

# Fast & furious: accelerating weighted NMF using random projections

Farouk Yahaya\*, Matthieu Puigt, Gilles Delmaire, and Gilles Roussel

Univ. Littoral Côte d’Opale, LISIC – EA 4491, F-62228 Calais, France.

**Abstract**— Random projections have been successfully applied to accelerate Nonnegative Matrix Factorization (NMF). However, they are not suited to 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. We experimentally show the proposed framework to significantly speed-up state-of-the-art NMF methods under some mild conditions.

## 1 Introduction

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]

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

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

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

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

$$W \circ X \simeq W \circ (G \cdot F), \quad (3)$$

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<sup>1</sup> [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, 20, 21, 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.

## 2 Proposed method

As for NMF, WNMF iteratively and alternately updates  $G$  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<sup>2</sup> 0 and 1. In the E-step, the unknown entries of  $X$  are estimated by their best estimates, derived from the product  $G \cdot F$ , i.e.,

$$X^{\text{comp}} = W \circ X + (\mathbb{1}_{n,m} - W) \circ (G \cdot F), \quad (4)$$

where  $\mathbb{1}_{n,m}$  is the  $n \times m$  matrix of ones. The M-step then consists of applying standard NMF update rules to  $X^{\text{comp}}$  in order to derive  $G$  and  $F$ . Once NMF converged to a given solution [10] or after a given number  $\text{Max}_{\text{Outlier}}$  of iterations [15],  $X^{\text{comp}}$  is updated in another E-step using the last estimates of  $G$  and  $F$  in Eq. (4). In practice, the EM strategy is less sensitive to initialization but is slower than the direct optimization of Eq. (3) with classical NMF solvers [10]. This last drawback was solved in [15] by applying the Nesterov iterations [24], where the EM extension of NeNMF [18] was found to be far more efficient than its direct weighted extension.

At this stage, it should be noticed that the above EM strategy replaces  $W \circ X$  in the WNMF problem by  $X^{\text{comp}}$ , thus allowing to directly applying any solver, e.g., Multiplicative Update or Alternating Least Squares (ALS). Still, it needs to process the whole matrices  $X^{\text{comp}}$ ,  $G$  and  $F$ , which might be computationally expensive in some large-scale problems. To solve this issue, we propose to apply *random projections* to  $X^{\text{comp}}$  at each E-step, so that the NMF iterations in the M-step are sped up.

Applied to unweighted 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. Among the proposed strategies to design  $L$  and  $R$ , structured random projections—denoted Randomized Power Iterations (RPIs) in [25]—are well-suited to NMF. From a realization of a Gaussian random matrix  $\Omega_L$ ,  $L$  is computed as the QR decomposition of  $(X X^T)^q \cdot X \cdot \Omega_L$ , where  $q$  is a given integer, e.g.,  $q = 4$  in [20].  $R$  is similarly obtained [20]. Please note that we found in [22] that Randomized Subspace Iterations (RSIs)—a variant of RPIs which is less sensitive to round-off errors [25]—provided a slightly better NMF performance than RPIs. Moreover, as  $L$  and  $R$  have no sign constraint, the matrices  $X_L$ ,  $G_L$ ,  $X_R$ , and

\*F. Yahaya greatly acknowledges the Région Hauts-de-France to partly fund his Ph.D. fellowship. Experiments presented in this paper were carried out using the CALCULCO computing platform, supported by SCoSI/ULCO.

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

<sup>2</sup>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.

$F_R$  can get negative entries. Since  $G$  and  $F$  remain nonnegative, their associated update rules are instances of semi-NMF [26]. Lastly, the NMF stopping criterion might be a target approximation error, a number of iterations, or a CPU time.

The overall structure of the proposed Randomized extension of EM-W-NMF—denoted REM-W-NMF below—is presented in Algorithm 1. The approach consists of a loop alternating E-steps and M-steps. Each M-step consists of an NMF outer loop which is run  $\text{Max}_{\text{Outer}}$  times. Then each update of the matrices  $G$  and  $F$  can be processed by, e.g., ALS (REM-W-NMF-ALS) or a Nesterov inner loop<sup>3</sup> (REM-W-NeNMF).

---

**Algorithm 1** Proposed REM-W-NMF structure

---

**Require:** initial matrices  $G$  and  $F$

**repeat**

  {E-step}

  Compute  $X^{\text{comp}}$  as in (4)

  Apply RPIs or RSIs to  $X^{\text{comp}}$  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**  $\text{compt}=1$  to  $\text{Max}_{\text{Outer}}$  **do**

    Define  $F_R \triangleq F \cdot R$  and solve  $\min_{G \geq 0} \|X_R^{\text{comp}} - G \cdot F_R\|_{\mathcal{F}}$

    Define  $G_L \triangleq L \cdot G$  and solve  $\min_{F \geq 0} \|X_L^{\text{comp}} - G_L \cdot F\|_{\mathcal{F}}$

