Time series recovery from partial observations via Non-negative Matrix Factorization

Anonymous authors
Paper under double-blind review

Abstract

In modern time series problems, one aims at forecasting multiple times series with possible missing and noisy values. In this paper, we introduce the Sliding Mask Method (SMM) for forecasting multiple nonnegative time series by means of nonnegative matrix completion: observed noisy values and forecast/missing values are collected into matrix form, and learning is achieved by representing its rows as a convex combination of a small number of nonnegative vectors, referred to as the archetypes. We introduce two estimates, the mask Archetypal Matrix factorization (mAMF) and the mask normalized Nonnegative Matrix Factorization (mNMF) which can be combined with the SMM method. We prove that these estimates recover the true archetypes with an error proportional to the noise. We use a proximal alternating linearized method (PALM) to compute the archetypes and the convex combination weights. We compared our estimators with state-of-the-art methods (Transformers, LSTM, SARIMAX...) in multiple time series forecasting on real data and obtain that our method outperforms them in most of the experiments.

1 Introduction

This article investigates forecasting multiple nonnegative times series with missing or noisy entries. We observe $N \geq 1$ time series $\mathbf{M}^{(1)}, \ldots, \mathbf{M}^{(N)} \in \mathbb{R}^T$ over a period of time of length $T \geq 1$ and we would like to forecast the next $F \geq 1$ future values by means of matrix completion, see Figure 1.

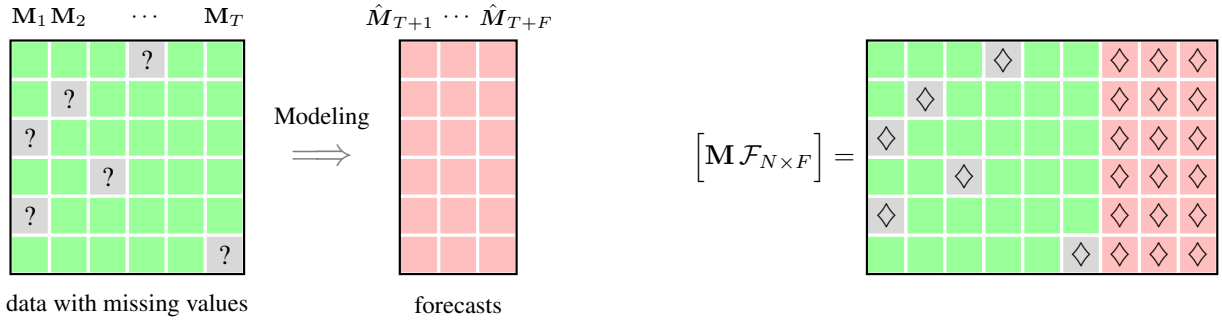

Figure 1: `[Left]` Consider noisy multiple time series observations (green) with possible missing entries (question mark) from $N \geq 1$ time series and their F forecast values in red. `[Right]` Matrix completion problem under consideration: missing and forecast values $\mathcal{F}_{N\times F}$ (gray and red diamonds) are not observed.

The matrix completion problem depicted on the right hand side of Figure 1 is ill-posed, it is not possible to complete the red values by any low-rank technique. To address this issue, we consider a linear transform $\mathbf{\Phi}$ defined as follows.

- We denote by $P \geq 1$ some time series length parameter referred to as the *periodicity*. However, our analysis does not assume any periodicity in the data and the practitioner is free to chose any value for P. If some periodicity appears in the data, she can choose P as a multiple of this periodicity. Note that our method works even if P is not chosen in this way.

- We partition the $T+F$ columns (right side of Figure 1) into B sub-blocks of length P so that $B = (T+F)/P$. Padding at most $P-1$ forecasts columns to the right of the matrix, one can assume, without loss of generality, that B is an integer.
- Now, the rows of the output matrix have length WP, gathering W consecutive row sub-blocks together. It amounts in sliding a window of length WP by jumps of length P on successive rows of the input matrix.
- We assume that W and P are such that $WP > F$. Up to row permutation, one obtain an output four blocks matrix whose bottom right block is the forecast values, see Figure 2.

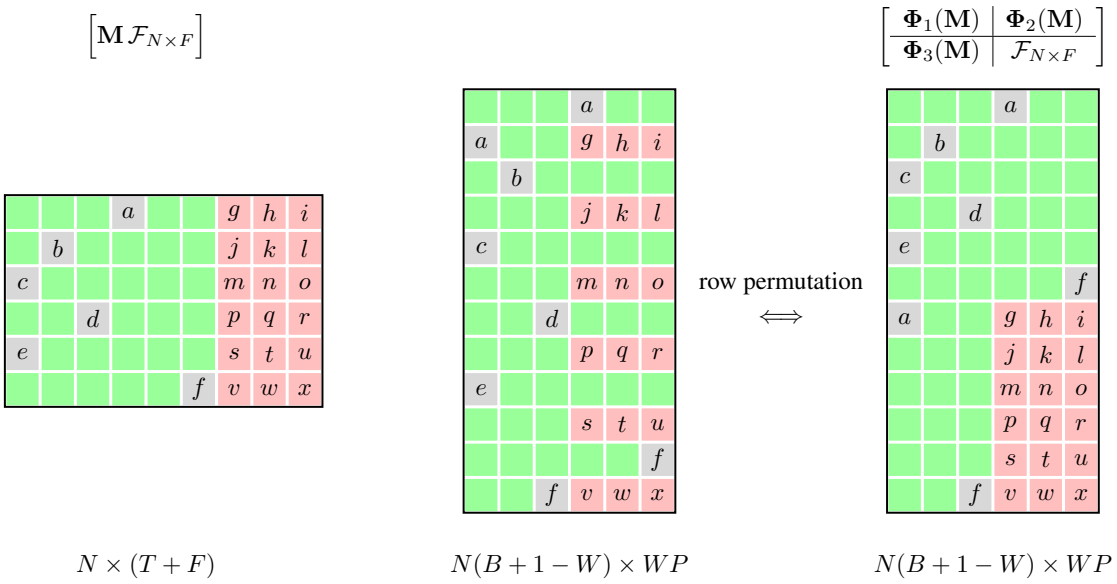

Figure 2: Given an input matrix of size $N \times (T+F)$, the output sliding mask matrix is composed of 4 blocks, the bottom right one being of size $N \times F$ with the F last columns of the input matrix, denoted by $\mathcal{F}_{N\times F}$. In this example, we consider a periodicity of $P = 3$, giving $B = 3$ sub-blocks per row of the input matrix and we gather $W = 2$ consecutive sub-blocks in an output row. To ease readability, we denote by $a, \ldots, f$ the missing values and by $g, \ldots, x$ the values to forecast. After row permutation, we obtain the output sliding mask matrix. The matrix $\mathbf{\Phi}_1(\mathbf{M})$ (resp. $\mathbf{\Phi}_2(\mathbf{M})$, $\mathbf{\Phi}_3(\mathbf{M})$) is a sub-matrix of size $N(B-W) \times (WP-F)$ (resp. $N(B-W) \times F$, $N \times (WP-F)$) of the input matrix $\mathbf{M}$.

Given the input multiple time series matrix $\mathbf{M}$ of size $N \times T$, define the observation matrix $\mathbf{X}$ as the matrix of size $N(B+1-W) \times WP$ obtained by dropping out the forecast and missing entries (gray and red entries in Figure 2) and keeping the observed values of $\mathbf{M}$ (green entries). Given any matrix $\mathbf{N}$ of size $N(B+1-W) \times WP$, the mask operator $\mathbf{T}(\mathbf{N})$ is defined by dropping out the forecast and missing entries (red and gray entries in Figure 2). By a slight abuse of notation, one has $\mathbf{X} = \mathbf{T}(\mathbf{\Phi}(\mathbf{M}))$.

