A Deep Structure-Enforced Nonnegative Matrix Factorization for Data Representation

Yijia Zhou[1] and Lijun Xu[2(✉)]

[1] Dalian Neusoft University of Information,
Dalian 116023, Liaoning, People's Republic of China
zhouyijia@neusoft.edu.cn

[2] Dalian Maritime University,
Dalian 116026, Liaoning, People's Republic of China
lijun_xu@dlmu.edu.cn

Abstract. In this paper, we focus on a deep structure-enforced nonnegative matrix factorization (DSeNMF) which represents a large class of deep learning models appearing in many applications. We present a unified algorithm framework, based on the classic alternating direction method of multipliers (ADMM). For updating subproblems, we derive an efficient updating rule according to its KKT conditions. We conduct numerical experiments to compare the proposed algorithm with state-of-the-art deep semi-NMF. Results show that our algorithm performs better and our deep model with different sparsity imposed indeed results in better clustering accuracy than single-layer model. Our DSeNMF can be flexibly applicable for data representation.

Keywords: Deep matrix fatorization · Alternating direction method
Data representation

1 Introduction

Matrix factorization techniques have found great utility in various data-related applications, such as in signal and image processing and in machine learning tasks, primarily because they often help reveal latent features in a dataset. In recent years, Non-negative Matrix Factorization (NMF) is a widely-used method for finding meaningful representations of nonnegative data and has been proven useful in dimension reduction of images, text data and signals, for example. The family of NMF algorithms has been successfully applied to a variety of areas, like environmetrics [1], microarray data analysis [2,3], document clustering [4], face recognition [5,6], speech recognition [7], hyperspectral image unmixing [8,9], blind audio source separation [10], etc. Moreover, NMF has been extended into

Supported by the Fundamental Research Funds for the Central Universities (3132018218).

J.-H. Lai et al. (Eds.): PRCV 2018, LNCS 11258, pp. 340–350, 2018.
https://doi.org/10.1007/978-3-030-03338-5_29

a number of variant forms, allowing for various structures or regularized models, most of which demonstrate distinct advantages in local feature extraction or data representation learning.

The work of Lee and Seung [11] demonstrates that NMF models tend to return part-based sparse representations of data, which has popularized the use of and research on NMF-related techniques. In particular, various NMF-inspired formulations add different regularization or penalty terms to promote desired properties, such as sparsity patterns or orthogonality in addition to nonnegativity (see [12–18], for example). Besides, graph-regularized NMF versions have also been explored. For example, Cai et al. [19] proposed a graph-regularized NMF by incorporating prior information of samples into the typical NMF. This helps to keep the original topological structure of data after being projected into a subspace and usually leads to better clustering results.

Semi Non-negative Matrix Factorization (Semi-NMF) [20], as one of the most popular variants of NMF, was proposed to extend NMF by relaxing the factorized basis matrix to be real values. This practice allows Semi-NMF to learn new lower-dimensional features from the data that have a convenient clustering interpretation and have a wider application in the real world than traditional NMF. Moreover, it has shown that it is equivalent to k-means clustering, and that in fact, this NMF variants are expected to perform better than k-means clustering particularly when the data is not distributed in a spherical manner.

Although there have been extensive variants of NMF, most of them remain to be single-layer models, hence can only capture one level of data features. Most recently, deep learning is becoming increasingly popular and has been demonstrated to be powerful in learning data representation. Inspired by the success of training deep architectures, Multi-layer NMF (see [21,22] for example), Deep Semi-NMF [23], Deep Orthogonal NMF [24], Sparse Deep NMF [25], Deep Nonsmooth NMF [26], etc. have been proposed by stacking one-layer variants of NMF into multiple layers to learn hierarchical relationships among features or hierarchical projections. Since these deep (multi-layer) models can extract high level data representations and yield intuitive interpretations for features generated in each layer, they have been successfully applied to many areas, such as recommender systems [27], image clustering [28], neural network [29], speech separation [30], matrix completion [31], for example. However, these models are only designed for specific problems with certain intuitive structures. In this paper, we focus on a unified deep structure-enforced NMF in data representation, which imposing desired properties (like sparsity, orthogonality, for example) in addition to nonnegativity. A specific algorithmic approach to solve the deep structure-enforced NMF is further studied and can be applicable to a range of easily projectable structures.

