Nonnegative matrix factorization with side information for time series recovery and prediction

Jiali Mei, Yohann De Castro, Yannig Goude, Jean-Marc Azaïs, and Georges Hébrail

Abstract—Motivated by the recovery and prediction of electricity consumption, we extend Nonnegative Matrix Factorization to take into account side information (column or row features). We consider general linear measurement settings, and propose a framework which models non-linear relationships between features and the response variables. We extend previous theoretical results to obtain a sufficient condition on the identifiability of NMF in this setting. Based on the classical Hierarchical Alternating Least Squares (HALS) algorithm, we propose a new algorithm (HALSX, or Hierarchical Alternating Least Squares with eXogeneous variables) which estimates the NMF model. The algorithm is validated on both simulated and real electricity consumption datasets as well as a recommendation system dataset, to show its performance in matrix recovery and prediction for new rows and columns.

Index Terms—Matrix Factorization, Recommender System, Time Series Analysis

1 Introduction

In recent years, a large number of methods have been developed to improve matrix completion methods using side information [1], [2], [3]. By including features linked to the row and/or columns of a matrix, also called "side information", these methods have a better performance at estimating the missing entries.

We generalize this idea to nonnegative matrix factorization (NMF, [4]), a low-rank matrix approximation method under nonnegativity constraint. Although more difficult to identify, NMF often has a better empirical performance, and provides factors that are more interpretable in an appropriate context. We propose an NMF method that takes into account side information. Given some (potentially partial) observations of a matrix, we propose to jointly estimate the nonnegative factors of the matrix, and regression models of these factors on the side information. This allows us to improve the matrix recovery performance of NMF. Moreover, using the regression models, we can predict the value of interest for new rows and columns that are previously unseen. We develop this method in the general matrix recovery context, where linear measurements, or linear combinations of matrix entries, are observed instead of matrix entries.

This choice is especially motivated by applications in electricity consumption modeling. We are interested in estimating and predicting the electricity load from temporal aggregates. In the context of load balancing in the power market, transmission system operators (TSO) of the electricity network are typically legally bound to estimate the electricity consumption and production at a small temporal scale (half-hourly or hourly), for all market participants within their perimeter, *i.e.* utility providers, traders, large consumers, groups of consumers, *etc.*. Most traditional electricity meters do not provide information at such a fine temporal scale. Although smart meters can record consumption locally every minute, the usage of such data can be extremely constrained for TSOs, because of the high cost of data transmission and processing and/or privacy issues. Nowadays, TSOs often use regulator-approved proportional consumption profiles to estimate market participants' load. In a previous work by the authors [5], we proposed to solve the estimation problem by NMF using only temporal aggregate data.

By introducing an additional regression level in NMF, we use exogenous variables as side information. Temporal aggregate data are used jointly with features that are known to have a correlation with electricity consumption, such as the temperature, the day of the week, or the type of client (see [6], [7] and references within for recent works in electricity load modeling). This not only improves the performance of load estimation, but also allows us to predict future load for users in the dataset, and estimate and predict the past and future consumption of new individuals. In electrical power networks, load prediction for new periods is useful for balancing supply-demand on the network, and prediction for new individuals is useful for network planning. Moreover, by examining the relationship between external features, the factors produced by NMF can be much more interpretable.

In the rest of this section, we introduce the general framework of this method, and the related literature. In Section 2, we deduce a sufficient condition on the side information for NMF to be unique. In Section 3, we present Hierarchical Alternating Least Squares with eXogeneous variables (HALSX), an algorithm which solves the NMF problem with side information, for which we prove a local

- *J. Mei, Y. Goude and G. Hébrail are with EDF Lab Paris-Saclay, 91120 Palaiseau, France.*
- *J. Mei, Y. De Castro, and Y. Goude are with Laboratoire des Mathématiques d'Orsay, Univ. Paris-Sud, CNRS, Université Paris-Saclay, 91405 Orsay, France.*
- *J-M. Azaïs is with Institut de Mathématiques, Université Paul Sabatier 31062 Toulouse, France*

convergence property. In Section 4, we present experimental results, applying HALSX on simulated and real datasets, both for the electricity consumption application, and for a standard task in collaborative filtering.

1.1 General model definition

We are interested in recovering a nonnegative matrix $\mathbf{V}^* \in \mathbb{R}_+^{n_1 \times n_2}$. In the electricity application, $\mathbf{V}^*$ is the electricity consumption of n_2 individuals through n_1 consecutive periods.

The matrix $\mathbf{V}^*$ is partially observed through N linear measurements,

$$\mathbf{b} = \mathcal{A}(\mathbf{V}^*) \in \mathbb{R}^N, \tag{1}$$

where $\mathcal{A} : \mathbb{R}^{n_1 \times n_2} \to \mathbb{R}^N$ is a linear operator. Formally, $\mathcal{A}$ can be represented by $\mathbf{A}_1$, ..., $\mathbf{A}_N$, N design matrices of dimension $n_1 \times n_2$, and each linear measurement can be represented by

$$b_i = \mathrm{Tr}(\mathbf{V}^* \mathbf{A}_i^T) = \langle \mathbf{V}^*, \mathbf{A}_i \rangle. \tag{2}$$

The design matrices $\mathbf{A}_1$, ..., $\mathbf{A}_N$ are called *masks*.

We suppose that $\mathbf{V}^*$ stems from a generative low-rank nonnegative model, in the following sense:

1) As is generally supposed in NMF, the matrix $\mathbf{V}^*$ is of *nonnegative rank* k, with $k \ll n_1, n_2$. This means, k is the smallest number so that we can find two nonnegative matrices $\mathbf{F}_r \in \mathbb{R}_+^{n_1 \times k}$ and $\mathbf{F}_c \in \mathbb{R}_+^{n_2 \times k}$ satisfying
$$\mathbf{V}^* = \mathbf{F}_r \mathbf{F}_c^T.$$
Note that this implies that $\mathbf{V}^*$ is of rank at most k, and therefore is of low rank.
2) There are d_1 row features $\mathbf{X}_r \in \mathbb{R}^{n_1 \times d_1}$ and d_2 column features $\mathbf{X}_c \in \mathbb{R}^{n_2 \times d_2}$ connected to each row and column of $\mathbf{V}^*$. We note by $\mathbf{x}_r^i$ (or $\mathbf{x}_c^i$) the i-th row of $\mathbf{X}_r$ (or $\mathbf{X}_c$). There are two link functions $f_r : \mathbb{R}^{d_1} \to \mathbb{R}^k$ and $f_c : \mathbb{R}^{d_2} \to \mathbb{R}^k$, so that
$$\mathbf{F}_r = (f_r(\mathbf{X}_r))_+,$$
$$\mathbf{F}_c = (f_c(\mathbf{X}_c))_+,$$
where $f_r(\mathbf{X}_r) \in \mathbb{R}^{n_1 \times k}$ is the matrix obtained by stacking row vectors $f_r(\mathbf{x}_r^i)$, for $1 \leq i \leq n_1$ (*idem* for $f_c(\mathbf{X}_c) \in \mathbb{R}^{n_2 \times k}$), and $(\cdot)_+$ is the ramp function which corresponds to thresholding all entries at 0 in a matrix or a vector.