We introduce the mask normalized Nonnegative Matrix Factorization (mNMF):

$$\min_{\substack{\mathbf{W1}=\mathbf{1},\mathbf{W}\geq\mathbf{0} \\ \mathbf{H}\geq\mathbf{0} \\ \mathbf{T}(\mathbf{N})=\mathbf{X}}} \|\mathbf{N} - \mathbf{WH}\|_F^2 \,, \tag{mNMF}$$

where $\|\cdot\|_F$ is the Frobenius norm, $\mathbf{1}$ is the vector of ones, *i.e.* solutions $\mathbf{N}$ are such that $\mathbf{T}(\mathbf{N}) = \mathbf{X}$, the values of $\mathbf{N}$ corresponding to the green entries in Figure 2 are equal to observed values. We also introduce the mask Archetypal Matrix Factorization (mAMF):

$$\min_{\substack{\mathbf{W}\geq\mathbf{0},\mathbf{W1}=\mathbf{1} \\ \mathbf{V}\geq\mathbf{0},\mathbf{V1}=\mathbf{1} \\ \mathbf{T}(\mathbf{N})=\mathbf{X}}} \|\mathbf{N} - \mathbf{WH}\|_F^2 + \lambda\|\mathbf{H} - \mathbf{VN}\|_F^2 \,, \tag{mAMF}$$

where $\lambda \geq 0$ is a tuning parameter.

The weight matrix $\mathbf{W}$ has size $n \times K$ where $n = N(B+1-W)$ and $K \geq 1$ is referred to as the nonnegative rank. The matrix $\mathbf{W}$ satisfies the constraint $\mathbf{W} \geq \mathbf{0}$ and $\mathbf{W1} = \mathbf{1}$, this being later referred to as *normalization*. Its rows $(w_{i,1}, \ldots, w_{i,K})$ are convex combination weights and each row of $\mathbf{WH}$ is a convex combination of the K rows of $\mathbf{H}$. The matrix $\mathbf{H}$ has size $K \times p$ where $p = WP$, referred to as the nonnegative basis (resp. archetypes basis) in (mNMF) (resp. (mAMF)). Hence we get the following decomposition of the i^{th} row of $\mathbf{WH}$,

$$(\mathbf{WH})^{(i)} = \sum_{k=1}^{K} w_{i,k} \mathbf{H}_k \,. \tag{1}$$

Forecasts $\hat{\mathbf{M}} \in \mathbf{R}^{N \times F}$ are given by the bottom right $N \times F$ sub-matrix of $\mathbf{WH}$, see Figure 2.

1.1 Mask nonnegative matrix completion statistical guarantees

Consider the *mask* operator $\mathbf{T}(\mathbf{N})$ that sets to zero $N \times F$ values of a $n \times p$ matrix $\mathbf{N}$. Namely, given $\mathbf{N} \in \mathbb{R}^{n \times p}$, we define

$$\mathbf{T}(\mathbf{N}) = \left[\begin{array}{c|c} \mathbf{N}_1 & \mathbf{N}_2 \\ \hline \mathbf{N}_3 & \mathbf{0}_{N \times F} \end{array}\right] \quad \text{and} \quad \mathbf{T}^{\perp}(\mathbf{N}) = \left[\begin{array}{c|c} \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{N}_4 \end{array}\right] \quad \text{where } \mathbf{N} = \left[\begin{array}{c|c} \mathbf{N}_1 & \mathbf{N}_2 \\ \hline \mathbf{N}_3 & \mathbf{N}_4 \end{array}\right] .$$

For sake of readability, we did not consider missing values in the mask operator (entries $a, \ldots, f$ in Figure 2), but our results easily extend to them, changing the definition of $\mathbf{T}$ by also zeroing out entries corresponding to the missing values of $\mathbf{\Phi}(\mathbf{M})$.

Our goal is to solve the following nonnegative matrix completion problem: We observe a matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ containing the multiple time values, with $n = N(B+1-W)$ and $p = WP$, given by the transformation presented in Figure 2. The missing values and the forecast values are arbitrarily set to zero, *i.e.*, $\mathbf{T}^{\perp}(\mathbf{X}) = \mathbf{0}$. This choice is not restrictive since the values of $\mathbf{X}$ corresponding to the missing and forecast entries are not observed and since our study is insensitive to the values of these entries. Given a normalized nonnegative rank K approximation of $\mathbf{X}$, we introduce:

$$\mathbf{X}_0 := \mathbf{W}_0 \mathbf{H}_0 \in \arg \min_{\substack{\mathbf{W}_0 \mathbf{1} = \mathbf{1} \\ \mathbf{W}_0 \geq \mathbf{0} \\ \mathbf{H}_0 \geq \mathbf{0}}} \left\{ \left\| \mathbf{X} - \mathbf{T}(\mathbf{W}_0 \mathbf{H}_0) \right\|_F^2 \right\}, \tag{2a}$$

where $\mathbf{W}_0 \in \mathbb{R}^{n \times K}$ and $\mathbf{H}_0 \in \mathbb{R}^{K \times p}$. As K grows, the approximation error $\|\mathbf{X} - \mathbf{T}(\mathbf{W}_0 \mathbf{H}_0)\|_F$ decreases. The matrix $\mathbf{X}_0$ is referred to as the best normalized nonnegative rank K approximation of $\mathbf{X}$. The goal is to recover the matrices $\mathbf{W}_0$ (weights) and $\mathbf{H}_0$ (archetypes) from the observation matrix $\mathbf{X}$. The observation can be written as

$$\mathbf{X} = \mathbf{T}(\mathbf{X}_0) + \mathbf{F} \,, \tag{2b}$$

where $\mathbf{F}$ is some additive error term, referred to as the noise.

Contributions SMM inputs the forecast values and it can be viewed as a nonnegative matrix completion algorithm under low nonnegative rank assumption. This framework raises two issues. A first question is the uniqueness of the decomposition, also referred to as *identifiability* of the model. In Theorem 3, we introduce a new condition that ensures uniqueness from partial observation of the target matrix. An other challenge, as pointed out by Vavasis (2009) for instance, is that solving *exactly* NMF decomposition problem is NP-hard. Nevertheless NMF-type problems can be solved efficiently using (accelerated) proximal gradient descent method Parikh & Boyd (2013) for block-matrix coordinate descent in an *alternating projection scheme*, *e.g.*, Javadi & Montanari (2020a) and references therein. We rely on these techniques to introduce algorithms inputting the forecast values based on NMF decomposition, see Section 3. Theorem 6 complements the theoretical analysis by proving the robustness of NMF-type algorithms when entries are missing or corrupted by noise. Our main theoretical contributions are as follows:

- A uniqueness decomposition result (Theorem 3) showing that the decomposition $\mathbf{W}_0 \mathbf{H}_0$ is unique *given partial observations*, namely

$$\text{If} \quad \mathbf{T}(\mathbf{WH}) = \mathbf{T}(\mathbf{W}_0 \mathbf{H}_0) \quad \text{then} \quad (\mathbf{W}, \mathbf{H}) \equiv (\mathbf{W}_0, \mathbf{H}_0) \,, \tag{$\mathbb{P}_u$}$$

  where $\equiv$ means up to positive scaling and permutation[1].

[1] if an entry-wise nonnegative pair $(\mathbf{W}, \mathbf{H})$ is given then $(\mathbf{WPD}, \mathbf{D}^{-1}\mathbf{P}^{\top}\mathbf{H})$ is also a nonnegative decomposition $\mathbf{WH} = \mathbf{WPD} \times \mathbf{D}^{-1}\mathbf{P}^{\top}\mathbf{H}$, where $\mathbf{D}$ scales and $\mathbf{P}$ permutes the columns (resp. rows) of $\mathbf{W}$ (resp. $\mathbf{H}$)

