Forecasting Nonnegative Time Series via Sliding Mask Method (SMM) and Latent Clustered Forecast (LCF)

YOHANN DE CASTRO[⋆] AND LUCA MENCARELLI[†]

[⋆]*Institut Camille Jordan, École Centrale de Lyon, Écully, France*
[†]*CERMICS, École des Ponts ParisTech, Marne la Vallée, France*

March 3, 2021

Abstract

We consider nonnegative time series forecasting framework. Based on recent advances in Nonnegative Matrix Factorization (NMF) and Archetypal Analysis, we introduce two procedures referred to as Sliding Mask Method (SMM) and Latent Clustered Forecast (LCF). SMM is a simple and powerful method based on time window prediction using Completion of Nonnegative Matrices. This new procedure combines low nonnegative rank decomposition and matrix completion where the hidden values are to be forecasted. LCF is two stage: it leverages archetypal analysis for dimension reduction and clustering of time series, then it uses any black-box supervised forecast solver on the clustered latent representation. Theoretical guarantees on uniqueness and robustness of the solution of NMF Completion-type problems are also provided for the first time. Finally, numerical experiments on real-world and synthetic data-set confirms forecasting accuracy for both the methodologies.

1 Introduction

This article deals with forecasting and/or clustering of times series. We focus on applications where one knows that times series at hand have some intrinsic structure, such as entry-wise *nonnegativity*. In that case one can exploit Nonnegative Matrix Factorization (NMF) approaches which have been introduced by [Paatero and Tapper, 1994] for spectral unmixing problems in analytical chemistry and popularized by [Lee and Seung, 1999]. For further details we refer the interested reader to the surveys [Wang and Zhang, 2013, Gillis, 2015, Gillis, 2017] and references therein. NMF has been widely applied in many different contexts: document analysis [Xu et al., 2003, Essid and Fevotte, 2013], hidden Markov chain [Fu et al., 1999], representation learning in image recognition [Lee and Seung, 1999], community discovery [Wang et al., 2011], and clustering methods [Turkmen, 2015]. This paper introduces two NMF-like procedures for forecasting and clustering of time series. Forecasting for temporal time series has been previously done before through a mixed linear regression and matrix factorization in [Yu et al., 2016], matrix completion for one temporal time serie in [Gillard and Usevich, 2018], and tensor factorization [de Araujo et al., 2017, Yokota et al., 2018, Tan et al., 2016].

Sliding Mask Method (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 challenge is uniqueness of the decomposition, also referred to as *identifiability* of the model. In Theorem 2, we introduce a new condition that ensures uniqueness from partial observation of the target matrix $\mathbf{M}$. An other challenge, as pointed 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 and Boyd, 2013] for block-matrix coordinate descent in an *alternating projection scheme*, *e.g.*, [Javadi and Montanari, 2020a] and references therein. We rely on these techniques to introduce algorithms inputting the forecast values based on NMF decomposition, see Section 2.3. Theorem 3 complements proving robustness of solutions of NMF-type algorithms when entries are missing or corrupted by noise.

In practical situation, one may face a large number of time series to forecast, *e.g.* supply chain optimization, electricity consumption forecast... In this case, one cannot forecast each time series separately and/or clustering the set of time series without facing large algorithmic complexity. We can address this issue using Latent Clustered Forecast (LCF). This method uses NMF decomposition $\mathbf{M} = \mathbf{W}\mathbf{H}$ as a dimension reduction step. As we will see in Section 3, it performs forecast on rows of $\mathbf{H}$ (using any black-box supervised forecast solver) and/or clustering on the rows of $\mathbf{W}$.

Acronym	Name	Objective	Constraints: $\mathbf{W} \geq \mathbf{0}$ +
NMF	Nonnegative Matrix Factorization [Cichocki and Zdunek, 2006]	$\mathbf{F}_1$	$\mathbf{H} \geq \mathbf{0}$
SNMF	Semi NMF [Gillis and Kumarg, 2015]	$\mathbf{F}_1$	
NNMF	Normalized NMF	$\mathbf{F}_1$	$\mathbf{H} \geq \mathbf{0}$, $\mathbf{W1} = \mathbf{1}$
SNNMF	Semi Normalized NMF	$\mathbf{F}_1$	$\mathbf{W1} = \mathbf{1}$
AMF	Archetypal Matrix Factorization [Javadi and Montanari, 2020a]	$\mathbf{F}_2$	$\mathbf{W1} = \mathbf{1}$, $\mathbf{V} \geq \mathbf{0}$, $\mathbf{V1} = \mathbf{1}$
ANMF	Archetypal NMF	$\mathbf{F}_2$	$\mathbf{H} \geq \mathbf{0}$, $\mathbf{V} \geq \mathbf{0}$, $\mathbf{V1} = \mathbf{1}$
ANNMF	Archetypal Normalized NMF	$\mathbf{F}_2$	$\mathbf{W1} = \mathbf{1}$, $\mathbf{H} \geq \mathbf{0}$, $\mathbf{V} \geq \mathbf{0}$, $\mathbf{V1} = \mathbf{1}$
mNMF	Mask NNMF	$\mathbf{F}_3$	$\mathbf{TN} = \mathbf{X}, \mathbf{W1} = \mathbf{1}, \mathbf{H} \geq \mathbf{0}$
mAMF	Mask AMF	$\mathbf{F}_4$	$\mathbf{TN} = \mathbf{X}, \mathbf{W1} = \mathbf{1}$, $\mathbf{V} \geq \mathbf{0}$, $\mathbf{V1} = \mathbf{1}$

Table 1: The seven block convex programs achieving matrix factorization of nonnegative matrices. The objectives are $\mathbf{F}_1 := \|\mathbf{M}-\mathbf{WH}\|_F^2$ and $\mathbf{F}_2 := \|\mathbf{M}-\mathbf{WH}\|_F^2 + \lambda\|\mathbf{H}-\mathbf{VM}\|_F^2$. The two last lines are SMM procedures with sliding operator $\mathbf{\Pi}$ and objectives $\mathbf{F}_3 := \|\mathbf{N}-\mathbf{WH}\|_F^2$ and $\mathbf{F}_4 := \|\mathbf{N}-\mathbf{WH}\|_F^2 + \lambda\|\mathbf{H}-\mathbf{VN}\|_F^2$.

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

1.1 The time series forecasting problem

This article considers $N \geq 1$ times series on the same temporal period with $T \geq 1$ timestamps in a setting where $N \geq T$ and possibly $N \gg T$. We would like to forecast the next $F \geq 1$ times. Additionally, one may also aim at clustering these N time series, and/or reduce the ambient dimension $N \times T$ while maintaining a good approximation of these times series. The observed times series can be represented as a matrix $\mathbf{M}$ of size $N \times T$. A row $\mathbf{M}^{(i)}$ of $\mathbf{M}$ represents a time series and a column $\mathbf{M}_j$ of $\mathbf{M}$ represents a timestamp record. We assume that there exists a *target* matrix $\mathbf{M}^\star$ with

$$\mathbf{M} = \mathbf{M}_T^\star + \mathbf{E},$$

where $\mathbb{E}$ is some noise term and $\mathbf{M}_T^\star \in \mathbb{R}_+^{N\times T}$ is a sub-matrix

$$\mathbf{M}^\star := \Big[\underbrace{\mathbf{M}_T^\star}_{\text{past}} \ \underbrace{\mathbf{M}_F^\star}_{\text{future}} \Big]$$

of the *target matrix* of size $N \times (T + F)$ that can be split into timestamps up today $\mathbf{M}_T^\star \in \mathbb{R}_+^{N\times T}$ and future timestamps $\mathbf{M}_F^\star \in \mathbb{R}_+^{N\times F}$ to be forecast.

The statistical task is the following: given the observation $\mathbf{M}$ predict the future target values $\mathbf{M}_F^\star$, and incidentally the "denoised" $\mathbf{M}_T^\star$.

1.2 Nonnegative and Archetypal Analysis

We aim to decompose nonnegative matrix $\mathbf{M} \in \mathbb{R}_+^{N\times T}$ as the product of nonnegative matrix $\mathbf{W} \in \mathbb{R}^{N\times K}$ and matrix $\mathbf{H} \in \mathbb{R}^{K\times T}$ by minimizing the Frobenius norm of the difference between $\mathbf{M}$ and the reconstructed matrix $\widehat{\mathbf{M}} := \mathbf{WH}$ [Cichocki et al., 2009]:

$$\min_{\mathbf{W}\geq 0, \mathbf{H}\geq 0} \|\mathbf{M}-\mathbf{WH}\|_F^2 \,. \tag{NMF}$$

Another approach consists in the *Archetypal Analysis*:

$$\min_{\substack{\mathbf{W}\geq 0, \mathbf{W1}=\mathbf{1} \\ \mathbf{V}\geq 0, \mathbf{V1}=\mathbf{1}}} \|\mathbf{M}-\mathbf{WH}\|_F^2 + \lambda\|\mathbf{H}-\mathbf{VM}\|_F^2 \,, \tag{AMF}$$

where $\lambda > 0$ is a tuning parameter, see for instance [Javadi and Montanari, 2020a]. Different normalisation and constraints can be considered, we exhibit 7 variants in Table 1. We will be particularly interested in

$$\min_{\substack{\mathbf{W}\geq 0, \mathbf{W1}_K=\mathbf{1}_N \\ \mathbf{H}\geq \mathbf{0}}} \|\mathbf{M}-\mathbf{WH}\|_F^2 \,. \tag{NNMF}$$

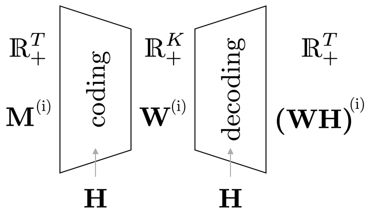

Figure 1: Nonnegative and Archetypal representations performs dimension reduction using $\mathbf{H}$ as decoding, while coding is based on conic (resp. barycentric) coordinates (w.r.t. rows of $\mathbf{H}$) in the non-normalized case (resp. normalized case $\mathbf{W}\mathbf{1}_K = \mathbf{1}_N$).