In this general setting, the features $\mathbf{X}_r$ and $\mathbf{X}_c$, the measurement operator $\mathcal{A}$, and the measurements $\mathbf{b}$ are observed. The objective is to estimate the true matrix $\mathbf{V}^*$ as well as the factor matrices $\mathbf{F}_r$ and $\mathbf{F}_c$, by estimating the link functions f_r and f_c.

Optimization problem

To obtain a solution to this estimation problem, we minimize the quadratic error of the matrix factorization. In Section 3, we will propose an algorithm for the following optimization problem:

$$\begin{aligned} \min_{\mathbf{V}, f_r \in F_r^k, f_c \in F_c^k} \quad & \|\mathbf{V} - (f_r(\mathbf{X}_r))_+ (f_c(\mathbf{X}_c))_+^T\|_F^2 \\ \text{s.t.} \quad & \mathcal{A}(\mathbf{V}) = \mathbf{b}, \quad \mathbf{V} \geq \mathbf{0}, \end{aligned} \tag{3}$$

where $F_r \subseteq (\mathbb{R})^{\mathbb{R}^{d_1}}$ and $F_c \subseteq (\mathbb{R})^{\mathbb{R}^{d_2}}$ are some functional spaces in which the row and column link functions are to be searched.

In the electricity consumption application, typical row features are the temperature and calendar variables such as the day of the week and the hour of the day. Typical column features are demographic variables of the individuals such as the age, occupation, and the type of housing. We allow the functional spaces to be quite general, as state-of-art studies in electricity modeling established that the relationship between these variables and electricity consumption is often non-linear (see [6] and references within a more detailed description).

Notice that without side information and with complete observations, Problem (3) becomes

$$\min_{\mathbf{F}_r, \mathbf{F}_c} \quad \|\mathbf{V} - (\mathbf{F}_r)_+ (\mathbf{F}_c)_+^T\|_F^2. \tag{4}$$

This is equivalent to the classical NMF problem,

$$\begin{aligned} \min_{\mathbf{F}_r, \mathbf{F}_c} \quad & \|\mathbf{V} - \mathbf{F}_r (\mathbf{F}_c)^T\|_F^2 \\ \text{s.t.} \quad & \mathbf{F}_r \geq \mathbf{0}, \quad \mathbf{F}_c \geq \mathbf{0}, \end{aligned} \tag{5}$$

in the sense that for any solution $(\mathbf{E}_r, \mathbf{E}_c)$ to (4), $((\mathbf{E}_r)_+, (\mathbf{E}_c)_+)$ is a solution to (5).

A more immediate generalization of (5) to include exogenous variables would be in the form

$$\begin{aligned} \min_{\mathbf{V}, f_r \in F_r^k, f_c \in F_c^k} \quad & \|\mathbf{V} - f_r(\mathbf{X}_r)(f_c(\mathbf{X}_c))^T\|_F^2 \\ \text{s.t.} \quad & \mathcal{A}(\mathbf{V}) = \mathbf{b}, \quad \mathbf{V} \geq \mathbf{0}, \\ & f_r(\mathbf{X}_r) \geq \mathbf{0}, \quad f_c(\mathbf{X}_c) \geq \mathbf{0}. \end{aligned} \tag{6}$$

Solving (6) would involve identifying the subset of F_r and F_c that only produce nonnegative value on the row and column features, which could be difficult.

By using (3), we actually shifted the search space F_r and F_c to $(F_r)_+$ and $(F_c)_+$ which consists of composing all functions of F_r and F_c with the ramp function (thresholding at 0). In a word, (3) is equivalent to

$$\begin{aligned} \min_{\mathbf{V}, f_r \in (F_r)_+^k, f_c \in (F_c)_+^k} \quad & \|\mathbf{V} - f_r(\mathbf{X}_r)(f_c(\mathbf{X}_c))^T\|_F^2 \\ \text{s.t.} \quad & \mathcal{A}(\mathbf{V}) = \mathbf{b}, \quad \mathbf{V} \geq \mathbf{0}. \end{aligned} \tag{7}$$

Problem (7) also helps us to reason on the identifiability of (3). In a sense, (3) is not well-identified: two distinct elements in $F_r^k \times F_c^k$ have the same evaluation value of the objective function, if they only differ on their negative parts. In fact, this does not affect the interpretation of the model, because these distinct elements correspond to the same element in $(F_r)_+^k \times (F_c)_+^k$. Since we are only going to use the positive parts of the function both in recovery and prediction, this becomes a parameterization choice which has no consequence on the applications.

We note that an alternative problem,

$$\min_{f_r \in F_r^k, f_c \in F_c^k} \quad \|\mathbf{b} - \mathcal{A}((f_r(\mathbf{X}_r))_+ (f_c(\mathbf{X}_c))_+^T)\|_2^2, \tag{8}$$

can be used to solve the same matrix recovery problem. Instead of using the data in a linear matrix equation, Problem (8) minimizes the sampling error of an exactly low-rank matrix. In the literature of matrix factorization without side information, both have been studied [8], [9]. We argue

that Problem (3) is superior to Problem (8) as is shown in Sections 3.7 and 4.4.

By specializing $\mathbf{X}_r$, $\mathbf{X}_c$, and $\mathcal{A}$, or restricting the search space of f_r and f_c, this general model includes a number of interesting applications, old and new.

The masks $\mathbf{A}_1$, ..., $\mathbf{A}_N$

- **Complete observation**: $N = n_1 n_2, \mathbf{A}_{i_1,i_2} = \mathbf{e}_{i_1}\mathbf{e}_{i_2}^T$, where $\mathbf{e}_i$ is the i-th canonical vector. This means every entry of $\mathbf{V}^*$ is observed.
- **Matrix completion** [10]: the set of masks is a subset of complete observation masks, with $N < n_1 n_2$.
- **Matrix sensing** [8]: the design matrices $\mathbf{A}_i$ are random matrices, sampled from a certain probability distribution. Typically, the probability distribution needs to verify certain conditions, so that with a large probability, $\mathcal{A}$ verifies the Restricted Isometry Property [8].
- **Rank-one projections** [11], [12]: the design matrices are random rank-one matrices, that is $\mathbf{A}_i = \mathbf{b}_i\boldsymbol{\beta}_i^T$, where $\mathbf{b}_i$ and $\boldsymbol{\beta}_i$ are respectively random vectors of dimension n_1 and n_2. The main advantage to this setting is that much less memory is needed to store the masks, since we can store the vectors $\mathbf{b}_i$ and $\boldsymbol{\beta}_i$ (dimension-$(n_1 + n_2)$) instead of $\mathbf{A}_i$ (dimension-$(n_1 \times n_2)$). In [11], [12], theoretical properties are proved for the case where $\mathbf{b}_i$ and $\boldsymbol{\beta}_i$ are vectors with independent Gaussian entries and/or drawn uniformly from the vectors of the canonical basis.
- **Temporal aggregate measurements**: in this case, the matrix is composed of n_1 time series concerning n_2 individuals, and each measure is a temporal aggregate of the time series of an individual. The design matrices are defined as $\mathbf{A}_i = \sum_{t=t_0(i)+1}^{t_0(i)+h(i)} \mathbf{e}_t\mathbf{e}_{s_i}^T$, where s_i is the individual concerned by the i-th measure, $t_0(i) + 1$ the first period covered by the measure, and $h(i)$ the number of periods covered by the measure.

