How to Apply Random Projections to Nonnegative Matrix Factorization with Missing Entries?

Farouk Yahaya, Matthieu Puigt, Gilles Delmaire, Gilles Roussel
Univ. Littoral Côte d'Opale, LISIC – EA 4491, F-62228 Calais, France
firstname.lastname@univ-littoral.fr

Abstract—**Random projections belong to the major techniques to process big data and have been successfully applied to Nonnegative Matrix Factorization (NMF). However, they cannot be applied in the case of missing entries in the matrix to factorize, which occurs in many actual problems with large data matrices. In this paper, we thus aim to solve this issue and we propose a novel framework to apply random projections in weighted NMF, where the weight models the confidence in the data (or the absence of confidence in the case of missing data). We experimentally show the proposed framework to significantly speed-up state-of-the-art NMF methods under some mild conditions. In particular, the proposed strategy is particularly efficient when combined with Nesterov gradient or alternating least squares.**

Index Terms—**Nonnegative matrix factorization, missing data, random projections, low-rank matrix completion, blind source separation, big data**

I. INTRODUCTION

Modern latent variable analysis methods—e.g., sparse approximation, robust principal component analysis, dictionary learning—have been massively investigated for more than two decades and were successfully applied to signal, image, or video processing, and to machine learning. Among these techniques, Nonnegative Matrix Factorization (NMF) attracted a lot of interest from the scientific community since the pioneering work in [1], [2]. Indeed, it usually provides more interpretable results than methods without any sign constraint (e.g., independent component analysis) [3] and it was successfully applied to many fields, e.g., audio signals [4], hyperspectral unmixing [5], or environmental data processing [6]. NMF consists of estimating two $n \times p$ and $p \times m$ nonnegative matrices G and F, respectively, from a $n \times m$ nonnegative matrix X such that [7]

$$X \simeq G \cdot F. \qquad (1)$$

NMF usually consists of solving alternating subproblems, i.e.,

$$\hat{G} = \arg\min_{G \geq 0} ||X - G \cdot F||_{\mathcal{F}}, \qquad (2)$$

$$\hat{F} = \arg\min_{F \geq 0} ||X - G \cdot F||_{\mathcal{F}}, \qquad (3)$$

where $||\cdot||_{\mathcal{F}}$ denotes the Frobenius norm. Weigthed NMF (WNMF) adds a confidence measure w_{ij} associated to the data point x_{ij} and consists of solving

$$\min_{G, F \geq 0} ||W \circ X - W \circ (G \cdot F)||_{\mathcal{F}}, \qquad (4)$$

where W and $\circ$ denote the matrix of the above weights w_{ij}, and the Hadamard product, respectively. WNMF was successfully applied to, e.g., image [8] and audio processing [9], collaborative filtering [10], chemical source separation [11], mobile sensor calibration [12], or nonnegative matrix completion[1] [15].

When $\min(n, m) \gg p$ is large, the computational time for estimating G and F in Eq. (1) using classical NMF techniques is prohibitive. To solve such an issue, several strategies were proposed, e.g., distributed computations [16], partial update through online optimization [17], the use of optimal solvers [18] and/or of random projections [19]–[22]. It should be noticed that even if most of these strategies can be combined, all of them cannot be used in WNMF. Indeed, to the best of the authors' knowledge, using random projections to speed up the WNMF computations was never investigated. This paper addresses this issue, by proposing a novel framework which can be applied to any NMF method as well as any matrix factorization technique without sign constraint, e.g., [23] for low-rank matrix completion.

The remainder of the paper reads as follows. We recall the principles of compressed NMF in Sect. II. Section III introduces our proposed compressed and weighted NMF method whose performance is investigated in Sect. IV. Lastly, we conclude and discuss about future directions in Sect. V.

II. NMF WITH RANDOM PROJECTIONS

We now briefly recall the principles of NMF using random projections. Random projections became a popular tool for lage-scale machine learning and signal processing—see, e.g., [24], [25] for comprehensive reviews—and are built on the Johnson-Lindenstrauss lemma [26]. The latter states that some points which lie in a high dimensional Euclidian space can be projected into a low-dimensional one with a negligible pairwise distance error.