Algorithms	mAMF		mNMF		BasisFormer		RFR		EXP		SARIMAX		LSTM		GRU	
Metrics	RRMSE	RMPE	RRMSE	RMPE	RRMSE	RMPE	RRMSE	RMPE	RRMSE	RMPE	RRMSE	RMPE	RRMSE	RMPE	RRMSE	RMPE
daily electricity	14.42%	36.85%	15.86%	46.66%	**7.56%**	**6.64%**	12.16%	47.78%	11.25%	43.83%	9.85%	43.16%	12.42%	46.49%	12.03%	45.90%
weekly electricity	14.80%	17.50%	11.09%	13.79%	8.76%	9.07%	**7.25%**	8.61%	10.07%	7.98%	9.05%	**7.42%**	27.85%	15.64%	26.04%	15.92%
gas	**21.71%**	**18.55%**	37.46%	42.79%	57.45%	52.10%	66.80%	71.61%	63.35%	68.16%	45.58%	52.83%	62.97%	68.38%	62.87%	67.90%
Istanbul	15.67%	17.80%	**14.18%**	16.77%	14.83%	**12.54%**	15.37%	18.32%	15.46%	18.64%	14.75%	17.01%	16.22%	20.96%	20.01%	26.87%
ETTh1	**10.24%**	15.23%	12.30%	14.16%	14.57%	**13.61%**	12.96%	17.98%	12.37%	13.65%	13.36%	15.94%	14.86%	18.78%	14.71%	18.85%
ETTh2	9.42%	13.07%	**4.87%**	**6.66%**	54.66%	53.66%	6.47%	7.60%	14.06%	13.67%	12.76%	13.03%	14.17%	13.75%	14.44%	14.36%
ETTm1	10.12%	15.22%	**9.94%**	**12.25%**	13.58%	12.31%	12.81%	17.42%	11.45%	14.20%	12.29%	16.45%	13.39%	17.96%	14.13%	18.63%
ETTm2	8.19%	11.65%	**5.08%**	7.41%	55.52%	54.95%	5.81%	**7.16%**	13.18%	12.88%	13.16%	12.95%	14.29%	13.89%	14.46%	14.03%
electricity1	**6.59%**	13.17%	11.11%	15.28%	26.93%	28.19%	12.75%	16.09%	38.27%	34.44%	>100.00%	>100.00%	8.51%	10.13%	7.19%	**9.61%**
electricity2	**8.09%**	16.82%	8.82%	**12.17%**	35.38%	39.73%	12.05%	15.67%	47.40%	40.36%	43.38%	38.98%	9.05%	12.83%	10.30%	13.28%
electricity3	**10.57%**	13.95%	12.43%	14.04%	34.30%	37.25%	12.45%	14.14%	40.37%	33.48%	37.05%	33.01%	10.70%	11.62%	10.70%	**11.02%**
electricity4	11.02%	24.30%	25.07%	29.71%	39.42%	40.66%	23.16%	19.50%	54.42%	43.63%	63.08%	46.59%	12.18%	13.88%	**9.53%**	**11.05%**
electricity5	9.52%	19.05%	**7.72%**	**15.48%**	46.22%	49.60%	25.96%	26.84%	56.76%	49.31%	*	*	21.92%	28.79%	20.73%	27.02%
electricity6	10.11%	17.04%	14.30%	18.62%	45.50%	46.86%	13.81%	16.26%	51.87%	37.35%	52.10%	40.56%	7.58%	11.74%	**7.13%**	**10.32%**
electricity7	**8.34%**	16.75%	37.49%	30.03%	40.17%	43.20%	29.51%	22.96%	53.00%	45.95%	*	*	17.74%	14.72%	16.55%	**14.19%**
electricity8	**10.03%**	17.49%	23.81%	20.59%	30.64%	30.99%	19.33%	17.98%	36.83%	40.46%	38.54%	41.23%	12.16%	**15.73%**	13.89%	17.08%
electricity9	19.45%	38.90%	21.15%	41.72%	34.88%	38.85%	18.18%	**37.53%**	35.90%	38.65%	>100.00%	>100.00%	**18.00%**	37.77%	18.80%	38.21%
electricity10	**5.13%**	12.53%	5.40%	11.29%	29.78%	31.79%	12.11%	13.42%	33.88%	34.89%	36.55%	38.92%	7.66%	10.25%	7.77%	**9.94%**
synthetic1	6.40%	9.30%	5.81%	**8.01%**	8.06%	12.32%	5.79%	9.44%	**5.73%**	9.32%	5.76%	8.81%	6.67%	11.87%	6.77%	11.89%
synthetic2	19.04%	20.28%	20.35%	25.30%	31.32%	47.88%	**17.12%**	**20.24%**	21.09%	25.87%	28.41%	35.53%	21.55%	29.04%	21.48%	28.94%
low-noise	**0.10%**	**0.26%**	0.10%	0.26%	23.71%	51.94%	8.65%	22.76%	16.97%	48.02%	0.19%	0.33%	16.52%	46.03%	16.76%	46.46%
medium-noise	2.41%	5.23%	**1.92%**	**4.81%**	21.68%	48.48%	8.42%	21.95%	15.94%	44.25%	1.97%	4.94%	15.66%	42.68%	15.67%	42.98%
high-noise	12.69%	28.04%	**10.39%**	**26.43%**	18.44%	45.77%	11.73%	30.26%	13.02%	33.27%	15.24%	27.37%	13.03%	33.67%	12.97%	33.47%

Figure 3: Best results in **bold**, and second best results underlined. Some values have not been evaluated due to numerical errors on matrix inversion, reported as a star symbol $*$. Datasets include: electricity consumption (daily, weekly, electricity1-10), gas sensor measurements (gas), stock exchange returns (Istanbul), electricity transformer temperatures (ETTh1, ETTh2, ETTm1, ETTm2), and various synthetic datasets, see Section 4 for details.

- A robustness result (Theorem 6) showing that (mNMF) and (mAMF) recover $\mathbf{H}_0$ with an error proportional to $\|\mathbf{F}\|_F$.

Our analysis is completed by an algorithmic and numerical study that

- introduces a Proximal Alternating Linearized Minimization (PALM) method to solve (mAMF) and shows that PALM reaches a stationary point (Theorem 7).

- reports a performance improvement of (mNMF) and (mAMF) against state-of-the-art algorithms on real datasets for RRMSE and RMPE (Table 3). The relative root-mean-squared error (RRMSE) and the relative mean-percentage error (RMPE) are defined by

$$\text{RRMSE} = \frac{\|\mathbf{M}_F - \mathbf{M}_F^\star\|_F}{\|\mathbf{M}_F^\star\|_F}\,,\ \text{RMPE} = \frac{\|\mathbf{M}_F - \mathbf{M}_F^\star\|_1}{\|\mathbf{M}_F^\star\|_1}\,.$$

  where $\mathbf{M}_F^\star$ are the true values and $\mathbf{M}_F$ the forecasts (see Section 4).

Comments on low rank modeling Sparse or Low-Rank representations are ubiquitous in applications and well studied in the literature. In our analysis a time series is cut into several smaller W sub-blocks time series with the same length $p = WP$. For instance, observing sales over a period of one year, one can consider 52 weekly time series (one per week). These observations are the rows of our observed matrix $\mathbf{X}$. The normalized nonnegative low rank hypothesis assumes that the p-length multiple time series of the dataset can be decomposed as a sum of K basis time series $\mathbf{H}$ plus an error term. Of course, this error term can incorporate the model approximation error as depicted in (2a). The K basis time series $\mathbf{H}$ are learned on the entire dataset $\mathbf{X}$. This technique can be seen as dimension reduction, each observation can be summarized as K weights $\mathbf{W}$ such that the resulting convex combination of basis time series (1) is a good approximation of the observation $\mathbf{X}$.

The relevance of such a hypothesis on real data cannot be proven beforehand. Our numerical study on real data shows that we improve results in prediction, better than standard methods in time series analysis: Seasonal AutoRegressive Integrated Moving Average with eXogenous variables model (SARIMAX), EXPonential moving average (EXP), Random Forest Regressor (RFR), Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), BasisFormer (Attention-based Time Series Forecasting with Learnable and Interpretable Basis). It suggests that the low rank assumption is reasonable for the datasets studied in the paper.

1.2 Related works

Our proposed method, the Sliding Mask Method (SMM) deeply exploits Nonnegative Matrix Factorization (NMF) approaches, see for instance Paatero & Tapper (1994) and Lee & Seung (1999). For further details, we refer the interested reader to the surveys Wang & Zhang (2013); Gillis (2015; 2017) and references therein. NMF has been widely used in the contexts of document analysis Xu et al. (2003); Essid & Fevotte (2013), hidden Markov chain Fu et al. (1999), representation learning Lee & Seung (1999), community discovery Wang et al. (2011), and clustering Turkmen (2015). Forecasting time series has been previously done before through a mixed linear regression and matrix factorization as in Yu et al. (2016), matrix completion for one temporal time series as in Gillard & Usevich (2018), tensor factorization as in de Araujo et al. (2017); Yokota et al. (2018); Tan et al. (2016), and multiple time series with auto-corelation regularization Mei et al. (2017) or side information Mei et al. (2018).