An important parameter is the so-called *nonnegative rank* $K \geq 1$ [Gillis and Glineur, 2012b] that governs dimension reduction performed by these matrix factorization, see Figure 1. Note that the left hand side matrix $\mathbf{W}$ is always nonnegative and may be or may not be normalized so that $\mathbf{W}\mathbf{1}_K = \mathbf{1}_N$. Each row of the observation $\mathbf{M}$ is then a weighted sum of rows of $\mathbf{H}$, these weights being encoded by $\mathbf{W}$. Hence, each time series (row $\mathbf{M}^{(i)}$) is encoded by $\mathbf{W}^{(i)}$ a row of $\mathbf{W}$. Decoding is performed multiplying $\mathbf{W}^{(i)}$ by $\mathbf{H}$. When $\mathbf{W}$ is normalized, the aforementioned factorisation algorithms search for the best polytope encapsulating the cloud of point given by the rows of $\mathbf{M}$. The points generating this polytope are the rows of $\mathbf{H}$ while each cloud point (here a row $\mathbf{M}^{(i)}$ of $\mathbf{M}$) is localized thanks to its barycentric coordinates given by $\mathbf{W}^{(i)}$, for further details see [Gillis, 2014]. When $\mathbf{W}$ is not normalized, the aforementioned factorisation algorithms search for the best cone encapsulating the cloud of point given by the rows of $\mathbf{M}$. The rays generating this cone are the rows of $\mathbf{H}$ while each cloud point (here a row $\mathbf{M}^{(i)}$ of $\mathbf{M}$) is localized thanks to its conic coordinates given by $\mathbf{W}^{(i)}$, for further details see [Ge and Zou, 2015].

The parameter λ in the archetypal analysis enforces some kind of reciprocity. In NMF, the cloud of points given by the rows of $\mathbf{M}$ should be encapsulated by the rows of $\mathbf{H}$. Archetypal analysis penalises the reciprocal: the rows of $\mathbf{H}$ should be encapsulated in the convex/conic volume given by the cloud of points of the rows of $\mathbf{M}$. Both parameters K and λ can be tuned by cross-validation [Arlot and Celisse, 2010], as done in our experiments, see Section 4.

1.3 Contribution: Towards Nonnegative Matrix Completion for Time series

We are interested in a *Matrix Completion* problem using Nonnegative Matrix Factorization. We consider $n \times p$ matrices $\mathbf{X}_0, \mathbf{X}^\star, \mathscr{X}$ which, in the Sliding Mask Method (SMM), are linear transformations of matrices $\mathbf{M}^\star, \mathbf{M}^\star_T, \mathbb{E}$ respectively, see Section 2. Note that $n := (B - W + 1)N > N$ and $p := WP > F$ where B, W, F will be introduced in Section 2.

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

$$\mathbf{T}(\mathbf{N}) = \left[\begin{array}{c|c} \mathbf{N}_1 & \mathbf{N}_2 \\ \hline \mathbf{N}_3 & \mathbf{0}_{N\times F} \end{array}\right],$$

where $(\mathbf{N}_i)_{i=1}^4$ are blocks of $\mathbf{N} = \left[\begin{array}{c|c} \mathbf{N}_1 & \mathbf{N}_2 \\ \hline \mathbf{N}_3 & \mathbf{N}_4 \end{array}\right]$.

Our goal is the following matrix completion problem: Given a noisy and incomplete observation

$$\mathbf{X} := \mathbf{T}(\mathbf{X}_0) + \mathscr{F}, \tag{1}$$

where $\mathscr{F}$ is some noise term, find a good estimate of the target $\mathbf{X}_0$. One can consider the *Nonnegative Matrix Completion* referred to as Mask NNMF:

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

where solutions $\mathbf{N} \in \mathbb{R}^{n\times p}$ are such that $\mathbf{TN} = \mathbf{X}$ (observed values) and $\mathbf{T}^\perp\mathbf{N} = \mathbf{T}^\perp(\mathbf{WH})$ (forecast values). This latter formulation is an instance of Matrix Completion [Nguyen et al., 2019]. Forecasting problem reduces to Matrix Completion problem, whose aim is finding the nonnegative matrix factorization $\mathbf{N} \simeq \mathbf{WH}$ of observed matrix $\mathbf{X}$ such that $\mathbf{TN} = \mathbf{X}$.

Remark 1. *Problem (mNMF) is NNMF when* $\mathbf{T} = \mathbb{I}$, *where* $\mathbb{I}$ *is the identity operator.*

Dropping $\mathbf{H} \geq 0$, another approach is the *Archetypal Matrix Completion* referred to as Mask AMF:

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

Remark 2. *When* $\mathbf{T} = \mathbb{I}$, *Problem (mAMF) reduces to standard AMF formulation (AMF).*

1.3.1 Uniqueness from partial observations

Define the mask $\mathbf{X}^\star$ of $\mathbf{X}_0$ by

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

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 $\mathbf{X}_0$. Let us consider

$$\begin{aligned}
&\mathbf{T}_{\text{train}}(\mathbf{X}_0) := [\mathbf{X}_1\ \mathbf{X}_2], \quad && \mathbf{T}_{\text{test}}(\mathbf{X}_0) := [\mathbf{X}_3\ \mathbf{0}_{N\times F}],\\
&\mathbf{T}_T(\mathbf{X}_0) := \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_3 \end{bmatrix}, && \mathbf{T}_F(\mathbf{X}_0) := \begin{bmatrix} \mathbf{X}_2 \\ \mathbf{0}_{N\times F} \end{bmatrix}.
\end{aligned}$$

Remark 3. *Let* $\mathbf{X}_0 := \mathbf{W}_0\mathbf{H}_0$*,* $\mathbf{H}_0 := [\mathbf{H}_{0T}\ \mathbf{H}_{0F}]$*, and* $\mathbf{W}_0^\top := [\mathbf{W}_{0\text{train}}^\top\ \mathbf{W}_{0\text{test}}^\top]$*, then*

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

At first glance, the coding-decoding scheme of Figure 1 can be ill-posed and/or not robust to noise. The first issue is the *uniqueness* of the decomposition $\mathbf{W}_0\mathbf{H}_0$ *given partial observations*, namely proving that *Partial Observation Uniqueness* (**POU**) property holds:

$$\begin{aligned}
&\text{If } \mathbf{T}(\mathbf{W}\mathbf{H}) = \mathbf{T}(\mathbf{W}_0\mathbf{H}_0)\\
&\text{Then } (\mathbf{W},\mathbf{H}) \equiv (\mathbf{W}_0,\mathbf{H}_0)\,,
\end{aligned} \tag{POU}$$

where $\equiv$ means up to positive scaling and permutation: if an entry-wise nonnegative pair $(\mathbf{W},\mathbf{H})$ is given then $(\mathbf{W}\mathbf{P}\mathscr{D}, \mathscr{D}^{-1}\mathbf{P}^\top\mathbf{H})$ is also a nonnegative decomposition $\mathbf{W}\mathbf{H} = \mathbf{W}\mathbf{P}\mathscr{D} \times \mathscr{D}^{-1}\mathbf{P}^\top\mathbf{H}$, where $\mathscr{D}$ scales and $\mathbf{P}$ permutes the columns (resp. rows) of $\mathbf{W}$ (resp. $\mathbf{H}$). When we observe the full matrix $\mathbf{X}_0 = \mathbf{W}_0\mathbf{H}_0$, the uniqueness issue 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 and Stodden, 2004], and *Separability* of [Recht et al., 2012]. A necessary and sufficient condition exists:

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*

$$\mathrm{Cone}(\mathbf{W}_0^\top) \subseteq \mathscr{C} \subseteq \mathrm{Cone}(\mathbf{H}_0)\,,$$

where $\mathrm{Cone}(\mathbf{A})$ *is the cone generated by the rows of* $\mathbf{A}$*.*

Our first main assumption is that:

- (**A1**) *In the set given by the union of sets:*

$$\begin{aligned}
&\{\mathscr{C}\ :\ \mathrm{Cone}(\mathbf{W}_{0\text{train}}{}^\top) \subseteq \mathscr{C} \subseteq \mathrm{Cone}(\mathbf{H}_0)\}\\
&\bigcup \{\mathscr{C}\ :\ \mathrm{Cone}(\mathbf{W}_0^\top) \subseteq \mathscr{C} \subseteq \mathrm{Cone}(\mathbf{H}_{0T})\}\,,
\end{aligned}$$

the nonnegative orthant is the only K*-simplicial cone.*

It is clear that this property is implied by the following one, (**A'1**) $\Rightarrow$ (**A1**).