This paper is organized as follows. In Sect. 2, we introduce the deep structure-enforced NMF (DSeNMF) and propose a new ADMM-based algorithm framework for solving DSeNMF. Section 3 contains several numerical experiments comparing the proposed algorithm with Deep Semi-NMF and single-layer matrix factorization on MNIST digit dataset. Finally, we conclude this paper in Sect. 4.

2 Deep Structure-Enforced Nonnegative Matrix Fatorization Model

The general structured-enforced matrix factorization (SeMF) model (1) is firstly proposed in the earlier work in [32]. That is, decomposing a given data matrix $M \in \mathbb{R}^{p\times n}$ into two factors $Z \in \mathbb{R}^{p\times k}$ and $H \in \mathbb{R}^{k\times n}$ which belong to $\mathcal{Z}$ and $\mathcal{H}$, respectively,

$$\min_{Z,H} \frac{1}{2}\|M - ZH\|_F^2 \ \text{ s.t. } \ Z \in \mathcal{Z},\ H \in \mathcal{H}, \tag{1}$$

where $\|\cdot\|_F$ is Frobenius norm, and $\mathcal{Z}$ and $\mathcal{H}$ are subsets of $\mathbb{R}^{p\times k}$ and $\mathbb{R}^{k\times n}$, respectively. Obviously, the model (1) is a single-layer matrix factorization. Thus, it can only do one-layer feature extraction even utilizing more structures. In practice, it is common that complex data objects have hierarchical features, each of which denotes a different level of abstract understanding of the objects. It is therefore meaningful to develop corresponding models with a deep architecture, which allows to discover the hierarchy of data. It is well known that NMF is widely used both in single-layer and in multi-layer data representation. To this end, we propose a deep structure-enforced version for nonnegative matrix factorization by extending model (1).

Similar to the general multi-layer framework, the Deep Structure-enforced NMF (DSeNMF) model is presented to factorize $M \in \mathbb{R}^{p\times n}$ into the multiplier of m + 1 nonnegative matrices, as follows:

$$\min_{\{Z_i\geq 0\}_{i=1}^m, H_m\geq 0} \frac{1}{2}\|M - Z_1Z_2\cdots Z_mH_m\|_F^2 \ \text{ s.t. } \ Z_i \in \mathcal{Z}_i,\ H_m \in \mathcal{H}, \tag{2}$$

where $Z_1 \in \mathbb{R}^{p\times k_1}$, $\{Z_i \in \mathbb{R}^{k_{i-1}\times k_i}\}_{i=2}^m$, $H_m \in \mathbb{R}^{k_m\times n}$, $\{\mathcal{Z}_i\}_{i=1}^m$ and $\mathcal{H}$ are structure subsets with proper dimensions. In our model, prior knowledge are explicitly enforced as constraint sets $\{\mathcal{Z}_i\}_{i=1}^m$ and $\mathcal{H}$ whose members possess desirable matrix structures allowing "easy projection". In practice, the most useful structures of this kind include, but are not limited to, nonnegativity, normality and various sparsity patterns. Many deep NMF models can be represented by the DSeNMF (2) with different structure constraints, see Sparse Deep NMF, Deep Orthogonal NMF, Deep Semi-NMF as mentioned above, for example.