Uniqueness of NMF can be tracked back to Thomas (1974); Donoho & Stodden (2004); Recht et al. (2012) with a simplicial polyhedral cone analysis. This paper extends this analysis to matrix completion framework, where some entries are not observed. Uniqueness and robustness of archetypal analysis have been studied in Javadi & Montanari (2020a) for simplicial polyhedral cone approximation of a dataset, denoted in data matrix form by $\mathbf{X} \in \mathbb{R}^{n\times p}$ in this paper. This paper extends this latter analysis to the case where some data entries might be missing and some data blocks are not observed (forecast, red values in Fig. 2).

Notation Denote by $\mathbf{A}^\top$ the transpose of matrix $\mathbf{A}$. Denote by $\|\mathbf{A}\|_F^2 = \mathrm{tr}(\mathbf{A}^\top\mathbf{A})$ the Frobenius norm of $\mathbf{A}$. We use $\mathbb{R}_+^{n\times p}$ to denote $n \times p$ nonnegative matrices. It would be useful to consider the columns description $A_k \in \mathbb{R}^{n_1}$ of matrix $\mathbf{A} = [A_1 \cdots A_{n_2}]$ and the row decomposition $A^{(k)} \in \mathbb{R}^{n_2}$ of a matrix $\mathbf{A}$ using $\mathbf{A}^\top = [(A^{(1)})^\top \cdots (A^{(n_1)})^\top]$ for $\mathbf{A} \in \mathbb{R}^{n_1\times n_2}$. Notation $A_{i,j}$ indicates the elements of matrix $\mathbf{A}$; $[n]$ represents the set $\{1, 2, \ldots, n\}$; $\mathbf{1}_d$ is the all-ones vector of size d; and $\mathbb{1}_{\mathcal{A}}$ is the indicator function of $\mathcal{A}$, such that $\mathbb{1}_{\mathcal{A}} = 0$ if condition $\mathcal{A}$ is verified, ∞ otherwise.

Proofs Unless otherwise stated, all the proofs are given in Supplement Material.

2 Uniqueness and estimation guarantees

2.1 The train and test paradigm, link with forecasting multiple nonnegative time series

The model under consideration is presented in Equations (2). Our goal is to estimate the K-best normalized nonnegative approximation $\mathbf{X}_0$, defined in Equation (2a), from the partial and noisy observation $\mathbf{X}$. We denote by $\mathbf{X}^\star$ is the *mask* of $\mathbf{X}_0$, namely

$$\mathbf{X}^\star := \mathbf{T}(\mathbf{X}_0) = \left[\begin{array}{c|c} \mathbf{X}_1 & \mathbf{X}_2 \\ \hline \mathbf{X}_3 & \mathbf{0}_{N\times F} \end{array}\right], \tag{3a}$$

where $\mathbf{X}_1 \in \mathbb{R}^{(n-N)\times(p-F)}$, $\mathbf{X}_2 \in \mathbb{R}^{(n-N)\times F}$, and $\mathbf{X}_3 \in \mathbb{R}^{N\times(p-F)}$ are blocks of $\mathbf{X}_0$. Note that $\mathbf{X} = \mathbf{X}^\star + \mathbf{F}$, where $\mathbf{F}$ is the *noise* term, see Equation (2b).

These blocks can be be gathered into a train and test paradigm. Note that we observe the full sub-matrix $\mathbf{T}_{\mathrm{train}}(\mathbf{X}_0) := [\mathbf{X}_1\ \mathbf{X}_2]$ which we refer to as the training part. We would like to predict (forecast) the $\mathbf{0}_{N\times F}$ block of the sub-matrix $\mathbf{T}_{\mathrm{test}}(\mathbf{X}_0) := [\mathbf{X}_3\ \mathbf{0}_{N\times F}]$ which we refer to as the test part of our observation $\mathbf{X}$. Looking at Figure 2, we define

$$\mathbf{T}_T(\mathbf{X}_0) := \left[\begin{array}{c} \mathbf{X}_1 \\ \mathbf{X}_3 \end{array}\right] \quad \text{and} \quad \mathbf{T}_F(\mathbf{X}_0) := \left[\begin{array}{c} \mathbf{X}_2 \\ \mathbf{0}_{N\times F} \end{array}\right]. \tag{3b}$$

Our notation (subscripts T and F) stems from the sliding mask method for multiple time series forecast. Note that $\mathbf{T}_T(\mathbf{X})$ gathers all the information observed up to time T, and we would like to forecast the $\mathbf{0}_{N\times F}$ block of $\mathbf{T}_F(\mathbf{X})$. Now, we know by design that $\mathbf{X}_0 := \mathbf{W}_0\mathbf{H}_0$. Hence, denoting $\mathbf{H}_0 =: [\mathbf{H}_{0T}\ \mathbf{H}_{0F}]$, and $\mathbf{W}_0^\top =: [\mathbf{W}_{0\mathrm{train}}^\top\ \mathbf{W}_{0\mathrm{test}}^\top]$, we get that

$$\begin{aligned} \mathbf{T}_{\mathrm{train}}(\mathbf{X}_0) &= \mathbf{W}_{0\mathrm{train}}\mathbf{H}_0\,, & \mathbf{X}_3 &= \mathbf{W}_{0\mathrm{test}}\mathbf{H}_{0T}\,, \\ \mathbf{T}_T(\mathbf{X}_0) &= \mathbf{W}_0\mathbf{H}_{0T}\,, & \mathbf{X}_2 &= \mathbf{W}_{0\mathrm{train}}\mathbf{H}_{0F}\,. \end{aligned} \tag{3c}$$

In light of Figures 1 and 2, the multiple forecasts $\hat{\mathbf{M}}_{T+1}, \ldots, \hat{\mathbf{M}}_{T+F}$ are given by the estimation of $\mathbf{W}_{0\text{test}}\mathbf{H}_{0F}$. Observe that an estimation of $\mathbf{W}_{0\text{test}}$ gives the weights learnt on the test sub-matrix while an estimation of $\mathbf{H}_{0F}$ is the forecast of the archetypes, see the decomposition (1).

2.2 Uniqueness from partial observations

When we observe the full matrix $\mathbf{X}_0 = \mathbf{W}_0\mathbf{H}_0$, the issue on uniqueness has been addressed under some sufficient conditions on $\mathbf{W}, \mathbf{H}$, *e.g.*, *Strongly boundary closeness* of Laurberg et al. (2008), *Complete factorial sampling* of Donoho & Stodden (2004), and *Separability* of Recht et al. (2012). A necessary and sufficient condition exists as given by the following theorem. We recall that that the K-dimensional positive orthant is the set $\{x \in \mathbf{R}^K \, : \, x_i \geq 0\,, \, \forall i \in [K]\}$ and a K-simplicial cone is the set which the conic hull of K linearly independent vectors of $\mathbf{R}^K$.

Theorem 1 (Thomas (1974)) *The decomposition $\mathbf{X}_0 := \mathbf{W}_0\mathbf{H}_0$ is unique up to permutation and positive scaling of columns (resp. rows) of $\mathbf{W}_0$ (resp. $\mathbf{H}_0$)* **if and only if** *the K-dimensional positive orthant is the only K-simplicial cone verifying $\mathrm{Cone}(\mathbf{W}_0^\top) \subseteq \mathcal{C} \subseteq \mathrm{Cone}(\mathbf{H}_0)$ where $\mathrm{Cone}(\mathbf{A})$ is the cone generated by the rows of $\mathbf{A}$.*

Our first assumption is following.

Assumption 1 *In the set given by the union of sets:*

$$\{\mathcal{C} \, : \, \mathrm{Cone}({\mathbf{W}_{0\text{train}}}^\top) \subseteq \mathcal{C} \subseteq \mathrm{Cone}(\mathbf{H}_0)\} \bigcup \{\mathcal{C} \, : \, \mathrm{Cone}(\mathbf{W}_0^\top) \subseteq \mathcal{C} \subseteq \mathrm{Cone}(\mathbf{H}_{0T})\}\,, \tag{$\mathbb{A}_1$}$$

the nonnegative orthant is the only K-simplicial cone. Note that this assumption is implied by the following one: In the set

$$\{\mathcal{C} \, : \, \mathrm{Cone}({\mathbf{W}_{0\text{train}}}^\top) \subseteq \mathcal{C} \subseteq \mathrm{Cone}(\mathbf{H}_{0T})\} \tag{$\mathbb{A}_1'$}$$

the nonnegative orthant is the only K-simplicial cone.

