An efficient ADMM-type algorithm for deep semi-nonnegative matrix factorization

Yijia Zhou\(^1\) and Lijun Xu\(^{2,3}\)

\(^1\)School of Computer & Software, Dalian Neusoft University of Information, Dalian 116023, Liaoning, China
\(^2\)School of Science, Dalian Maritime University, Dalian 116026, Liaoning, China
\(^3\)Email: lijun_xu@dlmu.edu.cn

Abstract. In this paper, we focus on deep semi-nonnegative matrix factorization (DSemiNMF) which has a wider application in the real world than traditional NMF. We propose an efficient algorithm based on the classic alternating direction method of multipliers (ADMM) for DSemiNMF. By utilizing structures in DSemiNMF, we derive an efficient updating rule for updating subproblems according to its KKT conditions. Numerical experiments are conducted to compare the proposed algorithm with state-of-the-art deep semi-NMF algorithm. Results show that our algorithm performs better and the deep model indeed results in better clustering accuracy than single-layer model.

1. Introduction
Matrix factorization techniques have found particularly useful in various data-related applications, such as in traditional signal and image processing and in modern machine learning tasks, primarily because they often help interpret latent properties in data analysis. In recent years, Non-negative Matrix Factorization (NMF) which decomposing a nonnegative data matrix \(M\) into nonnegative factors \(Z\) and \(H\) is a widely-used method for finding meaningful features of data. NMF has been proven great useful in dimension reduction of large scaled signals, images, text data, etc. The family of NMF has been successfully applied to numerous 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 a number of variant forms, allowing for different 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. Semi-Non-negative Matrix Factorization (SemiNMF) [12], 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 SemiNMF 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, deep NMF models (see [13-21], for example) 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 (or 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 [16], image clustering [17], speech separation [18]. However, these models are only designed for specific problems with certain intuitive structures. In this paper, we focus on one type of deep matrix factorization models, i.e. deep semi-NMF. A specific algorithm for deep semi-NMF is further studied and can be applied to data representation.

The main contributions of this paper contain that we design an ADMM algorithm for the deep (or multi-layer) semi-NMF problem via the approach of variable splitting and propose an efficient way to solve each subproblem in the ADMM framework.

This paper is organized as follows. In Sect. 2, we introduce semi-NMF and deep semi-NMF and related algorithmic frameworks. In Sect. 3, we propose a new ADMM-type algorithm for solving DsemiNMF. Section 4 contains numerical experiments comparing the proposed algorithm with two existing algorithms for DsemiNMF and single-layer matrix factorization on MNIST digit dataset. Finally, we conclude this paper in Sect. 5.

2. Semi-NMF and Deep semi-NMF

In this section, we present a brief overview of the semi-NMF and deep semi-NMF. Assuming that the input data is provided in a matrix form $M \in \mathbb{R}^{p \times n}$ containing a collection of $n$ data vectors as columns, general matrix factorization decomposes $M$ into two factors $Z \in \mathbb{R}^{p \times k}$ and $H \in \mathbb{R}^{k \times n}$. NMF is a class of constrained matrix factorization assuming both data matrix and factors to be nonnegative, and has been known very well as follows,

$$\min_{Z,H} \frac{1}{2} \| M - ZH \|_F^2 \quad \text{s.t.} \quad Z \succeq 0, \ H \succeq 0,$$

where $\| \cdot \|_F$ is Frobenius norm. That is, NMF aims at finding factors of nonnegative $M$ that satisfy nonnegative constraints. The application area of NMF has grown significantly during the past years. However, when data matrix $M$ is not strictly nonnegative, the factors may not be nonnegative either. Later on, researchers extended the applicability of NMF by imposing different constraints on two factors with certain properties. In this work, we only focus on one of these variants, semi-NMF introduced in [12], that imposes nonnegativity constraints only on the second factor $H$, but allows mixed signs in both the data matrix $M$ and the first factor $Z$.

2.1. Semi-NMF

Ding et al. [12] introduced semi-NMF that relaxes the non-negativity constraints on $Z$ and hence on the data matrix $M$ like factorizations of the form $M^\pm \approx Z^\pm H^\pm$. This form can be viewed as the least square optimization problem of matrix approximation,

$$\min_{Z,H} \frac{1}{2} \| M - ZH \|_F^2 \quad \text{s.t.} \quad H \succeq 0.$$

Their motivation was based on the case of clustering with $Z$ representing the cluster centers and $H$ denoting the cluster indicators. In fact, if $H$ was imposed not only non-negative but also every column vector possessing only one positive element, then semi-NMF can be proved equivalent to well-known k-means clustering problems.

