

2018 INFORMS International Conference
Taipei, Taiwan, June 17–20, 2018
© 2018 INFORMS|ISBN 978-0-9906153-1-6
https://doi.org/10.1287/infp.2018.XXXX
pp. 000–000

Continuously Updating Nonnegative Matrix Factorization

Nikolai D. Lipscomb
Statistics and Operations Research, UNC-Chapel Hill, nikolail@live.unc.edu

Austin Z. Gratton
Mathematics and Statistics, UNC-Wilmignton azg5169@uncw.edu

Yaw Chang
Mathematics and Statistics, UNC-Wilmigton, changy@uncw.edu

Cuixian Chen
Mathematics and Statistics, UNC-Wilmignton chenc@uncw.edu

Yishi Wang
Mathematics and Statistics, UNC-Wilmignton wangy@uncw.edu

Abstract Nonnegative matrix factorization (NMF) is a problem that decomposes a real matrix with nonnegative elements into a product of two real, nonnegative matrices. This decomposition is often used as a dimension reduction tool by choosing a small value for the inner dimension of the product. NMF has been widely applied in information retrieval, computer vision and pattern recognition. While various methods has been proposed to improve the quality of the factorization, no research to address how to update the factorization when new data stream coming in. We proposed a novel approach called Continuously updating nonnegative matrix factorization (CUNMF) to address the updating issue. The proposed method can also be applied to large data set to improve the quality of final factorization.

Keywords Non-negative Matrix Factorization, Continuouslu updating Non-negative Matrix Factorization, Clustering

1. Introduction

Nonnegative matrix factorization (NNF) is one of the popular tools to incorporate the non-negativity constraints to decompose high dimensional data into low-rank structure, which yields a parts-based representation. NMF is a numerical optimization problem that gained popularity around the beginning of the 21st century. Statisticians and artificial intelligence (AI) researchers are often interested in matrix factorizations that reveal latent traits in data.

NMF was first introduced by [14] under the name of Positive Matrix Factorization. In 1999, Lee and Seung [10] published a paper in *Letters to Nature* that drew attention to the features of NMF, due to their simple yet effective algorithm and the intuitive illustration of parts-based representation example. As nonnegative data arise in many fields such as image analysis [7], bioinformatics [2], and text mining [1], the forcing of a nonnegative factorization gives interesting results. On the other hand, with additive parts-based underlying structures such as a face can be represented conceptually by different parts: eyes, nose, mouth and cheeks, NMF has been demonstrated to be more likely to capture such structure effectively [10, 12, 8].

Various methods has been proposed to improve the quality of the factorization [18, 9, 6, 13]. However, there is no research to address how to update the factorization when new data stream coming in. Thus a novel approach called Continuously updating nonnegative matrix factorization (CUNMF) is proposed in this paper to address the updating issue.

This paper is organized as follows. The development of nonnegative matrix factorization in a mathematical context is introduced in Section 2. In Section 3, a novel algorithm Continuously Updating Nonnegative Matrix Factorization is proposed. Experiments on document clustering and image clustering are illustrated in Section 4. Conclusion and future work are given in Section 5.

2. Nonnegative Matrix Factorization

Definition 1. Let $\boldsymbol{V}$ be an $m \times n$ matrix with real, nonnegative entries. The nonnegative matrix factorization of rank-k is the matrix product $\boldsymbol{WH}$, where $\boldsymbol{W}$ is an $m \times k$ matrix with real, nonnegative entries, $\boldsymbol{H}$ is a $k \times n$ matrix with real, nonnegative entries, and $\boldsymbol{V} \approxeq \boldsymbol{WH}$.

