ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2.
Available from http://www.i6doc.com/en/.

Image completion via nonnegative matrix factorization using HALS and B-splines

Cécile Hautecoeur and François Glineur*

Université catholique de Louvain - CORE and ICTEAM Institute
B-1348 Louvain-la-Neuve - Belgium

Abstract. When performing image completion, it is common to assume that images are smooth and low-rank, when viewed as matrices of pixel intensities. In this work, we use nonnegative matrix factorization to successively refine the image by representing alternatively rows and columns as smooth signals using splines. Previous work solved this model using an alternating direction method of multipliers. Instead, we propose to use a version of the hierarchical alternating least squares algorithm adapted to handle splines, and show in numerical experiments that it outperforms the existing method. Performance can be further improved by increasing progressively the size of used splines. We also introduce a non iterative algorithm using the same NMF approach, where factorization is computed in a fast and accurate way but for which convergence is harder to achieve.

1 Introduction

Nonnegative matrix factorization (NMF) is a powerful tool for data analysis that relies on expressing a nonnegative matrix as the product of two nonnegative low-rank matrices. This factorization is for instance able to filter noise by reducing the rank (or dimensionality) of the input matrix [1, 2]. It is quite popular for many applications, including image completion, where images are represented as matrices of nonnegative pixel intensities. The low rank assumption, combined to smoothness conditions, is used in many state-of-the art methods for image completion. The smoothness is enforced via regularization, either on the produced image [3], or on its factors [4]. Recently, Sadowski and Zdunek proposed a different technique where the smoothness constraint is imposed on one factor of the NMF problem by constraining it to be a linear combination of smooth elements, namely B-Splines [5]. This approach leads to promising results, but at the cost of a relatively high computational time. Our goal is to use a similar idea for image completion while keeping a low computational effort.

2 Image completion with smooth NMF

NMF with spline factors, denoted smooth-NMF, consists in the following problem: given a matrix $Y \in \mathbb{R}^{m\times n}$, a factorization rank r and a (discretized) B-Spline basis with d splines $S \in \mathbb{R}^{m\times d}$, find matrices $A \in \mathbb{R}^{m\times r}$ and $X \in \mathbb{R}^{r\times n}$

*This work was supported by the Fonds de la Recherche Scientifique - FNRS and the Fonds Wetenschappelijk Onderzoek - Vlaanderen under EOS Project no 30468160.

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

minimizing

$$||Y - AX||_F^2 \quad \text{such that} \quad A = SB \geq 0, X \geq 0. \tag{1}$$

To use smooth-NMF for image completion, Sadowski and Zdunek proposed in [5] Algorithm 1, inspired from [4]. This algorithm successively solves a smooth-NMF problem based on a guess Y of the image to be recovered. This guess is updated using the low-rank matrix AX produced by NMF, where known pixels are replaced by their true value. Iterations are stopped once Y is close enough to AX. To ensure smoothness on both the rows and the columns, smooth-NMF is performed on the image and on its transpose[1]. The error at line 8 is computed with respect to the initial guess of the image (Y) because the image can be smoother in its rows or its column, and the error can thus increase between line 5 and 7. Nevertheless, we want this error to decrease after each full iteration.

Algorithm 1 B-Splines-based Algorithm for Image Completion (BSA-IC)

Input: $M \in \mathbb{R}^{m \times n}$ is the incomplete matrix with support Ω. Y is a first guess for M, by default $0^{m \times n}$, $A \in \mathbb{R}^{m \times r}$ and $X \in \mathbb{R}^{r \times n}$ are initialized randomly if not provided. A is initialized to contain nonnegative splines.

function BSA-IC(M, Ω, Y, $A \geq 0$, $X \geq 0$, $\delta = 10^{-1}$)
2: $Y_\Omega = M_\Omega$, $Z = Y - AX$, $Z_{\bar{\Omega}} = 0$ ▷ Z contains the initial error
$e_2 = ||Z||$, $e_1 = e_2 + \delta + 1$
4: **while** $e_1 - e_2 > \delta$ **do**
$(A, X) =$ `Smooth-NMF`(Y, A, X) ▷ Smooth rows
6: $Y_2 = AX$, $Y_{2\ \Omega} = M_\Omega$
$(X^\top, A^\top) =$ `Smooth-NMF`$(Y_2^\top, X^\top, A^\top)$ ▷ Smooth columns
8: $Z = Y - AX$, $Z_{\bar{\Omega}} = 0$ ▷ Error with respect to Y
$e_1 = e_2$, $e_2 = ||Z||$
10: $Y = AX$, $Y_\Omega = M_\Omega$
end while
12: **return** Y
end function