The features $\mathbf{X}_r$ and $\mathbf{X}_c$

- **Individual features**: $\mathbf{X}_r = \mathbf{I}_{n_1}, \mathbf{X}_c = \mathbf{I}_{n_2}$. This is the case of NMF without side information. The row individuals and column individuals are each different.
- **General numeric features**: $\mathbf{X}_r \in \mathbb{R}^{n_1 \times d_1}$ and $\mathbf{X}_c \in \mathbb{R}^{n_2 \times d_2}$.
- **Features generated from a kernel** [2], [13], [14]: certain information about the row and column individuals may not be in the form of a numeric vector. For example, if the row individuals are vertices of a graph, their connection to each other is interesting information for the problem, but it is difficult to encode as real vectors. In this case, features can be generated through a transformation defined by a kernel function.

The link functions f_r and f_c

- **Linear**: $\mathbf{F}_r = f_r(\mathbf{X}_r) = \mathbf{X}_r\mathbf{B}_r$, and $\mathbf{F}_c = f_c(\mathbf{X}_c) = \mathbf{X}_c\mathbf{B}_c$. In this case, we need to estimate $\mathbf{B}_r$ and $\mathbf{B}_c$ to fit the model. With identity matrices as row and column features, this case is reduced to the traditional matrix factorization model with

$$\mathbf{F}_r = \mathbf{B}_r, \quad \mathbf{F}_c = \mathbf{B}_c, \quad \mathbf{V}^* = \mathbf{F}_r\mathbf{F}_c^T = \mathbf{B}_r\mathbf{B}_c^T.$$

  When the features are generated from a kernel function, even a linear link function permits non-linear relationship between the features and the factor matrices.
- **General regression models**: when the relationship between the features and the variable of interest is not linear, any off-the-shelf regression methods can be plugged in to search for a non-linear link function.

1.2 Prior works

Table 1 shows a taxonomy of matrix factorization models with side information, by the mask, the link function and the features used as side information.

There is an abundant literature that studies the general matrix factorization problem under various measurement operators, when no additional information is provided (see [8], [10], [12], [24] for various masks considered, and [25] for a recent global convergence result with RIP measurements). The NMF with general linear measurements is studied in various applications [5], [9], [26].

On the other hand, with complete observations, the multiple regression problem taking into account the low-rank structure of the (multi-dimensional) variable of interest is known as reduced-rank regression. This approach was first developed very early (see [15] for a review). Recent developments on rank selection [16], adaptive estimation procedures [27], using non-parametric link function [19], often draw the parallel between reduced-rank regression and the matrix completion problem. However, measurement operators other than complete observations or matrix completion are rarely considered in this community.

Building on theoretical boundaries on matrix completion, the authors of [1], [21], [22] showed that by providing side information (the matrix $\mathbf{X}$), the number of measurements needed for exact matrix completion can be reduced. Moreover, the number of measurements necessary for successful matrix completion can be quantified by measuring the quality of the side information [22].

Collaborative filtering with side information has received much attention from practitioners and academic researchers alike, for its huge impact in e-commerce and web applications [1], [2], [3], [20], [21], [22]. One of the first methods for including side information in collaborative filtering systems (matrix completion masks) was proposed by [14]. The authors generalized collaborative filtering into a operator estimation problem. This method allows more general feature spaces than a numerical matrix, by applying a kernel function to side information. [3] proposed choosing the kernel function based on the goal of the application. [18] applied the kernel-based collaborative filtering framework to electricity price forecasting. Their kernel choice is determined by multi-kernel learning methods.

To the best of our knowledge, matrix factorization (non-negative or not) with side information, from general linear measurements has rarely been considered, nor is general non-linear functions other than with features obtained from kernels. This article aims at proposing a general approach which fills this gap.

Table 1
Classification of matrix factorization with side information by the mask, the link function, and the features included as side information, with some problems previously addressed in the literature.

	Link function	Linear			Other regression methods
	Features	Identity	General numeric features	Kernel features	General numeric features
Mask	Identity	Matrix factorization	Reduced-rank regression [15], [16], [17]	Multiple kernel learning [18]	Nonparametric RRR [19]
	Matrix completion	Matrix completion [10]	Inductive matrix completion (IMC) [1], [20], [21], [22]	Graph-regularized matrix factorization (GRMF) [2], [3], [14]	
	Rank-one projections	[11], [12]			
	Temporal aggregates	[5]			
	General masks	Matrix recovery [8], [23]			

2 Identifiability of nonnegative matrix factorization with side information

Matrix factorization is not a well-identified problem: for one pair of factors $(\mathbf{F}_r, \mathbf{F}_c)$, with $\mathbf{V}^* = \mathbf{F}_r\mathbf{F}_c^T$, any invertible matrix $\mathbf{R}$ produces another pair of factors, since $(\mathbf{F}_r\mathbf{R})(\mathbf{F}_c(\mathbf{R}^{-1})^T)^T$ is also equal to $\mathbf{V}^*$. In order to address this identifiability problem, one has to introduce extra constraints on the factors.

When the nonnegativity constraint is imposed on $\mathbf{F}_r$ and $\mathbf{F}_c$, however, it has been shown that sometimes the only invertible matrices that verify $\mathbf{F}_r\mathbf{R} \geq 0$ and $\mathbf{R}^{-1}\mathbf{F}_c \geq 0$ are the composition of a permutation matrix and a diagonal matrix with strictly positive diagonal elements. A nonnegative matrix factorization is said to be "identified" if the factors are unique up to permutation and scaling. The identifiability conditions for NMF are a hard problem, because sufficient and necessary conditions are computationally difficult to check (see [28]). In this section, we develop a sufficient condition for NMF identifiability in the context of linear numerical features, extending a classical result from the literature [29].

By studying the identifiability of NMF with side information, we show that model identification is not impaired by introducing side information. Moreover, if we wish to find interpretation of the factors obtained in NMF, it is desirable if the factors are "unique".

In order to simplify our analysis, we focus on the complete observation case in this section (every entry in $\mathbf{V}^*$ is observed). Without loss of generality, we derive the sufficient condition for row features. That is, we will derive conditions on $\mathbf{V}^* \in \mathbb{R}_+^{n_1 \times n_2}$ and $\mathbf{X}_r \in \mathbb{R}^{n_1 \times d_1}$, so that the nonnegative matrix factorization $\mathbf{V}^* = \mathbf{X}_r\mathbf{B}_r\mathbf{F}_c^T$, with $\mathbf{X}_r\mathbf{B}_r \geq 0, \mathbf{F}_c \geq 0$, is unique. A generalization to column features can be easily obtained. In this section, we assume that in addition to be of nonnegative rank k, matrix $\mathbf{V}^*$ is also exactly of *rank* k.