An iterative updating algorithm for semi-NMF was proposed in [12], i.e., alternatively updating $Z$ and $H$:

$$\begin{cases} Z \leftarrow MH(H^TH)^{-1} \quad \text{or} \quad MH^T, \\
H \leftarrow H \odot \frac{[z^TM]^+[z^TZ]^+H}{[z^TM]^+[z^TZ]^+H^+}, \end{cases}$$

\(3\)
where $H^\dagger$ is the Moore-Penrose pseudo-inverse of $H$, and $[A]^+([A]^-)$ is a matrix that has the negative (positive) elements of matrix $A$ replaced with 0, and $\bigodot$ denotes component multiplications. These updating formulas (3) are very popular in applications of semi-NMF, such as semi-NMF PCA [22-23], max-margin semi-NMF [24], graph regularized semi-NMF [25], speech emotion recognition [26]. Moreover, most usage of semi-NMF is to generate more representative feature expressions and a low-dimensional representation, then do data classification or clustering based on the feature representation.

2.2. Deep semi-NMF

In semi-NMF the model (2) is a single-layer matrix factorization only to construct one representation between the original data and its lower-dimensional basis. In many cases the real-world data is often rather complex and has a collection of distinct, often unknown, features. In the work of [15] for example, human faces data was studied where the variability in the data does not only stem from the difference in the appearance of the subjects, but also from other attributes. As demonstrated in [15], the multi-attribute nature of real data calls for a hierarchical framework that is better at representing it than a shallow semi-NMF. We focus on one popular class of deep models: deep semi-NMF.

Note that Cichocki and Zdunek have proposed a simple and primary multilayer NMF [15–17] to improve the performance of blind signal separation (BSS). The idea is performing the basic decomposition (1) firstly to $M$, then to each $H_i^+$ sequentially. The process can be repeated arbitrary many times until some stopping criteria are satisfied. Later on, this idea is popularized to “deep” versions of NMF variants. To make it more intuitive, take the concentrated deep semi-NMF for example, one can write the process into the following factorizations:

\[
    M \approx Z_1 H_1^+, \\
    H_1^+ \approx Z_2 H_2^+, \\
    \vdots \\
    H_{m-1}^+ \approx Z_m H_m^+,
\]

where $\{H_i^+\}_{i=1}^m$ are nonnegative. Most multi-layer and deep matrix factorizations are derived from the formulation (4), actually a sequential decomposition framework. In fact, (3) equivalents to the following multi-factors matrix decomposition:

\[
    M \approx Z_1 Z_2 \cdots Z_m H_m^+.
\]

To measure the approximation of the above factorization, the least square cost function is adopted by most researchers. As in most work, we consider the following deep semi-NMF model

\[
    \min_{Z_i, H_m} \frac{1}{2} \| M - Z_1 Z_2 \cdots Z_m H_m \|_F^2 \quad \text{s.t.} \quad H_m \succeq 0,
\]

where $Z_i \in \mathbb{R}^{p \times k_i}$, $(Z_i \in \mathbb{R}^{l_i \times k_i})_{i=2}^m$, $H_m \in \mathbb{R}^{k_m \times n}$.

There are two main kinds of algorithms for deep semi-NMF according to formulations (4) and (6). Firstly, most algorithms are designed by solving (4) layer by layer, that is applying iterative updating algorithm (3) to each approximate semi-NMF in (4). However, these algorithms are inefficient since the factor matrices in former layers are useless for subsequent layer factorizations. In other words, these works do not consider to update the factors globally and not the mechanism to keep the good fit of model to the original data, see [13-14] for example. Later works in [15-21] improve the above algorithms by utilizing layer by layer technique as initialization or pre-training, then fine-tuning all layers (factors) by alternating updating factor matrices sequentially.