To make it more intuitive, one can split the model (2) into the following factorizations:

$$\begin{aligned} M &\approx Z_1H_1, \\ H_1 &\approx Z_2H_2, \\ &\vdots \\ H_{m-1} &\approx Z_mH_m, \end{aligned} \tag{3}$$

where $\{Z_i\}_{i=1}^m$ and $\{H_i\}_{i=1}^m$ satisfy proper constraints, respectively. This formulation can intuitively illustrate that deep model (2) allows for a hierarchy of m layers of implicit representations of data. In other words, not only most multi-layer and deep matrix factorizations is derived from the formulation (3), but also most algorithms for (2) are designed by solving (3) layer by layer. In the beginning of approaches, the objective data matrix are multi-factorized only by solving (3) one round layer by layer. Obviously, these approaches are inefficient since the factor matrices in former layers are useless for subsequent layer factorizations. Therefore, the popular scheme is utilizing the layer by layer technique as initialization or pre-training, then fine-tuning all layers by alternating updating factor matrices one by one. Now, we propose a novel approach based on alternating direction algorithm framework to solve the non-convex problem (2).

2.1 An Alternating Direction Algorithm for the Proposed DSeNMF

As introduced in the work [32,33], an alternating direction and projection method solves single layer structure-enforced matrix factorization (SeMF) efficiently. Motivated by the algorithms in [32,33], we propose a novel way to tackle multi-layer or deep matrix factorizations. To facilitate an efficient use of alternating minimization, we introduce auxiliary variables $\{U_i\}_{i=1}^m$ and V_m in order to separate $\{Z_i\}_{i=1}^m$ and H_m from structure constraints $\{\mathcal{Z}_i\}_{i=1}^m$ and $\mathcal{H}$, respectively. Consider the following model equivalent to (2),

$$\begin{aligned} \min_{\{Z_i\geq 0, U_i\}_{i=1}^m, H_m\geq 0, V_m} & \frac{1}{2}\|M - Z_1 Z_2 \cdots Z_m H_m\|_F^2 \\ \text{s.t.} \;\; & Z_i - U_i = 0,\; U_i \in \mathcal{Z}_i, i = 1, \cdots, m, \\ & H_m - V_m = 0,\; V_m \in \mathcal{H}, \end{aligned} \tag{4}$$

where $\{U_i\}_{i=1}^m$ and V_m have the same dimension size with $\{Z_i\}_{i=1}^m$ and H_m, respectively. The augmented Lagrangian function of (4) is

$$\begin{aligned} & \mathcal{L}_A(\{Z_i, U_i, \Lambda_i\}_{i=1}^m, H_m, V_m, \Pi) \\ = & \tfrac{1}{2}\|M - Z_1 Z_2 \cdots Z_m H_m\|_F^2 + \\ & \textstyle\sum_{i=1}^m \Lambda_i \bullet (Z_i - U_i) + \Pi \bullet (H_m - V_m) \\ + & \textstyle\sum_{i=1}^m \frac{\alpha_i}{2}\|Z_i - U_i\|_F^2 + \frac{\beta}{2}\|H_m - V_m\|_F^2, \end{aligned} \tag{5}$$

where $\{\Lambda_i\}_{i=1}^m, \Pi$ are Lagrangian multipliers with equal-size of $\{Z_i\}_{i=1}^m, H_m$, respectively, and $(\{\alpha_i\}_{i=1}^m, \beta) \geq 0$ are penalty parameters for equality constraints, respectively. Note that the scalar product "$\bullet$" of two equal-size matrices X and Y is the sum of all element-wise products, i.e., $X \bullet Y = \sum_{i,j} X_{ij} Y_{ij}$.

The alternating direction method of multiplier (ADMM) [34,35] for (4) is derived by successively minimizing the augmented Lagrangian function $\mathcal{L}_A$ with respect to $\{Z_i\}_{i=1}^m, H_m$, $\{U_i\}_{i=1}^m$ and V_m, one at a time while fixing others at their most recent values, and then updating the multipliers after each sweep of such alternating minimization. The introduction of the auxiliary variables $\{U_i\}_{i=1}^m$ and V_m makes it easy to carry out each of the alternating minimization steps. Specifically, these steps can be written in the following forms,