Applied to NMF, random projections consist of designing two compression matrices L and R to be left and right multiplied to X, respectively. The resulting matrices—denoted X_L and X_R, respectively—are far smaller than X and allow to fasten the NMF computations, as shown in Algorithm 1. Please note that as L and R have no sign constraint, the matrices X_L, G_L, X_R, and F_R can get negative entries. Since

---

[1]Please note that most low-rank matrix completion techniques find their roots in [13], [14] and are thus not based on matrix factorization.

Algorithm 1 Compressed NMF strategy

Require: initial matrices G and F, compression matrices L and R.
    Define $X_L \triangleq L \cdot X$ and $X_R \triangleq X \cdot R$
    repeat
        Define $F_R \triangleq F \cdot R$
        Update G as in (2) by replacing X and F by X_R and F_R, respectively
        Define $G_L \triangleq L \cdot G$
        Update F as in (3) by replacing X and G by X_L and G_L, respectively
    until a stopping criterion

G and F remain nonnegative, their associated update rules in Algorithm 1 are instances of semi-NMF [27]. Lastly, the NMF stopping criterion might be a target approximation error, a number of iterations, or a reached CPU time.

Algorithm 2 RPIs for NMF [20]

Require: a target rank ν (with $p \leq \nu \ll \min(n, m)$) and an integer q (e.g., $q = 4$ in [20])
    Draw Gaussian random matrices $\Omega_L \in \mathbb{R}^{m \times \nu}$ and $\Omega_R \in \mathbb{R}^{\nu \times n}$
    Define $B_L \triangleq (XX^T)^q \cdot X \cdot \Omega_L$ and $B_R \triangleq \Omega_R \cdot X \cdot (X^T X)^q$
    Derive $L \in \mathbb{R}^{\nu \times n}$ and $R \in \mathbb{R}^{m \times \nu}$ by QR decompositions of B_L and B_R, respectively.

Several designs for L and R have been investigated in the literature. Indeed, the authors in [19] proposed to use Gaussian realizations as tentative compression matrices, thus following the proof of the Johnson-Lindenstrauss lemma. The authors in [20]–[22] found instead that deriving compression matrices L and R which exhibit orthonormal columns and rows, respectively, provided a much higher enhancement. For that purpose, Randomized Power Iterations (RPIs) were used in [20], [21] while we used a Randomized Subspace Iteration method (RSI) in [22]. Both the RPI and RSI techniques are provided in Algorithms 2 and 3, respectively. In practice, the computation of $(XX^T)^q$ and $(X^T X)^q$ in RPIs are done in a loop, in the same way as proposed in RSIs, except that there is no intermediate QR decomposition in the RPI algorithm. As a consequence, both randomized methods are equivalent in theory but RSIs are less sensitive to round-off errors [24].

III. PROPOSED METHOD

In this section, we propose a WNMF strategy with random projections. As for NMF, WNMF is iteratively performed by alternatingly updating G and F. To that end, in the literature, two main strategies allow to take into account the weight matrix W (both were tested in [15]), i.e., (i) keeping W in the update rules [8] or (ii) removing it within an Expectation-Maximization (EM) technique [10]. The latter assumes the entries of W to be between[2] 0 and 1. Denoting

---

[2]Such an assumption is not an issue, as it is possible to scale any non-null matrix W so that its maximum value is 1.