3. A novel ADMM-type algorithm for deep semi-NMF

As introduced in the work [27-29], alternating direction and projection methods solve single layer and deep structure-enforced matrix factorization efficiently. Motivated by these algorithms, we propose a novel way to tackle deep semi-NMF model (6). To facilitate an efficient use of alternating minimization, we introduce an auxiliary variable $V_m$ in order to separate $H_m$ from nonnegative constraints. Consider the following model equivalent to (6),

\[
    \min_{\{Z_i\}_{i=1}^m, H_m, V_m} \frac{1}{2} \| M - Z_1 Z_2 \cdots Z_m H_m \|_F^2 \quad \text{s.t.} \quad H_m - V_m = 0, V_m \succeq 0,
\]
where $V_m$ have the same dimension size with $H_m$. The augmented Lagrangian function of (7) is

$$L_A((Z_i)_{i=1}^m, H_m, V_m, \Pi) = \frac{1}{2} \| M - Z_1 Z_2 \cdots Z_m H_m \|_F^2 + \Pi \cdot (H_m - V_m) + \frac{\beta}{2} \| H_m - V_m \|_F^2,$$

(8)

where $\Pi$ is the Lagrange multiplier with equal-size of $H_m$, and $\beta \geq 0$ is the penalty parameter. Note that the scalar product “•” of X and Y is the sum of all element-wise products, i.e., $X \cdot Y = \sum_{i,j} X_{ij} Y_{ij} = \text{trace}(X, Y)$.

The alternating direction method of multiplier (ADMM)[30,31] for (8) is derived by successively minimizing the augmented Lagrange function $L_A$ with respect to $(Z_i)_{i=1}^m, H_m$ and $V_m$, one at a time while fixing others at their most recent values, and then updating the multiplier after each sweep of such alternating minimization. The introduction of the auxiliary variable $V_m$ makes it easy to carry out each of the alternating minimization steps. Specifically, these steps can be written in the following forms,

$$Z_i^+ = \arg\min_{Z_i} L_A((Z_i)_{i=1}^m, H_m, V_m, \Pi), j = 1, 2, \cdots, m,$$

(9)

$$H_m^+ = \arg\min_{H_m} L_A((Z_i)_{i=1}^m, H_m, V_m, \Pi),$$

(10)

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

(11)

$$\Pi^+ = \Pi + \beta(H_m^+ - V_m^+).$$

(12)

where $\mathcal{P}_+$ stands for the projection onto the nonnegative matrix in Frobenius norm, and the superscript “+” is used to denote iterative values at the new iteration.

We use the following practical stopping criterion: for given tolerance $\text{tol} > 0$, $|f_{k-1} - f_k|/|f_k| \leq \text{tol}$, where $f_k = \| X - Z_1^k Z_2^k \cdots Z_m^k H_m^k \|_F^2$, $Z_i^k$ 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.

### 3.1. 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 (9) actually can be rewritten to an unconstraint optimization as

$$\min_{Z_j} \frac{1}{2} \| M - \Phi_j Z_j \Psi_j \|_F^2,$$

(13)

where $\Phi_j = Z_1 Z_2 \cdots Z_{j-1}$ and $\Psi_j = Z_{j+1} \cdots Z_m H_m$. By setting $\frac{\partial \mathcal{L}}{\partial Z_j} = \Phi_j^T \Phi_j Z_j \Psi_j \Psi_j^T - \Phi_j^T M \Psi_j^T = 0$, we get the $Z_j$ updates:

$$Z_j = (\Phi_j^T \Phi_j)^{-1} \Phi_j^T M \Psi_j^T (\Psi_j \Psi_j^T)^{-1}.$$

(14)

### 3.2. Updating rule for $H_m$

Similarly, we can formulate the update rule for $H_m$ by (10) which is also an unconstraint minimization, namely,

$$\min_{H_m} \frac{1}{2} \| M - \Phi H_m \|_F^2 + \Pi \cdot (H_m - V_m) + \frac{\beta}{2} \| H_m - V_m \|_F^2,$$

(15)

where $\Phi = Z_1^+ Z_2^+ \cdots Z_m^+$. We set $\frac{\partial \mathcal{L}}{\partial H_m} = \Phi^T M + \beta V_m - \Pi + \Phi^T \Phi H_m + \beta H_m = 0$, which gives us the $H_m$ updates:

$$H_m^+ = (\Phi^T \Phi + \beta I)^{-1} (\Phi^T M + \beta V_m - \Pi).$$

(16)

### 3.3. Updating rule for $V_m$ and $\Pi$

The updating rule for the variable $V_m$ in (11) actually is the easy projection onto the nonnegative matrix, that is, $\mathcal{P}_+(X) = \max(0, X)$. As for $\Pi$, it’s just the usual update of Lagrange multiplier.