- (**A'1**) *In the set*

$$\{\mathscr{C}\ :\ \mathrm{Cone}(\mathbf{W}_{0\text{train}}{}^\top) \subseteq \mathscr{C} \subseteq \mathrm{Cone}(\mathbf{H}_{0T})\}$$

the nonnegative orthant is the only K*-simplicial cone.*

Given the standard definition:

Definition 1 ([Javadi and 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 that:

- (**A2**) *Assume that*

$$\mathrm{conv}(\underbrace{\mathbf{T}_{\text{train}}(\mathbf{X}_0)}_{=\mathbf{W}_{0\text{train}}\mathbf{H}_0}) \text{ has internal radius } \mu > 0\,. \tag{A2}$$

Theorem 2. *Condition* (**A1**) *is sufficient for* (**POU**).
If (**A1**) *and* (**A2**) *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.*

Proof. Proofs are given in Supplement Material. □

Remark 4. *By Theorem 1, observe that* (**A'1**) *is a necessary and sufficient condition for the uniqueness of the decomposition* $\mathbf{X}_1 = \mathbf{W}_{0\text{train}}\mathbf{H}_{0T}$. *Then, using* (**A'1**) $\Rightarrow$ (**A1**)*, we understand that if decomposition of* $\mathbf{X}_1 = \mathbf{W}_{0\text{train}}\mathbf{H}_{0T}$ *is unique then* (**POU**) *holds.*

1.3.2 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 and 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 $\|\mathscr{F}\|_F$ is small enough.

In the normalized case (*i.e.*, $\mathbf{W1} = \mathbf{1}$), both issues (uniqueness and robustness) can be handle with the notion of α-uniqueness, introduced by [Javadi and 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 $\mathscr{B} \in \mathbb{R}^{n_b \times p}$ with same row dimension, and $\mathscr{C} \in \mathbb{R}^{n_a \times n_b}$, define the divergence $\mathscr{D}(\mathbf{A}, \mathscr{B})$ as

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

which is the squared distance between rows of $\mathbf{A}$ and $\operatorname{conv}(\mathscr{B})$, the convex hull of rows of $\mathscr{B}$.

For $\mathscr{B} \in \mathbb{R}^{n \times p}$ define

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

Definition 2 ($\mathbf{T}_\alpha$-unique)**.** *Given* $\mathbf{X}_0 \in \mathbb{R}^{n \times p}, \mathbf{W}_0 \in \mathbb{R}^{n \times K}, \mathbf{H}_0 \in \mathbb{R}^{K \times p}$*, 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* $\operatorname{conv}(\mathbf{X}_0) \subseteq \operatorname{conv}(\mathbf{H})$*:*

$$\begin{aligned}\widetilde{\mathscr{D}}(\mathbf{H}, \mathbf{X}_0)^{1/2} \geq\ & \widetilde{\mathscr{D}}(\mathbf{H}_0, \mathbf{X}_0)^{1/2} \\ & + \alpha \left\{ \mathscr{D}(\mathbf{H}, \mathbf{H}_0)^{1/2} + \mathscr{D}(\mathbf{H}_0, \mathbf{H})^{1/2} \right\} .\end{aligned}$$

Our third main assumption is given by:

- (**A3**) *Assume that*

$$\mathbf{X}_0 = \mathbf{W}_0\mathbf{H}_0 \text{ is } \mathbf{T}_\alpha\text{-unique} \tag{A3}$$

Theorem 3. *If* (**A2**) *and* (**A3**) *hold then there exists positive reals* Δ *and* Λ *(depending on* $\mathbf{X}_0$*) such that, for all* $\mathscr{F}$ *such that* $\|\mathscr{F}\|_F \leq \Delta$ *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* (1) *is such that:*

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

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

Proof. Proofs are given in Supplement Material. □

1.4 Outline

The rest of the paper is organized as follows. In Section 2 we discuss the *Sliding Mask Method* (SMM), while Section 3 is devoted to describe the *Latent Clustered Forecast* (LCF). Numerical experiments and conclusions are presented in Sections 4 and 5, respectively. A repository on the numerical expermients can be found at `https://github.com/Luca-Mencarelli/Nonnegative-Matrix-Factorization-Time-Series`.

2 The Sliding Mask Method

2.1 Sliding window as forecasting

One is given N time series $\mathbf{M}^{(1)},\dots,\mathbf{M}^{(N)} \in \mathbb{R}^T$ over a period of T dates. Recall $\mathbf{M} \in \mathbb{R}^{N\times T}$ is the matrix of observation such that $\mathbf{M}^\top = [(\mathbf{M}^{(1)})^\top \cdots (\mathbf{M}^{(N)})^\top]$ and assumed entry-wise nonnegative. We assume some *periodicity* in our time series, namely that $\mathbf{M}^\star$ can be split into B matrix blocks of size $N \times P$ where $P = (T+F)/B$, see Figure 2.

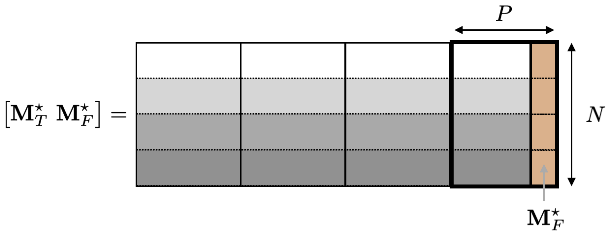

Figure 2: Target matrix $\mathbf{M}^\star$ can be split into B blocks of same time length P.

Given $W \geq 1$ and a $T \times N$ matrix $\mathbf{M}$, we define ″($\mathbf{M}$) the linear operator that piles up W consecutive sub-blocks in a row, as depicted in Figure 3. This process looks at W consecutive blocks in a *sliding* manner. Note that ″($\mathbf{M}$) is an *incomplete* matrix where the missing values are depicted in orange in Figure 3, they correspond to the time-period to be forecasted. Unless otherwise specified, these unobserved values are set to zero. Remark that ″($\mathbf{M}$) has W columns blocks, namely WP columns and $(B-W+1)N$ rows. By an abuse of notation, we also denote

$$'' : \mathbb{R}^{N\times(T+F)} \to \mathbb{R}^{(B-W+1)N\times WP}$$

the same one-to-one linear matrix operation on matrices of size $N \times (T+F)$. In this case, $\mathbf{X}_0 :=$ ″($\mathbf{M}^\star$) is a *complete* matrix where the orange values have been implemented with the future values of the target $\mathbf{M}_F^\star$.

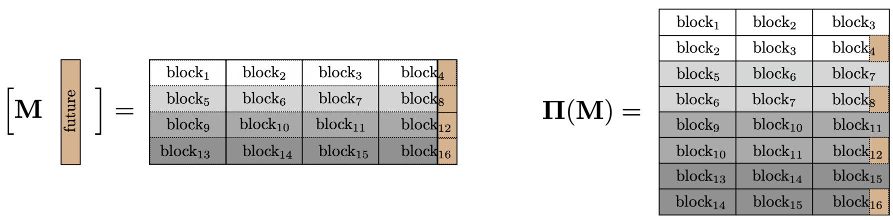

Figure 3: The operator $\mathbf{\Pi}(\mathbf{M})$ outputs an incomplete $(B-W+1)N \times WP$ matrix given by a mask where the NF orange entries are not observed. These entries corresponds to future times that should be forecasted.

The rationale behind is recasting the forecasting problem as a supervised learning problem where one observes, at each line of $\mathbf{\Pi}(\mathbf{M})$, the $WP-F$ first entries and learn the next F entries. The training set is given by rows $1,3,5,7$ in $\mathbf{\Pi}(\mathbf{M})$ of Figure 3 and the validation set is given by rows $2,4,6,8$ where one aims at predicting the F missing values from the $WP-F$ first values of these rows.

2.2 Mask NNMF and Mask AMF

Consider a matrix completion version of NMF with observations

$$\mathbf{X} := ''(\mathbf{M}) = \underbrace{''(\mathbf{M}_T^\star)}_{\mathbf{X}^\star} + \underbrace{''(\mathbb{E})}_{\mathscr{Z}},$$

and

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

where the "mask" operator $\mathbf{T}$ is defined by zeroing the "future" values (in orange in Figure 3). Note that

$$\mathbf{T}(\underbrace{\mathbf{\Pi}(\mathbf{M}^\star)}_{\mathbf{X}_0}) = \mathbf{\Pi}(\mathbf{M}_T^\star) = \mathbf{X}^\star\,.$$

Moreover, note that Problem (2) is equivalent to mask NNMF (mNMF). If we drop the nonnegative constraints on $\mathbf{H}$ and consider the archetypal approach, we obtain mask AMF (mAMF). In particular, Theorem 3 applies proving that (mNMF) and (mAMF) are robust to small noise.

2.3 Algorithms

2.3.1 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 1), when an alternative minimization procedure is performed and matrix $\mathbf{WH}$ is projected onto the linear subspace $\mathbf{TN}=\mathbf{X}$ by means of operator $\mathscr{P}_{\mathbf{X}}$, as follows:

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

Algorithm 1: ALS for mNMF

1: **Initialization:** choose $\mathbf{H}^0 \geq \mathbf{0}, \mathbf{W}^0 \geq 0$, set $\mathbf{N}^0 := \mathscr{P}_{\mathbf{X}}(\mathbf{H}^0\mathbf{W}^0)$ and $i := 0$.
2: **while** stopping criterion is not satisfied **do**
3: $\quad \mathbf{H}^{i+1} := \min_{\mathbf{H}\geq 0} ||\mathbf{N}^i - \mathbf{W}^i\mathbf{H}||_F^2$
4: $\quad \mathbf{W}^{i+1} := \min_{\mathbf{W}\geq 0, \mathbf{W1}=\mathbf{1}} ||\mathbf{N}^i - \mathbf{WH}^{i+1}||_F^2$
5: $\quad$ set $\mathbf{N}^{i+1} := \mathscr{P}_{\mathbf{X}}(\mathbf{W}^{i+1}\mathbf{H}^{i+1})$
6: $\quad i := i+1$
7: **end while**

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 and Glineur, 2012a] (see Supplement Material).