We often choose a value of k such that $k \ll \min(m, n)$ in order to produce dimension reduction/compression results. Deriving a NMF is done by numerically solving a matrix optimization problem in two basic forms:

$$\min_{\boldsymbol{W},\boldsymbol{H}} \left\|\boldsymbol{V} - \boldsymbol{WH}\right\|_F^2, \text{ or } \min_{\boldsymbol{W},\boldsymbol{H}} D\big(\boldsymbol{V}||\boldsymbol{WH}\big), \tag{1}$$

subject to the constraint $\boldsymbol{W}_{ij}, \boldsymbol{H}_{ij} \geq 0, \ \forall \ i, j$. Notice that the squared Frobenius norm, $\|\cdot\|_F^2$, is defined as

$$\left\|\boldsymbol{A} - \boldsymbol{B}\right\|_F^2 = \sum_{ij} \left(\boldsymbol{A}_{ij} - \boldsymbol{B}_{ij}\right)^2, \tag{2}$$

and the Kullback-Leibler divergence, $D(\cdot||\cdot)$, is defined as

$$D\big(\boldsymbol{A}||\boldsymbol{B}\big) = \sum_{ij} \left(\boldsymbol{A}_{ij} \log \frac{\boldsymbol{A}_{ij}}{\boldsymbol{B}_{ij}} - \boldsymbol{A}_{ij} + \boldsymbol{B}_{ij}\right). \tag{3}$$

Several other constraints and altered penalty functions have been developed throughout the years in order to produce different types of factorizations [16]; popular ones include sparse [8], orthogonal[4], and semi-NMF [5]. The difficulty with solving the NMF problem lies in the nonconvexity of the problem. This makes the attainment of a global minimum difficult as many efficient nonlinear optimization algorithms converge to a nearby local minimum at best. However, in many cases, a local minimum provides good factorization results for many purposes, such as dimension reduction prior to performing classification.

Lee and Seung [10, 11] proposed a set of multiplicative update rules for minimizing either of the cost functions presented in Problem (1). Over the course of this paper, we will refer to these update rules as the *Basic NMF* algorithm. These rules are nonincreasing with respect to their cost functions and only fail to improve if and only if $\boldsymbol{W}$ and $\boldsymbol{H}$ are at stationary points. These algorithms make use of the Hadamard product; the $\odot$ symbol will represent element-wise multiplication between two matrices of equal dimensions whereas the $\oslash$ symbol will represent element-wise division.

Algorithm 1 Basic NMF (Frobenius Norm) [10, 11]

procedure : FIND MATRICIES $\boldsymbol{W}, \boldsymbol{H}$ GIVEN MATRIX $\boldsymbol{V}$, FACTORIZATION RANK k, RELATIVE ERROR TOLERANCE TOL, AND MAXIMUM ITERATIONS M, SUCH THAT $\boldsymbol{V} \approx \boldsymbol{W}\boldsymbol{H}$

set initial guesses for $\boldsymbol{W}$ and $\boldsymbol{H}$
$\epsilon = \frac{\|\boldsymbol{V}-\boldsymbol{W}\boldsymbol{H}\|_F}{\|\boldsymbol{V}\|_F}$ ▷ Frobenius norm-based relative error
$\mu = 0$ ▷ μ is the iteration counter
while $\epsilon > TOL$ and $\mu < M$ **do**
    $\boldsymbol{H} = \boldsymbol{H} \odot \left(\boldsymbol{W}^T\boldsymbol{V}\right) \oslash \left(\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{H}\right)$
    $\boldsymbol{W} = \boldsymbol{W} \odot \left(\boldsymbol{V}\boldsymbol{H}^T\right) \oslash \left(\boldsymbol{W}\boldsymbol{H}\boldsymbol{H}^T\right)$
    $\epsilon = \frac{\|\boldsymbol{V}-\boldsymbol{W}\boldsymbol{H}\|_F}{\|\boldsymbol{V}\|_F}$
    $\mu = \mu + 1$
end while
return $\boldsymbol{W}, \boldsymbol{H}$
end procedure

3. Continuously Updating Nonnegative Matrix Factorization

Some factors need to be considered before introducing the continuously updating nonnegative matrix factorization (CUNMF) algorithm. First, NMF is performed on data matrices to extract latent features and/or compress the data. However, a comprehensive data set is seldom available for immediate use; in fact, in problems such as facial recognition and email classification, an ongoing feed of new data is to be expected. Since a factorization is performed on a static matrix, what approach can we use to incorporate new data?