$$Z_j^+ \approx \arg\min_{Z_j \geq 0} \mathcal{L}_A(\{Z_i, U_i, \Lambda_i\}_{i=1}^m, H_m, V_m, \Pi), j = 1, 2, \cdots, m, \tag{6a}$$

$$H_m^+ \approx \arg\min_{H_m \geq 0} \mathcal{L}_A(\{Z_i^+, U_i, \Lambda_i\}_{i=1}^m, H_m, V_m, \Pi), \tag{6b}$$

$$U_j^+ = \mathcal{P}_{\mathcal{Z}_j}(Z_j^+ + \Lambda_j/\alpha_j), j = 1, 2, \cdots, m, \tag{6c}$$

$$V_m^+ = \mathcal{P}_{\mathcal{H}}(H_m^+ + \Pi/\beta), \tag{6d}$$

$$\Lambda_j^+ = \Lambda_j + \alpha_j(Z_j^+ - U_j^+), j = 1, 2, \cdots, m, \tag{6e}$$

$$\Pi^+ = \Pi + \beta(H_m^+ - V_m^+). \tag{6f}$$

where $\mathcal{P}_{\mathcal{Z}_j}$ ($\mathcal{P}_{\mathcal{H}}$) stands for the projection onto the set $\mathcal{Z}_j$ ($\mathcal{H}$) in Frobenius norm, and the superscript "+" is used to denote iterative values at the new iteration.

Updating Rule for Z_j. We fix the rest of the factor matrices and minimize the cost function with respect to Z_j. The Z_j-updating subproblem (6a) actually can be rewritten as

$$\begin{aligned} \min_{Z_j} \ & \frac{1}{2}\|M - \Phi_j Z_j \Psi_j\|_F^2 + \Lambda_j \bullet (Z_j - U_j) + \frac{\alpha_j}{2}\|Z_j - U_j\|_F^2 \\ \text{s.t.} \ & Z_j \geq 0, \end{aligned} \tag{7}$$

where $\Phi_j = Z_1 Z_2 \cdots Z_{j-1}$ and $\Psi_j = Z_{j+1} \cdots Z_m H_m$. Let Γ be the lagrangian multiplier for constraint $Z_j \geq 0$, the Lagrangian function of (7) is

$$\mathcal{L} = \frac{1}{2}\|M - \Phi_j Z_j \Psi_j\|_F^2 + \Lambda_j \bullet (Z_j - U_j) + \frac{\alpha_j}{2}\|Z_j - U_j\|_F^2 + \Gamma \bullet Z_j.$$

The partial derivative of $\mathcal{L}$ with respect to Z_j is

$$\frac{\partial \mathcal{L}}{\partial Z_j} = \Phi_j^{\mathrm{T}} \Phi_j Z_j \Psi_j \Psi_j^{\mathrm{T}} + \alpha_j Z_j - \Phi_j^{\mathrm{T}} M \Psi_j^{\mathrm{T}} - \alpha_j U_j + \Lambda_j + \Gamma.$$

Using the Karush-Kuhn–Tucker (KKT) conditions $\Gamma_{ik} {Z_j}_{ik} = 0$, we get the following equations respect to the (i, k)-th element:

$$\left(\Phi_j^{\mathrm{T}} \Phi_j Z_j \Psi_j \Psi_j^{\mathrm{T}} + \alpha_j Z_j - \Phi_j^{\mathrm{T}} M \Psi_j^{\mathrm{T}} - \alpha_j U_j + \Lambda_j\right)_{ik} (Z_j)_{ik} = 0.$$

This equation leads to the following updating rule:

$$(Z_j^+)_{ik} = (Z_j)_{ik} \frac{(\Phi_j^{\mathrm{T}} M \Psi_j^{\mathrm{T}} + \alpha_j U_j - \Lambda_j)_{ik}}{(\Phi_j^{\mathrm{T}} \Phi_j Z_j \Psi_j \Psi_j^{\mathrm{T}} + \alpha_j Z_j)_{ik}}, \tag{8}$$

and it can be rewritten as

$$Z_j^+ = Z_j \odot [(\Phi_j^{\mathrm{T}} M \Psi_j^{\mathrm{T}} + \alpha_j U_j - \Lambda_j) \oslash (\Phi_j^{\mathrm{T}} \Phi_j Z_j \Psi_j \Psi_j^{\mathrm{T}} + \alpha_j Z_j)], \tag{9}$$

where $\odot$ and $\oslash$ denote component multiplications and divisions,respectively.