Algorithm 3 RSIs for NMF [22]

Require: a target rank ν (with $p \leq \nu \ll \min(n, m)$) and an integer q (e.g., $q = 4$)
    Draw Gaussian random matrices $\Omega_L \in \mathbb{R}^{m \times \nu}$ and $\Omega_R \in \mathbb{R}^{\nu \times n}$
    Form $\mathbb{X}_L^{(0)} \triangleq X \cdot \Omega_L$ and $\mathbb{X}_R^{(0)} \triangleq \Omega_R \cdot X$
    Compute their respective orthonormal bases $Q_L^{(0)}$ and $Q_R^{(0)}$, by QR decomposition of $\mathbb{X}_L^{(0)}$ and $\mathbb{X}_R^{(0)}$, respectively
    for $k = 1$ **to** q **do**
        Define $\tilde{\mathbb{X}}_L^{(k)} \triangleq X^T \cdot Q_L^{(k-1)}$ and $\tilde{\mathbb{X}}_R^{(k)} \triangleq Q_R^{(k-1)} \cdot X^T$
        Derive their respective orthonormal bases $\tilde{Q}_L^{(k)}$ and $\tilde{Q}_R^{(k)}$
        Compute $\mathbb{X}_L^{(k)} \triangleq X \cdot \tilde{Q}_L^{(k)}$ and $\mathbb{X}_R^{(k)} \triangleq \tilde{Q}_R^{(k)} \cdot X$
        Derive their respective orthonormal bases $Q_L^{(k)}$ and $Q_R^{(k)}$
    end for
    Derive $L \triangleq \tilde{Q}_L^{(q)}$ and $R \triangleq \tilde{Q}_R^{(q)}$, respectively.

$\overline{W} \triangleq (\mathbb{1}_{n,m} - W)$—where $\mathbb{1}_{n,m}$ is the $n \times m$ matrix of ones—X_u the matrix with unknown entries, and $(t - 1)$ the current iteration, the EM strategy aims to maximize $\mathbb{E}[\log \mathbb{P}(W \circ X, \overline{W} \circ X_u | G \cdot F), W \circ X | (G \cdot F)^{(t-1)}]$ [10]. In the E-step, the unknown entries of X are estimated as:

$$X^{\text{comp}} = W \circ X + \overline{W} \circ (G \cdot F)^{(t-1)}. \tag{5}$$

The M-step then consists of applying standard NMF update rules to X^{comp} in order to derive G and F. Once NMF converged to a given solution [10] or after a given number $\text{Max}_{\text{OutIter}}$ of iterations [15], X^{comp} is updated in another E-step using the last estimates of G and F in Eq. (5). In practice, the EM strategy is less sensitive to initialization but is slower than the direct optimization of Eq. (4) with classical NMF solvers [10]. This last drawback was solved in [15] by applying the Nesterov iterations [28] to NMF (NeNMF). Indeed, the direct weighted extension of NeNMF—denoted W-NeNMF—was found not to be efficient in [15] because of some Hadamard products involving the weight matrix which drastically slow down the method. On the contrary, the EM version of W-NeNMF—denoted EM-W-NeNMF—is far more efficient, except when the proportion of missing entries in X is large, i.e., 90% in [15]. Indeed, in that case, the estimation of X^{comp} in Eq. (5) is not accurate, thus propagating inaccuracies in the estimation of G and F along iterations.

At this stage, it should be noticed that the above EM strategy replaces $W \circ X$ in the WNMF problem by X^{comp}, thus allowing to directly applying any solver, e.g., Mulplicative Update (MU), Alternating Least Squares (ALS), Projected Gradient (PG), or the Nesterov optimal gradient. Still, it needs to process the whole matrices X^{comp}, G and F, which might be computationally expensive in some large-scale problems. To solve this issue, we propose to apply random projections to X^{comp}—using either RPIs or RSIs (see Algorithms 2 or 3)—at each E-step, so that the NMF iterations in the M-step are sped up. However, please note that contrary to unweighted NMF, the matrices L and R are not computed once but after each estimation of X^{comp} in the E-step. As a consequence,

the computational time earned during the M-step might be counterbalanced by the time needed to compress the matrices in the E-step. This issue is investigated in the next section.

The overall structure of the proposed Randomized extension of EM-W-NMF—denoted REM-W-NMF below—is presented in Algorithm 4. The approach consists of a loop alternating E-steps and M-steps. Each M-step consists of an NMF outer loop which is run $\text{Max}_{\text{OutIter}}$ times. Then each update of the matrices G and F can be processed by, e.g., MU (REM-W-NMF-MU), ALS (REM-W-NMF-ALS), PG (REM-W-NMF-PG) or a Nesterov inner loop[3] (REM-W-NeNMF). Let us stress again that, to the best of the authors' knowledge, this strategy is the very first one to apply random projections to weighted matrix factorization.