**end for**

**until** a stopping criterion

---

### 3 Experiments and conclusion

To assess the performance of the proposed strategy, we repeat 15 times the following experiment: we randomly generate non-negative factor matrices  $G^{\text{theo}}$  and  $F^{\text{theo}}$ , with  $n = m = 10000$  and  $p = 5$ . Its product provides the whole observed data matrix  $X^{\text{theo}}$  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 ALS—that we found in preliminary tests to be much faster than alternating nonnegative least squares [27]—and Nesterov optimal gradient [15]. For each test, each method is initialized with the same random matrices  $F$  and  $G$  and is run during<sup>4</sup> 60 s using Matlab R2018a on a laptop with an Intel Core i7-4800MQ Quad Core processor and 32 GB RAM memory. For each method, we consider two performance indices, i.e., the Relative Reconstruction Error (RRE) which is defined as

$$\text{RRE} \triangleq \|X^{\text{theo}} - G \cdot F\|_{\mathcal{F}}^2 / \|X^{\text{theo}}\|_{\mathcal{F}}^2, \quad (5)$$

and which is useful for validating the matrix-completion accuracy of the tested techniques, and the Signal-to-Interference Ratio (SIR) which reads

$$\text{SIR} = \sum_{j=1}^p 10 \log_{10} \left( \left\| \hat{f}_j^{\text{coll}} \right\|^2 / \left\| \hat{f}_j^{\text{orth}} \right\|^2 \right), \quad (6)$$

where  $\hat{f}_j^{\text{coll}}$  and  $\hat{f}_j^{\text{orth}}$  are the collinear and orthogonal parts of an estimated source vector  $\hat{f}_j$  to the associated theoretical vector in  $F^{\text{theo}}$ , respectively.

<sup>3</sup>This loop is run at most  $\text{Max}_{\text{iter}}$  times (or when a given bound in the gradient computations is reached), as in unweighted NeNMF [18].

<sup>4</sup>While not being classical in the literature, limiting the computations to a given available CPU time is a crucial constraint in practical applications.

We only use RSIs in these tests and we set  $q$  to  $q = 4$ . However, the number  $\text{Max}_{\text{Outer}}$  of outer iterations is not fixed but is set to 50 and 100, respectively, in order to investigate its effects to the WNMF 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  $\text{Max}_{\text{inner}} = 500$  updates or if the gradient term is 1000 times below the one estimated at the beginning of the inner loop.

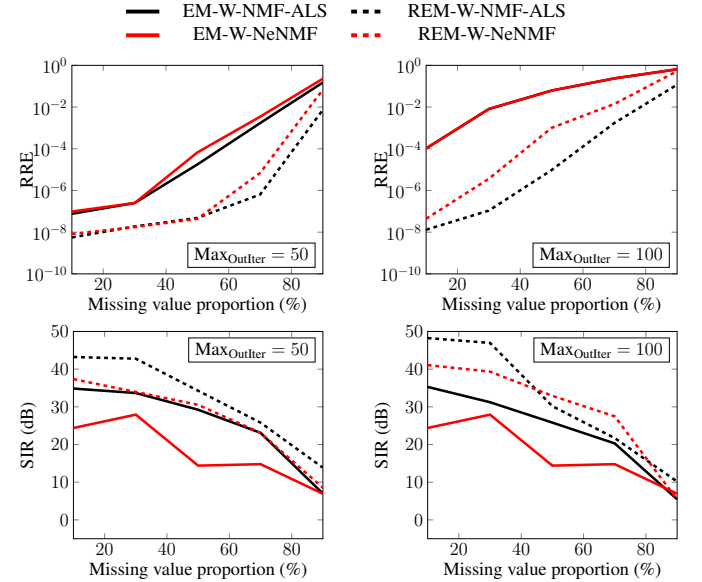


Figure 1: RRE (top) and SIR (bottom) versus the missing value proportion. (left): 50 outer loop iterations per M-step. (right): 100 iterations.

The obtained results are gathered on Fig. 1. The upper and lower plots show the obtained RREs and SIRs with respect to the missing value proportion, respectively, when the number of outer loops in the M-step is set to 50 (left) and 100 (right). The plain and dashed curves show the median performance of the vanilla EM-W-NMF and the proposed REM-W-NMF techniques, respectively. These plots show the relevance of the proposed methods, as except when the missing value proportion is high—i.e., 90%—the RREs (respectively the SIRs) obtained with the compressed methods are significantly lower (respectively higher) than those obtained with the vanilla ones.

It should be noticed that, as the random projections are applied at each E-step, the computational time needed by the proposed REM-W-NMF techniques at this stage ( $\approx 6.7$  s in these tests) is almost 3 times higher than for the original EM-W-NMF methods that they extend ( $\approx 2.4$  s in these tests). However, the median CPU time for 1 outer loop in the M-step is 10 and 100 times faster for the Nesterov-based and the ALS-based randomized extensions than for their corresponding vanilla methods, respectively. This implies that the proposed strategy does not speed-up classical EM-W-NMF techniques if the number of outer loops in the M-step is not high enough. This shows that the random projection stage is the bottleneck of the proposed strategy, which might be solved by using some specific hardware dedicated to these computations [28].

In future work, we will aim to extend these techniques to an informed framework in order to apply them to sensor calibration [12] or chemical source separation [11, 29]. We will also aim to validate their performance on real datasets.

## 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.
- [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*. SIAM, 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*. SIAM, 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] Y. Nesterov, “A method of solving a convex programming problem with convergence rate  $O(1/k^2)$ ,” in *Soviet Mathematics Doklady*, 1983, vol. 27, pp. 372–376.
- [25] 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.
- [26] 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.
- [27] Y. D. Kim and S. Choi, “Weighted nonnegative matrix factorization,” in *Proc. ICASSP’09*, April 2009, pp. 1541–1544.
- [28] 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.
- [29] G. Delmaire, M. Omidvar, M. Puigt, F. Ledoux, A. Limem, G. Roussel, and D. Courcot, “Informed weighted non-negative matrix factorization using  $\alpha\beta$ -divergence applied to source apportionment,” vol. 21, no. 3, March 2019.