



HAL
open science

Fast Nonnegative Matrix Factorization and Completion Using Nesterov Iterations

Clément Dorffer, Matthieu Puigt, Gilles Delmaire, Gilles Roussel

► **To cite this version:**

Clément Dorffer, Matthieu Puigt, Gilles Delmaire, Gilles Roussel. Fast Nonnegative Matrix Factorization and Completion Using Nesterov Iterations. 13th International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA 2017), Feb 2017, Grenoble, France. pp.26-35, 10.1007/978-3-319-53547-0_3 . hal-01469366

HAL Id: hal-01469366

<https://hal.science/hal-01469366>

Submitted on 3 Dec 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Fast Nonnegative Matrix Factorization and Completion using Nesterov Iterations

Clément Dorffer, Matthieu Puigt, Gilles Delmaire, and Gilles Roussel

Univ. Littoral Côte d’Opale, LISIC – EA 4491, F-62228 Calais, France
clement.dorffer@univ-littoral.fr, matthieu.puigt@univ-littoral.fr
gilles.delmaire@univ-littoral.fr, gilles.roussel@univ-littoral.fr

Abstract. In this paper, we aim to extend Nonnegative Matrix Factorization with Nesterov iterations (Ne-NMF)—well-suited to large-scale problems—to the situation when some entries are missing in the observed matrix. In particular, we investigate the Weighted and Expectation-Maximization strategies which both provide a way to process missing data. We derive their associated extensions named W-NeNMF and EM-W-NeNMF, respectively. The proposed approaches are then tested on simulated nonnegative low-rank matrix completion problems where the EM-W-NeNMF is shown to outperform state-of-the-art methods and the W-NeNMF technique.

Keywords: Low-rank matrix completion; nonnegative matrix factorization; Nesterov iterations; gradient descent

1 Introduction

Estimating missing entries in a low-rank matrix—known as Matrix Completion (MC)—received a lot of attention since some pioneering work in [8, 2]. Indeed, such a problem finds many applications [1] including environmental monitoring [17, 26] for example. Mathematically, it consists of estimating a matrix $M \in \mathbb{R}^{m \times n}$ —partially known on a subset Ω of entries—by a low-rank matrix X , i.e.,

$$\min_X \text{rank}(X) \quad \text{s.t.} \quad \mathcal{P}_\Omega(X) = \mathcal{P}_\Omega(M), \quad (1)$$

where $\mathcal{P}_\Omega(\cdot)$ is the sampling operator of M .

Among the many approaches recently proposed to solve MC, matrix factorization techniques emerged as a powerful tool, especially when the rank of the data matrix M is known [11, 31]. Moreover in some applications—e.g., blind sensor calibration [5, 6]— M is nonnegative, thus yielding a Nonnegative MC (NMC) problem which can be tackled by Nonnegative Matrix Factorization (NMF). NMF is widely used in, e.g., signal processing or machine learning, and consists of finding two nonnegative matrices $G \in \mathbb{R}_+^{m \times r}$ and $F \in \mathbb{R}_+^{r \times n}$ such that

$$X \approx G \cdot F. \quad (2)$$

Most NMF algorithms consider two separate convex sub-problems that respectively and alternately estimate G for a fixed F and F for a fixed G . Such estimates are obtained using, e.g., Multiplicative Updates (MU) [19], Alternating Nonnegative Least Squares (ANLS) [15], or Projected Gradient (PG) [22]. Additionally, some authors incorporated some extra-information in the NMF model [30], e.g., weights [13, 10], sum-to-one constraints [18, 21, 4], sparsity assumptions [14, 6, 12], or known and/or bounded values [22, 3, 20, 21, 5, 6, 12]. With the Big Data era, computational time reduction of NMF is particularly investigated, e.g., through optimal solvers [9], distributed strategies [23], online estimation [24], or randomization [33, 29].