2.3.2 Projected Gradient for (mAMF)

Proximal Alternative Linear Minimization (PALM) method, introduced in [Bolte et al., 2014] and applied to AMF by [Javadi and Montanari, 2020a], can be also generalized to (mAMF) (see Algorithm 2).

Algorithm 2: PALM for mAMF

1: **Initialization:** chose $\mathbf{H}^0$, $\mathbf{W}^0 \geq \mathbf{0}$ such that $\mathbf{W}^0\mathbf{1}=\mathbf{1}$, set $\mathbf{N}^0 := \mathscr{P}_{\mathbf{X}}(\mathbf{W}^0\mathbf{H}^0)$ and $i := 0$.
2: **while** stopping criterion is not satisfied **do**
3: $\quad \widetilde{\mathbf{H}}^i := \mathbf{H}^i - \frac{1}{\gamma_1^i}\mathbf{W}^{i\top}\left(\mathbf{H}^i\mathbf{W}^i - \mathbf{N}^i\right)$
4: $\quad \mathbf{H}^{i+1} := \widetilde{\mathbf{H}}^i - \frac{\lambda}{\lambda+\gamma_1^i}\left(\widetilde{\mathbf{H}}^i - \mathscr{P}_{\mathrm{conv}(\mathbf{N}^i)}(\widetilde{\mathbf{H}}^i)\right)$
5: $\quad \mathbf{W}^{i+1} := \mathscr{P}_{\Delta}\left(\mathbf{W}^i - \frac{1}{\gamma_2^i}\left(\mathbf{H}^{i+1}\mathbf{W}^i - \mathbf{N}^i\right)\mathbf{H}^{i+1\top}\right)$
6: $\quad \mathbf{N}^{i+1} := \mathscr{P}_{\mathbf{X}}\left(\mathbf{N}^i + \frac{1}{\gamma_3^i}\left(\mathbf{H}^{i+1}\mathbf{W}^{i+1} - \mathbf{N}^i\right)\right)$
7: $\quad i := i+1$
8: **end while**

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

Theorem 4. *Let $\varepsilon > 0$. If $\gamma_1^i > ||\mathbf{W}^{i\top}\mathbf{W}^i||_F$, $\gamma_2^i > \max\left\{||\mathbf{H}^{i+1}\mathbf{H}^{i+1\top}||_F, \varepsilon\right\}$, 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 2 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:*

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

Proof. Proof is given in Supplement Material. □

Finally, the inertial PALM (iPALM) method, introduced for NMF in [Pock and Sabach, 2016], is generalized to (mAMF) in Algorithm 3.

Remark 5. *If, for all iterations i, $\alpha_1^i = \alpha_2^i = 0$ and $\beta_1^i = \beta_2^i = 0$, iPALM reduces to PALM.*

Algorithm 3: iPALM for mAMF

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

2.3.3 Stopping criterion for normalized NMF

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

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

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

Remark 6. *Numerically to obtain a robust estimation 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 in 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
$$\|\mathbf{R}(\mathbf{W}^{i+1})\|_F + \|\mathbf{R}(\mathbf{H}^{i+1})\|_F \leq \varepsilon_{\mathbf{R}}$$
where matrices $\mathbf{R}(\mathbf{W})$ and $\mathbf{R}(\mathbf{H})$ are defined as
$$\begin{aligned}\mathbf{R}(\mathbf{W})_{i,j} &:= |(\mathbf{W}\mathbf{H} - \mathbf{N})\mathbf{H}^\top)_{i,j} + t_i|\mathbb{1}_{\{W_{i,j}\neq 0\}}\,,\\ \mathbf{R}(\mathbf{H})_{i,j} &:= |\mathbf{W}^\top(\mathbf{W}\mathbf{H} - \mathbf{N}))_{i,j}|\mathbb{1}_{\{H_{i,j}\neq 0\}}\,.\end{aligned}$$

2.4 Large-scale data-set

Assume the observed matrix $\mathbf{X} = \mathbf{\Pi}(\mathbf{M})$ 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 sub-matrix $\mathbf{N}_r \in \mathbb{R}^{r\times T}$ of $K \leq r \ll N$ rows of $\mathbf{N} \in \mathbb{R}^{n\times T}$, and 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 sub-matrix 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 2 as follows:

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

Same approach is exploited for Algorithm 3.

3 Latent Clustered Forecast

In this section we describe a second approach to time series forecasting problem, namely *Latent Clustered Forecast* (LCF) (see Algorithm 4). Let

$$\mathbf{M}^\star = \mathbf{W}^\star\mathbf{H}^\star = \mathbf{W}^\star\Big[\underbrace{\mathbf{H}_T^\star}_{\text{past}}\ \underbrace{\mathbf{H}_F^\star}_{\text{future}}\Big] = \Big[\underbrace{\mathbf{M}_T^\star}_{\text{past}}\ \underbrace{\mathbf{M}_F^\star}_{\text{future}}\Big].$$

Rather than directly forecasting the time series future data $\mathbf{M}_F^\star$, we firstly forecast the profiles $\mathbf{H}_F$ and we secondly reconstruct $\mathbf{M}_F^\star$. LCF initially cluster the N time series in $I \ll N$ groups. For each cluster $i \in [I]$, matrix $(\mathbf{M}_T^\star)_i$ is defined by selecting the time series belonging to cluster and is factorized as $\mathbf{W}_i(\mathbf{H}_T)_i$. Then, $(\mathbf{H}_F)_i$ ($i \in [I]$) are forecasted by means of a black-box procedure, *e.g.*, random forest regression. Finally, the forecasted value are retrieved as $\mathbf{M}_F := \mathbf{W}_i(\mathbf{H}_F)_i$.

Algorithm 4: LCF for time series forecasting

1: factorize $\mathbf{M}_T^\star \simeq \mathbf{W}_0\mathbf{H}_0$
2: divide the rows of $\mathbf{M}_T^\star$ in I clusters
3: **for each** $i \in [I]$ **do**
4:    define $(\mathbf{M}_T^\star)_i$ by selecting the time series in i
5:    factorize $(\mathbf{M}_T^\star)_i \simeq \mathbf{W}_i(\mathbf{H}_T)_i$
6:    forecast $(\mathbf{H}_F)_i$
7: **end for**
8: $\mathbf{M}_F := \mathbf{W}_0\mathbf{H}_F$

Remark 7. *Algorithm 4 defines a class of algorithms, whose instances are defined by choosing clustering strategy at Steps 2, matrix factorization approach at Steps 1 and 5, and forecasting algorithm at Step 6. Algorithm 4 can be trivially parallelized by implementing multi-threading for loop at Steps 4-6.*

We implemented two specific procedures for clustering and forecasting steps, detailed in next subsections.

3.1 Clustering (Step 2)

We performs initially a (nonnegative) factorization of observed matrix $\mathbf{M}_T^\star \simeq \mathbf{W}_0\mathbf{H}_0$. The clustering of time series is based on weight matrix $\mathbf{W}_0$: the rationale of this strategy is considering time series as "similar" if the corresponding linear combination of the columns of $(\mathbf{H}_T)_0$ has "similar weights". Hence, the full dendrogram of matrix $\mathbf{W}_0$ is computed via hierarchical clustering the rows of $\mathbf{W}_0$ with complete linkage and ℓ_1 affinity, implemented in the Python `sklearn 0.23` package under the routine `AgglomerativeClustering(compute_full_tree=True)` [Pedregosa et al., 2011].

The dendrogram is explored from top to the bottom by looking at the children of each node (see Algorithm 5). The node is split into two children if and only if at least one of its children has size at least d. If it is not the case, the node is the leaf of the output tree. Algorithm 5 is executed at the root node of the full dendrogram and returns the set of time series clusters.

Note that other clustering approaches might have been considered here. In practice, one need to pay attention that clusters should have maximal size d. The choice of the size d is discussed for the numerical experiments of Section 4.

Algorithm 5: Exploring dendrogram

1: **Input**: node s, cluster maximal dimension d
2: let child_1 and child_2 be the children of s
3: **if** size of child_1 $\leq d$ and size of child_2 $\leq d$ **then**
4:    **return** the set of all children of node s
5: **else**
6:    **for all** children c of s **do**
7:       **if** size of $c \leq d$ **then**
8:          **return** the set of all children of node c
9:       **else**
10:          recursively execute Algorithm 5 on node c
11:       **end if**
12:    **end for**
13: **end if**

Algorithm	RRMSE(K,W)	RMPE(K,W)
SMM on mAMF	5.24%(4,5)	**6.60%**(4,5)
SMM on mNMF	**5.15%**(4,5)	7.63%(4,4)
LCF	18.40%(10,–)	25.84%(4,–)
RFR	8.10%	7.98%

Table 2: Indices for weekly consumption data-set.

Algorithm	RRMSE(K,W)	RMPE(K,W)
SMM on mAMF	14.35%(4,5)	**34.50%**(4,20)
SMM on mNMF	**14.17%**(4,5)	41.90%(4,40)
LCF	21.22%(10,–)	52.97%(10,–)
RFR	14.50%	47.72%

Table 3: Indices for daily consumption data-set.

3.2 Forecasting (Step 6)

In order to forecast $\mathbf{H}_F$ we apply deep learning approach. The samples are defined via *sliding window transformation*: the time series are subdivided in overlapping intervals of time window $D \geq 1$ and, in supervised learning fashion, the relationship between the values of each interval (*inputs*) and the consecutive F (*outputs*) values is learnt by mean of a regression algorithm, *e.g.*, random forest regression.