Updating Rule for H_m. We can derive the H_m-updating rule of (6b) in a similar way. We omit the derivative procedure and directly write updating rule for (i, k)-th component of H_m:

$$(H_m^+)_{ik} = (H_m)_{ik} \frac{(\Phi^{\mathrm{T}} M + \beta V_m - \Pi)_{ik}}{(\Phi^{\mathrm{T}} \Phi H_m + \beta H_m)_{ik}}, \tag{10}$$

where $\Phi = Z_1^+ Z_2^+ \cdots Z_m^+$. Namely,

$$H_m^+ = H_m \odot [(\Phi^{\mathrm{T}} M + \beta V_m - \Pi) \oslash (\Phi^{\mathrm{T}} \Phi H_m + \beta H_m)], \tag{11}$$

where $\odot$ and $\oslash$ denote component multiplications and divisions, respectively.

Since we update Z_j and H_m by component multiplications and divisions instead of involving inverse matrices, the dominant computational tasks at each iteration are the matrix multiplications. Therefore, our updating scheme posses much lower complexity than inverting matrices.

Based on the formulas in (6), (9) and (11), we can implement the following ADMM algorithmic framework so long as we can compute the projections in steps (6c) and (6d).

Algorithm 1. ADMM Framework for DSeNMF

Input: M, each layer dimension $k_i, i = 1, \cdots, m$, $maxiter > 0$ and $tol > 0$.
Output: $\{Z_i\}_{i=1}^m$ and H_m.
Set $\{\alpha_i\}_{i=1}^m, \beta > 0$.
$H_0 = M$;
for $i = 1$ *to* m **do**
  $Z_i, H_i \leftarrow \mathrm{SeMF}(H_{i-1}, k_i)$ \\\\ *Initialization.*
end
for $k = 1$ *to maxiter* **do**
  Update $(\{Z_i, U_i, \Lambda_i\}_{i=1}^m, H_m, V_m, \Pi)$ by the formulas in (6), (9) and (11).
  if *stopping criterion (12) is met* **then**
    output $\{Z_i\}_{i=1}^m$ and H_m, and exit.
  end
end

We use the following practical stopping criterion: for given tolerance $tol > 0$,

$$\frac{|f_k - f_{k+1}|}{|f_k|} \leq tol, \tag{12}$$

where $f_k = \|X - Z_1^k Z_2^k \cdots Z_m^k H_m^k\|_F$, Z_i^k is the k-th iterate for the variable Z_i, and so on. For the sake of robustness, in our implementation we require that the above condition be satisfied at three consecutive iterations. In other words, we stop the algorithm when data fidelity does not change meaningfully in three consecutive iterations.

3 Experimental Results

In this section we test the proposed model on MNIST dataset to show that our Deep SeNMF is able to learn better high-level representations of data than a single one-layer structure-enforced NMF. In addition, we compare the performance of the proposed DSeNMF with recently Deep Semi-NMF on the task of clustering analysis and consuming time. Note that we consider to impose several sparse constraints on our DSeNMF model (2).

To better understand the proposed model, we introduce three way to impose sparsity on H_m. One is adding sparsity not only during initialization but also in subsequential updating and denote this case as `DSeNMF(sparse)`. The other way is imposing sparsity only in step (6d), that is, using standard NMF to initialize each layer matrix, and is denoted as `DSeNMF(semi-sparse)`. The last one will not impose sparsity and denote this case as `DSeMF(no sparse)`. To illustrate deep model and single-layer factorization distinct, we also consider single-layer structure-enforced matrix factorization and denote as `SingleSeMF`.

