Orthogonal Nonnegative Matrix Factorization for Blind Image Separation

Andri Mirzal

Faculty of Computing N28-439-03,
Universiti Teknologi Malaysia,
81310 UTM Johor Bahru, Malaysia
Tel.: +60-755-32062
andrimirzal@utm.my

Abstract. This paper describes an application of orthogonal nonnegative matrix factorization (NMF) algorithm in blind image separation (BIS) problem. The algorithm itself has been presented in our previous work as an attempt to provide a simple and convergent algorithm for orthogonal NMF, a type of NMF proposed to improve clustering capability of the standard NMF. When we changed the application domain of the algorithm to the BIS problem, surprisingly good results were obtained; the reconstructed images were more similar to the original ones and pleasant to view compared to the results produced by other NMF algorithms. Good results were also obtained when another dataset that consists of unrelated images was used. This practical use along with its convergence guarantee and implementation simplicity demonstrate the benefits of our algorithm.

Keywords: nonnegative matrix factorization, convergent algorithm, blind image separation, orthogonality constraint.

1 Introduction

Orthogonal nonnegative matrix factorization (ONMF) is an NMF objective that imposes orthogonality constraints on its factors. This objective was first introduced by Ding at al. [1] as an attempt to improve clustering capability of the standard NMF (SNMF) proposed by Lee and Seung [2,3]. In ref. [4,5], we proposed a convergent algorithm for ONMF based on the work of Lin [6]. We also showed that our ONMF algorithm can outperform the SNMF algorithm in document clustering task.

In this paper, we will show that our ONMF algorithm can also be used in blind image separation (BIS) problem; a task of recovering original images from image mixtures. This finding, thus demonstrates another application domain of the algorithm.

2 BIS Problem Statement in NMF

Let $\mathbf{W} \in \mathbb{R}_{+}^{M \times R}$ denotes M-by-R nonnegative matrix which each of its column $\mathbf{w}_r$ contains an original image (for this purpose, every image must be of the

H. Badioze Zaman et al. (Eds.): IVIC 2013, LNCS 8237, pp. 25–35, 2013.

same size and has been reshaped into the corresponding vector). In this work, we assume that the mixing process is linear and involves only the original images. Thus, this process can be modeled with:

$$\mathbf{A} = \mathbf{W}\mathbf{H},$$

where $\mathbf{H} \in \mathbb{R}_+^{R \times N}$ denotes column normalized nonnegative mixing matrix, i.e., $\sum_r h_{rn} = 1$, for $\forall n$. To recover $\mathbf{W}$ from $\mathbf{A}$, the NMF technique can be employed:

$$\mathbf{A} \approx \mathbf{B}\mathbf{C}, \tag{1}$$

where $\mathbf{B} \in \mathbb{R}_+^{M \times R}$ and $\mathbf{C} \in \mathbb{R}_+^{R \times N}$ are nonnegative approximations to $\mathbf{W}$ and $\mathbf{H}$ respectively. Thus, the task of recovering the original images turns into the task of decomposing the mixture matrix $\mathbf{A}$ into basis matrix $\mathbf{B}$ and coefficient matrix $\mathbf{C}$ which can be done using NMF algorithms.

3 A Convergent Algorithm for ONMF

A brief description of the algorithm will be presented in this section. More details including convergence analysis and experimental results in document clustering can be found in ref. [4,5].

The algorithm was proposed to solve the following objective:

$$\begin{aligned} &\min_{\mathbf{B},\mathbf{C}} J(\mathbf{B},\mathbf{C}) = \frac{1}{2}\|\mathbf{A} - \mathbf{B}\mathbf{C}\|_F^2 + \frac{\alpha}{2}\|\mathbf{C}\mathbf{C}^T - \mathbf{I}\|_F^2 \\ &\text{s.t. } \mathbf{B} \geq \mathbf{0}, \mathbf{C} \geq \mathbf{0}, \end{aligned} \tag{2}$$

where $\|\mathbf{X}\|_F$ denotes the Frobenius norm of $\mathbf{X}$, the first component of the right hand side part denotes the SNMF objective, the second component denotes the orthogonality constraint imposed on the rows of $\mathbf{C}$, $\mathbf{I}$ denotes a compatible identity matrix, and α denotes a regularization parameter to adjust the degree of orthogonality of $\mathbf{C}$. Algorithm 1 shows the convergent algorithm proposed in [4,5] for finding a solution to the problem.