Under the assumption that a data matrix $\boldsymbol{V}$ has different observations along its columns and different features along its rows, it is noted that the two factors of a nonnegative matrix factorization for $\boldsymbol{V}$ are the basis matrix $\boldsymbol{W}$ and a weight matrix $\boldsymbol{H}$. $\boldsymbol{W}$ consists of k column vectors whose span via positive linear combinations of column elements from $\boldsymbol{H}$ approximately encapsulates the span of the columns of $\boldsymbol{V}$. As $\boldsymbol{W}$ is often composed of useful features with respect to the data, $\boldsymbol{H}$ contains the information about what features contribute the most and least to approximate the associated observation from $\boldsymbol{V}$.

Assume we receive a new observation $\boldsymbol{v}_\nu$. This new observation, if appended to $\boldsymbol{V}$, requires its corresponding weighting column $\boldsymbol{h}_\nu$ to be appended to $\boldsymbol{H}$. One could run the factorization from scratch; however, if we wish to preserve the existing basis in $\boldsymbol{W}$, we set the elements of $\boldsymbol{h}_\nu$ as the projection of $\boldsymbol{v}_\nu$ into the column space of $\boldsymbol{W}$: $\boldsymbol{h}_\nu = (\boldsymbol{W}^T\boldsymbol{W})^{-1}(\boldsymbol{W}^T\boldsymbol{v}_\nu)$. This leads to one of two outcomes:

(1) $\boldsymbol{W}\boldsymbol{h}_\nu = \boldsymbol{v}_\nu$, if $\boldsymbol{v}_\nu$ already lies in $W_s = \{w | w = \sum_{i=1}^{k} \alpha_i \boldsymbol{w}_i, \alpha_i \geq 0\}$, i.e. the cone generated by the columns of $(\boldsymbol{W})$; or
(2) $\boldsymbol{v}_\nu$ lies outside of W_s. In this case, at least one of the components in $\boldsymbol{h}_\nu$ is less than 0. In this case, we replace these components by 0.

As a result, we notice that $\boldsymbol{h}_\nu$ will be a nonnegative vector.

We will call the new matrix $\boldsymbol{H}_\nu = [\boldsymbol{H} \quad \boldsymbol{h}_\nu]$. Similarly, let $\boldsymbol{V}_\nu = [\boldsymbol{V} \quad \boldsymbol{v}_\nu]$. Clearly, given that we wish to preserve $\boldsymbol{W}$, $\boldsymbol{V}_\nu$ is well approximated by $\boldsymbol{W}\boldsymbol{H}_\nu$. However, there lies the consideration that perhaps a better basis matrix, $\boldsymbol{W}_\nu$ exists, allowing us to get an even better matrix approximation given the new information from $\boldsymbol{v}_\nu$. This brings into consideration the initialization problem for NMF algorithms: our proposed update rule requires us to run a low-iteration NMF update algorithm for $\boldsymbol{V}_\nu$ given the existing matrices of $\boldsymbol{W}$ and $\boldsymbol{H}_\nu$ as the initialization.