4 Numerical Experiments

The interested reader may find a github repository on the numerical experiments at `https://github.com/Luca-Mencarelli/Nonnegative-Matrix-Factorization-Time-Series`

4.1 Real-world Data-sets

The numerical experiments refer to the weekly and daily electricity consumption data-set of $N = 370$ Portuguese customers during the period 2011-2014 ($T + F = 208$ and $T + F = 1456$ for weekly and daily measurements, respectively) [Trindade, 2016]. We chose the the last four weeks of 2014 as forecasting period ($F = 4$ and $F = 28$ for weekly and daily consumption, respectively).

We tests both SMM and LCF with a random initialization of matrices $\mathbf{H}^0, \mathbf{W}^0$. Each entry in $\mathbf{H}^0$ is randomly selected in $[0, h]$ where $h > 0$ is chosen by practitioner. Each row of matrix $\mathbf{W}^0$ is randomly generated in the corresponding standard simplex.

For SMM we implement both HALS for (mNMF) and PALM for (mAMF). Moreover, we consider two different strategies to define matrix $\mathbf{\Pi}(\mathbf{M})$: with *non-overlapping* and *overlapping* sliding intervals (for the overlapping case, we consider intervals with a common period of a week).

For LCF we consider different regression algorithms to forecast profiles $\mathbf{H}_F$, *e.g.*, *Long Short-Term Memory* (LSTM) and *Gated Recurrent Units* (GRU) deep neural networks with preliminary data standardization [Shewalkar et al., 2019] and *Autoregressive Integrated Moving Average* (ARIMA) models [Douc et al., 2014]: the best results in terms of forecasting quality and elapsed time are produced by *random forest regression*. Maximal dimension d in Algorithm 5 is set equal to nonnegative rank. For the matrix factorizations we apply PALM algorithm to Semi Normalized NMF (SNNMF)

Moreover, we have compared our method with other existing time series forecasting methods such as *Random Forest Regression* (RFR) and *EXponential Smoothing* (EXS).

The quality of the forecasted matrix $\mathbf{M}_F$ is measured by the relative root-mean-squared error (RRMSE) and the relative mean-percentage error (RMPE):

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

We run all the test on a MacBook Pro mounting macOS Sierra with 2.6 GHz Intel Core i5 processor and 8 GB 1600 MHz DDR3 memory.

Tables 2-3 report the cross-validated RRMSE and RMPE on observed values obtained during the computational tests for each methods for the weekly and daily consumption data-set, respectively. In parenthesis we detail the corresponding values of the eventual hyper-parameters K and W. SMM outperforms RFR, while performances of LCF are comparable with RFR.

Algorithm	RRMSE(K,W)	RMPE(K,W)	CPU (sec.)
SMM on mAMF	**0.28%(20,5)**	**0.27%(20,5)**	210.07
SMM on mNMF	0.28%(20,5)	0.27%(20,5)	867.83
LCF with RFR	36.32%	35.78%	393.58
LCF with EXP	49.47%	49.73%	86.39
RFR	23.89%	22.88%	953.67
EXS	48.44%	48.11%	**34.39**

Table 4: Indices for low noise data-set.

Algorithm	RRMSE(K,W)	RMPE(K,W)	CPU (sec.)
SMM on mAMF	0.79%(20,5)	0.83%(20,5)	5216.62
SMM on mNMF	**0.59%(20,5)**	**0.57%(20,5)**	5372.65
LCF with RFR	34.55%	34.57%	332.98
LCF with EXS	46.53%	46.02%	90.14
RFR	23.30%	21.75%	776.40
EXP	45.39%	44.10%	**29.84**

Table 5: Indices for medium noise data-set.

4.2 Synthetic Data-sets

Further computational experiments have been realized by consider three synthetic data-sets. Each data-set has been generated by replicating 5000 small 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 three data-sets as "low noise", "medium noise", and "high noise".

We tested the SMM both for mAMF and mNMF, and LCF with random forest, exponential smoothing, SARIMAX model, and LSTM or GRU deep learning network as regression algorithm to extend the archetypes. We set a time limit of 2 hours for all the methods: SARIMAX, LSTM and GRU exceed time limit. We compare the proposed methodologies against the same regression methods applied to forecast each time series separately.

In Tables 4-6 we report the performance indexes for each method. RFR stands for random forest, while EXS for exponential smoothing. For the LCF methodology the best results is obtained with rank equal to 10 for the first matrix factorization and equal to 3 for the matrix factorization corresponding to the matrix factorization for each cluster. Moreover, we set to 600 the number of clusters, obtaining 1800 archetypes to forecast and significantly reducing the number of forecast time series.

For all the data-sets the SMM outperforms all the other methods in terms of performance indexes. As the noise increases, the LCF methodology produces a forecast with performance indexes comparable with the ones corresponding to the plain forecasting of each single time series.

The CPU time of SMM increases with the noise, but it significantly depend on the rank. For instance, in the medium noise data-set a forecast comparable with the best one (reported in Table 5) is obtained for rank equal to 10 in a CPU time of 1386.74 seconds.

EXS is the fastest method but one of the less accurate with respect to reconstruction error. LCF with RFR is faster than the plain RFR by taking advantage from the smaller number of time series to extends. Unfortunately, LCF with EXS is slower that EXS since the small number of extended time series does not compensate the time devoted to performs the matrix factorization for each cluster.

5 Conclusions and Perspectives

In this paper, we have introduced and described two novel approaches for the time series forecasting problem relying on nonnegative matrix factorization. We apply these algorithms to a realistic data-sets, namely the daily and weekly Portuguese electricity consumption data-set, and synthetics data-sets, showing the forecasting capabilities of the proposed methodology.

Moreover, we have shown several uniqueness and robustness theoretical results for the solution of the matrix factorization problems faced by the two algorithms, namely the *Sliding Mask Method* and the *Latent Clustered Forecast*.

The strength of the proposed methodology consists in its relatively loose assumptions, mainly by supposing that time series matrix can be efficiently described by a low rank nonnegative decomposition, and that the time series are periodic for the *Sliding Mask Method*.

Algorithm	RRMSE(K,W)	RMPE(K,W)	CPU (sec.)
SMM on mAMF	**30.85%(10,5)**	**29.86(10,5)**	2120.79
SMM on mNMF	38.83%(10,5)	30.23%(10,5)	4576.86
LCF with RFR	35.40%	32.85%	427.90
LCF with EXS	37.51%	33.66%	191.94
RFR	33.84%	30.63%	772.89
EXP	37.13%	33.33%	**28.86**

Table 6: Indices for high noise data-set.

Future works consists in embedding side information in the forecasting procedure by extending algorithms in [Mei et al., 2019] to the *Sliding Mask Method* and the *Latent Clustered Forecast*.

References

[Arlot and Celisse, 2010] Arlot, S. and Celisse, A. (2010). A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4:40–79.

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

[Cichocki and Zdunek, 2006] Cichocki, A. and Zdunek, R. (2006). *NMFLAB for signal processing*.

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

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

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

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

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

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

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

[Ge and Zou, 2015] Ge, R. and Zou, J. (2015). Intersecting faces: Non-negative matrix factorization with new guarantees. In Bach, F. R. and Blei, D. M., editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2295–2303.

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

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

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

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

[Gillis and Glineur, 2012a] Gillis, N. and Glineur, F. (2012a). Accelerated multiplicative updates and hierarchical ALS algorithms for nonnegative matrix factorization. *Neural Computation*, 24(4):1085–1105.

[Gillis and Glineur, 2012b] Gillis, N. and Glineur, F. (2012b). On the geometric interpretation of the nonnegative rank. *Linear Algebra and its Applications*, 437(11):2685–2712.

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

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

[Javadi and Montanari, 2020b] Javadi, H. and Montanari, A. (2020b). Supplement To "Non-negative Matrix Factorization via Archetypal Analysis". *Journal of the American Statistical Association*, 115(530):896–907.

[Laurberg et al., 2008] Laurberg, H., Christensen, M., Plumbley, M., Hansen, L., and Jensen, S. (2008). Theorems on positive data: On the uniqueness of NMF. *Computational Intelligence and Neuroscience*, 2008:1–9.

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

[Mei et al., 2019] Mei, J., De Castro, Y., Goude, Y., Azaïs, J.-M., and Hébrail, G. (2019). Nonnegative matrix factorization with side information for time series recovery and prediction. *IEEE Transactions on Knowledge and Data Engineering*, 31(3):493–506.

[Nguyen et al., 2019] Nguyen, L. T., Kim, J., and Shim, B. (2019). Low-rank matrix completion: A contemporary survey. *IEEE Access*, 7:94215–94237.

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

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

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[Pock and Sabach, 2016] Pock, T. and Sabach, S. (2016). Inertial proximal alternating linearized minimization (iPALM) for nonconvex and nonsmooth problems. *SIAM Journal on Imaging Sciences*, 9(4):1756–1787.

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

[Shewalkar et al., 2019] Shewalkar, A., Nyavanandi, D., and Ludwig, S. A. (2019). 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.

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

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

[Trindade, 2016] Trindade, A. (2016). UCI Maching Learning Repository - ElectricityLoadDiagrams20112014 Data Set. http://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014. [Online; accessed 13-August-2020].

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

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

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

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

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

[Xu et al., 2003] Xu, W., Liu, X., and Gong, Y. (2003). 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*, pages 267–273.

[Yokota et al., 2018] Yokota, T., Erem, B., Guler, S., Warfield, S. K., and Hontani, H. (2018). 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*, pages 8251–8259.

[Yu et al., 2016] Yu, H.-F., Rao, N., and Dhillon, I. S. (2016). Temporal regularized matrix factorization for high-dimensional time series prediction. In *NIPS*, pages 847–855.

A Proofs

A.1 Proof of Theorem 2

• We start by proving that *Condition* (**A1**) *is sufficient for* (**POU**).