Next, we apply models to the testing data in an unsupervised way to clustering. We opt the digits from 0 to 4 in MNIST which constitute a 784×5139 matrix M. In this test, we choose the number of layers to be 3 and dimension size of each layer is 300, 15 and 50, respectively. Besides, set the maximum number of iteration $maxiter = 500$ and tolerance $tol = 1\text{e}{-6}$. We factorize data matrix M using Deep Semi-NMF (`DSemiNMF`) in [23], `DSeNMF(sparse)`, `DSeNMF(semi-sparse)` and `SingleSeMF`, respectively. Then we cluster columns of the final H_m according to the approach in [23] and output the clustering accuracy as AC.

Table 1. Results comparison with different deep NMF models

Method	`DSeNMF (sparse)`	`DSeNMF (semi-sparse)`	`DSeNMF (no sparse)`	`DSemiNMF` [23]	`SingleSeMF` [32]
AC	0.57	0.68	0.48	0.40	0.33
Time(s)	64.18	64.38	66.57	292.54	29.39
RMSE	37.3688	37.3693	37.3676	37.4621	24.0117

In Table 1, we tabulate the average clustering accuracy (AC), average running time (in second) and average root mean square error (RMSE). We see from the table that our deep structure-enforced NMF performs well both in accuracy and in time consuming. It should be note that our algorithm only need about one fifth running time comparing with deep semi-NMF algorithm. In addition, note that the last column in Table 1, we use the SeMF algorithm in [32] to decompose M into multiplication of $Z \in \mathbb{R}^{784\times 50}$ and $H \in \mathbb{R}^{50\times 5139}$ which is indeed a single-layer nonnegative matrix factorization. Obviously, `SingleSeMF` obtain the best data fidelity, but get the worst clustering accuracy meanwhile. It confirms that all the DSeNMF models are able to learn better high-level representations of data than a single one-layer structure-enforced NMF. Among results of our proposed model with three different structure constraints, we note that `DSeNMF(sparse)` and `DSeNMF(semi-sparse)` obtain better clustering results than `DSeNMF(no sparse)` since imposing sparsity on H_m. More interestingly, comparing `DSeNMF(sparse)` with `DSeNMF(semi-sparse)`, the former gets lower clustering accuracy even though considering sparsity in initialization. It demonstrates that imposing structure constraints earlier could not obtain a better initialization. It makes sense that some properties in real data should be considered step by step rather than completely utilized at the beginning.

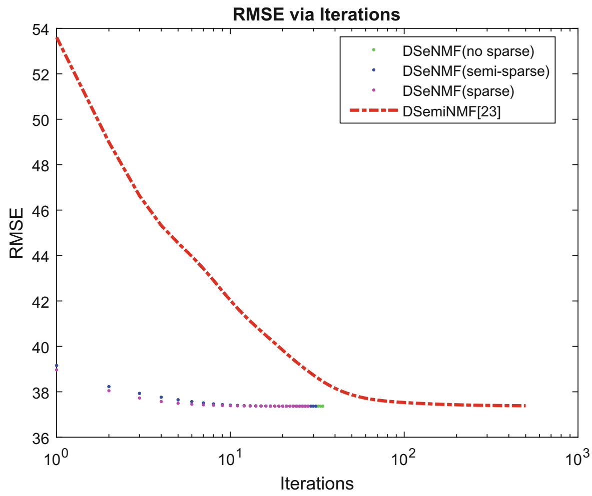

Fig. 1. RMSE comparison with different deep NMF models

Figure 1 presents RMSE curves of four deep models. It shows that our algorithm for solving deep NMF models needs much less (about 50) iterations than the algorithm in [23] (around 500 iterations). It will be evident that our proposed model and algorithm are efficient for the class of deep structured NMF.