Algorithm 2 New Information NMF Update Rule

procedure : FIND MATRICIES $\boldsymbol{W}_\nu, \boldsymbol{H}_\nu$ GIVEN MATRIX $\boldsymbol{V}$, NEW INFORMATION VECTOR $\boldsymbol{v}_\nu$, INITIALIZATION MATRICES $\boldsymbol{W}$ AND $\boldsymbol{H}$, NMF UPDATE RULE U, SUCH THAT $\boldsymbol{V}_\nu \approx \boldsymbol{W}_\nu \boldsymbol{H}_\nu$
  $\boldsymbol{V}_\nu = [\boldsymbol{V} \quad \boldsymbol{v}_\nu]$
  $\boldsymbol{h}_\nu = (\boldsymbol{W}^T \boldsymbol{W})^{-1} (\boldsymbol{W}^T \boldsymbol{v}_\nu)$
  $k =$ the number of columns of $\boldsymbol{W}$ ▷ factorization rank
  for $i = 1$ to k **do**
    if $h_{\nu_i} < 0, h_{\nu_i} = 0$
  end for
  $\boldsymbol{W}_\nu = \boldsymbol{W}$
  $\boldsymbol{H}_\nu = [\boldsymbol{H} \quad \boldsymbol{h}_\nu]$
  $(\boldsymbol{W}_\nu, \boldsymbol{H}_\nu) = U(\boldsymbol{V}_\nu, k, (\boldsymbol{W}_\nu, \boldsymbol{H}_\nu))$
  return $\boldsymbol{W}_\nu, \boldsymbol{H}_\nu$
end procedure

Next, we will consider the following definition needed for our proposed rule. Note that this algorithm will be dependent on a general NMF update algorithm, U. We expect U to be dependent on a maximum number of iterations and an error tolerance as stopping criteria. We generally expect the number of iterations required to calculate the new matrices to be low. Since $\boldsymbol{h}_\nu$ is clearly the best approximation with the existing basis matrix, the main purpose of Algorithm (2) is to develop a slight improvement for the basis matrix. Depending on the quality of the information provided by $\boldsymbol{v}_\nu$, the new basis matrix, $\boldsymbol{W}_\nu$ may contain higher quality features. Further, we have only considered how to add a single vector of new information at a time. If multiple information vectors are provided, they do not have to be projected, appended, and updated one at a time; instead, new information vectors can be added in batches. It is worth considering how varying the batch-size can affect the convergence of the update rule.

Definition 2. Let $\mathbb{R}^*_{m\times n}$ be the set of nonnegative, real, $m \times n$ matrices and $\mathbb{Z}^+$ be the set of positive integers. $U : \mathbb{R}^*_{m\times n} \times \mathbb{Z}^+ \times (\mathbb{R}^*_{m\times k} \times \mathbb{R}^*_{k\times n}) \to \mathbb{R}^*_{m\times k} \times \mathbb{R}^*_{k\times n}$ is called an NMF update algorithm if $U(\boldsymbol{V}, k, (\boldsymbol{A}, \boldsymbol{B})) = (\boldsymbol{W}, \boldsymbol{H})$ where $\boldsymbol{V}_{m\times n} \approx \boldsymbol{W}_{m\times k} \boldsymbol{H}_{k\times n}$ is a nonnegative matrix factorization. $\boldsymbol{A}_{m\times k}$ and $\boldsymbol{B}_{k\times n}$ are called the initialization matrices for U and $k \in \mathbb{Z}^+$ is the rank of the factorization.

We will now consider our *New Information NMF Update Rule* in Algorithm (2), which also raises the question as to whether this type of approach is useful for developing a nonnegative matrix factorization from the start. We will first present a new algorithm, *Continuously Updating Nonnegative Matrix Factorization* (CUNMF), then examine its components.