When applied to a data matrix M which is partially known on a subset Ω of entries, NMF consists of estimating both matrices G and F which satisfy

$$\min_{G, F \geq 0} \|\mathcal{P}_\Omega(M) - \mathcal{P}_\Omega(G \cdot F)\|_{\mathcal{F}}, \quad (3)$$

where $\|\cdot\|_{\mathcal{F}}$ is the Frobenius norm. Solving Eq. (3) is usually done either by applying Weighted NMF (WNMF) [13, 10] to M or by using an Expectation-Maximization (EM) strategy [32, 16]. However, these approaches use MU [32, 13] or ANLS iterations [16], and are limited to relatively small-sized matrices. As NMC may be applied to very large-scale problems, efficient WNMF methods must be proposed. As a consequence, in this paper we investigate NMF with Nesterov iterations (Ne-NMF) [9] that we propose to extend to the NMC problem. Indeed, NeNMF is improving PG-NMF [22] by replacing the line search of the optimal step in the update rules—known to be time consuming—by Nesterov iterations [25]. NeNMF is thus much faster than MU-NMF or PG-NMF¹. Moreover, NeNMF can be easily extended to semi-NMF, which is of particular interest in some sensor calibration problems [7]. We thus propose to extend the NeNMF algorithm to deal with missing entries in a nonnegative data matrix M . In particular, we discuss the respective advantages and drawbacks of extensions using the above weighted and EM strategies. We experimentally show that the latter outperforms both the state-of-the-art methods and the proposed weighted extension in simulations of NMC problems.

The remainder of the paper is organized as follows. We recall the principles of the NeNMF method in Section 2. We then extend such an approach to the case of an observed matrix with missing entries in Section 3. The enhancement of the proposed methods is investigated in Section 4 while we conclude and discuss about future work in Section 5.

2 Standard NeNMF

As explained above, for a fixed $m \times n$ nonnegative data matrix X , NMF consists of finding both the $m \times r$ and $r \times n$ matrices G and F which provide the best

¹ See for example the CPU-time consumptions of [27] at <https://github.com/andrewssobral/lrslibrary>.

low-rank approximation of X , as shown in Eq. (2). The NeNMF method [9] iteratively and alternately updates G and F . For that, it solves

$$F = \arg \min_{\tilde{F} \geq 0} \mathcal{J}(G, \tilde{F}) = \arg \min_{\tilde{F} \geq 0} \frac{1}{2} \cdot \left\| X - G \cdot \tilde{F} \right\|_{\mathcal{F}}^2, \quad (4)$$

and

$$G = \arg \min_{\tilde{G} \geq 0} \mathcal{J}(\tilde{G}, F) = \arg \min_{\tilde{G} \geq 0} \frac{1}{2} \cdot \left\| X - \tilde{G} \cdot F \right\|_{\mathcal{F}}^2, \quad (5)$$

respectively, by applying in an inner loop the Nesterov accelerated gradient descent [25]. In order to update a factor, say F , the latter initializes $Y_0 \triangleq F^t$ —where t is an NeNMF outer iteration—and considers a series α_k defined as

$$\alpha_0 = 1, \quad \alpha_{k+1} = \frac{1 + \sqrt{4\alpha_k^2 + 1}}{2}, \quad \forall k \in \mathbb{N}. \quad (6)$$

For each inner loop index k , the Nesterov gradient descent then computes

$$F_k = \left(Y_k - \frac{1}{L} \nabla_F \mathcal{J}(G, Y_k) \right)^+, \quad (7)$$

and

$$Y_{k+1} = F_k + \frac{\alpha_k - 1}{\alpha_{k+1}} (F_k - F_{k-1}), \quad (8)$$

where $(\cdot)^+$ is the operator which projects any negative entries to zero, $\nabla_F \mathcal{J}(G, \cdot)$ is the gradient of $\mathcal{J}(G, \cdot)$ in Eq. (4), and L is a Lipschitz constant equal to