The following gives definitions for some notations used in the algorithm:

$$\bar{b}_{mr}^{(k)} \equiv \begin{cases} b_{mr}^{(k)} \text{ if } \nabla_{\mathbf{B}} J\big(\mathbf{B}^{(k)}, \mathbf{C}^{(k)}\big)_{mr} \geq 0 \\ \max(b_{mr}^{(k)}, \sigma) \text{ if } \nabla_{\mathbf{B}} J\big(\mathbf{B}^{(k)}, \mathbf{C}^{(k)}\big)_{mr} < 0 \end{cases},$$

$$\bar{c}_{rn}^{(k)} \equiv \begin{cases} c_{rn}^{(k)} \text{ if } \nabla_{\mathbf{C}} J\big(\mathbf{B}^{(k+1)}, \mathbf{C}^{(k)}\big)_{rn} \geq 0 \\ \max(c_{rn}^{(k)}, \sigma) \text{ if } \nabla_{\mathbf{C}} J\big(\mathbf{B}^{(k+1)}, \mathbf{C}^{(k)}\big)_{rn} < 0 \end{cases},$$

denote the modifications to avoid the zero locking with σ is a small positive number, $\bar{\mathbf{B}}$ and $\bar{\mathbf{C}}$ denote matrices that contain $\bar{b}_{mr}$ and $\bar{c}_{rn}$ respectively,

$$\nabla_{\mathbf{B}} J(\mathbf{B}^{(k)}, \mathbf{C}^{(k)}) = \mathbf{B}^{(k)}\mathbf{C}^{(k)}\mathbf{C}^{(k)T} - \mathbf{A}\mathbf{C}^{(k)T},$$

Algorithm 1. A convergent algorithm for ONMF [4,5]

Initialization, $\mathbf{B}^{(0)} \geq \mathbf{0}$ and $\mathbf{C}^{(0)} \geq \mathbf{0}$.
for $k = 0, \ldots, K$ **do**

$$b_{mr}^{(k+1)} \longleftarrow b_{mr}^{(k)} - \frac{\bar{b}_{mr}^{(k)} \times \nabla_{\mathbf{B}} J(\mathbf{B}^{(k)}, \mathbf{C}^{(k)})_{mr}}{\mathbf{P}_{mr} + \delta}$$

$$\delta_{\mathbf{C}}^{k} \longleftarrow \delta$$

repeat

$$c_{rn}^{(k+1)} \longleftarrow c_{rn}^{(k)} - \frac{\bar{c}_{rn}^{(k)} \times \nabla_{\mathbf{C}} J(\mathbf{B}^{(k+1)}, \mathbf{C}^{(k)})_{rn}}{\mathbf{Q}_{rn} + \delta_{\mathbf{C}}^{(k)}}$$

$$\delta_{\mathbf{C}}^{(k)} \longleftarrow \delta_{\mathbf{C}}^{(k)} \times \text{step}$$

until $J\big(\mathbf{B}^{(k+1)}, \mathbf{C}^{(k+1)}\big) \leq J\big(\mathbf{B}^{(k+1)}, \mathbf{C}^{k}\big)$

end for

$$\begin{aligned}\nabla_{\mathbf{C}} J(\mathbf{B}^{(k+1)}, \mathbf{C}^{(k)}) = {} & \mathbf{B}^{(k+1)T}\mathbf{B}^{(k+1)}\mathbf{C}^{(k)} - \mathbf{B}^{(k+1)T}\mathbf{A} + \\ & \alpha\mathbf{C}^{(k)}\mathbf{C}^{(k)T}\mathbf{C}^{(k)} - \alpha\mathbf{C}^{(k)},\end{aligned}$$

$$\begin{aligned}\mathbf{P} &= \bar{\mathbf{B}}^{(k)}\mathbf{C}^{(k)}\mathbf{C}^{(k)T}, \\ \mathbf{Q} &= \mathbf{B}^{(k+1)T}\mathbf{B}^{(k+1)}\bar{\mathbf{C}}^{(k)} + \alpha\bar{\mathbf{C}}^{(k)}\bar{\mathbf{C}}^{(k)T}\bar{\mathbf{C}}^{(k)},\end{aligned}$$

δ denotes a small positive number to avoid division by zero, and step denotes a positive constant that determines how fast $\delta_{\mathbf{C}}^{(k)}$ grows in order to satisfy the nonincreasing property.