The identifiability conditions of NMF is first studied by [4] and then further tightened by [29]. Here we extend [29, Theorem 5], by proposing a sufficient condition for the NMF problem with side information to have a unique solution.

We rely on the following two definitions.

Definition 1 (Separability, [4])**.** Suppose $m \leq n$. A nonnegative matrix $\mathbf{M} \in \mathbb{R}_+^{m \times n}$ is said to be *separable* if there is a m-by-m permutation matrix Π which verifies

$$\mathbf{M} = \Pi \begin{pmatrix} \mathbf{D}_n \\ \mathbf{M}_0 \end{pmatrix},$$

where $\mathbf{D}_n$ is a n-by-n diagonal matrix with only strictly positive coefficients on the diagonal and zeros everywhere else, and the $(m-n)$-by-n matrix $\mathbf{M}_0$ is a collection of the other $m-n$ rows of $\mathbf{M}$.

Definition 2 (Strongly Boundary Closeness, [29])**.** A nonnegative matrix $\mathbf{M} \in \mathbb{R}_+^{m \times n}$ is said to be *strongly boundary close* if the following conditions are satisfied.

1) $\mathbf{M}$ is *boundary close*: for all $i, j \in \{1, ..., n\}, i \neq j$, there is a row $\mathbf{m}$ in $\mathbf{M}$ which satisfies $m_i = 0, m_j > 0$;
2) There is a permutation of $\{1, ..., n\}$ such that for all $i \in \{1, ..., n-1\}$, there are $n-i$ rows $\mathbf{m}^1, ..., \mathbf{m}^{n-i}$ in $\mathbf{M}$ which satisfy
   a) $m_i^j = 0, \sum_{s=i+1}^n m_i^j > 0$ for all $j \in \{1, ..., n-i\}$;
   b) the square matrix $(m_s^j)_{1 \leq j \leq n-i, i+1 \leq s \leq n}$ is of full rank $(n-i)$.

Strongly boundary closeness demands, *modulo* a permutation in $\{1, ..., n\}$, that for each $1 \leq i \leq n-1$, there are $n-i$ rows $\mathbf{m}^1, ..., \mathbf{m}^{n-i}$ of $\mathbf{M}$ that have the following form,

$$\begin{pmatrix} \mathbf{m}^1 \\ \vdots \\ \mathbf{m}^{n-i} \end{pmatrix}^T = \begin{pmatrix} \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \\ m_{i+1}^1 & \cdots & m_{i+1}^{n-i} \\ \vdots & \ddots & \vdots \\ m_n^1 & \cdots & m_n^{n-i} \end{pmatrix} \begin{matrix} \\ (i-1) \text{ first rows} \\ i\text{-th row is all zero} \\ \\ \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} (n-i) \times (n-i) \text{ full rank matrix} \end{matrix} \tag{9}$$

These row vectors, $\mathbf{m}^1, ..., \mathbf{m}^{n-i}$, all have 0 on the i-th element, and its lower square matrix of is of full rank. There are therefore enough linearly independent points on each $n-1$-dimensional facet $\mathbb{R}_+^n$, which shows that cone$(\mathbf{M}^T)$ is "maximal" in $\mathbb{R}_+^n$.

The NMF with linear row features, $\mathbf{V}^* = \mathbf{X}_r\mathbf{B}_r\mathbf{F}_c^T$, is said to be *unique*, if for all matrix pairs $(\tilde{\mathbf{B}}_r, \tilde{\mathbf{F}}_c) \in \mathbb{R}^{d_1\times k} \times \mathbb{R}^{n_2\times k}$ that verifies

$$\mathbf{X}_r\tilde{\mathbf{B}}_r \geq 0, \quad \tilde{\mathbf{F}}_c \geq 0, \quad \mathbf{V}^* = \mathbf{X}_r\tilde{\mathbf{B}}_r\tilde{\mathbf{F}}_c,$$

we have $\tilde{\mathbf{B}} = \mathbf{B}_r$, $\tilde{\mathbf{F}}_c = \mathbf{F}_c$ up to permutation of columns and scaling.

For a given full-rank matrix $\mathbf{X} \in \mathbb{R}^{n_1\times d_1}$, consider the following two sets of matrices:

$$E = \{\mathbf{M} \in \mathbb{R}_+^{n_1\times k} | \text{The columns of } \mathbf{M} \text{ are strongly boundary close}\};$$
$$F(\mathbf{X}) = \{\mathbf{M} \in \mathbb{R}_+^{n_1\times k} | \text{rank}(\mathbf{M}) = k, \text{span}(\mathbf{M}) \in \text{span}(\mathbf{X})\}.$$

Theorem 1. *If $E \bigcap F(\mathbf{X}_r) \neq \varnothing$, and $\mathbf{B}_r \in (\mathbf{X}_r^T\mathbf{X}_r)^{-1}\mathbf{X}_r^T(E \bigcap F(\mathbf{X}_r))$, and $\mathbf{F}_c$ is separable, then the factorization $\mathbf{V}^* = \mathbf{X}_r\mathbf{B}_r\mathbf{F}_c^T$ is unique.*

Proof. Notice that for $\mathbf{B}_r \in (\mathbf{X}_r^T\mathbf{X}_r)^{-1}\mathbf{X}_r^T(E \bigcap F(\mathbf{X}_r))$, the nonnegative matrix $\mathbf{X}_r\mathbf{B}_r$ is strongly boundary close. The factorization $(\mathbf{X}_r\mathbf{B}_r, \mathbf{F}_c)$ is therefore unique. The model identifiability follows immediately ([29, Theorem 5]), since $\mathbf{X}_r$ is of full rank. □

Example of $\mathbf{X}_r$ that verifies $E \bigcap F(\mathbf{X}_r) \neq \varnothing$

For this theorem to have practical consequences, one needs to find appropriate row features so that $E \bigcap F(\mathbf{X}_r) \neq \varnothing$.

Here we provide a family of matrices $\mathbf{X}_r$ so that $E \bigcap F(\mathbf{X}_r) \neq \varnothing$.

With a fixed $k \geq 2$, suppose that $\mathbf{X}_r$ has $k(k-1)/2$ columns, and at least $k(k-1)/2$ rows, with the first $k(k-1)/2+1$ rows defined as the following:

- the first row and column have 0 on the first entry and positive entries elsewhere;
- for $2 \leq i \leq k$, $\mathbf{X}_r$ has strictly positive entries on the first $((i-1)(i-2)/2+1)$ columns, from Row $(i-1)(i-2)/2+3$ to Row $(i-1)(i-2)/2+1+i$, and zero entries everywhere else. These $(k-1)$ rows are linearly independent.

Then we have $E \bigcap F(\mathbf{X}_r) \neq \varnothing$, because the following $k(k-1)/2$-by-k matrix $\mathbf{B}_r$ is in this set:

- for $1 \leq i \leq k$, $\mathbf{B}_K^*$ has i consecutive strictly positive entries on the i-th column, between Row $i(i-1)/2+1$ and Row $i(i-1)/2+i$.