We consider the following standard definition.

Definition 2 (Javadi & Montanari (2020a)) *The convex hull $\mathrm{conv}(\mathbf{X}_0)$ has an internal radius $\mu > 0$ if it contains an $K-1$ dimensional ball of radius μ.*

Our second main assumption is the following.

Assumption 2 *Assume that*

$$\mathrm{conv}(\underbrace{\mathbf{T}_{\text{train}}(\mathbf{X}_0)}_{=\mathbf{W}_{0\text{train}}\mathbf{H}_0}) \text{ has internal radius } \mu > 0\,. \tag{$\mathbb{A}_2$}$$

Theorem 3 *Assumption 1 implies Property ($\mathbb{P}_\text{u}$). Moreover, if ($\mathbb{A}_1$) and ($\mathbb{A}_2$) holds, $\mathbf{T}(\mathbf{WH}) = \mathbf{T}(\mathbf{W}_0\mathbf{H}_0)$ and $\mathbf{W}_0\mathbf{1} = \mathbf{W}\mathbf{1} = \mathbf{1}$ then $(\mathbf{W}, \mathbf{H}) = (\mathbf{W}_0, \mathbf{H}_0)$ up to permutation of columns (resp. rows) of $\mathbf{W}$ (resp. $\mathbf{H}$), and there is no scaling.*

Corollary 4 *If decomposition of $\mathbf{X}_1 = \mathbf{W}_{0\text{train}}\mathbf{H}_{0T}$ is unique then ($\mathbb{P}_\text{u}$) holds.*

Proof. By Theorem 1, observe that ($\mathbb{A}_1'$) is a necessary and sufficient condition for the uniqueness of the decomposition $\mathbf{X}_1 = \mathbf{W}_{0\text{train}}\mathbf{H}_{0T}$. Observe that ($\mathbb{A}_1'$) implies ($\mathbb{A}_1$) and invoke Theorem 1 and Theorem 3. ■

This corollary shows that the uniqueness of the decomposition of the top left block $\mathbf{X}_1$ (which is fully observed) implies the uniqueness of normalized decomposition of $\mathbf{X}_0$ given partial observations (the bottom right block is not observed).

2.3 Robustness under partial observations

The second issue is *robustness to noise*. To the best of our knowledge, all the results addressing this issue assume that the noise error term is small enough, *e.g.*, Laurberg et al. (2008), Recht et al. (2012), or Javadi & Montanari (2020a).

In this paper, we extend these stability result to the nonnegative matrix completion framework (partial observations) and we also assume that noise term $\|\mathbf{F}\|_F$ is small enough.

In the normalized case (*i.e.*, $\mathbf{W1} = \mathbf{1}$), both issues (uniqueness and robustness) can be handled with the notion of α-uniqueness, introduced by Javadi & Montanari (2020a). This notion does not handle the matrix completion problem we are addressing. To this end, let us introduce the following notation. Given two matrices $\mathbf{A} \in \mathbb{R}^{n_a \times p}$ and $\mathbf{B} \in \mathbb{R}^{n_b \times p}$ with same row dimension, and $\mathbf{C} \in \mathbb{R}^{n_a \times n_b}$, define the divergence $\mathcal{D}(\mathbf{A}, \mathbf{B})$ as

$$\begin{aligned}\mathcal{D}(\mathbf{A}, \mathbf{B}) &:= \min_{\mathcal{C} \geq \mathbf{0}\,,\, \mathcal{C}\mathbf{1}_{n_b} = \mathbf{1}_{n_a}} \sum_{a=1}^{n_a} \left\| A^{(a)} - \sum_{b=1}^{n_b} C_{ab} B^{(b)} \right\|_F^2 , \\ &= \min_{\mathcal{C} \geq \mathbf{0}\,,\, \mathcal{C}\mathbf{1}_{n_b} = \mathbf{1}_{n_a}} \|\mathbf{A} - \mathbf{C}\mathbf{B}\|_F^2 . \end{aligned} \tag{4a}$$

which is the squared distance between rows of $\mathbf{A}$ and $\mathrm{conv}(\mathbf{B})$, the convex hull of rows of $\mathbf{B}$. For $\mathbf{B} \in \mathbb{R}^{n \times p}$ define

$$\widetilde{\mathcal{D}}(\mathbf{A}, \mathbf{B}) := \min_{\substack{\mathbf{C} \geq \mathbf{0}\,,\, \mathbf{C}\mathbf{1}_n = \mathbf{1}_{n_a} \\ \mathbf{T}(\mathbf{N} - \mathbf{B}) = 0}} \|\mathbf{A} - \mathbf{C}\mathbf{N}\|_F^2 . \tag{4b}$$

Definition 5 ($\mathbf{T}_\alpha$-unique, Javadi & Montanari (2020a)) *Given $\mathbf{X}_0 \in \mathbb{R}^{n \times p}, \mathbf{W}_0 \in \mathbb{R}^{n \times K}$, and $\mathbf{H}_0 \in \mathbb{R}^{K \times p}$, the factorization $\mathbf{X}_0 = \mathbf{W}_0\mathbf{H}_0$ is $\mathbf{T}_\alpha$-unique with parameter $\alpha > 0$ if for all $\mathbf{H} \in \mathbb{R}^{K \times p}$ with $\mathrm{conv}(\mathbf{X}_0) \subseteq \mathrm{conv}(\mathbf{H})$:*

$$\widetilde{\mathcal{D}}(\mathbf{H}, \mathbf{X}_0)^{1/2} \geq \widetilde{\mathcal{D}}(\mathbf{H}_0, \mathbf{X}_0)^{1/2} + \alpha \left\{ \mathcal{D}(\mathbf{H}, \mathbf{H}_0)^{1/2} + \mathcal{D}(\mathbf{H}_0, \mathbf{H})^{1/2} \right\} . \tag{4c}$$

Our third main assumption is given by:

Assumption 3 *Assume that*

$$\mathbf{X}_0 = \mathbf{W}_0\mathbf{H}_0 \text{ is } \mathbf{T}_\alpha\text{-unique} \tag{$\mathbb{A}_3$}$$

Theorem 6 (Archetypes estimation) *If ($\mathbb{A}_2$) and ($\mathbb{A}_3$) hold then there exist positive reals Δ and Λ (depending on $\mathbf{X}_0$) such that, for all $\mathbf{F}$ such that $\|\mathbf{F}\|_F \leq \Delta$ and $0 \leq \lambda \leq \Lambda$, any solution $(\widehat{\mathbf{W}}, \widehat{\mathbf{H}})$ to (mAMF) (if $\lambda \neq 0$) or (mNMF) (if $\lambda = 0$) with observation (2b) is such that:*

$$\sum_{\ell \leq K} \min_{\ell' \leq K} \|\mathbf{H}_{0\ell} - \widehat{\mathbf{H}}_{\ell'}\|_2^2 \leq c\, \|\mathbf{F}\|_F^2 ,$$

where c is a constant depending only on $\mathbf{X}_0$.

By Theorem 6, when the noise is sufficiently small, there exists a permutation σ on $[K]$ such that

$$\|\mathbf{H}_0 - \hat{\mathbf{H}}_\sigma\|_F^2 := \sum_{\ell \leq K} \|\mathbf{H}_{0\ell} - \hat{\mathbf{H}}_{\sigma(\ell)}\|_2^2 \leq c\, \|\mathbf{F}\|_F^2 \tag{4d}$$

where $\hat{\mathbf{H}}_\sigma$ is a permutation of the row of $\hat{\mathbf{H}}$.

3 Solving masked nonnegative/archetypal matrix factorization

3.1 Alternating Least Squares for (mNMF)

The basic algorithmic framework for matrix factorization problems is *Block Coordinate Descent* (BCD) method, which can be straightforwardly adapted to (mNMF) (see Supplement Material). BCD for (mNMF) reduces to *Alternating Least Squares* (ALS) algorithm (see Algorithm 4 in Appendix), when an alternative minimization procedure is performed and matrix $\mathbf{WH}$ is projected onto the linear subspace $\mathbf{T}(\mathbf{N}) = \mathbf{X}$ by means of operator $\mathcal{P}_\mathbf{X}$, as follows:

$$\mathbf{N} := \mathcal{P}_\mathbf{X}(\mathbf{WH}) : \mathbf{T}(\mathbf{N}) = \mathbf{X} \text{ and } \mathbf{T}^\perp(\mathbf{N}) = \mathbf{WH} .$$

Hierarchical Alternating Least Squares (HALS) is an ALS-like algorithm obtained by applying an exact coordinate descent method Gillis (2014). Moreover, an accelerated version of HALS is proposed in Gillis & Glineur (2012) (see Supplement Material).