Numerically, as long as α is sufficiently small, then for each k-th iteration, the inner **repeat** – **until** loop will only be executed once. Since in practice usually α is set to be small, this inner loop can be opened to reduce the computational cost. The code below gives a quick implementation of the algorithm in Matlab codes for small α.

As shown, the computational complexity of the algorithm is $O(MNR)$ per iteration, thus is the same as the SNMF algorithm's.

```
function [B,C] = nmfOrtho(A,r)
[m,n] = size(A); alpha = 0.1; maxiter = 100;
B = rand(m,r);
C = rand(r,n);
sigma = 1.0e-9; delta = sigma;

for iteration = 1 : maxiter
  CCt   = C*C';
  gradB = B*CCt - A*C';
  Bm    = max(B,(gradB<0)*sigma);
  B     = B - Bm./(Bm*CCt+delta).*gradB;
```

```
  BtB   = B'*B;
  gradC = BtB*C + alpha1*CCt*C - alpha1*C - B'*A;
  Cm    = max(C,(gradC<0)*sigma);
  CmCmt = Cm*Cm';
  C     = C - Cm./(BtB*Cm + alpha1*CmCmt*Cm + delta).*gradC;
end
```

4 Experimental Results

The experiments were conducted using two image datasets. Each dataset consists of four images of 400x350 pixels. The first dataset is the dataset that contains related images, and the second contains unrelated images. Fig. 1 and 3 show the first and the second datasets respectively.

To create a mixture matrix $\mathbf{A}$ for each dataset, we generated a column normalized matrix $\mathbf{H} \in \mathbb{R}_{+}^{4\times 8}$ randomly. The resulting mixed images for the first and the second datasets are shown in fig. 2 and 4 respectively.

To recover the original images from the mixtures, the SNMF algorithm, the constrained NMF (CNMF) algorithm [7]—a widely used NMF algorithm in image unmixing research, extended Lee-Seung (ENMF) algorithm [8], and our ONMF algorithm were used. We also tried a block principal pivoting based NMF algorithm [9] which has a good convergence property and SMART algorithm [8] which was specially designed for blind source separation problem. However, we didn't include the results due to the poor performances in our datasets. The appendix shows the implementation of these algorithms with δ is defined as in algorithm 1, β in CNMF denotes the regularization parameter similar to α in ONMF, and η_B and η_C in ENMF denote the learning rates.

As there are parameters in CNMF, ONMF and ENMF, we repeatedly conducted the experiments using different parameter values until satisfactory results were obtained. Table 1 shows the values used in the experiments. The unmixed images are shown in fig. 5 and fig. 6 for dataset I and dataset II respectively.

For dataset I, ONMF produced the most pleasant images to view for all cases, followed by ENMF, CNMF and then SNMF. As shown in last row of fig. 5 every image produced by ONMF has minimum interference from other images resulting in the clearest images. ONMF also showed the best ability in detecting edges (boundaries of distinct regions) and distinct regions as both components are perceptually easiest to identify due to the more pronounced contrasts between

Table 1. Parameter values used in the experiments

Dataset	α	β	η_B	η_C
Dataset I	0.03	0.5	1	1
Dataset II	0.03	1	1	1

(a) (b) (c) (d)