If $E \bigcap F(\mathbf{X}) \neq \varnothing$, for any invertible matrix $\mathbf{R} \in \mathbb{R}^{K\times K}$, $E \bigcap F(\mathbf{X}\mathbf{R}) \neq \varnothing$.

As a conclusion to this section, we notice that since the sufficient condition proposed by Theorem 1 is based on a known uniqueness condition of classical NMF, this does not make the problem more identifiable. Rather, we derived a sufficient condition on the feature matrix and the coefficient matrix ($\mathbf{X}_r$ and $\mathbf{B_r}$) for the NMF with side information to be unique. This generalization can be relied on as a criterion for the type of side information we introduce in NMF.

3 HALSX ALGORITHM

In this section, we propose HALSX, or *Hierarchical Alternating Least Squares with eXogeneous variables*, which estimates the nonnegative matrix factorization problem with side information, from linear measurement, by solving (3). It is an extension to a popular NMF algorithm: Hierarchical Alternating Least Squares (HALS) (see [30], [31]).

Before discussing HALSX, we will first present a result on the local convergence of Gauss-Seidel algorithms. This result guarantees that any legitimate limiting points generated by HALSX are stationary points of (3).

While presenting specific methods to estimate link functions, we will only discuss row features, as the generalization to column features is immediate.

3.1 Relaxation of convexity assumption for the convergence of Gauss-Seidel algorithm

To show that all legitimate limiting points of HALSX are stationary points, we first extend a classical result concerning Gauss-Seidel algorithm [32, Proposition 4], also called Block Coordinate Descent.

Consider the minimization problem,

$$\begin{aligned} \min \quad & g(x) \\ \text{s.t.} \quad & x \in X = X_1 \times X_2 \times ... \times X_m \subseteq \mathbb{R}^n, \end{aligned} \tag{10}$$

where g is a continuously differentiable real-valued function, and the feasible set X is the Cartesian product of closed, nonempty and convex subsets $X_i \subset \mathbb{R}^{n_i}$, for $1 \leq i \leq m$, with $\sum_i n_i = n$. Suppose that the global minimum is reached at a point in X. The m-block Gauss-Seidel algorithm is defined as Algorithm 1.

Algorithm 1 Gauss-Seidel algorithm

Initialize $x^0 \in X, t = 0$
while Stopping criterion is not satisfied **do**
  for $i = 1, 2, ..., m$ **do**
    Calculate $x_i^{t+1} = \arg\min_{y_i \in X_i} g(x_1^{t+1}, ..., y_i, ..., x_m^t)$
  end for
  Set $x^{t+1} = (x_1^{t+1}, ..., x_m^{t+1})$
  $t = t + 1$
end while

Define formally the notion of component-wise quasi-convexity.

Definition 3. Let $i \in \{1, 2, ..., m\}$. The function g is *quasi-convex* with respect to the i-th component on X if for every $x \in X$ and $y_i \in X_i$, we have

$$\begin{aligned} &g(x_1, x_2, ...tx_i + (1-t)y_i, ..., x_m) \\ \leq &\max\{g(x), g(x_1, x_2, ..., y_i, ...x_m)\} \end{aligned}$$

for all $t \in [0, 1]$. The function g is said to be *strictly quasi-convex* with respect to the i-th component, if with the additional assumption that $y_i \neq x_i$, the previous inequality holds strictly.

It has been shown that if g is strictly quasi-convex with respect to the first $m-2$ blocks of components on X, then a limiting point produced by a Gauss-Seidel algorithm is a critical point [32].

This result is not directly applicable for the HALS algorithm. Typically, if $\mathbf{f}_{c,i}$, the $i-th$ column of $\mathbf{F}_c$, is identically zero, the loss function is flat with respect to $\mathbf{f}_{r,i}$, the i-th column of $\mathbf{F}_r$. Therefore the loss function is not strictly quasi-convex. In order to avoid this scenario, [31] suggests

thresholding at a small positive number ϵ instead of at 0, when updating each column of the factor matrices.

In fact the convexity assumption of [32] can be slightly relaxed to directly apply to HALS, as demonstrated by the following theorem.

Theorem 2. *Suppose that the function g is quasi-convex with respect to x_i on X, for $i = 1, ..., m-2$. Suppose that some limit points $\bar{x}$ of the sequence $\{x^t\}_{(t\in\mathbb{N})}$ verify that g is strictly quasi-convex with respect to x_i on the product set $\{\bar{x}_1\} \times \{\bar{x}_2\} \times ... \times X_i \times, ... \times \{\bar{x}_m\}$, for $i = 1, ..., m-2$. Then every such limiting point is a critical point of Problem (10).*

Compared to the result of [32], this shows that the strict convexity with respect to one block does not have to hold universally for feasible regions of other blocks. It only needs to hold at the limiting point.

This theorem can be established following the proof of Proposition 5 of [32], using the following lemma instead of [32, Proposition 4].

Lemma 3. *Suppose that the function g is quasi-convex with respect to x_i on X, for some $i \in \{1, ..., m\}$. Suppose that some limit points $\bar{y}$ of $\{y^t\}$ verify that g is strictly quasi-convex with respect to x_i on $\{\bar{y}_1\} \times \{\bar{y}_2\} \times ... \times X_i \times, ... \times \{\bar{y}_m\}$. Let $\{v^t\}$ be a sequence of vectors defined as follows:*

$$v_j^t = \begin{cases} y_j^t & \text{if } j \neq i, \\ \arg\min_{z_i \in X_i} g(y_1^t, ..., z_i, ..., y_m^t) & \text{if } j = i. \end{cases}$$

Then, if $\lim_{t\to+\infty} g(y^t) - g(v^t) = 0$, we have $\lim_{t\to+\infty} ||v_i^t - y_i^t|| = 0$. That is $\lim_{t\to+\infty} ||v^t - y^t|| = 0$.

Proof. (The proof of the lemma is based on [33].)

Suppose on the contrary that $||v_i^t - y_i^t||$ does not converge to 0. Define $\tau_k = ||v_i^t - y_i^t||$. Restricting to a subsequence, we can obtain that $\tau_k \geq \tau_0 > 0$. Define $s^t = \frac{v^t - y^t}{\tau_k}$. Notice that $\{s^t\}$ is of unit norm, and $v^t = y^t + \tau_k s^t$. Since $\{s^t\}$ is on the unit sphere, it has a converging subsequence. By restricting to a subsequence again, we could suppose that $\{s^t\}$ converges to $\bar{s}$.

For all $\epsilon \in [0, 1]$, we have $0 \leq \epsilon\tau_0 \leq \tau_k$, which implies $y^t + \epsilon\tau_0 s^t \in X$ is on the segment $[y^t, v^t]$. This segment has strictly positive dimension in the subspace corresponding to X_i.