3.2 Projected Gradient for (mAMF)

Proximal Alternating Linearized Minimization (PALM) method, introduced in Bolte et al. (2014) and applied to AMF by Javadi & Montanari (2020a), can be also generalized to (mAMF) (see Algorithm 1).

Algorithm 1 PALM for mAMF

Initialization: chose $\mathbf{H}^0$, $\mathbf{W}^0 \geq \mathbf{0}$ such that $\mathbf{W}^0\mathbf{1} = \mathbf{1}$, set $\mathbf{N}^0 := \mathcal{P}_{\mathbf{X}}(\mathbf{W}^0\mathbf{H}^0)$ and $i := 0$.
while stopping criterion is not satisfied **do**
  $\widetilde{\mathbf{H}}^i := \mathbf{H}^i - \frac{1}{\gamma_1^i}{\mathbf{W}^i}^\top\left(\mathbf{W}^i\mathbf{H}^i - \mathbf{N}^i\right)$ ▷ Gradient step on $\mathbf{H}$, objective first term
  $\mathbf{H}^{i+1} := \widetilde{\mathbf{H}}^i - \frac{\lambda}{\lambda+\gamma_1^i}\left(\widetilde{\mathbf{H}}^i - \mathcal{P}_{\mathrm{conv}(\mathbf{N}^i)}(\widetilde{\mathbf{H}}^i)\right)$ ▷ Gradient step on $\mathbf{H}$, objective second term
  $\mathbf{W}^{i+1} := \mathcal{P}_{\Delta}\left(\mathbf{W}^i - \frac{1}{\gamma_2^i}\left(\mathbf{W}^i\mathbf{H}^{i+1} - \mathbf{N}^i\right){\mathbf{H}^{i+1}}^\top\right)$ ▷ Projected gradient step on $\mathbf{W}$
  $\mathbf{N}^{i+1} := \mathcal{P}_{\mathbf{X}}\left(\mathbf{N}^i + \frac{1}{\gamma_3^i}\left(\mathbf{W}^{i+1}\mathbf{H}^{i+1} - \mathbf{N}^i\right)\right)$ ▷ Projected gradient step on $\mathbf{N}$
  $i := i + 1$
end while

Where $\mathcal{P}_{\mathrm{conv}(\mathbf{A})}$ is the projection operator onto $\mathrm{conv}(\mathbf{A})$ and $\mathcal{P}_{\Delta}$ is the projection operator onto the $(N-1)$-dimensional standard simplex Δ^N. The two projections can be efficiently computed by means of, *e.g.*, Wolfe algorithm Wolfe (1976) and active set method Condat (2016) respectively.

Theorem 7 *If* $\gamma_1^i > \|{\mathbf{W}^i}^\top\mathbf{W}^i\|_F$, $\gamma_2^i > \max\left\{\|\mathbf{H}^{i+1}{\mathbf{H}^{i+1}}^\top\|_F, \varepsilon\right\}$ *for some* $\varepsilon > 0$, *and* $\gamma_3^i > 1$, *for each iteration* i, *then the sequence* $\left(\mathbf{H}^i, \mathbf{W}^i, \mathbf{N}^i\right)$ *generated by Algorithm 1 converges to a stationary point of* $\Psi(\mathbf{H}, \mathbf{W}, \mathbf{N}) := f(\mathbf{H}) + g(\mathbf{W}) + p(\mathbf{N}) + h(\mathbf{H}, \mathbf{W}, \mathbf{N})$, *where:*

$$f(\mathbf{H}) = \lambda\mathcal{D}(\mathbf{H}, \mathbf{N})\,, \qquad g(\mathbf{W}) = \textstyle\sum_{k=1}^{K} \mathbb{1}_{\{W_k \in \Delta\}}\,,$$
$$p(\mathbf{N}) = \mathbb{1}_{\{\mathbf{N} = \mathcal{P}_{\mathbf{X}}(\mathbf{WH})\}}\,, \qquad h(\mathbf{H}, \mathbf{W}, \mathbf{N}) = \|\mathbf{N} - \mathbf{WH}\|_F^2\,.$$

Proof. Proof is given in Supplement Material. ■

Finally, the inertial PALM (iPALM) method, introduced for NMF in Pock & Sabach (2016), is generalized to (mAMF) in Algorithm 2.

Algorithm 2 iPALM for mAMF

Initialization: $\mathbf{H}^0$, $\mathbf{W}^0 \geq 0$ such that $\mathbf{W}^0\mathbf{1} = \mathbf{1}$, set $\mathbf{N}^0 := \mathcal{P}_{\mathbf{X}}(\mathbf{W}^0\mathbf{H}^0)$, $\mathbf{H}^{-1} := \mathbf{H}^0$, $\mathbf{W}^{-1} := \mathbf{W}^0$, $\mathbf{N}^{-1} := \mathbf{N}^0$, and $i := 0$.
while stopping criterion is not satisfied **do**
  $\mathbf{H}_1^i := \mathbf{H}^i + \alpha_1^i\left(\mathbf{H}^i - \mathbf{H}^{i-1}\right)$, $\mathbf{H}_2^i := \mathbf{H}^i + \beta_1^i\left(\mathbf{H}^i - \mathbf{H}^{i-1}\right)$ ▷ Inertial $\mathbf{H}$
  $\widetilde{\mathbf{H}}^i := \mathbf{H}_1^i - \frac{1}{\gamma_1^i}{\mathbf{W}^i}^\top\left(\mathbf{H}_2^i\mathbf{W}^i - \mathbf{N}^i\right)$ ▷ Gradient step on $\mathbf{H}$, objective first term
  $\mathbf{H}^{i+1} := \widetilde{\mathbf{H}}^i - \frac{\lambda}{\lambda+\gamma_1^i}\left(\widetilde{\mathbf{H}}^i - \mathcal{P}_{\mathrm{conv}(\mathbf{N}^i)}(\widetilde{\mathbf{H}}^i)\right)$ ▷ Gradient step on $\mathbf{H}$, objective second term
  $\mathbf{W}_1^i := \mathbf{W}^i + \alpha_2^i\left(\mathbf{W}^i - \mathbf{W}^{i-1}\right)$, $\mathbf{W}_2^i := \mathbf{W}_1^i + \beta_2^i\left(\mathbf{W}^i - \mathbf{W}^{i-1}\right)$ ▷ Inertial $\mathbf{W}$
  $\mathbf{W}^{i+1} := \mathcal{P}_{\Delta}\left(\mathbf{W}_1^i - \frac{1}{\gamma_2^i}\left(\mathbf{W}_2^i\mathbf{H}^{i+1} - N^i\right){\mathbf{H}^{i+1}}^\top\right)$ ▷ Projected gradient step on $\mathbf{W}$
  $\mathbf{N}_1^i := \mathbf{N}_1^i + \alpha_3^i\left(\mathbf{N}^i - \mathbf{N}^{i-1}\right)$, $\mathbf{N}_2^i := \mathbf{N}_1^i + \beta_3^i\left(\mathbf{N}^i - \mathbf{N}^{i-1}\right)$ ▷ Inertial $\mathbf{N}$
  $\mathbf{N}^{i+1} := \mathcal{P}_{\mathbf{X}}\left(\mathbf{N}_1^i + \frac{1}{\gamma_3^i}\left(\mathbf{W}^{i+1}\mathbf{H}^{i+1} - \mathbf{N}_2^i\right)\right)$ ▷ Projected gradient step on $\mathbf{N}$
  $i := i + 1$
end while

Remark 8 *If, for all iterations* i, $\alpha_1^i = \alpha_2^i = 0$ *and* $\beta_1^i = \beta_2^i = 0$, *iPALM reduces to PALM.*

Stopping criterion For (mNMF), KKT conditions regarding matrix $\mathbf{W}$ are the following (see Supplement Material):

$$\mathbf{W} \circ \left((\mathbf{WH} - \mathbf{N})\mathbf{H}^\top + t\,\mathbf{1}_K^\top\right) = 0\,.$$

By complementary condition, it follows that, $\forall j$, $t_i = ((\mathbf{WH} - \mathbf{N})\mathbf{H}^\top)_{i,j}$. Hence, we compute t_i by selecting, for each row $W^{(i)}$, any positive entry $W_{i,j} > 0$.