Let $\mathbf{H}_0 := [\mathbf{H}_{0T}\ \mathbf{H}_{0F}]$, $\mathbf{W}_0^\top := [\mathbf{W}_{0\text{train}}^\top\ \mathbf{W}_{0\text{test}}^\top]$, $\mathbf{H} := [\mathbf{H}_T\ \mathbf{H}_F]$, and $\mathbf{W}^\top := [\mathbf{W}_{\text{train}}^\top\ \mathbf{W}_{\text{test}}^\top]$. Assumption (**A1**) implies that decomposition $\mathbf{W}_{0\text{train}}\mathbf{H}_0$ and $\mathbf{W}_0\mathbf{H}_{0T}$ are unique. By Theorem 1, it holds

$$
\begin{aligned}
\mathbf{W}_{0\text{train}}\mathbf{H}_0 = \mathbf{W}_{\text{train}}\mathbf{H} &\Longrightarrow (\mathbf{W}_{0\text{train}}, \mathbf{H}_0) \equiv (\mathbf{W}_{\text{train}}, \mathbf{H}) \\
\mathbf{W}_0\mathbf{H}_{0T} = \mathbf{W}\mathbf{H}_T &\Longrightarrow (\mathbf{W}_0, \mathbf{H}_{0T}) \equiv (\mathbf{W}, \mathbf{H}_T)\,,
\end{aligned}
$$

where $\equiv$ stands for equality up to permutation and positive scaling of columns (resp. rows) of $\mathbf{W}_0$ (resp. $\mathbf{H}_0$). Hence, if (**A1**) holds, then

$$
(\mathbf{W}_{0\text{train}}\mathbf{H}_0 = \mathbf{W}_{\text{train}}\mathbf{H}) \wedge (\mathbf{W}_0\mathbf{H}_{0T} = \mathbf{W}\mathbf{H}_T) \Longrightarrow (\mathbf{W}_0, \mathbf{H}_0) \equiv (\mathbf{W}, \mathbf{H})\,. \tag{3}
$$

Moreover, note $\mathbf{T}(\mathbf{W}_{0\text{train}}\mathbf{H}_0) = \mathbf{T}_{\text{train}}(\mathbf{X}_0) = \mathbf{W}_{0\text{train}}\mathbf{H}_0$ and $\mathbf{T}(\mathbf{W}_0\mathbf{H}_{0T}) = \mathbf{T}_T(\mathbf{X}_0) = \mathbf{W}_0\mathbf{H}_{0T}$ (same equations holds for $(\mathbf{W}, \mathbf{H})$). We deduce that $\mathbf{T}(\mathbf{W}_0\mathbf{H}_0) = \mathbf{T}(\mathbf{W}\mathbf{H})$ implies $(\mathbf{W}_{0\text{train}}\mathbf{H}_0 = \mathbf{W}_{\text{train}}\mathbf{H}) \wedge (\mathbf{W}_0\mathbf{H}_{0T} = \mathbf{W}\mathbf{H}_T)$. We deduce the result by (3).

• We prove that *If* (**A1**) *and* (**A2**) *holds,* $\mathbf{T}(\mathbf{W}\mathbf{H}) = \mathbf{T}(\mathbf{W}_0\mathbf{H}_0)$ *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.*

By the previous point, we now that (**A1**) implies $(\mathbf{W}_0, \mathbf{H}_0) \equiv (\mathbf{W}, \mathbf{H})$. So that there exist $\lambda_1, \ldots, \lambda_K$ positive and a permutation $\sigma(1), \ldots, \sigma(K)$ such that

$$
\forall i \in [n-N], \forall k \in [K], \quad (\mathbf{W})_k^{(i)} = \lambda_{\sigma(k)}(\mathbf{W}_0)_{\sigma(k)}^{(i)}\,.
$$

Recall that $\mathbf{W}\mathbf{1} = \mathbf{1}$ (resp. $\mathbf{W}_0\mathbf{1} = \mathbf{1}$) so that the rows of $\mathbf{W}$ (resp. $\mathbf{W}_0$) belongs to the affine space

$$
\mathscr{A}_{\mathbf{1}} := \left\{ w \in \mathbb{R}^K \ :\ \langle w, \mathbf{1} \rangle = 1 \right\}.
$$

Namely, for a given row $i \in [n-N]$, we have

$$
\begin{aligned}
(\mathbf{W}_0)^{(i)}\mathbf{1} = \mathbf{1} &\Rightarrow \sum_{k=1}^K (\mathbf{W}_0)_k^{(i)} = 1 \\
\mathbf{W}^{(i)}\mathbf{1} = \mathbf{1} &\Rightarrow \sum_{k=1}^K \lambda_{\sigma(k)}(\mathbf{W}_0)_{\sigma(k)}^{(i)} = 1
\end{aligned}
$$

Which proves that $(\mathbf{W}_0)^{(i)} \in \mathscr{A}_{\mathbf{1}} \cap \mathscr{A}_{\lambda_{\sigma^{-1}}}$, for all $i \in [n-N]$, where

$$
\mathscr{A}_{\lambda_{\sigma^{-1}}} := \left\{ w \in \mathbb{R}^K \ :\ \sum_{k=1}^K \lambda_{\sigma^{-1}(k)} w_k = 1 \right\},
$$

is the affine space orthogonal to $\mathbf{d} := (\lambda_{\sigma^{-1}(1)}, \ldots, \lambda_{\sigma^{-1}(K)})$. We deduce that the rows $(\mathbf{W}_0)^{(i)}$ belong to the affine space

$$
\mathscr{A} := \left\{ w \in \mathbb{R}^K \ :\ \langle w, \mathbf{1} \rangle = 1 \text{ and } \langle w, \mathbf{d} \rangle = 1 \right\}
$$

which is of:
• co-dimension 2 if $\mathbf{d}$ is not proportional to $\mathbf{1}$;
• co-dimension 1 if there exists $\lambda > 0$ such that $\mathbf{d} = \lambda\mathbf{1}$. In this latter case, $\lambda = 1$ and for all $k \in [K]$, $\lambda_k = 1$, namely there is no scaling of the columns.

If $\mathscr{A}$ is of co-dimension 2 then $\mathscr{A}$ is of dimension $K-2$ and $\mathrm{Conv}(\mathbf{W}_{0,\text{train}}) \subseteq \mathscr{A}$ cannot contain a ball of dimension $K-1$, which implies that $\mathrm{Conv}(\mathbf{T}_{\text{train}}(\mathbf{X}_0)) \subseteq \mathscr{A} \times H$ is of dimension at most $K-2$ and it cannot contain a ball of dimension $K-1$ (i.e., co-dimension 1), where $\mathscr{A} \times H = \{x \ :\ \exists a \in \mathscr{A} \text{ s.t. } x = a^\top H\}$. This latter is a contradiction under (**A2**). We deduce that $\mathscr{A}$ is of co-dimension 2, and so there is no scaling.

A.2 Proof of Theorem 3

This proof follows the pioneering work [Javadi and Montanari, 2020a]. In this latter paper, the authors consider neither masks $\mathbf{T}$ nor nonnegative constraints on $\mathbf{H}$ as in (mNMF). Nevertheless,
1/ considering the hard constrained programs (4) and (6) below;

2/ remarking that it holds $\widetilde{\mathscr{D}}(\mathbf{H},\mathbf{X}) \le \mathscr{D}(\mathbf{H},\mathbf{X})$ and $\overline{\mathscr{D}}(\mathbf{X},\mathbf{H}) \le \mathscr{D}(\mathbf{X},\mathbf{H})$;
then a careful reader can note that their proof extends to masks $\mathbf{T}$ and nonnegative constraints on $\mathbf{H}$. For sake of completeness we reproduce here the steps that need to be changed in their proof. A reading guide of the 60 pages proof of [Javadi and Montanari, 2020b] is given in Section B.

Step 1: reduction to hard constrained Programs (4) and (6)

Consider the constrained problem:

$$\begin{aligned}\widehat{\mathbf{H}} \in \arg\min_{\mathbf{H}} \ & \widetilde{\mathscr{D}}(\mathbf{H},\mathbf{X}) \\ \text{s.t. } & \overline{\mathscr{D}}(\mathbf{X},\mathbf{H}) \le \Delta_1^2 .\end{aligned} \tag{4}$$

where

$$\overline{\mathscr{D}}(\mathbf{X},\mathbf{H}) := \min_{\mathbf{W}\ge 0,\ \mathbf{W}\mathbf{1}=\mathbf{1}} \|\mathbf{T}(\mathbf{X}-\mathbf{W}\mathbf{H})\|_F^2$$

Then (mAMF) can be seen as Lagrangian formulation of this problem setting $\Delta_1^2 = \overline{\mathscr{D}}(\mathbf{X},\widehat{\mathbf{H}}_{(\mathrm{mAMF})})$, where $\widehat{\mathbf{H}}_{(\mathrm{mAMF})}$ is a solution to (mAMF). We choose Δ_1 so as to bound the noise level $\|\mathbf{F}\|_F$

$$\Delta_1^2 \ge \|\mathbf{F}\|_F^2 . \tag{5}$$

Consider the constrained problem:

$$\begin{aligned}\widehat{\mathbf{H}} \in \arg\min_{\mathbf{H}\ge 0} \ & \widetilde{\mathscr{D}}(\mathbf{H},\mathbf{X}) \\ \text{s.t. } & \overline{\mathscr{D}}(\mathbf{X},\mathbf{H}) \le \Delta_2^2 .\end{aligned} \tag{6}$$