Fig. 1. Dataset I: Related images

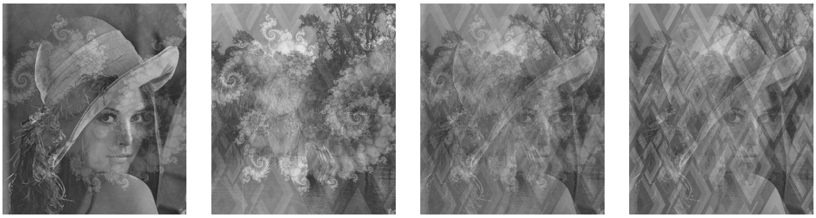
(a) (b) (c) (d)

(e) (f) (g) (h)

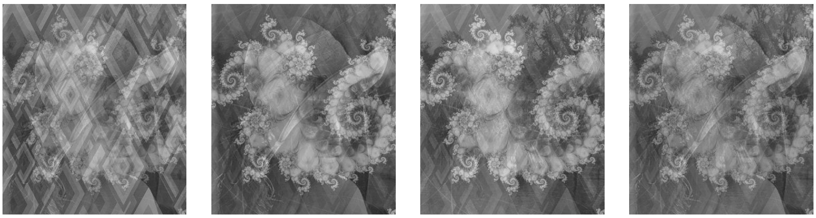
Fig. 2. Mixed images (dataset I)

(a) (b) (c) (d)

Fig. 3. Dataset II: Unrelated images

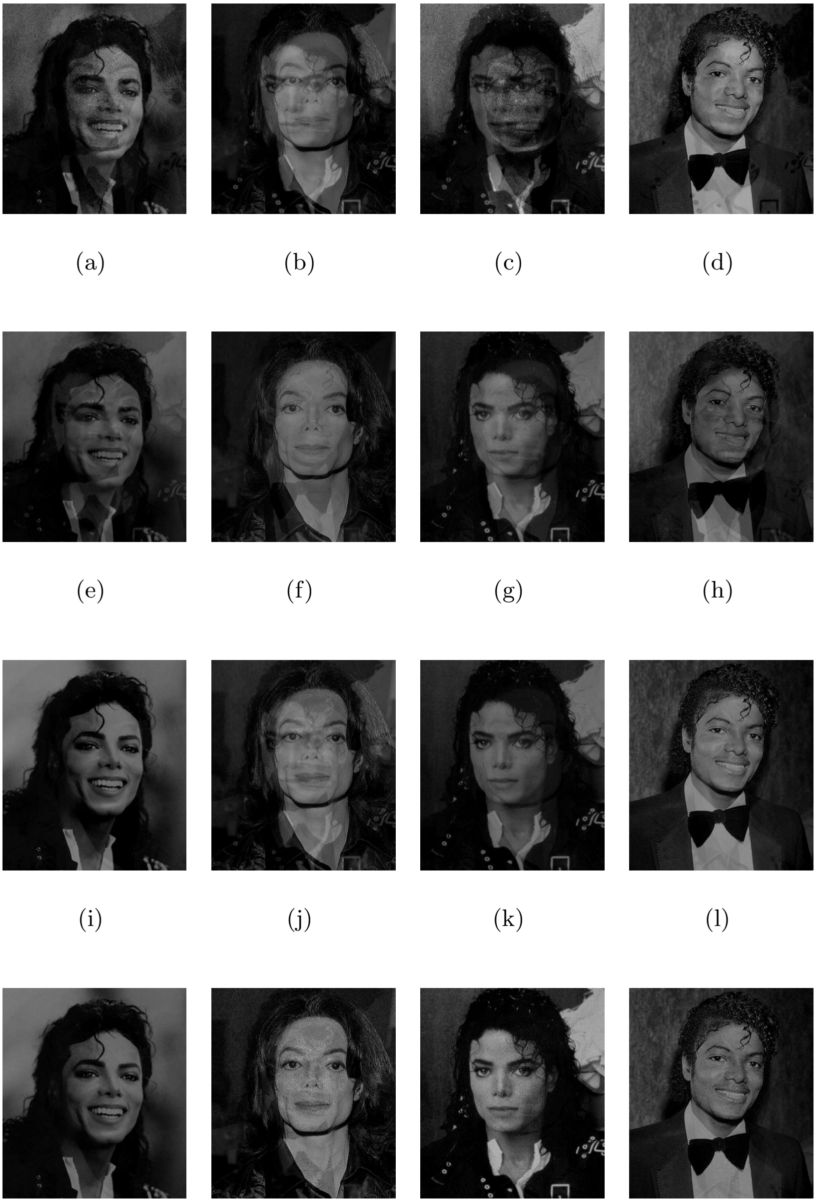


(a) (b) (c) (d)



(e) (f) (g) (h)

Fig. 4. Mixed images (dataset II)



(m) (n) (o) (p)

Fig. 5. Unmixed images for dataset I. The first, second, third, and fourth row correspond to SNMF, CNMF, ENMF, and ONMF respectively.