Remark 9 *Numerically to obtain a robust estimate of t_i, we can average the corresponding values calculated per entry $W_{i,j}$.*

Let $\varepsilon_{\mathbf{W}}$, $\varepsilon_{\mathbf{H}}$, and $\varepsilon_{\mathbf{R}}$ be three positive thresholds. The stopping criterion for the previous algorithms consists of a combination of:

1. the maximum number of iterations;
2. the Frobenius norm of the difference of $\mathbf{W}$ and $\mathbf{H}$ at two consecutive iterations, *i.e.*, the algorithm stops if $\|\mathbf{W}^{i+1} - \mathbf{W}^i\|_F \leq \varepsilon_{\mathbf{W}} \ \wedge \ \|\mathbf{H}^{i+1} - \mathbf{H}^i\|_F \leq \varepsilon_{\mathbf{H}}$;
3. a novel criterion based on KKT condition, *i.e.*, the algorithm stops if it holds that

$$\|\mathbf{R}(\mathbf{W}^{i+1})\|_F + \|\mathbf{R}(\mathbf{H}^{i+1})\|_F \leq \varepsilon_{\mathbf{R}} \,,$$

where matrices $\mathbf{R}(\mathbf{W})$ and $\mathbf{R}(\mathbf{H})$ are defined as

$$\begin{aligned} \mathbf{R}(\mathbf{W})_{i,j} &:= |(\mathbf{WH} - \mathbf{N})\mathbf{H}^\top)_{i,j} + t_i| \mathbb{1}_{\{W_{i,j} \neq 0\}} \\ \text{and } \mathbf{R}(\mathbf{H})_{i,j} &:= |\mathbf{W}^\top(\mathbf{WH} - \mathbf{N}))_{i,j}| \mathbb{1}_{\{H_{i,j} \neq 0\}} \end{aligned}$$

respectively.

3.3 Large-scale dataset

Assume the observed matrix $\mathbf{X} = \mathbf{T}(\mathbf{\Phi}(\mathbf{M}))$ is large-scaled, namely one has to forecast a large number N of times series (*e.g.* more than $100,000$) and possibly a large number of time stamps T. The strategy, described in Section 1.3.1 in Cichocki et al. (2009) for NMF, is to learn the $\mathbf{H} \in \mathbb{R}^{K \times T}$ matrix from a submatrix $\mathbf{N}_r \in \mathbb{R}^{r \times T}$ of $K \leq r \ll N$ rows of $\mathbf{N} \in \mathbb{R}^{n \times T}$, and to learn the $\mathbf{W} \in \mathbb{R}^{N \times K}$ matrix from a sub-matrix $\mathbf{N}_c \in \mathbb{R}^{N \times c}$ of $K \leq c \ll T$ columns of $\mathbf{N} \in \mathbb{R}^{N \times T}$. We denote by $\mathbf{H}_c$ the submatrix of $\mathbf{H}$ given by the columns appearing in $\mathbf{N}_c$ and $\mathbf{W}_r$ the sub-matrix of $\mathbf{H}$ given by the columns appearing in $\mathbf{N}_c$.

This strategy can be generalized to (mNMF) and (mAMF). For (mNMF) this generalization is straightforward, and for (mAMF) one need to change Steps 3-5 in Algorithm 1 as follows:

$$\begin{aligned} \widetilde{\mathbf{H}}^i &:= \mathbf{H}^i - \frac{1}{\gamma_1^i}(\mathbf{W}_r^i)^\top \left(\mathbf{W}_r^i \mathbf{H}^i - \mathbf{N}_r^i\right) \\ \mathbf{H}^{i+1} &:= \widetilde{\mathbf{H}}^i - \frac{\lambda}{\lambda + \gamma_1^i} \left(\widetilde{\mathbf{H}}^i - \mathcal{P}_{\mathrm{conv}(\mathbf{N}^i)}(\widetilde{\mathbf{H}}^i)\right) \\ \mathbf{W}^{i+1} &:= \mathcal{P}_\Delta \left(\mathbf{W}^i - \frac{1}{\gamma_2^i}\left(\mathbf{W}^i \mathbf{H}_c^{i+1} - \mathbf{N}_c^i\right)(\mathbf{H}_c^{i+1})^\top\right) . \end{aligned}$$

The same approach is used for Algorithm 2.

4 Numerical Experiments

We tested SMM on real-world datasets. Matrix $\mathbf{H}^0$ is initially selected as in Javadi & Montanari (2020a). Each row of matrix $\mathbf{W}^0$ is generated randomly in the corresponding standard simplex. For SMM we implemented both HALS for (mNMF) and iPALM for (mAMF).

Moreover, we have compared our method with other classically-designed mainstream time series forecasting methods such as *Random Forest Regression* (RFR) and *EXPonential smoothing* (EXP), *Long Short-Term Memory* (LSTM) and *Gated Recurrent Units* (GRU) deep neural networks with preliminary data standardization Shewalkar et al. (2019),

and *Seasonal Auto-Regressive Integrated Moving Average with eXogenous factors* (SARIMAX) models Douc et al. (2014).

The interested reader may find a Github repository on numerical experiments at **[link redacted to comply with double blind reviewing]**

4.1 Real-world datasets

The numerical experiments refer to the following real-world datasets: weekly and daily electricity consumption datasets of 370 Portuguese customers during the period 2011-2014, Trindade (2015); twin gas measurement dataset of five replicates of an 8-MOX gas sensor, Fonollosa (2016); Istanbul Stock Exchange returns with seven other international indexes for the period 2009-2011, Akbilgic (2013); daily electricity transformer temperature (ETT) measurements, Zhou et al. (2020). Figure 3 reports the cross-validated RRMSE and RMPE on observed values obtained during computational tests for each method.

In the majority of the cases, our method is the best or second best among all the approaches for all the dataset we tested in terms of RRMSE and RMPE indices (except for the "weekly electricity" dataset), and there is no other method performing better.

(mAMF) seems to be the most promising algorithm in terms of performances for the first five datasets, while (mNMF) is the best method for the last four ETT datasets.

4.2 Comparison with BasisFormer method

We performed additional computational experiments to compare our NMF-based methodology with state-of-the-art time series transformer models, which are suitable for large-scale time series forecasting problems. In particular, we consider the BasisFormer model recently described in Ni et al. (2023). We consider the same electricity dataset as in Ni et al. (2023) and split the whole dataset into 10 small sets of 960 time steps each. We collect our performance statistics, namely RRMSE and RMPE, on the original unscaled datasets. Note that in Ni et al. (2023), the performance statistics reported are the absolute errors on the scaled dataset obtained by applying `sklearn.preprocessing.StandardScaler` to the original data. For BasisFormer, we run the code in the repository `https://github.com/nzl5116190/Basisformer`.

As shown in Figure 3, our method outperforms the BasisFormer methodology and is competitive against the other methodologies (in particular, with respect to the deep learning approaches which seem the most promising methods for these datasets). We also perform additional computational experiments on scaled datasets, collecting our performance indices on relative errors and absolute errors as in Ni et al. (2023), and we obtain the same dominance results.

4.3 Synthetic datasets

Further computational experiments have been performed by considering additional synthetic datasets. In particular, we generated three datasets by replicating $1,000$ short time series (with 10 time periods) 10 times and adding white noise multiplied by a constant factor σ to each time series entry separately. We choose $\sigma \in \{0.005, 0.1, 1\}$. We refer to the these datasets as "low noise", "medium noise", and "high noise", respectively.

An additional synthetic dataset has been generated considering few probability vectors and computing the entire matrix $\mathbf{W}$ by randomly choosing a probability vector and adding white noise. A completely randomly generated matrix $\mathbf{H}$ is multiplied by $\mathbf{W}$ to obtain the whole matrix $\mathbf{M}^* := \mathbf{WH}$. We refer to this dataset as "synthetic1".