Then (mNMF) can be seen as Lagrangian formulation of this problem setting $\Delta_2^2 = \overline{\mathscr{D}}(\mathbf{X},\widehat{\mathbf{H}}_{(\mathrm{mNMF})})$, where $\widehat{\mathbf{H}}_{(\mathrm{mNMF})}$ is a solution to (mNMF). We choose Δ_1 so as to bound the noise level $\|\mathbf{F}\|_F$

$$\Delta_2^2 \ge \|\mathbf{F}\|_F^2 . \tag{7}$$

Step 2: First bound on the loss

Denote $\mathscr{D} := \left\{\mathscr{D}(\mathbf{H},\mathbf{H}_0)^{1/2} + \mathscr{D}(\mathbf{H}_0,\mathbf{H})^{1/2}\right\}$. By Assumption (**A2**) we have

$$\boldsymbol{z}_0 + \boldsymbol{U}B_{K-1}(\mu) \subseteq \mathrm{conv}(\boldsymbol{X}_0) \subseteq \mathrm{conv}(\boldsymbol{H}_0)\,,$$

where $\boldsymbol{z}_0 + \boldsymbol{U}B_{K-1}(\mu)$ is a parametrization of the ball of center $\boldsymbol{z}_0$ and radius μ described in Assumption (**A2**) with $\boldsymbol{U}$ a matrix whose columns are $K-1$ orthonormal vectors. Using Lemma 6, we get that

$$\mu\sqrt{2} \le \sigma_{\min}(\boldsymbol{H}_0) \le \sigma_{\max}(\boldsymbol{H}_0)\,,$$

where $\sigma_{\min}(\boldsymbol{H}_0), \sigma_{\max}(\boldsymbol{H}_0)$ denote its largest and smallest nonzero singular values. Then, since $\boldsymbol{z}_0 \in \mathrm{conv}(\boldsymbol{H}_0)$ we have $\boldsymbol{z}_0 = \boldsymbol{H}_0^\top \alpha_0$ for some α_0 s.t. $\alpha_0 \mathbf{1} = \mathbf{1}$. It holds,

$$\|\boldsymbol{z}_0\|_2 \le \sigma_{\max}(\boldsymbol{H}_0)\|\alpha_0\|_2 \le \sigma_{\max}(\boldsymbol{H}_0). \tag{8}$$

Note that

$$\sigma_{\max}(\widehat{\boldsymbol{H}} - \mathbf{1}\boldsymbol{z}_0^\top) \le \sigma_{\max}(\widehat{\boldsymbol{H}}) + \sigma_{\max}(\mathbf{1}\boldsymbol{z}_0^\top) = \sigma_{\max}(\widehat{\boldsymbol{H}}) + \sqrt{K}\|\boldsymbol{z}_0\|_2. \tag{9}$$

Therefore, using Lemma 8 we have

$$\mathscr{D} \le c\left[K^{3/2}\Delta_{1/2}\kappa(\boldsymbol{P}_0(\widehat{\boldsymbol{H}})) + \frac{\sigma_{\max}(\widehat{\boldsymbol{H}})\Delta_{1/2}K^{1/2}}{\mu} + \frac{K\Delta_{1/2}\|\boldsymbol{z}_0\|_2}{\mu}\right] + c\sqrt{K}\|\mathbf{F}\|_F\,, \tag{10}$$

where $\Delta_{1/2}$ equals Δ_1 for problem (4) and Δ_2 for problem (6), and $\kappa(\boldsymbol{A})$ stands for the conditioning number of matrix $\boldsymbol{A}$. In addition, Lemma 9 implies that

$$\mathscr{L}(\boldsymbol{H}_0,\widehat{\boldsymbol{H}})^{1/2} \le \frac{1}{\alpha}\max\left\{(1+\sqrt{2})\sqrt{K}, \sqrt{2}\kappa(\boldsymbol{H}_0)\right\}\mathscr{D}\,. \tag{11}$$

Step 3: Combining and final bound

By Lemma 10 it holds

$$\mathscr{D} \le c\Big[\frac{K^{3/2}\mathscr{D}\Delta_{1/2}}{\alpha(\mu-2\Delta_{1/2})\sqrt{2}} + \frac{K^2\sigma_{\max}(\boldsymbol{H}_0)\Delta_{1/2}}{(\mu-2\Delta_{1/2})\sqrt{2}} + \frac{\mathscr{D}K^{1/2}\Delta_{1/2}}{\alpha\mu} + \frac{\sigma_{\max}(\boldsymbol{H}_0)\Delta_{1/2}K}{\mu} + \frac{K\Delta_{1/2}\|\boldsymbol{z}_0\|_2}{\mu}\Big] + c\sqrt{K}\|\mathbf{F}\|_F. \tag{12}$$

We understand that $\mathscr{D} = \mathscr{O}_{\Delta_{1/2}\to 0}(\Delta_{1/2})$ and for small enough $\Delta_{1/2}$ there exists a constant $c > 0$ such that

$$\mathscr{D} \le c\Delta_{1/2} + c\sqrt{K}\|\mathbf{F}\|_F$$

By (5) and (7), it yields that for small enough noise error $\|\mathbf{F}\|_F$ one has

$$\mathscr{D} \le c\|\mathbf{F}\|_F\,,$$

for some (other) constant $c > 0$. Plugging this result in (11) we prove the result.

A.3 Proof of Theorem 4

$\mathbf{N} \mapsto \nabla_{\mathbf{N}} h(\mathbf{H}, \mathbf{W}, \mathbf{N})$ is Lipschitz continuous with moduli $L = 2$. The statement follows from Proposition 4.1 in [Javadi and Montanari, 2020a] and from Theorem 1 in [Bolte et al., 2014].

B Propositions and Lemmas

• Results that we can use directly from [Javadi and Montanari, 2020b]: Lemma B.1, Lemma B.2, Lemma B.3.
• Results of [Javadi and Montanari, 2020b] that has to be adapted: Lemme B.4 (done in Lemma 6), Lemma B.5 (done in Lemma 7), and Lemma B.6 (done in Lemma 8).

Proposition 5. *For $\widehat{\boldsymbol{H}}$ solution to (4) (or (6)) one has $\widetilde{\mathscr{D}}(\widehat{\boldsymbol{H}}, \mathbf{X}) \le \widetilde{\mathscr{D}}(\mathbf{H}_0, \mathbf{X})$.*

Proof. Observe that $\widetilde{\mathscr{D}}(\mathbf{X}, \mathbf{H}_0) = \|\mathbf{F}\|_F^2$. By (5) and (7), $\mathbf{H}_0$ is feasible for (4) (or (6)) then $\widetilde{\mathscr{D}}(\widehat{\boldsymbol{H}}, \mathbf{X}) \le \widetilde{\mathscr{D}}(\mathbf{H}_0, \mathbf{X})$ □

Lemma 6 (Adapted version of Lemma B.4 of [Javadi and Montanari, 2020b])**.** *If $\boldsymbol{H}$ is feasible for problem (4) (or (6)) and has linearly independent rows, then we have*

$$\sigma_{\min}(\boldsymbol{H}) \ge \sqrt{2}(\mu - 2\Delta_{1/2})\,, \tag{13}$$

where $\Delta_{1/2}$ equals Δ_1 for problem (4) and Δ_2 for problem (6).

Proof. Consider the notation and the outline of proof Lemma B.4 in [Javadi and Montanari, 2020b]. The adaptation is simple here. The trick is to only consider rows in the training set, $\mathbf{T}_{\text{train}}(\mathbf{X}_0)$: the indice i of proof of Lemma B.4 in [Javadi and Montanari, 2020b] correspond to the $n-N$ first rows in our case (the training set); and one should replace $\mathbf{X}_0$ by $\mathbf{T}_{\text{train}}(\mathbf{X}_0)$. This proof requires only feasibility of $\boldsymbol{H}$ and works no matter if a nonnegative constraint on $\boldsymbol{H}$ is active (as in Program (6)). □

Lemma 7 (Adapted version of Lemma B.5 of [Javadi and Montanari, 2020b])**.** *For $\widehat{\boldsymbol{H}}$ solution to (4) (or (6)), it holds*

$$\widetilde{\mathscr{D}}(\widehat{\boldsymbol{H}}, \mathbf{X}_0)^{1/2} \le \widetilde{\mathscr{D}}(\mathbf{H}_0, \mathbf{X}_0)^{1/2} + c\sqrt{K}\|\mathbf{F}\|_F\,.$$

Proof. Consider the notation and the outline of proof Lemma B.5 in [Javadi and Montanari, 2020b]. Note that Eq. (B.103) holds by Proposition 5. Form Eq. (B.104), the proof remains unchanged once one substitutes $\mathscr{D}$ by $\widetilde{\mathscr{D}}$. □

Lemma 8 (Adapted version of Lemma B.6 of [Javadi and Montanari, 2020b])**.** *For $\widehat{\boldsymbol{H}}$ the optimal solution of problem (4) (or (6)), we have*

$$\alpha(\mathscr{D}(\widehat{\boldsymbol{H}}, \boldsymbol{H}_0)^{1/2} + \mathscr{D}(\boldsymbol{H}_0, \widehat{\boldsymbol{H}})^{1/2}) \le c\left[K^{3/2}\Delta_{1/2}\kappa(\boldsymbol{P}_0(\widehat{\boldsymbol{H}})) + \frac{\Delta_{1/2}\sqrt{K}}{\mu}\sigma_{\max}(\widehat{\boldsymbol{H}} - \mathbf{1}\boldsymbol{z}_0^\top)\right] + c\sqrt{K}\|\mathbf{F}\|_F \tag{14}$$

where $\boldsymbol{P}_0 : \mathbb{R}^d \to \mathbb{R}^d$ is the orthogonal projector onto $\text{aff}(\boldsymbol{H}_0)$ (in particular, $\boldsymbol{P}_0$ is an affine map), and $\Delta_{1/2}$ equals Δ_1 for problem (4) and Δ_2 for problem (6).