$$L = \left\| G \cdot G^T \right\|_2 = \|G\|_2^2, \quad (9)$$

where $\|\cdot\|_2$ is the spectral norm. Nesterov gradient descent thus performs a single step of gradient to go to F_k and then slides it in the direction of F_{k-1} to derive Y_{k+1} . Using the Karush-Kuhn-Tucker (KKT) conditions, a stopping criterion—considering both a maximum number Max_{iter} of iterations and a gradient bound—is proposed in [9], thus yielding $F^{t+1} = Y_K$, where Y_K is the last iterate of the above inner iterative gradient descent. In practice, the gradient descent stops if $K = \text{Max}_{\text{iter}}$ or if

$$\left\| \nabla_F^+ \mathcal{J}(G^t, Y_K) \right\|_{\mathcal{F}} \leq \epsilon \cdot \left\| \left[\nabla_G^+ \mathcal{J}(G^1, F^1), \nabla_F^+ \mathcal{J}(G^1, F^1) \right] \right\|_{\mathcal{F}}, \quad (10)$$

where ϵ is a user-defined threshold, $\nabla_F^+ \mathcal{J}(G^t, \cdot)$ is the projected gradient of $\mathcal{J}(G^t, \cdot)$, and $\mathcal{J}(G^1, F^1)$ is the value of the cost function computed at the outer loop initialization. The same strategy is applied to G .

3 Extending NeNMF to missing entries

3.1 Weighted extension of NeNMF

As explained in Section 1, two main strategies exist to process missing data in matrix factorization, i.e., using weights or an EM framework. In the weighted

strategy, a binary matrix W is included in the NMF formalism. The (i, j) -th entry in W is defined as

$$W(i, j) \triangleq \begin{cases} 1 & \text{if } (i, j) \in \Omega, \\ 0 & \text{otherwise.} \end{cases} \quad (11)$$

Combining Eqs. (3) and (11), WNMF for NMC considers the following problem:

$$\min_{\tilde{G}, \tilde{F} \geq 0} \left\| W \circ (X - \tilde{G} \cdot \tilde{F}) \right\|_{\mathcal{F}}^2, \quad (12)$$

which is solved using the same splitting strategy as for NMF, e.g., for F :

$$F = \arg \min_{\tilde{F} \geq 0} \mathcal{J}_W(G, \tilde{F}) = \arg \min_{\tilde{F} \geq 0} \frac{1}{2} \cdot \left\| W \circ (X - G \cdot \tilde{F}) \right\|_{\mathcal{F}}^2. \quad (13)$$

Lemma 1. *The cost function $\mathcal{J}_W(G, \cdot)$ is convex.*

Lemma 2. *In the case of binary weights—and assuming that any row and any column of W contains at least one nonzero element—the gradient of $\mathcal{J}_W(G, \cdot)$ is L -Lipschitz continuous, where L is defined in Eq. (9).*

The proofs are provided in Appendix A.

Lemmas 1 and 2 allow the use of Nesterov iterations, which alternately compute updates (7) and (8), except that the gradient $\nabla_F \mathcal{J}(G, Y_k)$ is here replaced by

$$\nabla_F \mathcal{J}_W(G, Y_k) = G^T \cdot (W^2 \circ (G \cdot Y_k - X)), \quad (14)$$

where $W^2 = W \circ W$ and \circ is the Hadamard product. The stopping criterion for the inner loop is the same as in Section 2 for the standard approach, except that $\mathcal{J}(\cdot, \cdot)$ is here replaced by $\mathcal{J}_W(\cdot, \cdot)$.

It should be noticed that weighted gradient optimization was also investigated in matrix factorization with no sign constraint [28]. In particular, the authors explain that many local minima of $\mathcal{J}_W(G, F)$ exist but can be avoided in practice by gradient descent methods with a well-chosen step size. As a consequence, the major risk of the proposed W-NeNMF might be to converge to a local minimum.