Finally, the last synthetic dataset is obtained by generating a matrix $\mathbf{H}$ by replicating a small time series (with 50 time periods) 100 times and adding white noise multiplied by a constant factor $\sigma = 1$ and matrix $\mathbf{W}$ of suitable dimensions, whose rows are uniformly distributed over the corresponding dimensional simplex. Then, we set the matrix $\mathbf{M}^* := \mathbf{WH}$. We refer to this last dataset as "synthetic2".

Figure 3 reports the cross-validated RRMSE and RMPE indices referring to synthetically generated datasets. The more pronounced the periodicity of the time series or of the archetypes, the better the performances of our proposed NMF-like methods: in this case, the more realistic the hypothesis that the whole dataset can be expressed as convex combinations of a few archetypes, having a low-rank representation.

References

O. Akbilgic. Istanbul Stock Exchange. UCI Machine Learning Repository, 2013. DOI: https://doi.org/10.24432/C54P4J.

J. Bolte, S. Sabach, and M. Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1–2):459–494, 2014.

A. Cichocki and R. Zdunek. *NMFLAB for signal processing*, 2006.

A. Cichocki, R. Zdunek, A.H. Phan, and S.-I. Amari. *Nonnegative matrix and tensor factorization: Applications to exploratory multi-way data analysis and blind source separation*. John Wiley and Sons, 2009.

L. Condat. Fast projection onto the simplex and the l_1 ball. *Mathematical Programming*, 158(1–2):575–585, 2016.

M.R. de Araujo, P.M.P. Ribeiro, and C. Faloutsos. Tensorcast: Forecasting with context using coupled tensors (best paper award). In *2017 IEEE International Conference on Data Mining (ICDM)*, pp. 71–80. IEEE, 2017.

D. Donoho and V. Stodden. When does non-negative matrix factorization give a correct decomposition into parts? In S. Thrun, L. K. Saul, and B. Schölkopf (eds.), *Advances in Neural Information Processing Systems 16*, pp. 1141–1148. MIT Press, 2004.

R. Douc, E. Moulines, and D. Stoffer. *Nonlinear time series: Theory, methods, and applications with R examples*. Chapman & Hall/CRC, 2014.

S. Essid and C. Fevotte. Smooth nonnegative matrix factorization for unsupervised audiovisual document structuring. *IEEE Transactions on Multimedia*, 15(2), 2013.

J. Fonollosa. Twin gas sensor arrays. UCI Machine Learning Repository, 2016. DOI: https://doi.org/10.24432/C5MW3K.

X. Fu, K. Huang, N.D. Sidiropoulos, and W.-K. Ma. Nonnegative matrix factorization for signal and data analytics: Identifiability, algorithms, and applications. *IEEE Signal Processing Magazine*, 36(2):59–80, 1999.

J. Gillard and K. Usevich. Structured low-rank matrix completion for forecasting in time series analysis. *International Journal of Forecasting*, 34(4):582–597, 2018.

N. Gillis. The why and how of nonnegative matrix factorization. In J.A.K. Suykens, M. Signoretto, and A. Argyriou (eds.), *Regularization, optimization, kernels, and support vector machines*, Machine Learning and Pattern Recognition Series, pp. 257–291. Chapman & Hall/CRC, 2014.

N. Gillis. The why and how of nonengative matrix factorization. In J.A.K. Suykens, M. Signoretto, and A. Argyriou (eds.), *Regularization, optimization, kernels, and support vector machines*. Chapman & Hall/CRC, 2015.

N. Gillis. Introduction to nonnegative matrix factorization. *SIAG/OPT Views and News*, 25(1):7–16, 2017.

N. Gillis and F. Glineur. Accelerated multiplicative updates and hierarchical ALS algorithms for nonnegative matrix factorization. *Neural Computation*, 24(4):1085–1105, 2012. doi: 10.1162/NECO_a_00256. URL https://doi.org/10.1162/NECO_a_00256.

N. Gillis and A. Kumarg. Exact and heuristic algorithms for semi-nonnegative matrix factorization. *SIAM Journal on Matrix Analysis and Applications*, 36(4):1404–1424, 2015.

H. Javadi and A. Montanari. Nonnegative matrix factorization via archetypal analysis. *Journal of the American Statistical Association*, 115(530):896–907, 2020a.

H. Javadi and A. Montanari. Supplement To "Non-negative Matrix Factorization via Archetypal Analysis". *Journal of the American Statistical Association*, 115(530):896–907, 2020b. URL https://doi.org/10.1080/01621459.2019.1594832.

H. Laurberg, M.G. Christensen, M.D. Plumbley, L.K. Hansen, and S.H. Jensen. Theorems on positive data: On the uniqueness of NMF. *Computational Intelligence and Neuroscience*, 2008:1–9, 2008.

D.D. Lee and H.S. Seung. Learning the parts of objects by nonnegative matrix factorization. *Nature*, 401(6755): 788–791, 1999.

J. Mei, Y. De Castro, Y. Goude, and G. Hébrail. Nonnegative matrix factorization for time series recovery from a few temporal aggregates. In *Proceedings of the 34th International Conference on Machine Learning*. JMLR: W&CP, 2017.

J. Mei, Y. De Castro, Y. Goude, J.-M. Azaïs, and G. Hébrail. Nonnegative matrix factorization with side information for time series recovery and prediction. *IEEE Transactions on Knowledge and Data Engineering*, 31(3):493–506, 2018.

Z. Ni, H. Yu, S. Liu, J. Li, and W. Lin. BasisFormer: Attention-based time series forecasting with learnable and interpretable basis. *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, 2023.

P. Paatero and U. Tapper. Positive matrix factorization: A nonnegative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(1):111—126, 1994.

N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):123–231, 2013.

T. Pock and S. Sabach. Inertial proximal alternating linearized minimization (iPALM) for nonconvex and nonsmooth problems. *SIAM Journal on Imaging Sciences*, 9(4):1756–1787, 2016. doi: 10.1137/16M1064064. URL https://doi.org/10.1137/16M1064064.

B. Recht, C. Re,, J. Tropp, and V. Bittorf. Factoring nonnegative matrices with linear programs. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 25*, pp. 1214–1222. Curran Associates, Inc., 2012.

A. Shewalkar, D. Nyavanandi, and S.A. Ludwig. Performance evaluation of deep neural networks applied to speech recognition: RNN, LSTM and GRU. *Journal of Artificial Intelligence and Soft Computing Research*, 9:235–245, 2019.

H. Tan, Y. Wu, B. Shen, P.J. Jin, and B. Ran. Short-term traffic prediction based on dynamic tensor completion. *IEEE Transactions on Intelligent Transportation Systems*, 17(8):2123–2133, 2016.

L. Thomas. Solution to problem 73–14, rank factorizations of nonnegative matrices. *SIAM Review*, 16(1):393–394, 1974.

A. Trindade. ElectricityLoadDiagrams20112014. UCI Machine Learning Repository, 2015. DOI: https://doi.org/10.24432/C58C86.

A.C. Turkmen. A Review of nonnegative matrix factorization methods for clustering. https://arxiv.org/abs/1507.03194, 2015.

S.A. Vavasis. On the complexity of nonnegative matrix factorization. *SIAM Journal on Optimization*, 20(3):1364–1377, 2009. ISSN 1052–6234.

F. Wang, T. Li, X. Wang, S. Zhu, and C. Ding. Community discovery using nonnegative matrix factorization. *Data Mining and Knowledge Discovery*, 22(3):493–521, 2011.

Y.-X. Wang and Y.-J. Zhang. Nonnegative matrix factorization: A comprehensive review. *IEEE Transactions on Knowledge and Data Engineering*, 25(6):1336–1353, 2013.

T. Wolfe. Finding the nearest point in a polytope. *Mathematical Programming*, 11:128–149, 1976.

W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pp. 267–273, 2003.

T. Yokota, B. Erem, S. Guler, S.K. Warfield, and H. Hontani. Missing slice recovery for tensors using a low-rank model in embedded space. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8251–8259, 2018.

H.-F. Yu, N. Rao, and I.S. Dhillon. Temporal regularized matrix factorization for high-dimensional time series prediction. In *NIPS*, pp. 847–855, 2016.

H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *AAAI Conference on Artificial Intelligence*, 2020.