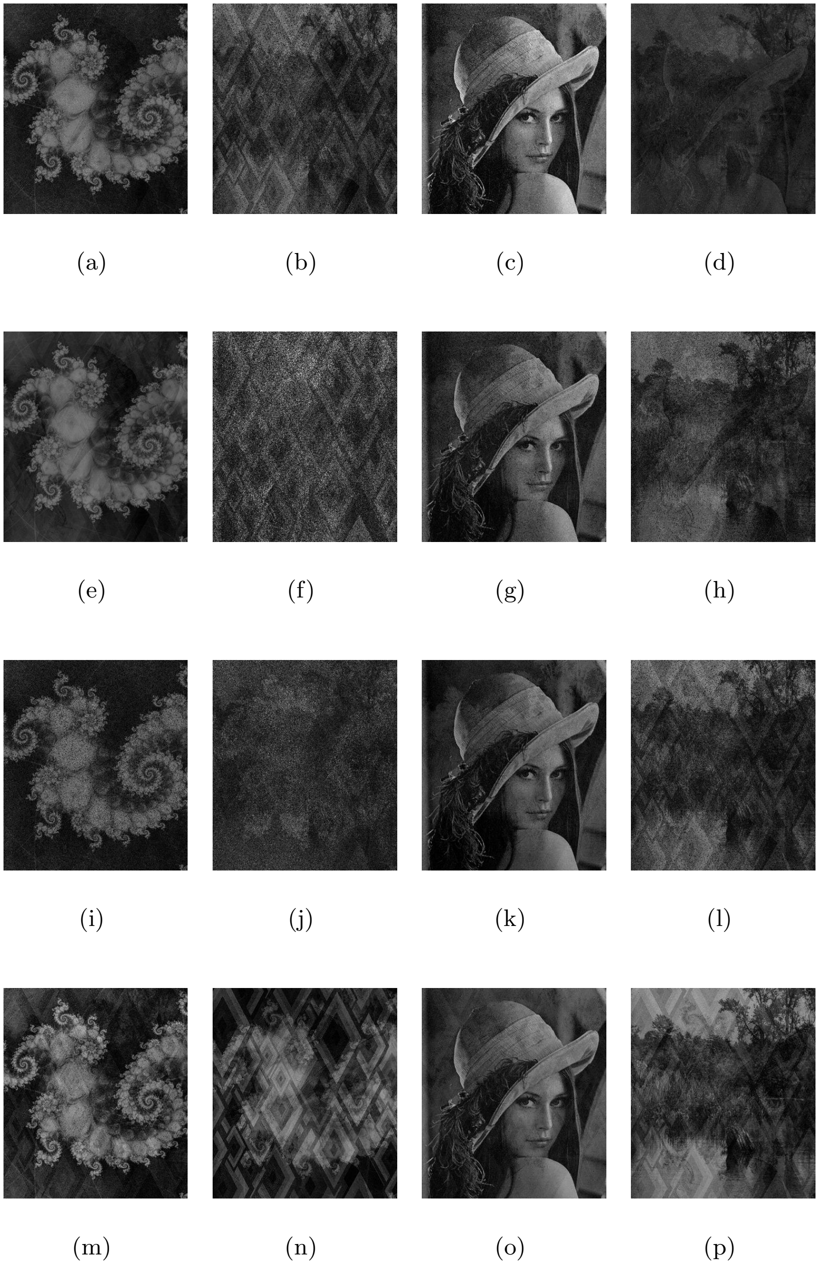


Fig. 6. Unmixed images for dataset II. The first, second, third, and fourth row correspond to SNMF, CNMF, ENMF, and ONMF respectively.