3.2 EM extension of NeNMF

As an alternative to the above weighted framework, the authors in [32] proposed an EM strategy which consists of an unweighted low-rank approximation of a weighted combination of a previously computed approximation. It turns out that for a given NMF iteration t , the EM problem aims to solve

$$\min_{G^{t+1}, F^{t+1} \geq 0} \left\| W \circ M + (\mathbb{1}_{m,n} - W) \circ (G^t \cdot F^t) - G^{t+1} \cdot F^{t+1} \right\|_{\mathcal{F}}^2, \quad (15)$$

where $\mathbb{1}_{m,n}$ is the $m \times n$ matrix of ones. Indeed, the term $(\mathbb{1}_{m,n} - W) \circ (G^t \cdot F^t)$ in Eq. (15) provides the expected missing values of M . The latter are derived

from the complete-data log-likelihood expression, with respect to the unknown data X and given the observed data matrix M and the current estimate of X , i.e., $G^t \cdot F^t$. Denoting $X^t \triangleq W \circ M + (\mathbb{1}_{m,n} - W) \circ (G^t \cdot F^t)$, Eq. (15) reads as a classical NMF problem, i.e.,

$$\min_{G^{t+1}, F^{t+1} \geq 0} \|X^t - G^{t+1} \cdot F^{t+1}\|_{\mathcal{F}}^2, \quad (16)$$

whose solution maximizes the previously computed likelihood [32]. As a consequence, the whole approach consists of:

E-step: Estimate the completed version X of M using

$$X \leftarrow W \circ M + (\mathbb{1}_{m,n} - W) \circ (G \cdot F). \quad (17)$$

M-step: Apply a standard NMF to update G and F using X instead of M .

The resulting EM extension of NeNMF then follows the same structure, except that the M-step runs Nesterov iterations instead of MU [32] or ANLS [16].

It should be noticed that the E-step is a key point in the method. Indeed, if the estimation of X is not accurate enough—which might happen in early NMF iterations or in noisy conditions—this might propagate inaccuracies in the estimation of G and F along iterations. We experimentally investigate such potential issues in Section 4.

Moreover, the authors in [32] state that the EM strategy combined with MU-NMF is less sensitive to initialization than MU-WNMF. However, they notice that MU-WNMF is faster than EM-NMF in their tests, because they run many MUs at each M-step. When applied to Nesterov iterations, such a strategy has to be rethought. As a consequence, we investigated in preliminary tests the number of Nesterov outer loops processed at each M-stage. Actually, this number is linked to the proportion of missing data in M . Indeed, when the proportion of missing data is low, the E-step is accurate enough to run many Nesterov outer loops and provide a good performance. On the contrary, when this proportion is high, we faced that running relatively few outer loops provides a better performance. As a trade-off, we run 2 outer loops per M-step.

4 Performance of the weighted NeNMF extensions

In this section, we investigate the completion performance of the proposed W-NeNMF and EM-W-NeNMF, with respect to the state-of-the-art WNMF [13] and EM-WNMF² [32] techniques³. For that purpose, we generate simulations which each consist of the generation of 500×10 and 10×500 random matrices G

² As explained above, the standard approach was shown to be slow to converge in [32], because many MUs were applied at each M-step. To prevent such an issue, at each M-stage, we here run one NMF update per matrix factor, i.e., one MU.

³ In some preliminary tests, we noticed that the EM-W-ANLS method [16] provided a performance similar to EM-WNMF [32]. We thus do not include it in these tests.