Algorithm 4 Proposed REM-W-NMF structure

Require: initial matrices G and F
  repeat
    {**E-step**}
    Compute X^{comp} as in (5)
    Apply RSIs or RPIs to X^{comp} to compute L and R.
    Define $X_L^{\text{comp}} \triangleq L \cdot X^{\text{comp}}$ and $X_R^{\text{comp}} \triangleq X^{\text{comp}} \cdot R$
    {**M-step**}
    for compt=1 **to** $\text{Max}_{\text{OutIter}}$ **do**
      Define $F_R \triangleq F \cdot R$ and solve $\min\limits_{G \geq 0} \|X_R^{\text{comp}} - G \cdot F_R\|_{\mathcal{F}}$
      Define $G_L \triangleq L \cdot G$ and solve $\min\limits_{F \geq 0} \|X_L^{\text{comp}} - G_L \cdot F\|_{\mathcal{F}}$
    end for
  until a stopping criterion

IV. EXPERIMENTAL VALIDATION

To assess the performance of the proposed strategy, we propose an experiment which is investigated with both a nonnegative matrix completion point of view—as in [15]—and a Blind Source Separation (BSS) one. Indeed, while the former focuses on the estimation of the missing entries of X from G and F, BSS investigates the quality of estimation of each matrix factor. The latter is challenging as a good low-rank approximation of X does not necessarily implies good estimates of G and F.

For that purpose, we repeat 15 times the following experiment: we randomly generate nonnegative factor matrices G^{theo} and F^{theo}, with $n = m = 10000$ and $p = 5$. Its product provides the whole observed data matrix X^{theo} that we randomly sample with a sampling rate varying from 10 to 90% (with a step-size of 20%). We compare the proposed REM-W-NMF strategy when compared to the original EM-W-NMF one using MU [10], ALS—that we found in preliminary tests to be much faster than alternating nonnegative least squares [29]—PG, and Nesterov optimal gradient [15]. For each test, each method is initialized with the same random matrices F and G and is run during[4] 60 s using Matlab R2018a on a laptop with

---

[3]This loop is run at most Max_{iter} times (or when a given bound in the gradient computations is reached), as in unweighted NeNMF [18].

[4]While not being classical in the literature, limiting the computations to a given available CPU time is a crucial constraint in practical applications.

---

an Intel Core i7-4800MQ Quad Core processor and 32 GB RAM memory. For a given solver, the fastest approach will run more iterations and should thus provide a better enhancement. For each method, we consider two performance indices, i.e., one classically used in matrix completion and one related to BSS. We thus consider the Relative Reconstruction Error (RRE) which is computed with respect to X^{theo}, i.e.,

$$\text{RRE} \triangleq \|X^{\text{theo}} - G \cdot F\|_{\mathcal{F}}^2 / \|X^{\text{theo}}\|_{\mathcal{F}}^2, \quad (6)$$

and which is useful for validating the matrix-completion accuracy of the tested techniques. We also consider the Signal-to-Interference Ratio (SIR) which compares the estimated matrix factor F with F^{theo}, up to permutation and scale ambiguities. In practice, the SIR is computed over each row of F and is averaged in order to provide a mean SIR (in dB) per experiment. It should be noticed that estimating the SIR is time demanding when the value p of the rank of X becomes large, because of the above permutation ambiguities. Moreover, and as discussed in the conclusion, we aim to extend the proposed REM-W-NMF strategy to mobile sensor calibration [12], [30] where the rank of X is known[5] and even smaller than in these experiments—i.e., $p = 2$ in [12] and $p = 3$ in [30]—hence the very small value of p which is tested in this paper. We found RSIs and RPIs to provide an almost similar performance, although the former sometimes yields a slightly better enhancement. As a consequence, we only use RSIs in these tests and we set q to $q = 4$ (see Algorithm 3). However, the number $\text{Max}_{\text{OutIter}}$ of outer iterations is not fixed but is set to 1, 20, and 50, respectively, in order to investigate its effects to the NMF performance. Lastly, as for the vanilla NeNMF and its weighted extensions, several parameters must be fixed in the (R)EM-W-NeNMF methods. The following values were chosen according to results in preliminary tests. The Nesterov inner loop is stopped after at most $\text{Max}_{\text{InIter}} = 500$ updates or if the gradient term is 1000 times below the one estimated at the beginning of the inner loop.