Note that most existing algorithms for deep variant NMF utilize (3) to update nonnegative factors by component-wise multiplications and divisions. The dominant computational tasks of existing algorithms at each iteration are the matrix multiplications. However, (3) actually is one step iterative solution approximately updating of $H$. Obviously, we can guess these algorithms may need a great
many iterations to converge. In contrary, we solve subproblems (9) and (10) more precisely in each iteration and hopefully get the solution quickly. This is demonstrated in next section.

4. Experiments
We compare the proposed algorithm with algorithms for deep semi-NMF in recent works [15] and [29]. In [15], an algorithm was designed to solve original model (6) and its updating rule for \( H_t \) is

\[
H_t \leftarrow H_t \bigodot \frac{[\Phi^T M + [\Phi^T \Phi] - H_t]}{[\Phi^T M]^{-1} + [\Phi^T \Phi]^{-1}}.
\]

which is the directly extension of updating scheme (3) for semi-NMF (single layer). In the work [29], an altering direction algorithm for a class of deep structure-enforced NMF was proposed which can be used to solve the deep semi-NMF problem. The key techniques of updating \( Z_t \) and \( H_m \) in [29] is approximately solve \( Z_t \) and \( H_m \) subproblems by iteratively component-wise multiplications and divisions instead of involving inverse matrices. Specifically,

\[
(Z^+)_{ik} = (Z_{ik})^{\frac{(\Phi^T M^T W + \alpha \beta_j - A_j)_{ik}}{(\Phi^T \Phi + \alpha \beta_j)_{ik}}} (H^+)_{ik} = (H_{ik})^{\frac{(\Phi^T M + \beta \nu_m - \Pi)_{ik}}{(\Phi^T \Phi + \beta H_m)_{ik}}}.
\]

4.1. Datasets and details
In this section, the widely used handwritten digit database MNIST [32] is used to evaluate the performance. As in [15,29], we opt the digits from 0 to 4 constitute a matrix \( \mathbf{M} \) is decomposed by deep (or single) semi-NMF algorithms. Then we cluster columns of the final \( H_m \) as the same approach as in [15]. One popular metric of the result is the clustering accuracy (AC) defined by \( \sum_{i=1}^n \delta(\gamma_i, \text{map}(\hat{y}_i))/N \), where \( \hat{y} \) is the cluster set of predictions, \( y \) is ground truth labels, \( \text{map} \) is a permutation function that maps each predicted cluster id to each corresponding ground truth. We calculate RMSE by \( \| M - Z_1Z_2 \cdots Z_nH_m \|_F \) to measure the objective value.

In this test, we choose 3 layers and dimension of each layer is 300, 15 and 50. Besides, set the maximum number of iteration \( \text{maxiter} = 500 \) and tolerance \( \text{tol} = 1e-6 \). We denote DsemiNMF_new, DsemiNMF_[15] and DsemiNMF_[29] as algorithms in this paper, [15] and [29], respectively. Note that we set the penalty parameter in (15) to the suitable value referring to the work in [27]. In addition, in the beginning of all algorithms, the factor matrices are initialized by pre-decomposing \( \mathbf{M} \) through solving (4) and denote as DsemiNMF_pre. As a contrast, we also compare with the single-layer algorithm (SemiNMF) in [27] by setting \( Z \in \mathbb{R}^{784 \times 50} \) and \( H \in \mathbb{R}^{50 \times 5139} \).