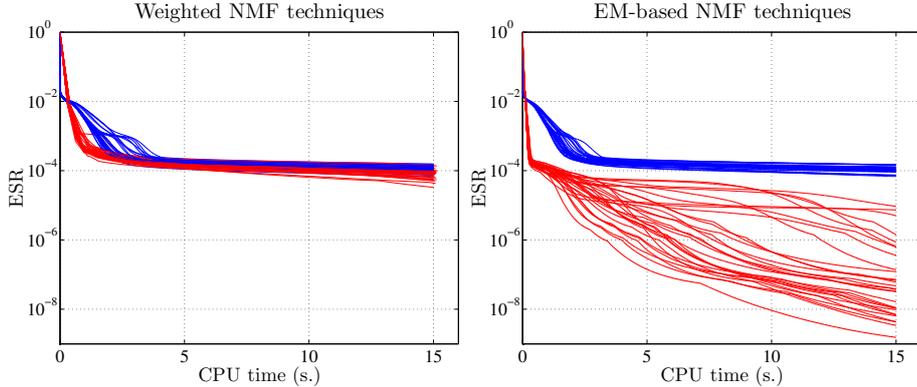


Fig. 1: Evolution of the NMC ESRs vs the CPU time, achieved with MU (in blue), and Nesterov iterations (in red).

and F , respectively. For each pair of matrices, we derive the rank-10 500×500 nonnegative matrix X and we randomly sample a percentage of data through an operator \mathcal{P}_Ω , where this percentage ranges from 10 to 90% (with a step-size of 20%). The number Max_{iter} of passes in the Nesterov inner loops is set to 100. Indeed, such a value was shown to be a good trade-off in preliminary tests where Max_{iter} was set to 1, 10, 100, and 1000. In particular, we noticed a very low performance when $\text{Max}_{\text{iter}} = 1$ —i.e., when the proposed extensions turn out to use a classical gradient descent instead of Nesterov iterations—while the achieved performance was close in the other cases, with a slightly better enhancement with 100 passes per inner loop for each extension. The user-defined threshold ϵ is initialized to 10^{-3} and is divided by 10 each time the stopping criterion (10) is reached within a very low number of inner iterations that we set to $\text{Min}_{\text{iter}} = 5$ in our tests.

All the methods are run using Matlab R2014a on a laptop with an Intel Core i7-4800MQ Quad Core processor, and 32 GB RAM memory. In particular, they are initialized with the same matrices and are stopped after 15 s. In order to compare their performance, we finally compute an Error Signal Ratio (ESR) defined as

$$\text{ESR}(X, G \cdot F) = \frac{\|X - G \cdot F\|_{\mathcal{F}}^2}{\|X\|_{\mathcal{F}}^2}. \quad (18)$$

Figure 1 shows the ESRs—reached by the four tested methods on 30 simulations made with 50% of missing entries in M —with respect to the CPU time expressed in seconds. Interestingly, the MU-based methods and the W-NeNMF method provide the same global ESR after 15 s. However, the W-NeNMF ESRs decrease faster in the early iterations. This shows the acceleration due to the Nesterov iterations. All these methods are outperformed by the EM-W-NeNMF approach whose much lower ESRs are still significantly decreasing. This implies that running the EM-W-NeNMF method more than 15 s should provide a much

better enhancement. On the contrary, the other methods provide slightly decreasing ESRs, which implies that they will need a lot of time to converge to a limit matrix.

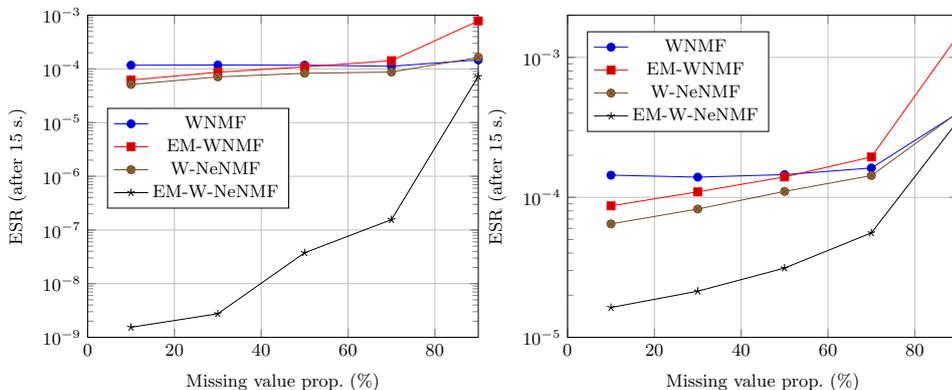


Fig. 2: Median NMC performance reached by the tested NMF methods: (left) noiseless data, (right) noisy data.