TABLE I
MEDIAN CPU TIME REACHED WITH THE DIFFERENT TESTED SOLVERS.

Method	1 E-step (in s)	1 outer loop in the M-step (in s)
REM-W-NeNMF	6.7359	0.0366
EM-W-NeNMF	2.3894	0.3275
REM-W-NMF-ALS	6.7039	0.0025
EM-W-NMF-ALS	2.4462	0.2778
REM-W-NMF-MU	6.6546	0.0043
EM-W-NMF-MU	2.3932	0.2773
REM-W-NMF-PG	6.4367	0.0718
EM-W-NMF-PG	2.4802	1.3917

Table I summarizes the computational cost needed at each stage of both the Vanilla EM-W-NMF and our proposed REM-W-NMF strategies, computed over all the tests when $\text{Max}_{\text{OutIter}} = 50$. One can notice that the median time needed in the E-step of all the randomized methods is almost 3 times

---

[5]Otherwise, a nuclear norm penalization [31] can easily be added to Eq. (4) to extend the proposed methods.

higher than those obtained with the Vanilla techniques. This shows that the bottleneck of our proposed framework is the repetitive use of random projections when the matrix X^{comp} is re-estimated. However and as expected, performing one outer loop in REM-W-NMF is 9 to 110 times faster than one in EM-W-NMF. This implies that the higher $\text{Max}_{\text{OutIter}}$, the more benefits there are to apply random projections. However, in practice, a trade-off must be found and an appropriate choice of NMF outer loops in the M-steps must be set. From Table I, one should notice that both tested gradient-based approaches get less benefits from the random projections, from a CPU time point of view. This might be due to the optimal step search in PG or to the number of Nesterov iterations which might vary up to a maximum number of iterations [15], [18].

Figure 1 shows the median RREs and SIRs obtained by the tested methods, with respect to the missing value proportion and the number of iterations in the outer loop. The plain lines show the performance of the vanilla EM-W-NMF techniques while the dashed ones show those reached by our proposed randomized extensions. Let us first investigate the RRE enhancement. When $\text{Max}_{\text{OutIter}} = 1$, the RREs provided by the REM-W-NMF techniques are higher than those provided by the vanilla ones. This is due to the fact that only 1 outer iteration is performed at each M-step, which implies—according to Table I—that all the REM-W-NMF run less iterations than their EM-W-NMF versions. Still, both randomized gradient-based approaches provide almost similar median RREs than their vanilla versions, when 50% (or less) of the data are missing. When $\text{Max}_{\text{OutIter}} = 20$ and except for the MU-based methods, all the REM-W-NMF techniques outperform their EM-W-NMF variants. However, it should be noticed that the enhancement provided by randomization is quite moderate, except for the PG-based approach. When $\text{Max}_{\text{OutIter}} = 50$, all the randomized techniques outperform their vanilla variants. In particular, both the REM-W-NMF-ALS and REM-W-NeNMF significantly outperfom all the tested methods: depending on the proportion of missing values in X, their RRE is 10 to 1000 lower than the already low one obtained by their vanilla versions. These results show the interest of the proposed strategy. However, it should be noticed that when the missing value proportion is 90%, the RREs obtained when $\text{Max}_{\text{OutIter}} = 20$ or 50 are much higher than when it is set to 1. This is probably due to the fact that the completed matrix X^{comp} is far from the theoretical one in early iterations, so that the estimation of F and G in the M-step does not get any significant benefits from increasing the number of M-step iterations. Lastly, we repeat the above experiment in a noisy scenario, which is not shown in the paper for space considerations. The obtained RREs are consistent with the above ones—reached in a noiseless case—except that all the RREs are higher in the former than in the latter, because of the additive noise.