Algorithm 3 Continuously Updating Nonnegative Matrix Factorization

procedure : FIND MATRICIES $\boldsymbol{W}, \boldsymbol{H}$ GIVEN MATRIX $\boldsymbol{V}$, NMF UPDATE RULE U, FACTORIZATION RANK k, INITIALIZATION SIZE S_I, INITIAL NUMBER OF TESTS t, BATCH SIZE S_u, SUCH THAT $\boldsymbol{V} \approx \boldsymbol{W}\boldsymbol{H}$
  $m =$ the number of rows of $\boldsymbol{V}$, $n =$ the number of columns of $\boldsymbol{V}$
  Let $\boldsymbol{V}_I$ be the submatrix of $\boldsymbol{V}$ containing the first S_I columns.
  Choose initialization for $\boldsymbol{W}_{m\times k}$ and $\boldsymbol{H}_{k\times S_I}$
  Solve $\boldsymbol{V} \approx \boldsymbol{W}\boldsymbol{H}$ by Algorithm 1.
  Create matrix $\boldsymbol{V}_R$ consisting of the remaining columns of $\boldsymbol{V}$ not in $\boldsymbol{V}_I$
  while $\boldsymbol{V}_R$ is nonempty **do**
    Select up to the first S_u columns of $\boldsymbol{V}_R$: $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{S_u}\}$
    Remove $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{S_u}\}$ from $\boldsymbol{V}_R$
    Let $\boldsymbol{V} = [\boldsymbol{V}_I \;\; \boldsymbol{v}_1 \cdots \boldsymbol{v}_{S_u}]$.
    $\boldsymbol{H} = [\boldsymbol{H} \;\; \hat{\boldsymbol{h}}_1 \cdots \hat{\boldsymbol{h}}_{S_u}]$ where $\hat{\boldsymbol{h}}_i$ is constructed as described in Algorithm 2.
    $(\boldsymbol{W}, \boldsymbol{H}) = U(\boldsymbol{V}, k, (\boldsymbol{W}, \boldsymbol{H}))$
  end while
  return $\boldsymbol{W}, \boldsymbol{H}$
end procedure

4. Numerical Experiments

For comparison, we test our proposed novel approach of CUNMF for document clustering and image clustering following [3]. For document clustering, it is assumed that a document corpus is comprised of k clusters each of which corresponding to a coherent topic. Xu *et al.* [17] suggested that it is ideal to project each document into a $k-$dimentional semantic space in which each axis corresponding to a particular topics. Because it is more natural to consider each document as an additive rather than subtractive mixture of underlying topics, the combination coefficients should all take non-negative values.

For appearance-based visual analysis, an image may be associated with hidden parts. For example,a face image can be thought as a combination of eyes, eye-brows, nose, mouth, etc.. It is also reasonable to require the combination coefficients are non-negative. This gives the main motivation to apply NMF on clustering.

4.1. Data Sets

Three data sets are used in the numerical experiments. Two of them are image data sets and the other is a document corpus. Here are the summarized statistics of these data sets (see also Table 1):

- COIL20 image library contains 1440 32×32 gray scale images of 20 subjects viewed from different angles. Each image is represented by a vector of length of 1024.
- CMU PIE face database contains 32×32 gray scale face image of 68 persons. Each person has 42 images under different light and illumination conditions. Again, each image is represented by a vector of length 1024.
- The NIST Topics Detection and Tracking (aka. TDT2) corpus consists of 11201 on-topic documents which are classified into 96 semantic categories. In this experiment, those documents appearing in two or more categories were removed and only the largest 30 categories were kept, thus leaving us with 9,394 documents in total.

Table 1. Summary statistics of data sets

Dataset	Dimension (m)	size (n)	# of class (k)
COIL20	1024	1440	20
PIE	1024	2856	68
TDT2	36771	9394	30

4.2. Comparison of Algorithms

To demonstrate how the clustering performance and quality of the factorization of proposed methos, we choose the following three algorithms:

- Nonnegative Matrix Factorization: We use the F-norm formulation and implement a normalized cut weight version of NMF as suggested in [17].
- Graph regularized Nonnegative Matrix Factorization (GNMF): Proposed in [3]. The algorithm incorporates a geometrically based regularizer into NMF with F-norm formulation.
- Continuously updating Nonnegative Matrix Factorization: We use the F-norm formulation with the underline updating NMF same as the first one.

Two metrics are used for evaluation: The normalized mutual information (NMI) for measuring the clustering performance and relative error (Rel-Err) for quality of the factorization result.