The left plot in Fig. 2 shows the median ESRs achieved by the tested methods after 15 s CPU time, with respect to the missing value proportion. Again, we notice that the EM-W-NeNMF outperforms all the other approaches. However, when only 10% of the data in M are available, the achieved performance is almost similar to the one obtained with both the WNMF and W-NeNMF methods. Actually, the weighed extensions of MU-NMF and NeNMF are not very sensitive to the sampling rate, in contrast to the EM methods which look very sensitive to this parameter. Indeed, when a lot of data is missing, the E-step does not yield an accurate estimate of X , thus providing a little enhancement in the M-step, which is consistent with [32]. This results in a slower strategy, which is particularly visible with the EM-WNMF performance and which is partly compensated by the speed of Nesterov gradient for the EM-W-NeNMF method.

The right plot in Fig. 2 shows the achieved ESRs in the same simulations when an additive noise is added to X , with an input SNR around 34 dB. We first notice that the median performance achieved by the WNMF, EM-WNMF, and W-NeNMF methods is almost not affected by the presence of noise. However, the obtained ESRs were already quite high in the noiseless case and thus remain high. Despite the fact that the achieved ESRs are higher in the noisy configuration than in the previous noiseless case, the EM-W-NeNMF still outperforms the other methods. The performance loss is probably due to the fact that the E-step is less efficient, which prevents accurate estimation of G and F in the M-step.

5 Conclusion

In this paper, we proposed to extend a fast NMF method using Nesterov gradient descent to solve large-scale low-rank nonnegative matrix completion. Indeed, contrary to classical gradient methods, Nesterov gradient descent does not need the line search of the optimal step in the update rules, which makes the approach really fast. In order to process missing data, we proposed two extensions using the weighted and the EM strategies, respectively. The latter—which consists of an unweighted optimization of a weighted problem—was found to be better suited than the weighted strategy, in noiseless and noisy NMC simulations, hence showing the relevance of the method.

In future work, we aim to extend such approaches to an informed framework [5, 6, 20]—where some entries of G and/or F are provided—and to apply such extensions to large-scale blind calibration problems [5–7].

Acknowledgments. This work was funded by the "OSCAR" project within the Région Hauts-de-France "Chercheurs Citoyens" Program.

A Proof of Lemmas 1 and 2

Following the structure of the proof in [9], the proof of Lemma 1 is shown by noticing that

$$\mathcal{J}_W(G, \lambda F_1 + (1 - \lambda)F_2) - \lambda \mathcal{J}_W(G, F_1) - (1 - \lambda) \mathcal{J}_W(G, F_2) = \frac{-\lambda(1-\lambda)}{2} \mathcal{J}_W(G, F_1 - F_2) \leq 0. \quad (19)$$

Let us now focus on the proof of Lemma 2. The Lipschitz continuity of $\nabla_F \mathcal{J}_W$ can be shown by extending the proof in [9] to the weighted situation considered in this paper. However, the key point lies in the estimation of the "best" Lipschitz constant. Indeed, let us first recall that Q is a Lipschitz constant of $\nabla_F \mathcal{J}_W(G, \cdot)$ if, for any matrices F_1 and F_2 ,

$$\|\nabla_F \mathcal{J}_W(G, F_1) - \nabla_F \mathcal{J}_W(G, F_2)\|_{\mathcal{F}} \leq Q \cdot \|F_1 - F_2\|_{\mathcal{F}}. \quad (20)$$

From Eq. (7), it is obvious that the larger is the majoring constant Q , the smaller is the associated gradient step size and thus the convergence speed.