The bottom plots in Fig. 1 show the obtained SIRs in the same conditions. Firstly, they do not monotonically vary with the missing value proportion. While one might expect to get better estimates of F when more entries in X are available,

this is not always the case. This behaviour is especially visible when $\text{Max}_{\text{OutIter}}$ is equal to 20 or 50, for a significant number of tested methods. This might be due to the fact that NMF is NP-hard and that a unique solution is not guaranteed in the general case[6] [3]. This might lower the median performance of the tested methods. As for the RREs, the achieved performance is low when $\text{Max}_{\text{OutIter}}$ is set to 1. As expected, the vanilla EM-W-NMF methods outperform their randomized extensions. It should be noticed that in that case, the best performance is achieved with PG or ALS. However, the NMF performance achieved by both the vanilla technique using MU (respectively, Nesterov iterations) and and its randomized extension are almost similar (but low). When $\text{Max}_{\text{OutIter}} = 20$, while the RREs provided by randomized techniques were already significantly lower than with vanilla techniques, such a behaviour is not obvious when we are interested in SIRs. Indeed, except when used with the Nesterov gradient, the random projections do not provide any noticeable separation enhancement and the highest SIRs are still obtained with the vanilla EM-W-NMF-ALS approach. When $\text{Max}_{\text{OutIter}}$ is set to 50, the performance reached by both the randomized and the vanilla techniques using MU or PG are quite similar. On the contrary, the other tested randomized methods clearly outperform their vanilla variants. These results show the relevance of the proposed technique. However, while the REM-W-NeNMF and the REM-W-NMF-ALS provided similar RREs, the latter provides much higher SIRs than the former. The ALS-based techniques thus provide the best low-rank matrix completion (with the Nesterov-based methods) and the best separation enhancement.

V. CONCLUSION

In this paper, we proposed a novel framework to combine random projection and weighted matrix factorization. It is based on an EM scheme and applies random projections at each E-step, on the completed version of the partially observed matrix. Provided there are enough outer iterations in the M-step, the proposed strategy allows to outperform non-randomized state-of-the-art EM techniques, especially for low-rank matrix completion. However, we noticed that the computation of the projection matrices in the E-step is the bottleneck of the proposed strategy (which can be counterbalanced by the reduced cost of the NMF updates in the M-step). Such an issue might be solved by using some specific hardware providing optical random projections [32]. In future work, we aim to apply the proposed strategy to informed and structured NMF techniques applied to mobile sensor calibration [12], [30].

REFERENCES

[1] P. Paatero and U. Tapper, "Positive matrix factorization: a non negative factor model with optimal utilization of error estimates of data values," *Environmetrics*, vol. 5, no. 2, pp. 111–126, 1994.

[2] D.D. Lee and H.S. Seung, "Learning the parts of objects by non negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[3] N. Gillis, "The why and how of nonnegative matrix factorization," in *Regularization, Optimization, Kernels, and Support Vector Machines*, pp. 257–291. Chapman and Hall/CRC, 2014.

