Faster-than-fast NMF using random projections and Nesterov iterations

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 recently implemented in Nonnegative Matrix Factorization (NMF) to speed-up the NMF computations, with a negligible loss of performance. In this paper, we investigate the effects of such projections when the NMF technique uses the fast Nesterov gradient descent (NeNMF). We experimentally show the randomized subspace iteration to significantly speed-up NeNMF.**

1 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.,

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

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

using, e.g., Multiplicative Updates (MU) [2], Hierarchical Alternating Least Squares (HALS) [8], Alternating Nonnegative Least Squares (ANLS) [9], or Projected Gradient (PG) [10]. Additionally, some authors incorporated some extra-information in the NMF model [7], e.g., weights [11, 12], sparsity assumptions [13, 14], sum-to-one constraints [15], specific matrix structures [16, 17], or information [10, 14, 18–21]. With the Big Data era, computational time reduction of NMF is particularly investigated, e.g., through optimal solvers [22], distributed strategies [23], online estimation [24], or randomization [25–27]. The latter consists of reducing the size of some matrices through random projections—see, e.g., [28] for a comprehensive review—and thus to speed up the NMF computations[1]. However, the methods in [25–27] are based on MU, PG, or HALS. In this paper, we aim to investigate the benefits of compressing the data when a fast NMF solver using Nesterov iterations (NeNMF) [22] is used. Indeed, this approach was found to be among the fastest techniques in [29][2].

---

[1] The authors in [26, 27] also proposed randomized techniques for separable NMF, which is out of the scope of this paper.

[2] See for example the CPU-time consumptions of [29] at `https://github.com/andrewssobral/lrslibrary`.

The remainder of the paper reads as follows. We recall the principles of NMF with Nesterov gradient descent in Sect. 2. Section 3 introduces our proposed compressed NeNMF method[3] whose performance is investigated in Sect. 4. Lastly, we conclude and discuss about future directions in Sect. 5.

2 NMF with Nesterov iterations

We firstly briefly recall the principles of the NeNMF method using Nesterov optimal gradient [22]. As explained above, for a fixed $n \times m$ nonnegative data matrix X, NMF consists of finding both the $n \times p$ and $p \times m$ matrices G and F which provide the best low-rank approximation of X (1). NeNMF [22] iteratively and alternately solves (2) and (3) by applying in an inner loop the Nesterov accelerated gradient descent [30]. To update a factor, say F, the latter initializes $Y_0 \triangleq F^t$—where t is an NeNMF outer iteration index—and considers a series α_k defined as $\alpha_0 = 1$, and $\alpha_{k+1} = \frac{1+\sqrt{4\alpha_k^2+1}}{2}$, $\forall k \in \mathbb{N}$. For each inner loop index k, the Nesterov gradient descent then computes an update F_k of F with a single gradient descent of Y_k, and then slides it in the direction of F_{k-1}—with weights from the series α_k—to derive Y_{k+1}. Using the Karush-Kuhn-Tucker conditions, a stopping criterion—considering both a maximum number Max_{iter} of iterations and a gradient bound—is proposed in [22], thus yielding $F^{t+1} = Y_K$, where Y_K is the last iterate of the above inner iterative gradient descent. The same strategy is applied to G. As shown in [22, 29], NeNMF is among the fastest state-of-the-art NMF techniques and is less sensitive to the matrix size than classical techniques, e.g., MU or PG.

3 NeNMF with random projections

We now introduce the principles of random projections for NMF. Assuming that the original data matrix X is low-rank, the key idea consists of estimating a smaller matrix—with the same main properties than X—whose reduced size allows to fasten the computations. While random projections were initially proposed for singular value decompositions [28], they were more recently applied to NMF in, e.g., [25–27]. Starting from a target rank ν (with $p \leq \nu \ll \min(n, m)$), the initial random projection technique consists of drawing scaled[4] Gaussian random matrices $L \in \mathbb{R}^{\nu \times n}$ and $R \in \mathbb{R}^{m \times \nu}$ whose product on the left side of X and G, or on the right side of X and F, respectively, allows to compress the matrices. The whole strategy is 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, so that the update rules in Algorithm 1 are instances of semi-NMF [31]. Lastly, the NMF stopping criterion might be a target approximation error, e.g., a reached CPU time.