By the definition of $\{v^t\}$, $g(v^t) \leq g(y_1^t, ..., z_i, ..., y_m^t)$, for all t, and for all $z_i \in X_i$. In particular,

$$g(v^t) \leq g(y^t + \epsilon\tau_0 s^t).$$

By quasi-convexity of g on X,

$$g(y^t + \epsilon\tau_0 s^t) \leq \max\{g(y^t), g(v^t)\} = g(y^t).$$

Taking the limit when t converges to $+\infty$ on both equalities, we obtain

$$\begin{aligned} g(\bar{y}) = \lim_{t\to+\infty} g(v^t) &\leq \lim_{t\to+\infty} g(y^t + \epsilon\tau_0 s^t) \\ = g(\bar{y} + \epsilon\tau_0 \bar{s}) &\leq \lim_{t\to+\infty} g(y^t) = g(\bar{y}). \end{aligned}$$

In other words, $g(\bar{y} + \epsilon\tau_0\bar{s}) = g(\bar{y})$, $\forall \epsilon \in [0, 1]$, which contradicts the strict quasi-convexity of g on $\{\bar{y}_1\} \times \{\bar{y}_2\} \times ... \times X_i \times, ... \times \{\bar{y}_m\}$. □

3.2 HALSX and its local convergence

To solve (3), we propose HALSX (Algorithm 2). When complete observations are available, the feature matrices are identity matrices, and when only linear functions are allowed as link functions, Algorithm 2 is equivalent to HALS [30].

Algorithm 2 Hierarchical Alternating Least Squares with eXogeneous variables for NMF (HALSX)

Require: Measurement operator $\mathcal{A}$, measurements $\mathbf{b}$, features $\mathbf{X}_r$ and $\mathbf{X}_c$, functional spaces F_r and F_c in which to search the link functions, and $1 \leq k \leq \min\{n_1, n_2\}$.
Initialize $\mathbf{F}_r^0, \mathbf{F}_c^0 \geq 0, t = 0$
while Stopping criterion is not satisfied **do**
  $\mathbf{V}^t = \arg\min_{\mathbf{V}|\mathcal{A}(\mathbf{V})=\mathbf{b}, \mathbf{V}\geq 0} \|\mathbf{V} - \mathbf{F}_r^t(\mathbf{F}_c^t)^T\|_F^2$
  $\mathbf{R}^t = \mathbf{V}^t - \mathbf{F}_r^t(\mathbf{F}_c^t)^T$
  for $i = 1, 2, ..., k$ **do**
    $\mathbf{R}_i^t = \mathbf{R}^t + \mathbf{f}_{r,i}^t(\mathbf{f}_{c,i}^t)^T$
    Calculate $f_{r,i}^{t+1} = \arg\min_{f\in F_r} \|\mathbf{R}_i^t - f(\mathbf{X}_r)(\mathbf{f}_{c,i}^t)^T\|_F^2$
    $\mathbf{f}_{r,i}^{t+1} = \max(0, f_{r,i}^{t+1}(\mathbf{X}_r))$
    $\mathbf{R}^t = \mathbf{R}_i^t - \mathbf{f}_{r,i}^{t+1}(\mathbf{f}_{c,i}^t)^T$
  end for
  for $i = 1, 2, ..., k$ **do**
    $\mathbf{R}_i^t = \mathbf{R}^t + \mathbf{f}_{r,i}^{t+1}(\mathbf{f}_{c,i}^t)^T$
    Calculate $f_{c,i}^{t+1} = \arg\min_{f\in F_c} \|\mathbf{R}_i^t - \mathbf{f}_{r,i}^{t+1} f(\mathbf{X}_c)^T\|_F^2$
    $\mathbf{f}_{c,i}^{t+1} = \max(0, f_{c,i}^{t+1}(\mathbf{X}_c))$
    $\mathbf{R}^t = \mathbf{R}_i^t - \mathbf{f}_{r,i}^{t+1}(\mathbf{f}_{c,i}^{t+1})^T$
  end for
  $t = t + 1$
end while
return $\mathbf{V}^t = \arg\min_{\mathbf{V}|\mathcal{A}(\mathbf{V})=\mathbf{b}, \mathbf{V}\geq 0} \|\mathbf{V} - \mathbf{F}_r^t(\mathbf{F}_c^t)^T\|_F^2$,
$\mathbf{F}_r^t \in \mathbb{R}_+^{n_1\times k}$, $f_{r,1}^t, ..., f_{r,k}^t \in F_r$,
$\mathbf{F}_c^t \in \mathbb{R}_+^{n_2\times k}$, $f_{c,1}^t, ..., f_{c,k}^t \in F_c$.

In this algorithm, at each elementary update step, we first look for a link function which minimizes the quadratic error, without concerning ourselves with its nonnegativity (lines 7 and 13). The obtained evaluation of the minimizer function is then thresholded at 0 to update the factors (lines 8 and 14).

To obtain that the limiting points of HALSX are stationary points, we need to ensure that for some functional spaces F_r and F_c, such an update solves a corresponding subproblem of (3). To do this, we will use the following proposition:

Proposition 4. *Suppose that $\mathbf{R} \in \mathbb{R}^{n_1\times n_2}$, $\mathbf{f}_c \in \mathbb{R}_+^{n_2}$ are not identically equal to zero, and $g : \mathbb{R}^d \to \mathbb{R}^{n_1}$, with $d \geq n_1$, is a convex differentiable function. Suppose*

$$\boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \|\mathbf{R} - g(\boldsymbol{\theta})(\mathbf{f}_c)^T\|_F^2.$$

If $\nabla g_{\boldsymbol{\theta}^*}$, the Jacobian matrix of g at $\boldsymbol{\theta}^*$, is of rank n_1, then $\boldsymbol{\theta}^*$ is also a solution to

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \|\mathbf{R} - (g(\boldsymbol{\theta}))_+(\mathbf{f}_c)^T\|_F^2. \tag{11}$$

Proof. Take $\mathbf{R} \in \mathbb{R}^{n_1\times n_2}$, $\mathbf{f}_c \in \mathbb{R}_+^{n_2}$ not identically equal to zero. We will note by L the loss function, so that $L(\mathbf{f}) = \|\mathbf{R} - (\mathbf{f})_+(\mathbf{f}_c)^T\|_F^2$ for all $\mathbf{f} \in \mathbb{R}^{n_1}$. The function L is convex.

Problem (11), which can be rewritten as

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} L(g(\boldsymbol{\theta})),$$

is also convex. The subgradient of the composition function $L \circ g$ at $\boldsymbol{\theta} \in \mathbb{R}^d$ is simply obtained by multiplying $\nabla g_{\boldsymbol{\theta}}$, the Jacobian matrix of g at $\boldsymbol{\theta}$, to each element of $\partial L_{g(\boldsymbol{\theta})}$, or $\partial L_{g(\boldsymbol{\theta})} \equiv \nabla g_{\boldsymbol{\theta}} \partial L_{g(\boldsymbol{\theta})} = \{\nabla g_{\boldsymbol{\theta}} \mathbf{y} | \mathbf{y} \in \partial L_{g(\boldsymbol{\theta})}\}$. Therefore $\forall \boldsymbol{\theta} \in \mathbb{R}^d$, $\boldsymbol{\theta}$ is a minimizer of (11), if and only if $\mathbf{0} \in \nabla g_{\boldsymbol{\theta}} \partial L_{g(\boldsymbol{\theta})}$.