---

[6]In preliminary tests, we found the proposed randomized methods to be as sensitive to the initialization as the vanilla ones they extend.
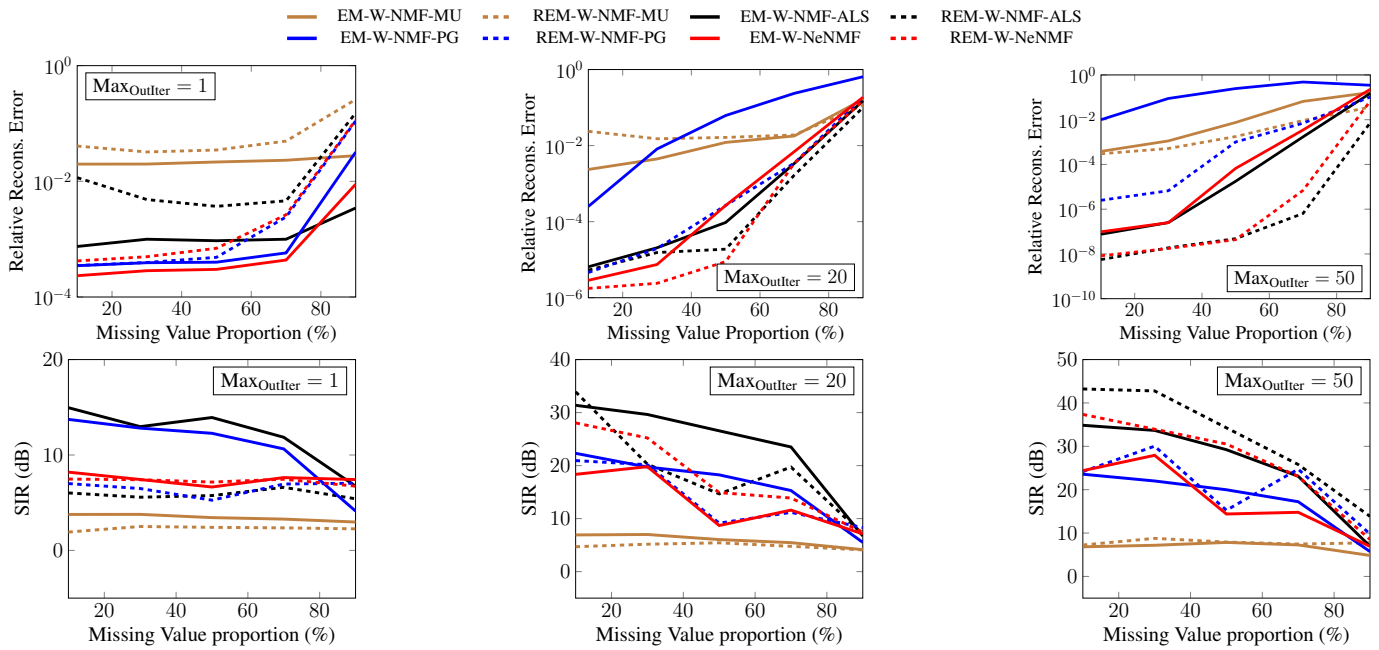
Fig. 1. RRE (top) and SIR (bottom) versus the missing value proportion. (left): 1 outer loop iteration per M-step. (middle): 20 iterations. (right): 50 iterations.

[4] C. Févotte, E. Vincent, and A. Ozerov, "Single-channel audio source separation with NMF: divergences, constraints and algorithms," in *Audio Source Separation*, pp. 1–24. Springer, 2018.

[5] J. M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot, "Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 2, pp. 354–379, 2012.

[6] M. Puigt, G. Delmaire, and G. Roussel, "Environmental signal processing: new trends and applications," in *Proc. ESANN'17*, 2017, pp. 205–214.

[7] Y. X. Wang and Y. J. Zhang, "Nonnegative matrix factorization: A comprehensive review," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 6, pp. 1336–1353, June 2013.

[8] N.-D. Ho, *Non negative matrix factorization algorithms and applications*, Phd thesis, Université Catholique de Louvain, 2008.

[9] T. O. Virtanen, "Monaural sound source separation by perceptually weighted non-negative matrix factorization," Tampere University of Technology, Tech. Rep, 2007.

[10] S. Zhang, W. Wang, J. Ford, and F. Makedon, "Learning from incomplete ratings using non-negative matrix factorization," in *Proc. SIAM ICDM'06*, 2006, pp. 549–553.

[11] A. Limem, G. Delmaire, M. Puigt, G. Roussel, and D. Courcot, "Non-negative matrix factorization under equality constraints—a study of industrial source identification," *Applied Numerical Mathematics*, vol. 85, pp. 1–15, Nov. 2014.