Table 2. SNR measures for dataset I

Algorithm	Fig 1	Fig 2	Fig 3	Fig 4	Total
SNMF	4.011	4.729	5.607	7.107	21.45
CNMF	3.909	5.322	5.487	4.530	19.25
ENMF	5.723	6.466	6.718	5.883	24.79
ONMF	**8.290**	**7.871**	**8.889**	**8.044**	**33.09**

Table 3. SNR measures for dataset II

Algorithm	Fig 1	Fig 2	Fig 3	Fig 4	Total
SNMF	5.338	3.361	**12.47**	3.086	24.26
CNMF	5.259	3.833	8.372	4.324	21.79
ENMF	5.891	4.165	10.56	5.010	25.63
ONMF	**7.364**	**5.654**	10.81	**6.915**	**30.74**

different regions and smoother pixels within a region. These facts can also be captured by using SNR (signal-to-noise-ratio). The following formula gives the definition of SNR.

$$\text{SNR} = -10\log_{10}\frac{\|\mathbf{w}_r - \mathbf{b}_r\|_F^2}{\|\mathbf{w}_r\|_F^2},$$

where $\mathbf{w}_r$ and $\mathbf{b}_r$ denote r-th column of $\mathbf{W}$ and $\mathbf{B}$ respectively. Table 2 shows SNR measures between the original images and the reconstructed images. As shown, in term of SNR, ONMF confidently outperformed other algorithms as well.

In dataset II where the images are unrelated, the recovering tasks were more difficult to perform as the mixed images are less informative than the mixed images of dataset I (one can easily point out that there are four different images from fig. 2, but it's not clear how many images represented by fig. 4). Regardless, as shown in fig. 6 all NMF algorithms seemed to be successful in unmixing those images (except ENMF which seemed to fail to recover the second image). The quality of unmixing, however, is different as reflected by the SNR values shown in table 3. In this case, overall our algorithm performed the best. Only in the Lena image, SNMF outperformed our algorithm.

5 Conclusion

We have shown an application of our ONMF algorithm in image unmixing problem in which the algorithm could work well both in the dataset of related images and dataset of unrelated images. In the dataset of related images, our algorithm worked very well in which it produced very clear images and hence are pleasant to view. In the dataset of unrelated images, in overall the algorithm still performed the best. However, since the source images fig. 4 are less informative than the source images fig. 2, all algorithms produced rather unclear images compared to the reconstructions of the first dataset.

Our algorithm also had the best ability to detect boundaries between distinct regions which were often blurred in the results of other algorithms. The more pronounced contrasts between different regions and smoother pixels within a region were also observed in the results of our algorithm which contributed to the overall quality of the reconstruction.

When the unmixing performance was quantified using SNR, in general our algorithm could confidently outperform other algorithms since only in the Lena image, the algorithm failed to perform the best.

The good performance of the proposed algorithm in the BIS problem can be thought as a result of enforcing orthogonality in the coefficient matrix **C** which in the same time also enforcing independency in the basis matrix **B** so that each column of **B** tends to contain more information of individual image from the mixtures.

Acknowledgement. The author would like to thank the reviewers for useful comments. This research was supported by Ministry of Higher Education of Malaysia and Universiti Teknologi Malaysia under Research University Grant Q.J130000.2628.07J82.