Considering the gradient expression (14), we derive

$$\|\nabla_F \mathcal{J}_W(G, F_1) - \nabla_F \mathcal{J}_W(G, F_2)\|_{\mathcal{F}} = \|G^T \cdot (W^2 \circ (G \cdot (F_1 - F_2)))\|_{\mathcal{F}}. \quad (21)$$

The singular value decomposition of G yields

$$\|\nabla_F \mathcal{J}_W(G, F_1) - \nabla_F \mathcal{J}_W(G, F_2)\|_{\mathcal{F}} \leq \|G\|_2 \|W^2 \circ (G \cdot (F_1 - F_2))\|_{\mathcal{F}}. \quad (22)$$

At this stage, we need to put W and G out of the norm to obtain a Lipschitz constant. This can be done by assuming that any column and any row of M

contains at least one element—i.e., any column and row of W contains at least one 1—and thus noticing that

$$\|W^2 \circ (G \cdot (F_1 - F_2))\|_{\mathcal{F}} \leq \|G \cdot (F_1 - F_2)\|_{\mathcal{F}}, \quad (23)$$

which provides, using another singular value decomposition of G :

$$\|\nabla_F \mathcal{J}_W(G, F_1) - \nabla_F \mathcal{J}_W(G, F_2)\|_{\mathcal{F}} \leq \|G\|_2^2 \|F_1 - F_2\|_{\mathcal{F}}, \quad (24)$$

i.e., the Lipschitz constant for W-NeNMF is L .

References

1. Candès, E.J., Plan, Y.: Matrix completion with noise. *Proc. of the IEEE* 98(6), 925–936 (June 2010)
2. Candès, E.J., Recht, B.: Exact matrix completion via convex optimization. *Foundations of Computational Mathematics* 9(6), 717–772 (2009)
3. Choo, J., Lee, C., Reddy, C.K., Park, H.: Weakly supervised nonnegative matrix factorization for user-driven clustering. *Data Mining and Knowledge Discovery* 29(6), 1598–1621 (Nov 2015)
4. Chreiky, R., Delmaire, G., Puigt, M., Roussel, G., Courcot, D., Abche, A.: Split gradient method for informed non-negative matrix factorization. In: Vincent, E., Yeredor, A., Koldovský, Z., Tichavský, P. (eds.) *Proc. of LVA/ICA*. pp. 376–383 (2015)
5. Dorffer, C., Puigt, M., Delmaire, G., Roussel, G.: Blind calibration of mobile sensors using informed nonnegative matrix factorization. In: Vincent, E., Yeredor, A., Koldovský, Z., Tichavský, P. (eds.) *Proc. of LVA/ICA*. pp. 497–505 (2015)
6. Dorffer, C., Puigt, M., Delmaire, G., Roussel, G.: Blind mobile sensor calibration using an informed nonnegative matrix factorization with a relaxed rendezvous model. In: *Proc. of ICASSP*. pp. 2941–2945 (2016)
7. Dorffer, C., Puigt, M., Delmaire, G., Roussel, G.: Nonlinear mobile sensor calibration using informed semi-nonnegative matrix factorization with a Vandermonde factor. In: *Proc. of SAM* (2016)
8. Fazel, M.: Matrix rank minimization with applications. Ph.D. thesis, Stanford University (2002)
9. Guan, N., Tao, D., Luo, Z., Yuan, B.: NeNMF: An optimal gradient method for nonnegative matrix factorization. *IEEE Trans. Sig. Proc.* 60(6), 2882–2898 (June 2012)
10. Guillamet, D., Vitrià, J., Schiele, B.: Introducing a weighted non-negative matrix factorization for image classification. *Pattern Recognition Letters* 24(14), 2447–2454 (2003)
11. Haldar, J.P., Hernando, D.: Rank-constrained solutions to linear matrix equations using powerfactorization. *IEEE Sig. Proc. Letters* 16(7), 584–587 (July 2009)
12. Hamon, R., Emiya, V., Févotte, C.: Convex nonnegative matrix factorization with missing data. In: *Proc. of MLSP* (2016)
13. Ho, N.D.: Nonnegative matrix factorizations algorithms and applications. Ph.D. thesis, Université Catholique de Louvain (2008)
14. Hoyer, P.O.: Non-negative matrix factorization with sparseness constraints. *J. Mach. Learn. Res.* 5, 1457–1469 (Dec 2004)

