48.

[85] Ma, H., Zhao, W., Tan, Q. and Shi, Z. (2010) Orthogonal nonnegative matrix tri-factorization for semi-supervised document co-clustering. In *Pacific-Asia Conf. Knowledge Discovery and Data Mining*, 189–200.

[86] Charikar, M. and Hu, L. (2021) Approximation algorithms for orthogonal non-negative matrix factorization. In *Int. Conf. Artificial Intelligence and Statistics*, 2728–2736.

[87] Peng, S., Ser, W., Chen, B. and Lin, Z. (2020) Robust orthogonal nonnegative matrix tri-factorization for data representation. *Knowl. Based Syst.*, 201, 106054.

[88] Xu, S., Liu, S. and Feng, L. (2021) Deep graph convolution neural network with non-negative matrix factorization for community discovery. *arXiv*, preprint arXiv:2103.05768.

[89] Jia, Y., Liu, H., Hou, J., Kwong, S. and Zhang, Q. (2021) Self-supervised symmetric nonnegative matrix factorization. *arXiv*, preprint arXiv:2103.01689.

[90] Luo, X., Liu, Z., Jin, L., Zhou, Y. and Zhou, M. (2021) Symmetric nonnegative matrix factorization-based community detection models and their convergence analysis. *IEEE Trans. Neural Netw. Learn. Syst.*

[91] He, Z., Xie, S., Zdunek, R., Zhou, G. and Cichocki, A. (2011) Symmetric nonnegative matrix factorization: algorithms and applications to probabilistic clustering. *IEEE Trans. Neural Netw.*, 22, 2117–2131.

[92] Jia, Y., Liu, H., Hou, J. and Kwong, S. (2020) Semisupervised adaptive symmetric non-negative matrix factorization. *IEEE Trans. Cybern.*

[93] Yan, W., Zhang, B., Yang, Z. and Xie, S. (2019) Similarity learning-induced symmetric nonnegative matrix factorization for image clustering. *IEEE Access*, 7, 166380–166389.

[94] Lu, H., Sang, X., Zhao, Q. and Lu, J. (2020) Community detection algorithm based on nonnegative matrix factorization and pairwise constraints. *Physica A Stat. Mech. Appl.*, 545, 123491.

[95] Zhu, X., Gan, J., Lu, G., Li, J. and Zhang, S. (2020) Spectral clustering via half-quadratic optimization. *World Wide Web*, 23, 1969–1988.

[96] Wang, G. and Yu, B. (2019) A new kernel method for nonnegative matrix factorization. In *ICAICA*, 454–458.

[97] He, C., Zhang, Q., Tang, Y., Liu, S. and Liu, H. (2019) Network embedding using semi-supervised kernel nonnegative matrix factorization. *IEEE Access*, 7, 92732–92744.

[98] Zhang, D., Zhou, Z.-H. and Chen, S. (2006) Non-negative matrix factorization on kernels. In *Pacific Rim Int. Conf. Artificial Intelligence*, 404–412.