Since

$$\boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \|\mathbf{R} - g(\boldsymbol{\theta})(\mathbf{f}_c)^T\|_F^2,$$

is a minimizer of a smooth convex problem,

$$\frac{\partial}{\partial \boldsymbol{\theta}} \|\mathbf{R} - g(\boldsymbol{\theta})(\mathbf{f}_c)^T\|_F^2(\boldsymbol{\theta}^*) = \nabla g_{\boldsymbol{\theta}^*}(\mathbf{R} - g(\boldsymbol{\theta}^*)(\mathbf{f}_c)^T)\mathbf{f}_c = \mathbf{0}.$$

This means $(\mathbf{R} - g(\boldsymbol{\theta}^*)(\mathbf{f}_c)^T)\mathbf{f}_c = \mathbf{0}$, because $\nabla g_{\boldsymbol{\theta}^*}$ is of full rank. Consequently

$$g(\boldsymbol{\theta}^*) = \frac{1}{\|\mathbf{f}_c\|_2^2}\mathbf{R}\mathbf{f}_c.$$

It has been shown in NMF literature (for example [31, Theorem 2]) that,

$$(g(\boldsymbol{\theta}^*))_+ = \arg\min_{\mathbf{f}\in\mathbb{R}_+^{n_1}} \|\mathbf{R} - \mathbf{f}(\mathbf{f}_c)^T\|_F^2.$$

This is equivalent to

$$g(\boldsymbol{\theta}^*) \in \arg\min_{\mathbf{f}\in\mathbb{R}^{n_1}} L(\mathbf{f}),$$

or

$$\mathbf{0} \in \partial L_{g(\boldsymbol{\theta}^*)}.$$

We therefore conclude with $\mathbf{0} \in \nabla g_{\boldsymbol{\theta}^*} \partial L_{g(\boldsymbol{\theta}^*)}$. □

In many regression methods, even when a non-linear transformation is applied to the data, the regression function is linear in its parameters. A non-exhaustive list of methods include linear regression ($g(\boldsymbol{\theta}) = \mathbf{X}_r\boldsymbol{\theta}$), spline regression ($g(\boldsymbol{\theta}) = \phi(\mathbf{X}_r)\boldsymbol{\theta}$), or support vector regression (SVR) ($g(\boldsymbol{\theta}) = K(\mathbf{X}_r, \mathbf{X}_r)\boldsymbol{\theta}$). In this case, g has a constant Jacobian matrix. In the case of linear and spline regression, the Jacobian matrix is of rank n_1 if there are no less features than examples. For SVR, this is true for any positive definite kernels. This allows us to apply the previous lemma to each column update step of Algorithm 2. By calculating $f_{r,i}^{t+1} = \arg\min_{f\in F_r} \|\mathbf{R}^t - f(\mathbf{X}_r)(\mathbf{f}_{c,i}^t)^T\|_F^2$ at Step t for Column i in $\mathbf{F}_r$, we actually have

$$f_{r,i}^{t+1} = \arg\min_{f\in F_r} \|\mathbf{R}^t - (f(\mathbf{X}_r))_+(\mathbf{f}_{c,i}^t)^T\|_F^2.$$

This shows that at each iteration, we solve the subproblems of (3).

In these cases, by rewriting the functional space F_r and F_c in a parametric form, the search space is actually $\mathbb{R}^{r_1}$ and $\mathbb{R}^{r_2}$, for some r_1 and r_2.

Proposition 5. *If $n_1 \le r_1$, $n_2 \le r_2$, every full-rank factorization produced by HALSX (Algorithm 2) is a critical point of Problem (3).*

3.3 Designs and HALSX

At each iteration of Algorithm 2, we need to project the working matrix $\mathbf{F}_r^t(\mathbf{F}_c^t)^T$ into the convex polytope defined by the measurements and nonnegativity:

$$\mathbf{V}^t = \arg\min_{\mathbf{V}|\mathcal{A}(\mathbf{V})=\mathbf{b},\mathbf{V}\ge 0} \|\mathbf{V} - \mathbf{F}_r^t(\mathbf{F}_c^t)^T\|_F^2. \tag{12}$$

In general, the polytope projection can be obtained by alternating projection. Namely, we can alternate between:

- $\mathbf{V} = \mathbf{V} + \mathcal{A}^\dagger(\mathbf{b} - \mathcal{A}(\mathbf{V}))$;
- $v_{i,j} = \max(0, v_{i,j})$,

where $\mathcal{A}^\dagger$ is the right pseudo-inverse of $\mathcal{A}$, viewed as an N-by-n_1n_2 matrix.

For some measurement operators, there are efficient ways to solve (12):

- Matrix completion mask:
  $v_{i,j} = \begin{cases} b_l, & \text{if } \exists 1 \le l \le N, \mathbf{A}_l = \mathbf{e}_i\mathbf{e}_j^T; \\ \max(0, v_{i,j}), & \text{if not.} \end{cases}$
- Temporal aggregate mask: simplex projection (see [5] for details).

3.4 HALSX with linear link functions

In this section, we consider HALSX with numeric row features and linear row link functions. That is, given $\mathbf{X}_r$ and $\mathbf{b} = \mathcal{A}(\mathbf{V}^*)$, we need to solve

$$\begin{aligned} \min_{\mathbf{V}\in\mathbb{R}^{n_1\times n_2}, \mathbf{B}_r\in\mathbb{R}^{d_1\times k}, \mathbf{F}_c\in\mathbb{R}^{n_2\times k}} \quad & \|\mathbf{V} - (\mathbf{X}_r\mathbf{B}_r)_+(\mathbf{F}_c)_+^T\|_F^2 \\ \text{s.t.} \quad & \mathcal{A}(\mathbf{V}) = \mathbf{b}, \quad \mathbf{V} \ge \mathbf{0}, \end{aligned} \tag{13}$$

Following Algorithm 2, we need to update the columns of $\mathbf{B}_r$ at each iteration. At the t-th step, for $1 \le i \le k$, we solve the subproblem

$$\arg\min_{\mathbf{b}_{r,i}} \|\mathbf{R}_i^t - \mathbf{X}_r\mathbf{b}_{r,i}(\mathbf{f}_{c,i}^t)^T\|_F^2,$$

where $\mathbf{R}_i^t = \mathbf{V}^t - \sum_{j=1,j\ne i}^k \mathbf{X}_r\mathbf{b}_{r,l}\mathbf{f}_{c,l}^T$. This minimization problem has a closed-form solution:

$$\mathbf{b}_{r,i}^{t+1} = \frac{1}{\|\mathbf{f}_{c,i}^t\|_2^2}(\mathbf{X}_r^T\mathbf{X}_r)^{-1}\mathbf{X}_r^T\mathbf{R}_i^t\mathbf{f}_{c,i}^t.$$

In order to accelerate the numerical algorithm, a QR decomposition of $\mathbf{X}_r = \mathbf{Q}\mathbf{R}$ is done before the iterations, where $\mathbf{Q}$ is an orthogonal matrix, and $\mathbf{R}$ is a square upper triangular matrix. When $\mathbf{X}_r$ is of full rank, $\mathbf{X}_r^T\mathbf{X}_r$ is invertible. We compute one time $(\mathbf{X}_r^T\mathbf{X}_r)^{-1}\mathbf{X}_r = \mathbf{R}^{-1}\mathbf{Q}^T$ before the iterations, and use the result at each iteration.