15. Kim, H., Park, H.: Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method. *SIAM Journal on Matrix Analysis and Applications* 30(2), 713–730 (2008)
16. Kim, Y.D., Choi, S.: Weighted nonnegative matrix factorization. In: Proc. of ICASSP. pp. 1541–1544 (April 2009)
17. Kumar, R., Da Silva, C., Akalin, O., Aravkin, A.Y., Mansour, H., Recht, B., Herrmann, F.J.: Efficient matrix completion for seismic data reconstruction. *Geophysics* 80(5), V97–V114 (2015)
18. Lantéri, H., Theys, C., Richard, C., Févotte, C.: Split gradient method for nonnegative matrix factorization. In: Proc. of EUSIPCO (2010)
19. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: NIPS, pp. 556–562 (2001)
20. Limem, A., Delmaire, G., Puigt, M., Roussel, G., Courcot, D.: Non-negative matrix factorization under equality constraints—a study of industrial source identification. *Applied Numerical Mathematics* 85, 1–15 (2014)
21. Limem, A., Puigt, M., Delmaire, G., Roussel, G., Courcot, D.: Bound constrained weighted NMF for industrial source apportionment. In: Proc. of MLSP (2014)
22. Lin, C.: Projected gradient methods for nonnegative matrix factorization. *Neural Computation* 19(10), 2756–2779 (Oct 2007)
23. Liu, C., Yang, H.C., Fan, J., He, L.W., Wang, Y.M.: Distributed nonnegative matrix factorization for web-scale dyadic data analysis on MapReduce. In: Proc. of WWW Conf. (April 2010)
24. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research* 11(January), 19–60 (2010)
25. Nesterov, Y.: A method of solving a convex programming problem with convergence rate $O(1/k^2)$. In: *Soviet Mathematics Doklady*. vol. 27, pp. 372–376 (1983)
26. Savvaki, S., Tsagkatakis, G., Panousopoulou, A., Tsakalides, P.: Application of matrix completion on water treatment data. In: Proc. of CySWater. pp. 3:1–3:6 (2015)
27. Sobral, A., Bouwmans, T., Zahzah, E.: LRSLibrary: Low-rank and sparse tools for background modeling and subtraction in videos. In: *Robust Low-Rank and Sparse Matrix Decomposition: Applications in Image and Video Processing*. CRC Press, Taylor and Francis Group.
28. Srebro, N., Jaakkola, T.: Weighted low-rank approximations. In: Proc. of ICML (2003)
29. Tepper, M., Sapiro, G.: Compressed nonnegative matrix factorization is fast and accurate. *IEEE Trans. Sig. Proc.* 64(9), 2269–2283 (May 2016)
30. Wang, Y.X., Zhang, Y.J.: Nonnegative matrix factorization: A comprehensive review. *IEEE Trans. on Knowledge and Data Eng.* 25(6), 1336–1353 (June 2013)
31. Wen, Z., Yin, W., Zhang, Y.: Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. *Mathematical Programming Computation* 4(4), 333–361 (2012)
32. Zhang, S., Wang, W., Ford, J., Makedon, F.: *Learning from Incomplete Ratings Using Non-negative Matrix Factorization*, chap. 58, pp. 549–553 (2006)
33. Zhou, G., Cichocki, A., Xie, S.: Fast nonnegative matrix/tensor factorization based on low-rank approximation. *IEEE Trans. Sig. Proc.* 60(6), 2928–2940 (June 2012)