To solve the smooth-NMF problem, the authors used the algorithm from [6], which solves the problem alternatively in matrices A and X. Matrix X is updated using the popular Hierarchical Alternating Least Squares (HALS) method [7], while matrix A is updated with and Alternating Direction Method of Multipliers (ADMM) [8]. As the HALS update of X is quite common, we only explain the ADMM update of A: it solves

$$\min_{A,B} \tfrac{1}{2}||Y - SBX||_F^2 + \Phi(A) \quad \text{such that} \quad SB = A \tag{2}$$

where $\Phi(A)$ is the indicator function of the nonnegative set ($\Phi(a) = 0$ if $a \geq 0$, $+\infty$ otherwise). Applying ADMM on this problem gives the following iterative

[1]Note that lines 5 to 7 where inferred from descriptions in [5], as the authors did not provide pseudocode of these elements. Moreover, the condition of the while loop has been slightly modified by removing the absolute value to stop the algorithm when the error increases.

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

scheme, using $[\xi]_+ = \max\{0, \xi\}$, $\tau > 0$ and the pseudo-inverse $S^\dagger$:

$$B_{k+1} = S^\dagger \big[YX^\top + \Lambda_k + \tau A_k\big]\big(XX^\top + \tau I_r\big)^{-1}$$
$$A_{k+1} = \big[SB_{k+1} - \tau^{-1}\Lambda_k\big]_+,\ \Lambda_{k+1} = \Lambda_k + \tau(A_{k+1} - SB_{k+1}).$$

Algorithm 1 is able to complete images properly, but at the cost of a high CPU time compared to existing approaches. We propose to tackle the smooth-NMF subproblem using S-HALS, a method introduced in [9] based on HALS. This method is potentially faster than the ADMM approach provided fast heuristics are used to project splines onto their nonnegative set. Indeed, it has been shown that using heuristics for the projection step could benefit the algorithm in terms of CPU time without impacting significantly its accuracy [10].

Our S-HALS method alternates between updates of matrix B (the coefficients of A = SB) and matrix X, as described in Algorithm 2. Projection in line 5 of `updateB` can be done exactly (see [9]), but instead we choose to compute it in an approximate but much faster way: we project each B-Spline coefficients on the nonnegative set, which ensures that the resulting spline is nonnegative (although not all nonnegative splines can be obtained in this way).

Algorithm 2 S-HALS

$$\Pi = S^\top S,\ Z = S^\top Y,\ \Pi_1 = \Pi^{-1} Z$$

1:	**function** UPDATEB(Π_1, B, X)	**function** UPDATEX(Z, Π, B, X)
2:	$P = \Pi_1 X^\top$, $Q = XX^\top$	$P = B^\top Z$, $Q = B^\top \Pi B$
3:	**for** $b_{:k}$ in B **do**	**for** $x_{k:}$ in X **do**
4:	$t = p_{:k} - \sum_{j \neq k} b_{:j} q_{jk}$	$t = p_{k:} - \sum_{j \neq k} q_{kj} x_{j:}$
5:	$b_{:k} \leftarrow$ `Projection`(t/q_{kk})	$x_{k:} \leftarrow \max(0, t/q_{kk})$
6:	**end for**	**end for**
7:	**return** B	**return** X
8:	**end function**	**end function**

3 Direct image completion (DIC) via HALS using splines

The image completion method described above relies on an input guess of the complete image, which is successively refined using smooth-NMF. Below we consider a different paradigm based on a single low-rank completion of the input.

Given a matrix $M \in \mathbb{R}^{m \times n}$ whose entries are known on support Ω, the problem of low-rank completion can be written as

$$\min_{A \in \mathbb{R}^{m \times r}, X \in \mathbb{R}^{r \times n}} C(A, X) = \sum_{(i,j) \in \Omega} (M_{i,j} - [AX]_{i,j})^2. \tag{3}$$

Since Algorithm 1 was alternating between imposing smoothness on the rows and on the columns of the image, we propose in our new model to impose both at the same time, by constraining both A and X to contain splines with nonnegative coefficients. We then solve (3) using a HALS approach similar to above,

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

applied to both factors and adapted to deal with an incomplete input matrix. Equating the gradient of the cost C with respect to $A_{p,q}$ to zero gives

$$\frac{\partial C}{\partial A_{p,q}} = -\sum_{(p,j)\in\Omega} 2(M_{p,j} - \sum_k A_{p,k}X_{k,j})X_{q,j}$$

$$\frac{\partial C}{\partial A_{p,q}} = 0 \Leftrightarrow A^*_{p,q} = \frac{\sum_{(p,j)\in\Omega}\left[M_{p,j} - \sum_{k\neq q} A_{p,k}X_{k,j}\right]X_{q,j}}{\sum_{(p,j)\in\Omega} X^2_{q,j}}$$