Let L be the label of clusters obtained from the ground truth and L' be the label obtained from clustering algorithm. Their multual information $MI(L, L')$ is defined as:

$$MI(L, L') = \sum_{l_i \in L, l'_j \in L'} p(l_i, l'_j) \cdot \log_2 \frac{p(l_i, l'_j)}{p(l_i) \cdot p(l'_j)},$$

where $p(l_i)$ and $p(l'_j)$ are the probability that a document randomly selected from the corpus belongs to clusters l_i and l'_j, respectively. $p(l_i, l'_j)$ is the joint probability that a randomly selected document belongs to clusters l_i, l'_j at the same time. The normalized mutual information $\overline{MI}$ is defined as:

$$\overline{MI}(L, L') = \frac{MI(L, L')}{\max(H(L), H(L')},$$

where $H(L)$ and $H(L')$ are the entropies of L and L', respectively. $\bar{M}I = 1$ implies that two sets of clusters are identical, and $\bar{M}I = 0$ implies the two sets are independent.

The relative error of $\boldsymbol{V} \approx \boldsymbol{W}\boldsymbol{H}$ is defined as:

$$\text{Rel-Err} = \frac{\|\boldsymbol{V} - \boldsymbol{W}\boldsymbol{H}\|_2}{\|\boldsymbol{V}\|_2} \times 100\%.$$

To simulate a continuous data stream, we set the size initial matrix $\boldsymbol{V}_I$ is about 50% of the original data with different incremental size. Figure 1 shows the performance results of CUNMF on COIL20 data set. (Note: Initial matrix size is 700.) Figure 2 shows the result of CUNMF verse the other two algorithms. It is understandable that GNMF has the best clustering result because it takes the advantage of graph regularizer, but CUNMF has the best relative error and similar clustering result as NMF.

Figure 3 shows the performance results of CUNMF on PIE data set with initial matrix size of 1400. Figure 4 shows the result of CUNMF verse the other two algorithms. The clustering result by CUNMF is much better than NMF in this data set. Again, CUNMF produced the best relative error.

Finally Figure 5 shows the performance results of CUNMF on TDT2 data set with initial matrix size of 4700. Figure 6 shows the result of CUNMF verse the other two algorithms. CUNMF does not show any particular improvement on both evaluation metrics. It may due to the sparsity of the data set.

FIGURE 1. CUNMF performance on COIL20

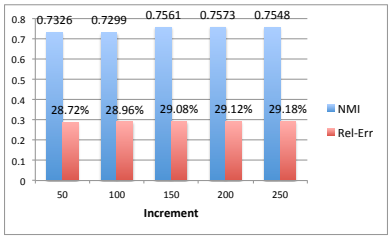


FIGURE 2. Performance comparsion on COIL20

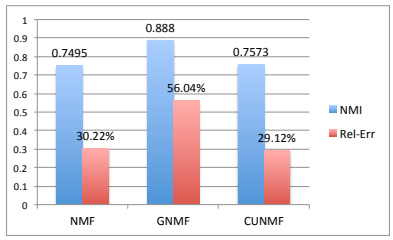


FIGURE 3. CUNMF performance on PIE

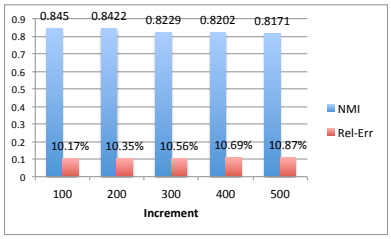


FIGURE 4. Performance comparsion on PIE

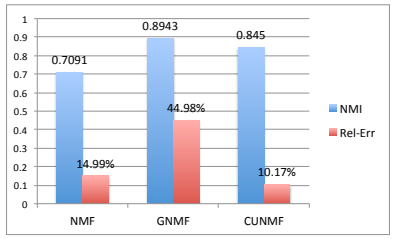


FIGURE 5. CUNMF performance on TDT2

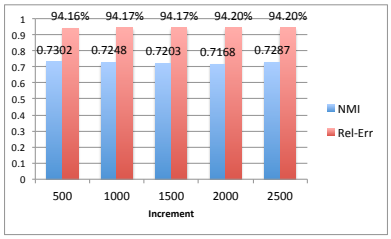
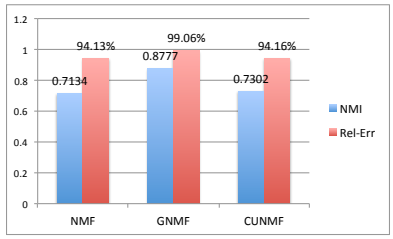