4 Conclusion and Future

We have introduced a kind of deep structure-enforced nonnegative matrix factorization and proposed a novel framework for solving the unified model. Although the proposed framework introduces many auxiliary variables, these variables aim to separate from complex structure constraints and split original factor matices. Further, it can facilitate the obtained model equivalently transformed to an ADMM-applicable model which is easy implemented. Numerical experiments also show the efficiency of the proposed algorithm and the applicable of our deep model for data representing problems.

Although deep structured matrix factorization problems are generally highly nonconvex, they widely and variously exist in real-world applications. Our next step is testing the proposed model and algorithm on more datasets and comparing it with other deep NMF algorithms. Another work will be focusing on how different decomposed dimension would affect clustering performance of deep non-negative matrix factorization.

References

1. Paatero, P., Tapper, U.: Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values. Environmetrics **5**(2), 111–126 (1994)
2. Brunet, J.-P., Tamayo, P., Golub, T.R., Mesirov, J.P.: Metagenes and molecular pattern discovery using matrix factorization. PNAS **101**(12), 4164–4169 (2004)
3. Devarajan, K.: Nonnegative matrix factorization: an analytical and interpretive tool in computational biology. PLoS Comput. Biol. **4**(7), e1000029 (2008)
4. Berry, M.W., Browne, M.: Email surveillance using nonnegative matrix factorization. Comput. Math. Organ. Theory **11**(3), 249–264 (2005)
5. Zafeiriou, S., Tefas, A., Buciu, I., Pitas, I.: Exploiting discriminant information in nonnegative matrix factorization with application to frontal face verification. TNN **17**(3), 683–695 (2006)
6. Kotsia, I., Zafeiriou, S., Pitas, I.: A novel discriminant nonnegative matrix factorization algorithm with applications to facial image characterization problems. TIFS **2**(3–2), 588–595 (2007)
7. Zdunek, R., Cichocki, A.: Non-negative matrix factorization with quasi-newton optimization. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J.M. (eds.) ICAISC 2006. LNCS (LNAI), vol. 4029, pp. 870–879. Springer, Heidelberg (2006). https://doi.org/10.1007/11785231_91
8. Wang, W., Li, S., Qi, H., Ayhan, B., Kwan, C., Vance, S.: Identify anomaly component by sparsity and low rank. In: IEEE Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensor (WHISPERS), Tokyo, Japan (2015)
9. Qu, Y., Guo, R., Wang, W., Qi, H., Ayhan, B., Kwan, C., Vance, S.: Anomaly detection in hyperspectral images through spectral unmixing and low rank decomposition. In: IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Beijing, pp. 1855–1858 (2016)
10. Weninger, F., Schuller, B.: Optimization and parallelization of monaural source separation algorithms in the openBliSSART toolkit. J. Signal Process. Syst. **69**(3), 267–C277 (2012)
11. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature **401**, 788–791 (1999)
12. Hoyer, P.O.: Non-negative sparse coding. In: IEEE Workshop on Neural Networks for Signal Processing, Martigny, Switzerland, pp. 557–565 (2002)
13. Feng, T., Li, S.Z., Shum, H.Y., Zhang, H.J.: Local non-negative matrix factorization as a visual representation. In: Proceedings of the 2nd International Conference on Development and Learning, pp. 178–183 (2002)
14. Hoyer, P.O., Dayan, P.: Non-negative matrix factorization with sparseness constraints. J. Mach. Learn. Res. **5**, 1457–1469 (2004)