---

[3] The Matlab code used in this paper is available at `https://gogs.univ-littoral.fr/puigt/Faster-than-fast_NMF`.

[4] Denoting Ω a $\nu \times n$ random matrix, its scaled version L reads $L = \Omega/\sqrt{\nu}$.
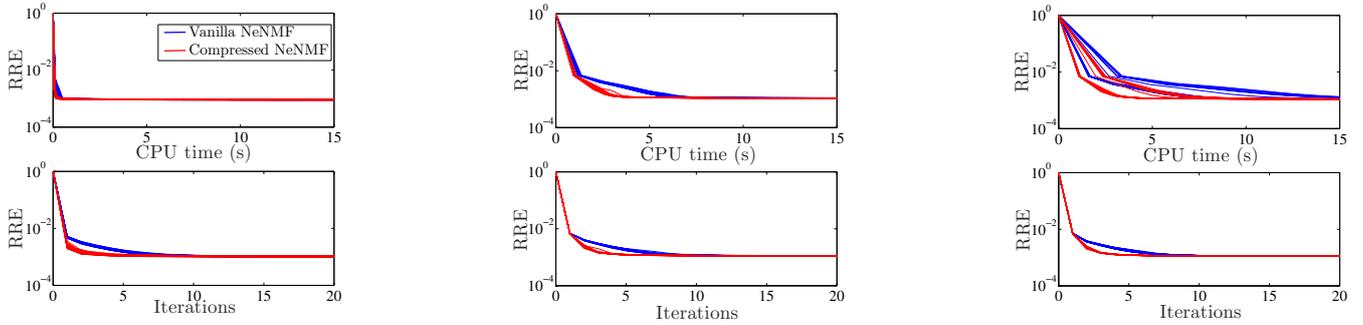
Figure 1: Vanilla and compressed NeNMF performance over CPU time (top) and iterations (bottom). Left: $n = 500$. Middle: $n = 5000$. Right: $n = 10000$.

Algorithm 1 Compressed NMF strategy

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

Actually, the design of randomized compression matrices L and R can be improved, e.g., using randomized power iterations or its stable variant named randomized subspace iteration [28]. Applied to NMF, the former was proposed in [26]—under the name of "structured random projection"—while the later is summarized in Algorithm 2. In this configuration, L and R exhibit orthonormal columns and rows, respectively. To the best of the authors' knowledge, randomized subspace iteration was never applied to NMF before and we investigate its behaviour in this paper.

As explained above, the NMF techniques applied to dense matrices X in [25–27] were using MU, PG, and HALS, respectively. In this paper, we propose to replace them by Nesterov iterations. The main interest of such an investigation consists of seeing whether/when random projections provide some benefits to NMF using an optimal solver.

4 Experimental validation

To investigate the performance of the proposed method, we draw random nonnegative matrices G and F, with $p = 15$ and $n = m$ (with $n \gg p$), such that X is a square low-rank matrix[5]. In our tests, we set n to 500, 5000, and 10000, respectively. For each tested value of n, we draw 40 different theoretical matrices G and F. We also add noise to the observed matrices so that the signal-to-noise ratio is around 30 dB.

We only investigate the performance of the NeNMF method without or with the randomized subspace iteration[6]. The target rank of the random matrices is set to $\nu = 25$. The tested meth-

---

Algorithm 2 Randomized subspace iterations for NMF

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.

---

ods are run during 15 s and, at each NMF iteration, we estimate the relative reconstruction error (RRE), defined as

$$\text{RRE} \triangleq ||X - G \cdot F||_{\mathcal{F}} / ||X||_{\mathcal{F}}, \qquad (4)$$

over the CPU time. All the methods are run using Matlab R2016a on a laptop with an Intel Core i7-4800MQ Quad Core processor, and 32 GB RAM memory.

Figure 1 shows the achieved performance by the vanilla NeNMF and its compressed extension, for the different tested values of n and when the maximum number of iterations per NeNMF inner loop is set to $\text{Max}_{\text{iter}} = 500$. In addition to the RRE evolution over the CPU time, we also plot the RRE versus the NMF iterations. The fast RRE decreasing which is visible in the early iterations in each plot shows the interest of applying random projections to an already fast NMF technique[7].

5 Conclusion

In this paper, we proposed an NMF method which combines random projections and optimal gradient descent. The proposed method is shown to be (much) faster than vanilla NeNMF. In future work, we aim to apply random projections to weighted [34] and informed [14] NMF.