FIGURE 6. Performance comparsion on TDT2



5. Conclusion and Future work

From the numerical results from previous section, CUNMF produces the best result on PIE data set. It may cause by the high number of categories ($k = 68$) or the data set itself (face image). We would like to apply the algorithm to a large face image data set. One particular data set we are interesting to study is the MORPH data set [15]. It is the largest public available longitudinal face data base. A subsets of MORPH Album contains 55,134 images of 13,000 individuals collected over five years. We will use CUNMF as dimension reduction tool to study the gender classification, age estimation, etc. with other machine learning techniques.

References

[1] Berry MW, Browne M, Langville AN, Pauca VP, Plemmons RJ (2007) Algorithms and applications for approximate nonnegative matrix factorization. *Computational statistics & data analysis* 52(1):155–173.

[2] Brunet JP, Tamayo P, Golub TR, Mesirov JP (2004) Metagenes and molecular pattern discovery using matrix factorization. *Proceedings of the national academy of sciences* 101(12):4164–4169.

[3] Cai D, He X, Han J, Huang TS (2011) Graph regularized nonnegative matrix factorization for data representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33(8):1548–1560.

[4] Ding C, Li T, Peng W, Park H (2006) Orthogonal nonnegative matrix t-factorizations for clustering. *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 126–135 (ACM).
[5] Ding CH, Li T, Jordan MI (2010) Convex and semi-nonnegative matrix factorizations. *IEEE transactions on pattern analysis and machine intelligence* 32(1):45–55.
[6] Gillis N, Vavasis SA (2014) Fast and robust recursive algorithmsfor separable nonnegative matrix factorization. *IEEE transactions on pattern analysis and machine intelligence* 36(4):698–714.
[7] Guillamet D, Vitria J (2002) Non-negative matrix factorization for face recognition. *Topics in artificial intelligence*, 336–344 (Springer).
[8] Hoyer PO (2004) Non-negative matrix factorization with sparseness constraints. *Journal of machine learning research* 5(Nov):1457–1469.
[9] Kong D, Ding C, Huang H (2011) Robust nonnegative matrix factorization using l21-norm. *Proceedings of the 20th ACM international conference on Information and knowledge management*, 673–682 (ACM).
[10] Lee DD, Seung HS (1999) Learning the parts of objects by non-negative matrix factorization. *Nature* 401(6755):788.
[11] Lee DD, Seung HS (2001) Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, 556–562.
[12] Li Z, Wu X, Peng H (2010) Nonnegative matrix factorization on orthogonal subspace. *Pattern Recognition Letters* 31(9):905–911.
[13] Mairal J, Bach F, Ponce J, Sapiro G (2010) Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research* 11(Jan):19–60.
[14] Paatero P, Tapper U (1994) Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics* 5(2):111–126.
[15] Ricanek K, Tesafaye T (2006) Morph: A longitudinal image database of normal adult age-progression. *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on*, 341–345 (IEEE).
[16] Wang YX, Zhang YJ (2013) Nonnegative matrix factorization: A comprehensive review. *IEEE Transactions on Knowledge and Data Engineering* 25(6):1336–1353.
[17] Xu W, Liu X, Gong Y (2003) Document clustering based on non-negative matrix factorization. *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, 267–273 (ACM).
[18] Zhang L, Chen Z, Zheng M, He X (2011) Robust non-negative matrix factorization. *Frontiers of Electrical and Electronic Engineering in China* 6(2):192–200.