Proof. Invoke the proof of Lemma B.6 in [Javadi and Montanari, 2020b] using the fact that $\widetilde{\mathscr{D}}(\mathbf{H},\mathbf{X}) \leq \mathscr{D}(\mathbf{H},\mathbf{X})$ and $\overline{\mathscr{D}}(\mathbf{X},\mathbf{H}) \leq \mathscr{D}(\mathbf{X},\mathbf{H})$. □

Lemma 9. *Let $\boldsymbol{H}, \boldsymbol{H}_0$ be matrices with linearly independent rows. We have*

$$\mathscr{L}(\boldsymbol{H}_0,\boldsymbol{H})^{1/2} \leq \sqrt{2}\kappa(\boldsymbol{H}_0)\mathscr{D}(\boldsymbol{H}_0,\boldsymbol{H})^{1/2} + (1+\sqrt{2})\sqrt{K}\mathscr{D}(\boldsymbol{H},\boldsymbol{H}_0)^{1/2}\,, \tag{15}$$

where $\kappa(\boldsymbol{A})$ stands for the conditioning number of matrix $\boldsymbol{A}$.

Proof. See Lemma B.2 in [Javadi and Montanari, 2020b] □

Lemma 10. *It holds*

$$\kappa(\boldsymbol{P}_0(\widehat{\boldsymbol{H}})) \leq \Big[\frac{\mathscr{D}}{\alpha(\mu-2\Delta_{1/2})\sqrt{2}} + \frac{K^{1/2}\sigma_{\max}(\boldsymbol{H}_0)}{(\mu-2\Delta_{1/2})\sqrt{2}}\Big].$$

Proof. The proof is given by Equations B.189-194 in [Javadi and Montanari, 2020b]. □

C Algorithms for mNMF

In this section we report *Block Coordinate Descend* (BCD) Algorithm (see Algorithm 6) and accelerated *Hierarchical Alternate Least Square* (HALS) for mNMF (see Algorithm 7), which is a generalization of Algorithm described in [Gillis and Glineur, 2012a] to the matrix factorization with mask.

Algorithm 6: BCD for mNMF

Initialization: choose $\mathbf{H}^0 \geq \mathbf{0}, \mathbf{W}^0 \geq \mathbf{0}$, and $\mathbf{N}^0 \geq \mathbf{0}$, set $i := 0$.
while stopping criterion is not satisfied **do**
  $\mathbf{H}^{i+1} := \text{update}(\mathbf{H}^i, \mathbf{W}^i, \mathbf{N}^i)$
  $\mathbf{W}^{i+1} := \text{update}(\mathbf{H}^{i+1}, \mathbf{W}^i, \mathbf{N}^i)$
  $\mathbf{N}^{i+1} := \text{update}(\mathbf{H}^{i+1}, \mathbf{W}^{i+1}, \mathbf{N}^i)$
  $i := i+1$
end while

Algorithm 7: accelerated HALS for mNMF

Initialization: choose $\mathbf{H}^0 \geq \mathbf{0}, \mathbf{W}^0 \geq \mathbf{0}$, *nonnegative rank* K, and $\alpha > 0$. Set $\mathbf{N}^0 = \mathscr{P}_{\mathbf{X}}(\mathbf{W}^0\mathbf{H}^0)$, $\rho_{\mathbf{W}} := 1 + n(m+K)/(m(K+1))$, $\rho_{\mathbf{H}} := 1 + m(n+K)/(n(K+1))$, and $i := 0$.
while stopping criterion is not satisfied **do**
  for $k \leq k_{\mathbf{W}} := \lfloor 1 + \alpha\rho_{\mathbf{W}} \rfloor$ **do**
    $\mathbf{A} := \mathbf{N}{\mathbf{H}^k}^\top$, $\mathscr{B} := \mathbf{H}^k{\mathbf{H}^k}^\top$
    for $\ell \in [K]$ **do**
      $C_\ell := \sum_{j=1}^{\ell-1} W_j^{k+1} B_{j\ell} + \sum_{j=\ell+1}^{K} W_j^k B_{j\ell}$
      $W_\ell^k := \max(0, (A_\ell - C_\ell)/B_{\ell\ell})$
    end for
    $\mathbf{W}^{k_{\mathbf{W}}} := \mathscr{P}_\Delta(\mathbf{W}^{k_{\mathbf{W}}})$
  end for
  for $k \leq k_{\mathbf{H}} := \lfloor 1 + \alpha\rho_{\mathbf{H}} \rfloor$ **do**
    $\mathbf{A} := \mathbf{W}^{k_{\mathbf{W}}}\mathbf{N}$, $\mathscr{B} := {\mathbf{W}^{k_{\mathbf{W}}}}^\top\mathbf{W}^{k_{\mathbf{W}}}$
    for $\ell \in [n]$ **do**
      $C_\ell := \sum_{j=1}^{\ell-1} H_j^{k+1} B_{j\ell} + \sum_{j=\ell+1}^{n} H_j^k B_{j\ell}$
      $H_\ell^k := \max(0, (A_\ell - C_\ell)/B_{\ell\ell})$
    end for
  end for
  $\mathbf{W}^{i+1} := \mathbf{W}^{k_{\mathbf{W}}}$, $\mathbf{H}^{i+1} := \mathbf{H}^{k_{\mathbf{H}}}$
  $\mathbf{N}^{i+1} := \mathscr{P}_{\mathbf{X}}(\mathbf{W}^{i+1}\mathbf{H}^{i+1})$
  $i := i+1$
end while

D KKT conditions for mNMF

In this section we determine the KKT condition for mNMF problem, namely

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

Let us introduce the dual variables $\mathbf{V}\geq\mathbf{0}$, $\mathscr{G}\geq\mathbf{0}$, $t\in\mathbb{R}^n$, and $\mathscr{Z}\in\text{range}(\Pi)$ such that $\mathbf{T}(\mathscr{Z})=\mathscr{Z}$. The Lagrangian of mNMF problem is

$$\mathscr{L}(\mathbf{N},\mathbf{W},\mathbf{H},\mathbf{V},\mathscr{G},t,\mathscr{Z}) = \mathscr{F}(\mathbf{N},\mathbf{W},\mathbf{H}) - \langle\mathbf{W},\mathbf{V}\rangle + \langle\mathbf{W}\mathbf{1}_K-\mathbf{1}_N,t\rangle - \langle\mathbf{H},\mathscr{G}\rangle - \langle\mathbf{N}-\mathbf{X},\mathscr{Z}\rangle\,.$$

The KKT condition are the following:

$$\nabla_{\mathbf{N}}\mathscr{L} = \mathbf{N}-\mathbf{WH}-\mathscr{Z}=\mathbf{0} \iff \mathbf{T}(\mathbf{N}-\mathbf{WH})=\mathscr{Z}\wedge\mathbf{T}^{\perp}(\mathbf{N}-\mathbf{WH})=\mathbf{0} \tag{16}$$

$$\nabla_{\mathbf{W}}\mathscr{L} = (\mathbf{WH}-\mathbf{N})\mathbf{H}^\top-\mathbf{V}-t\,\mathbf{1}_K^\top=\mathbf{0} \iff \mathbf{V}=(\mathbf{WH}-\mathbf{N})\mathbf{H}^\top-t\,\mathbf{1}_K^\top \tag{17}$$

$$\nabla_{\mathbf{H}}\mathscr{L} = \mathbf{W}^\top(\mathbf{WH}-\mathbf{N})-\mathscr{G} \iff \mathscr{G}=\mathbf{W}^\top(\mathbf{WH}-\mathbf{N}) \tag{18}$$

$$\langle\mathbf{W},\mathbf{V}\rangle=\mathbf{0} \iff \langle\mathbf{W},\nabla_{\mathbf{W}}\mathscr{F}-t\,\mathbf{1}_K^\top\rangle=\mathbf{0} \tag{19}$$

$$\langle\mathbf{H},\mathscr{G}\rangle=\mathbf{0} \iff \langle\mathbf{H},\nabla_{\mathbf{H}}\mathscr{F}\rangle=\mathbf{0} \tag{20}$$

From the complementarity conditions (19), it follows:

$$W_{i,j}>0 \Longrightarrow V_{i,j}=0 \Longrightarrow t_i=-(\nabla_W\mathscr{F})_{ij}\ \forall j$$

In order to compute t_i, we can select a row $W^{(}i)$, find any entry $W_{i,j}>0$ and apply the previous formula. In the practical implementation phase, in order to make numerically more stable the estimation of t_i's, we can adopt a slightly different strategy by averaging the values of t_i computed per row entry $W_{i,j}>0$.

E Details on electricity data-set tests

E.1 Sliding Mask Method

For the non-overlapping interval we test $W\in\{4,13,16\}$ and periodicity $P=4$ and $P=28$ for the weekly and daily data-set, respectively. For the overlapping intervals we test $W\in\{2,4,13,26\}$ with periodicity $P=8$ and $P=56$ for the weekly and daily data-set, respectively. We select nonnegative rank in $\{4,5,8,10,15,16,20,30,32\}$ for weekly data-set and in $\{5,10,20,30,40,50,60,70,80,90,100\}$ for daily data-set.

E.2 Latent Clustered Forecast

We consider nonnegative rank in $K\in\{5,10,15\}$ for the first and second nonnegative matrix factorization and the maximal dimension d in Explore dendrogram Algorithm equal to K. For the matrix factorization we apply PALM algorithm to Semi Normalized NMF (SNNMF).