---

[5] We also tested other ranks in some preliminary work without noticing any major differences.

[6] In preliminary tests, we compared its performance with MU-NMF, PG-NMF, and HALS-NMF and found it to be much faster. Moreover, we also found the standard random projections [25] to provide a lower enhancement than both other compression techniques, which is consistent with [26] for randomized power iteration. Lastly, we found the randomized subspace iteration to slightly outperform the randomized power iteration. Due to space restrictions, we cannot reproduce these results in this paper.

[7] It should be noticed that the compressed NeNMF is sensitive to the value of Max_{iter}. Indeed, when $\text{Max}_{\text{iter}} = 100$ in [32], both the compressed and vanilla NeNMF are faster (in terms of CPU time but not w.r.t. iterations) but the RRE is not always decreasing with the compressed version. This issue might be solved by adaptive restart strategies for example [33].

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. Lee and H. 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*, pp. 205–214, 2017.

[7] Y. X. Wang and Y. J. Zhang, "Nonnegative matrix factorization: A comprehensive review," *IEEE Trans. Knowl. Data Eng.*, vol. 25, pp. 1336–1353, June 2013.

[8] A. Cichocki, R. Zdunek, and S.-I. Amari, "Hierarchical ALS algorithms for nonnegative matrix and 3D tensor factorization," in *Proc. ICA'07*, pp. 169–176, 2007.

[9] J. Kim and H. Park, "Toward faster nonnegative matrix factorization: A new algorithm and comparisons," in *Proc. ICDM'08*, pp. 353–362, Dec 2008.

[10] C. Lin, "Projected gradient methods for nonnegative matrix factorization," *Neural Computation*, vol. 19, pp. 2756–2779, Oct 2007.

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

[12] D. Guillamet, J. Vitria, and B. Schiele, "Introducing a weighted non-negative matrix factorization for image classification," *Pattern Recognition Letters*, vol. 24, no. 14, pp. 2447–2454, 2003.

[13] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *J. Mach. Learn. Res.*, vol. 5, pp. 1457–1469, Dec. 2004.

[14] 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.*, to appear.

[15] H. Lantéri, C. Theys, C. Richard, and C. Févotte, "Split gradient method for nonnegative matrix factorization," in *Proc. EUSIPCO'10*, 2010.

[16] I. Meganem, Y. Deville, S. Hosseini, P. Déliot, and X. Briottet, "Linear-quadratic blind source separation using NMF to unmix urban hyperspectral images," *IEEE Trans. Signal Process.*, vol. 62, pp. 1822–1833, Apr. 2014.

[17] 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.

[18] J. Choo, C. Lee, C. K. Reddy, and H. Park, "Weakly supervised nonnegative matrix factorization for user-driven clustering," *Data Mining and Knowledge Discovery*, vol. 29, pp. 1598–1621, Nov. 2015.

[19] 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.

[20] M. Plouvin, A. Limem, M. Puigt, G. Delmaire, G. Roussel, and D. Courcot, "Enhanced NMF initialization using a physical model for pollution source apportionment," in *Proc. ESANN'14*, pp. 261–266, 2014.

[21] A. Limem, M. Puigt, G. Delmaire, G. Roussel, and D. Courcot, "Bound constrained weighted NMF for industrial source apportionment," in *Proc. MLSP'14*, 2014.

[22] N. Guan, D. 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.

[23] 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.

[24] 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.

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

[26] M. Tepper and G. Sapiro, "Compressed nonnegative matrix factorization is fast and accurate," *IEEE Trans. Signal Process.*, vol. 64, pp. 2269–2283, May 2016.

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

[28] 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.

[29] A. Sobral, T. Bouwmans, and E.-H. Zahzah, "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., 2015.

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

[31] C. H. 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.

[32] F. Yahaya, M. Puigt, G. Delmaire, and G. Roussel, "Do random projections fasten an already fast NMF technique using Nesterov optimal gradient?." poster presented during the 2018 "Journée Régionale des Doctorants en Automatique", available online at `https://hal.archives-ouvertes.fr/hal-01831111`, 2018.

[33] B. O'Donoghue and E. Candès, "Adaptive restart for accelerated gradient schemes," *Foundations of computational mathematics*, vol. 15, no. 3, pp. 715–732, 2015.

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