4.2. Results
We implement each algorithm on 50 trials under Matlab version 9.4 (R2018a) on a PC with an Intel Core i9 processor at 3.3GHz with 32GB RAM. The average results are presented in Table 1. It can been seen that the proposed algorithm performs well both in accuracy and in time consuming compared with algorithms in [15] and [29]. We should note that, for the same 3 layers model, lower RMSE is consistent with higher AC, namely the better algorithm. Remarkably, our algorithm need much less running time than the other deep algorithms. In addition, the result of single layer case is listed in the right column in Table 1. This is because the residue of 2-factors matrix factorization usually is less than the case of multi-factors, although 2-factors matrix factorization generally performs not as good as muti-factors matrix factorization in data representations.

| Method | DsemiNMF_new | DsemiNMF_[15] | DsemiNMF_[29] | DsemiNMF_pre | SemiNMF |
|--------|--------------|---------------|---------------|--------------|---------|
| AC     | 0.509        | 0.474         | 0.439         | 0.221        | 0.315   |
| Time(s)| 32.67        | 66.57         | 130.83        | 18.39        | 9.56    |
| RMSE   | 37.3676      | 37.3816       | 37.4057       | 42.2140      | 24.0150 |
Next, we compare three deep semi-NMF algorithms on iterations. Figure 1 shows RMSE with respect to each iteration of three algorithms, respectively. From Figure 1, we can clearly see that our algorithm outperforms the other two algorithms. The new algorithm only needs about 30 iterations which is much less than the algorithms in [29] (about 150 iterations) and in [15] (about 300 iterations).

The above results are consistent with our guess in section 3. This is because we get much more precise solutions in each iteration. Another point should be noted that solving (14) and (16) is actually not expensive since the inverse matrices usually have small size. It will be evident that our proposed algorithm is efficient for this class of deep semi-NMF.

![Figure 1. RMSE vs iterations comparison with different deep semi-NMF algorithms.](image)

5. Conclusion and future
In this work, we present a novel algorithm based on ADMM framework for solving deep semi-NMF model. Different from most existing methods which directly deal with by whether layer by layer or deep way, we separate factor matrix from constraints by introducing an auxiliary matrix. Deep semi-NMF can be transformed to an ADMM-applicable model and be easily implemented. Numerical experiments show the efficiency and applicability of the proposed algorithm for data representing. One of future works will be theoretically demonstrating some convergence results on the proposed algorithm. Besides, discussing how different decomposed dimension would affect clustering performance is another interesting work.

Acknowledgments
This work has been supported in part by Natural Science Foundation of Liaoning Province of China (NO. 2019-BS-013) and in part by the Fundamental Research Funds for the Central Universities (NO. 3132020171).

References
[1] Paatero P and Tapper U 1994 Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values *Environmetrics* 5(2) pp 111–126
[2] Brunet J P, Tamayo P, Golub, T R and Mesirov J P 2004 Metagenes and molecular pattern discovery using matrix factorization. *PNAS*. 101(12) pp 4164–4169
[3] Devarajan K 2008 Nonnegative matrix factorization: an analytical and interpretive tool in computational biology *PLoS computational biology* 4 (7) e1000029
[4] Berry M W and Browne M 2005 Email surveillance using nonnegative matrix factorization
Computational & Mathematical Organization Theory 11(3) pp 249–264

[5] Zafeiriou S, Tefas A, Buciu I and Pitas I 2006 Exploiting discriminant information in NMF with application to frontal face verification TNN. 17(3) pp 683–695

[6] Kotsia I, Zafeiriou S and Pitas I 2007 A novel discriminant nonnegative matrix factorization algorithm with applications to facial image characterization problems TIFS. 2 pp 588–595

[7] Zdunek R and Cichocki A 2006 Non-negative matrix factorization with quasi-Newton optimization Proc. 8th Int. Conf. on Artificial Intelligence and Soft Computing pp 870–879

[8] Wang W, Li S, Qi H, Ayhan B, Kwan C and Vance S 2015 Identify anomaly component by sparsity and low rank IEEE Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensor (WHISPERS, Tokyo, Japan)

[9] Qu Y, Guo R, Wang W, Qi H, Ayhan B, Kwan C and Vance S 2016 Anomaly detection in hyperspectral images through spectral unmixing and low rank decomposition IEEE International Geoscience and Remote Sensing Symposium (IGARSS, Beijing) pp 1855-1858

[10] Weninger F and Schuller B 2012 Optimization and parallelization of monaural source separation algorithms Journal of Signal Processing Systems 69(3) pp 267-277

[11] Lee D D and Seung H S 1999 Learning the parts of objects by non-negative matrix factorization Nature 401 pp 788–791