Values in column $A_{:,q}$ are independent from each other, so we can compute the optimal value $A^*_{:,q}$ of the whole column at once. However, we must also ensure that $A_{:,q}$ contains splines with nonnegative coefficients. Hence, in order to obtain the optimal update of column $A_{:,q}$, we must solve the following quadratic problem to find the optimal coefficients:

$$\text{argmin}_{g\in\mathbb{R}^d} g^\top S^\top DSg - 2A^{*,\top}_{:,q}DSg \quad \text{s.t.} \quad g \geq 0$$

where $S \in \mathbb{R}^{m\times d}$ contains the d considered B-Splines, and D is a diagonal matrix with diagonal components $d_p = \sum_{(p,j)\in\Omega} X^2_{q,j}$. The solution of this problem is approximated by projecting the best unconstrained coefficients ($g^* = (S^\top DS)^{-1}S^\top DA^*_{:,q}$) on the nonnegative set. The update of $A_{:,q}$ is then $S[g]_+$.

As the problem is symmetric in its two factors, we can use the fact that $Y^\top = X^\top A^\top$ to derive the updates of the rows of X from the updates of the columns of A. The algorithm iteratively updates A and X until the cost C stagnates ($\frac{C_{\text{previous}} - C_{\text{actual}}}{C_{\text{initial}}} < 10^{-3}$).

4 Experiments

We use cubic B-Splines in our tests on a 512×512 boat image (see Fig. 2) with 90% missing pixels. Choosing the number of B-splines to be used is an important component in all the proposed algorithms: one can either choose a fixed number (50 or 100), or perform a run with some number of splines and use its final image as input for a subsequent run with more splines (we tried $50 \rightarrow 100$ and $25 \rightarrow 50 \rightarrow 100$). Two more types of runs are reported: in an 'inner' run, as suggested in previous work [5], the number of B-splines is incremented progressively during the update of matrix A [6]. However we did not find that tweak, where the increase happens at each iteration of Algorithm 1, to be very effective. Instead we designed an 'outer' run, similar in spirit to the successive $50 \rightarrow 100$ and $25 \rightarrow 50 \rightarrow 100$ runs, where a different number of B-Splines is used at each iteration of Algorithm 1: we use $\min(3 * i + 10, 100)$ splines for the ith iteration (increasing the number of splines by 3 at each iteration prevents too quick stagnation).

Figure 1 (left) establishes that our methods (Alg. 1 with S-HALS, and DIC) are both more efficient than Alg. 1 with ADMM (up to four times as fast). The signal to interference ratio (SIR) with respect to the true image (right) is best for S-HALS whatever the number of splines used. We also see that

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

DIC does not really benefit from the subsequent runs in the $50 \to 100$ and $25 \to 50 \to 100$ cases, most probably because it does not make significant use of each intermediate input image. For this reason neither the 'inner' nor the 'outer' runs where tested on this algorithm. However, SIRs for both ADMM and S-HALS benefit from increasing progressively the number of splines and they benefit much more from 'outer increases' than from the 'inner' run. An explanation for this improvement could be that the image is first reconstructed from its low-rank elements, and then improved [4].

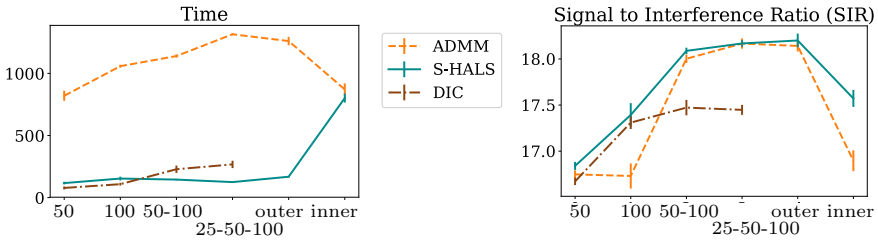

Fig. 1: Performance of the methods with varying number of B-Splines.

We also analyzed the methods with different choices of factorization rank r. In general, they obtain better results with higher rank, but at the cost of more computational effort. Moreover, the benefits for $r > 50$ are very small. Worse, the DIC method is not always able to converge when r is high ($r > 100$, see also discussion below). We also observed the influence of the percentage of missing pixels, and saw that the three methods are still able to recover a quite accurate image (with a SIR around 15.5 dB) even when 95% of pixels are missing.