15. Montano, A.P., Carazo, J.M., Kochi, K., Lehmann, D., Pascual-Marqui, R.D.: Nonsmooth nonnegative matrix factorization (nsNMF). IEEE Trans. Pattern Anal. **28**(3), 403–415 (2006)
16. Jenatton, R., Obozinski, G., Bach, F.: Structured sparse principal component analysis. In: International Conference on Artificial Intelligence and Statistics (AISTATS) (2010)
17. Peharz, R., Pernkopf, F.: Sparse nonnegative matrix factorization with ℓ_0-constraints. Neurocomputing **80**, 38–46 (2012)
18. Zheng, W.S., Lai, J.H., Liao, S.C., He, R.: Extracting non-negative basis images using pixel dispersion penalty. Pattern Recogn. **45**(8), 2912–2926 (2012)
19. Cai, D., He, X., Han, J.: Locally consistent concept factorization for document clustering. IEEE Trans. Knowl. Data Eng. **23**(6), 902–913 (2011)
20. Ding, C.H., Li, T., Jordan, M.I.: Convex and semi-nonnegative matrix factorizations. IEEE Trans. Pattern Anal. Mach. Intell. **32**(1), 45–55 (2010)
21. Ahn, J.H., Choi, S., Oh, J.: A multiplicative up-propagation algorithm. In: Proceedings of the 21st International Conference on Machine Learning, p. 3 (2004)
22. Song, H.A., Kim, B.K., Xuan, T.L., Lee, S.Y.: Hierarchical feature extraction by multi-layer non-negative matrix factorization network for classification task. Neurocomputing **165**, 63–74 (2015)
23. Trigeorgis, G., Bousmalis, K., Zafeiriou, S., Schuller, B.W.: A deep matrix factorization method for learning attribute representations. IEEE Trans. Pattern Anal. Mach. Intell. **39**(3), 417–429 (2017)
24. Lyu, B., Xie, K., Sun, W.: A deep orthogonal non-negative matrix factorization method for learning attribute representations. In: Liu, D., Xie, S., Li, Y., Zhao, D., El-Alfy, E.S. (eds.) ICONIP 2017. LNCS, vol. 10639, pp. 443–452. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70136-3_47
25. Guo, Z., Zhang, S.: Sparse deep nonnegative matrix factorization (2017). http://arxiv.org/abs/1707.09316
26. Yu, J., Zhou, G., Cichocki, A., Xie, S.: Learning the hierarchical parts of objects by deep non-smooth nonnegative matrix factorization (2018). http://arxiv.org/abs/1803.07226
27. Xue, H., Dai, X., Zhang, J., Huang, S., Chen, J.: Deep matrix factorization models for recommender systems. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence, Melbourne, Australia, pp. 3203–3209 (2017)
28. Zhao, H., Ding, Z., Fu Y.: Multi-view clustering via deep matrix factorization. In: Thirty-First AAAI Conference on Artificial Intelligence, pp. 2921–2927 (2017)
29. Flenner, J., Hunter, B.: A deep non-negative matrix factorization neural network (2017)
30. Le Roux, J., Hershey, J.R., Weninger, F.: Deep NMF for speech separation. In: IEEE International Conference on Acoustics, Speech and Signal Processing, South Brisbane, Australia, pp. 66–70 (2015)
31. Fan, J., Cheng, J.: Matrix completion by deep matrix factorization. Neural Netw. **98**, 34–41 (2017)
32. Xu, L., Yu, B., Zhang, Y.: An alternating direction and projection algorithm for structure-enforced matrix factorization. Comput. Optim. Appl. **68**(2), 333–362 (2017). https://doi.org/10.1007/s10589-017-9913-x

33. Xu, L., Zhou, Y., Yu, B.: Classification and clustering via structure-enforced matrix factorization. In: Sun, Y., Lu, H., Zhang, L., Yang, J., Huang, H. (eds.) IScIDE 2017. LNCS, vol. 10559, pp. 403–411. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-67777-4_35
34. Glowinski, R., Marroco, A.: Sur lapproximation, par elements finis dordre un, et la resolution, par penalisation-dualite dune classe de problemes de dirichlet non lineaires. Revue francaise dautomatique, informatique, recherche operationnelle. Analyse numerique **9**(2), 41–76 (1975)
35. Gabay, D., Mercier, B.: A dual algorithm for the solution of nonlinear variational problems via finite element approximation. Comput. Math. Appl. **2**(1), 17–40 (1976)