[12] Ding C H, Li T and Jordan M I 2010 Convex and semi-nonnegative matrix factorizations IEEE Trans. Pattern Anal. Mach. Intell. 32(1) pp 45–55

[13] Ahn J H, Choi S and Oh J 2004 A multiplicative up-propagation algorithm Proc. 21th Int. Conf. on Machine Learning pp 3

[14] Song H A, Kim B K, Xuan T L and Lee S Y 2015 Hierarchical feature extraction by multi-layer NMF network for classification task Neurocomputing 165 pp 63–74

[15] Trigeorgis G, Bousmalis K, Zafeiriou S and Schuller B W 2017 A deep matrix factorization method for learning attribute representations IEEE T. Pattern Anal. 39(3) pp 417–429

[16] Xue H, Dai X, Zhang J, Huang S and Chen J 2017 Deep Matrix Factorization Models for Recommender Systems. Proc. 26th Int. Conf. on Artificial Intelligence pp 3203–3209

[17] Zhao H, Ding Z and Fu Y 2017 Multi-View Clustering via Deep Matrix Factorization 31th AAAI Conf. on Artificial Intelligence pp 2921–2927

[18] Le Roux J, Hershey J R and Weninger F 2015 Deep NMF for speech separation IEEE Int. Conf. on Acoustics, Speech and Signal Processing (South Brisbane, Australia) pp 66–70

[19] 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. 9(5) pp 435

[20] Yang G, Li H, Yang W and Emery W 2018 Deep Semi-Nonnegative Matrix Factorization Based Unsupervised Change Detection of Remote Sensing Images IEEE International Geoscience and Remote Sensing Symposium (Valencia) pp 4917-4920.

[21] Imakura A, Inoue Y, Sakurai T and Futamura Y 2018 Parallel Implementation of the Nonlinear Semi-NMF Based Alternating Optimization Method for Deep Neural Networks. Neural Process Lett. 47 pp 815–827

[22] Allab K, Labiod L and Nadif M 2015 A Semi-NMF-PCA Unified Framework for Data Clustering IEEE Transactions on Knowledge and Data Engineering 29(1) pp 2–16

[23] Allab K, Labiod L and Nadif M 2015 Simultaneous Semi-NMF and PCA for Clustering IEEE Int. Conf. on Data Mining (Atlantic City, NJ) pp 679-684 doi: 10.1109/ICDM.2015.66

[24] Vijay B G, Kumar P and Irene K 2011 Max-Margin Semi-NMF. Proc. British Machine Vision Conf. (BMVA Press) 129 pp 1-11 http://dx.doi.org/10.5244/C.25.129

[25] Luo P and Peng J 2017 Group Sparsity and Graph Regularized Semi-Nonnegative Matrix Factorization with Discriminability for Data Representation Entropy 19(12) pp 627

[26] Song P, Ou S, Zhang X, Jin Y, Zheng W, Liu J and Yu Y 2016 Transfer Semi-Supervised NMF for Speech Emotion Recognition IEICE Trans. on Inform. & Syst. 99(10) pp 2647-2650

[27] Xu L, Yu B and Zhang Y 2017 An Alternating Direction and Projection Algorithm for Structure-enforced Matrix Factorization Comput. Optim. Appl. 68(2) pp 333–362
[28] Xu L, Zhou Y and Yu B 2017 Classification and Clustering via Structure-enforced Matrix Factorization. *Lecture Notes in Computer Science* (Springer Verlag) **10559** pp 403–411
[29] Zhou Y and Xu L 2018 A Deep Structure-Enforced Nonnegative Matrix Factorization for Data Representation *Lecture Notes in Computer Science* (Springer Verlag) **11258(3)** pp 340–350
[30] Glowinski R and Marroco A 1975 Sur l’approximation, par éléments finis d’ordre un, et la résolution, par pénalisation-dualité d’une classe de problèmes de Dirichlet non linéaires. *Revue Francaise d’Automatique, Informatique et Recherche Opérationelle* **9** pp 41–76.
[31] Gabay D and Mercier B 1976 A dual algorithm for the solution of nonlinear variational problems via finite element approximation *Comput. Math. Appl.* **2(1)** pp 17–40
[32] LeCun Y, Bottou L, Bengio Y and Haffner P 1998 Gradient-based learning applied to document recognition *Proceedings of the IEEE* **86(11)** pp 2278-2324