Finally, Figure 2 displays several examples of recovered images with 90 % missing pixels, with rank fixed to 50 for the 512×512 grayscale image and 25 for the 256×256 color images (for which algorithms were applied on each color channel independently). ADMM as well as S-HALS use an 'outer' run with a maximal number of splines equal to 100. We observe on all three images that S-HALS is significantly faster and slightly more accurate than ADMM, while DIC is the fastest method but also the least accurate.

5 Conclusion and discussion

The presented methods work with fixed rank. A way to find the best rank is to increase it progressively until the image is close enough to the known pixels [4]. This approach is quite easy to implement for Algorithm 1, and at each step the algorithm will take advantage from the image computed at the previous iteration. On other hand the DIC algorithm must be modified to better exploit information from a previous completion, which we leave for further research. This modification would also probably allow the algorithm to take advantage from a progressive increase in the number of splines.

Extending our approaches to tensors is another interesting topic for future work, which could be applied to colored images for which pixel intensities in each color channel are not independent, unlike what we did with our algorithms.

ESANN 2020 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Online event, 2-4 October 2020, i6doc.com publ., ISBN 978-2-87587-074-2. Available from http://www.i6doc.com/en/.

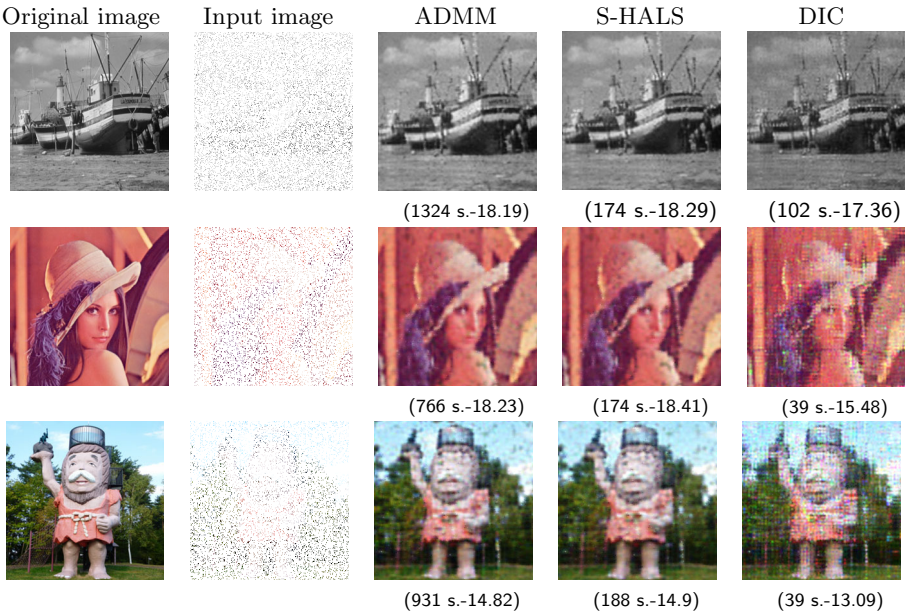

Fig. 2: Example of recovered images for 90% of missing pixels. (CPU time - SIR).

References

[1] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788, 1999.

[2] Andrzej Cichocki, Rafal Zdunek, Anh Huy Phan, and Shun-ichi Amari. *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009.

[3] Yipeng Liu, Zhen Long, and Ce Zhu. Image completion using low tensor tree rank and total variation minimization. *IEEE Transactions on Multimedia*, 21(2):338–350, 2018.

[4] Tatsuya Yokota, Qibin Zhao, and Andrzej Cichocki. Smooth parafac decomposition for tensor completion. *IEEE Transactions on Signal Processing*, 64(20):5423–5436, 2016.

[5] Tomasz Sadowski and Rafał Zdunek. Image completion with smooth nonnegative matrix factorization. In *International Conference on Artificial Intelligence and Soft Computing*, pages 62–72. Springer, 2018.

[6] Rafał Zdunek. Alternating direction method for approximating smooth feature vectors in nonnegative matrix factorization. In *2014 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2014.

[7] Andrzej Cichocki, Rafal Zdunek, and Shun-ichi Amari. Hierarchical ALS algorithms for nonnegative matrix and 3D tensor factorization. In *International Conference on Independent Component Analysis and Signal Separation*, pages 169–176. Springer, 2007.

[8] Tom Goldstein, Brendan O'Donoghue, Simon Setzer, and Richard Baraniuk. Fast alternating direction optimization methods. *SIAM Journal on Imaging Sciences*, 7(3):1588–1623, 2014.

[9] Cécile Hautecoeur and François Glineur. Nonnegative matrix factorization over continuous signals using parametrizable functions. *Neurocomputing*, 2019.

[10] Cécile Hautecoeur and François Glineur. Accelerating nonnegative matrix factorization over polynomial signals with faster projections. In *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2019.