[12] C. Dorffer, M. Puigt, G. Delmaire, and G. Roussel, "Informed nonnegative matrix factorization methods for mobile sensor network calibration," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 4, no. 4, pp. 667–682, Dec 2018.

[13] M. Fazel, *Matrix rank minimization with applications*, Ph.D. thesis, Stanford University, 2002.

[14] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational Mathematics*, vol. 9, no. 6, pp. 717–772, 2009.

[15] C. Dorffer, M. Puigt, G. Delmaire, and G. Roussel, "Fast nonnegative matrix factorization and completion using Nesterov iterations," in *Proc. LVA/ICA'17*, 2017, vol. LNCS 10179, pp. 26–35.

[16] C. Liu, H.-C. Yang, J. Fan, L.-W. He, and Y.-M. Wang, "Distributed nonnegative matrix factorization for web-scale dyadic data analysis on MapReduce," in *Proc. WWW Conf.'10*, April 2010.

[17] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *Journal of Machine Learning Research*, vol. 11, no. Jan, pp. 19–60, 2010.

[18] N. Guan, D.Vin Tao, Z. Luo, and B. Yuan, "NeNMF: An optimal gradient method for nonnegative matrix factorization," *IEEE Trans. Signal Process.*, vol. 60, no. 6, pp. 2882–2898, 2012.

[19] F. Wang and P. Li, "Efficient nonnegative matrix factorization with random projections," in *Proc. SIAM ICDM'10*, 2010, pp. 281–292.

[20] M. Tepper and G. Sapiro, "Compressed nonnegative matrix factorization is fast and accurate," *IEEE Trans. Signal Process.*, vol. 64, no. 9, pp. 2269–2283, May 2016.

[21] N. B. Erichson, A. Mendible, S. Wihlborn, and J. N. Kutz, "Randomized nonnegative matrix factorization," *Pattern Recognition Letters*, 2018.

[22] F. Yahaya, M. Puigt, G. Delmaire, and G. Roussel, "Faster-than-fast NMF using random projections and Nesterov iterations," in *Proc. iTWIST'18*, 2018.

[23] N. Srebro and T. Jaakkola, "Weighted low-rank approximations," in *Proc. ICML'03*, 2003, pp. 720–727.

[24] N. Halko, P.-G. Martinsson, and J. A Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM review*, vol. 53, no. 2, pp. 217–288, 2011.

[25] P. Drineas and M. W. Mahoney, "RandNLA: randomized numerical linear algebra," *CACM*, vol. 59, no. 6, pp. 80–90, 2016.

[26] W. B. Johnson and J. Lindenstrauss, "Extensions of Lipschitz mappings into a Hilbert space," *Contemp. Math.*, vol. 26, pp. 189–206, 1984.

[27] C. H. Q. Ding, T. Li, and M. I. Jordan, "Convex and semi-nonnegative matrix factorizations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 45–55, 2010.

[28] Y. Nesterov, "A method of solving a convex programming problem with convergence rate O(1/k2)," in *Soviet Mathematics Doklady*, 1983, vol. 27, pp. 372–376.

[29] Y. D. Kim and S. Choi, "Weighted nonnegative matrix factorization," in *Proc. ICASSP'09*, April 2009, pp. 1541–1544.

[30] C. Dorffer, M. Puigt, G. Delmaire, and G. Roussel, "Nonlinear mobile sensor calibration using informed semi-nonnegative matrix factorization with a Vandermonde factor," in *Proc. SAM'16*, 2016.

[31] R. Cabral, F. De la Torre, J. P. Costeira, and A. Bernardino, "Unifying nuclear norm and bilinear factorization approaches for low-rank matrix decomposition," in *Proc. ICCV'13*, 2013, pp. 2488–2495.

[32] A. Saade, F. Caltagirone, I. Carron, L. Daudet, A. Drémeau, S. Gigan, and F. Krzakala, "Random projections through multiple optical scattering: Approximating kernels at the speed of light," in *Proc. ICASSP'16*, 2016, pp. 6215–6219.