References

1. Ding, C., Li, T., Peng, W., Park, H.: Orthogonal nonnegative matrix tri-factorizations for clustering. In: 12th ACM SIGKDD Intl Conf. on Knowledge Discovery and Data Mining, pp. 126–135 (2006)
2. Lee, D., Seung, H.: Learning the parts of objects by non-negative matrix factorization. Nature 401, pp. 788–791 (1999)
3. Lee, D., Seung, H.: Algorithms for non-negative matrix factorization. In: Proc. Advances in Neural Processing Information Systems, pp. 556–562 (2000)
4. Mirzal, A.: A convergent algorithm for orthogonal nonnegative matrix factorization. Submitted to J Computational and Applied Mathematics
5. Mirzal, A.: Nonnegative matrix factorizations for clustering and LSI: Theory and programming. LAP LAMBERT Academic Publishing, Germany (2011)
6. Lin, C.: On the convergence of multiplicative update algorithms for nonnegative matrix factorization. IEEE Transactions on Neural Networks 18(6), pp. 1589–1596 (2007)
7. Pauca, V., Piper, J., Plemmons, R.: Nonnegative matrix factorization for spectral data analysis. Linear Algebra and Its Applications 416(1), pp. 29–47 (2006)
8. Cichocki, A., Amari, S., Zdunek, R., Kompass, R., Hori, G., He, Z.: Extended smart algorithms for non-negative matrix factorization. In: 8th intl conf. on Artificial Intelligence and Soft Computing, Springer-Verlag, Berlin, Heidelberg, pp. 548–562 (2006)
9. Kim, J., Park, H.: Toward faster nonnegative matrix factorization: A new algorithm and comparisons. In: 8th IEEE Intl Conf. on Data Mining, pp. 353–362 (2008)

Appendix

Algorithm 2. The SNMF algorithm [3].

Initialization, $\mathbf{B}^{(0)} > \mathbf{0}$ and $\mathbf{C}^{(0)} > \mathbf{0}$.
for $k = 0, \ldots, K$ **do**

$$b_{mr}^{(k+1)} \longleftarrow b_{mr}^{(k)} \frac{(\mathbf{A}\mathbf{C}^{(k)T})_{mr}}{(\mathbf{B}^{(k)}\mathbf{C}^{(k)}\mathbf{C}^{(k)T})_{mr} + \delta}$$

$$c_{rn}^{(k+1)} \longleftarrow c_{rn}^{(k)} \frac{(\mathbf{B}^{(k+1)T}\mathbf{A})_{rn}}{(\mathbf{B}^{(k+1)T}\mathbf{B}^{(k+1)}\mathbf{C}^{(k)})_{rn} + \delta}$$

end for

Algorithm 3. The CNMF algorithm [7].

Initialization, $\mathbf{B}^{(0)} > \mathbf{0}$ and $\mathbf{C}^{(0)} > \mathbf{0}$.
for $k = 0, \ldots, K$ **do**

$$b_{mr}^{(k+1)} \longleftarrow b_{mr}^{(k)} \frac{(\mathbf{A}\mathbf{C}^{(k)T})_{mr}}{(\mathbf{B}^{(k)}\mathbf{C}^{(k)}\mathbf{C}^{(k)T})_{mr} + \delta}$$

$$c_{rn}^{(k+1)} \longleftarrow c_{rn}^{(k)} \frac{(\mathbf{B}^{(k+1)T}\mathbf{A})_{rn}}{(\mathbf{B}^{(k+1)T}\mathbf{B}^{(k+1)}\mathbf{C}^{(k)} + \beta\mathbf{C}^{(k)})_{rn} + \delta}$$

end for

Algorithm 4. The ENMF algorithm [8].

Initialization, $\mathbf{B}^{(0)} > \mathbf{0}$ and $\mathbf{C}^{(0)} > \mathbf{0}$.
for $k = 0, \ldots, K$ **do**

$$b_{mr}^{(k+1)} \longleftarrow b_{mr}^{(k)} - \eta_B b_{mr}^{(k)} \frac{(\mathbf{B}^{(k)}\mathbf{C}^{(k)T} - \mathbf{A}\mathbf{C}^{(k)T})_{mr}}{(\mathbf{B}^{(k)}\mathbf{C}^{(k)}\mathbf{C}^{(k)T})_{mr} + \delta}$$

$$c_{rn}^{(k+1)} \longleftarrow c_{rn}^{(k)} - \eta_C c_{rn}^{(k)} \frac{\left(\mathbf{B}^{(k+1)T}\left(\mathbf{B}^{(k+1)T}\mathbf{C}^{(k)} - \mathbf{A}\right)\right)_{rn}}{(\mathbf{B}^{(k+1)T}\mathbf{B}^{(k+1)}\mathbf{C}^{(k)})_{rn} + \delta}$$

end for