Stopping criterion

As in classical NMF algorithms, we will use the Karush–Kuhn–Tucker conditions (KKT) to provide a stopping criterion. The KKT conditions of (13) are,

$$\begin{aligned} &\mathbf{V} \ge \mathbf{0}, \quad \mathbf{V} - (\mathbf{X}_r\mathbf{B}_r)_+(\mathbf{F}_c)_+^T \ge \mathbf{0}, \quad \mathcal{A}(\mathbf{V}) = \mathbf{b}, \\ &\mathbf{V} - (\mathbf{X}_r\mathbf{B}_r)_+(\mathbf{F}_c)_+^T \circ \mathbf{V} = \mathbf{0}, \\ &\nabla_{\mathbf{F}_c}\|\mathbf{V} - (\mathbf{X}_r\mathbf{B}_r)_+(\mathbf{F}_c)_+^T\|_F^2 \ni \mathbf{0}, \\ &\nabla_{\mathbf{B}_r}\|\mathbf{V} - (\mathbf{X}_r\mathbf{B}_r)_+(\mathbf{F}_c)_+^T\|_F^2 \ni \mathbf{0}, \end{aligned}$$

where $\mathbf{A} \circ \mathbf{B}$ is the entry-wise product (Hadamard product) for $\mathbf{A}, \mathbf{B}$ of the same dimension, and $\nabla_x f(x_0)$ is the subgradient of the function f at point x_0, with respect to

the variable x. Note that $\mathbf{V} \geq \mathbf{0}$ and $\mathcal{A}(\mathbf{V}) = \mathbf{b}$ are always satisfied at the end of an iteration.

Similar to classical NMF algorithms, we will stop the algorithm when the norm of the subgradient is smaller than ϵ times its initial value, with a small ϵ. By calculating the subgradient of the minimization function with respect to $\mathbf{F}_c$ and $\mathbf{B}_r$, the norm of following vector is used as stopping criterion:

$$\begin{aligned}&[vect((\mathbf{V} - (\mathbf{X}_r\mathbf{B}_r)_+(\mathbf{F}_c)_+^T)_-)^T,\\ &\ vect(\mathbf{V} - (\mathbf{X}_r\mathbf{B}_r)_+(\mathbf{F}_c)_+^T \circ \mathbf{V})^T,\\ &\ vect((\mathbf{X}_r\mathbf{B}_r)_+^T(\mathbf{V} - (\mathbf{X}_r\mathbf{B}_r)_+(\mathbf{F}_c)_+^T) \circ \mathbb{1}_{(\mathbf{F}_c)>\mathbf{0}})^T,\\ &\ vect(\mathbf{X}_r^T((\mathbf{V} - (\mathbf{X}_r\mathbf{B}_r)_+(\mathbf{F}_c)_+^T) \circ \mathbb{1}_{\mathbf{X}_r\mathbf{B}_r>\mathbf{0}}))^T].\end{aligned}$$

For the algorithms presented in the next sections, this stopping criterion is generalized quite easily.

3.5 HALSX with smoothing splines

The computation considered above can estimate an NMF with linear features fairly efficiently. However, in real applications, linear link functions are too restrictive. In the following, we will estimate non-linear link functions that are Generalized Additive Models (GAM, [34]).

A Generalized Additive Model is a generalization to Generalized Linear Model (GLM) which includes additive non-linear components. Consider n observations $\mathbf{x}_i, y_i$, for $1 \leq i \leq n$, where $\mathbf{x}_i$ is the vector of features, and y_i is an observation of a random variable Y_i. Suppose that $Y_i = \mu_i + \epsilon_i$, where ϵ_i are independent identically distributed zero-mean Gaussian variables, and $\mu_i = \mathbf{E}(Y_i)$ has the following relationship to the features:

$$g(\mu_i) = \mathbf{x}_i^T\boldsymbol{\theta} + h_1(x_{i,1}) + h_2(x_{i,2}) + h_3(x_{i,3}, x_{i,4}) + ...$$

where $\boldsymbol{\theta}$ is the vector of parametric model components, g is a known, monotonic, twice-differentiable function, h_1, h_2, h_3, ..., are the non-linear functions to be estimated.

We note by $\mathbf{X}$ the matrix grouping the features of all observations. We use penalized regression spline to fit the GAMs. For $j = 1, 2, 3, ...$, define a spline basis $\mathbf{a}^j = (a_1^j, a_2^j, ...)$ in which h_j, the j-th component of the GAM, is to be estimated. Practically, we search for h_j in the L-dimensional vector space

$$H(\mathbf{a}^j, L_j) = \{\sum_{l=1}^{L_j} \beta_l^j a_l^j | \boldsymbol{\beta}^j = (\beta_1^j, ...\beta_{L_j}^j) \in \mathbb{R}^{L_j}\}.$$

Noting by $\mathbf{X}^j = \{a_l^j(\mathbf{x}_i)\}_{i,l}$ the design matrix, for $h_j = \sum_{l=1}^{L_j} \beta_l^j a_l^j \in H(\mathbf{a}^j, L_j)$, an element of the functional space, we have

$$h_j(\mathbf{X}) = \mathbf{X}^j\boldsymbol{\beta}^j.$$

The whole model of g, can then be represented linearly:

$$\begin{aligned}g(\boldsymbol{\mu}) &= \mathbf{X}\theta + (\mathbf{X}^1, \mathbf{X}^2, ...)\begin{pmatrix}\boldsymbol{\beta}^1\\ \boldsymbol{\beta}^2\\ :\end{pmatrix}\\ &= \mathbf{X}\boldsymbol{\theta} + \sum_j \mathbf{X}^j\boldsymbol{\beta}^j.\end{aligned}$$

The dimension of $H(\mathbf{a}^j, L_j)$, L_j, controls the smoothness of the functions to be estimated. As little information is available on the degree of smoothness of the functions, we use a rather large L_j, and add a penalty on the wiggliness, $\int (h_j'')^2 dx$, as in [34]. The least squares estimator of this model is therefore

$$\arg\min_{\boldsymbol{\theta},\boldsymbol{\beta}^1,\boldsymbol{\beta}^2,...} \|g(\boldsymbol{\mu}) - \mathbf{X}\boldsymbol{\theta} - \sum_j \mathbf{X}^j\boldsymbol{\beta}^j\|^2 + \sum_j \lambda^j(\boldsymbol{\beta}^j)^T\mathbf{S}^j\boldsymbol{\beta}^j,$$

where λ_j is the penalization parameter of the j-th non-linear component, and $\mathbf{S}^j$ is a positive definite matrix depending on $\mathbf{X}$ and $\mathbf{a}^j$. The penalization parameter, λ^j, is chosen by a generalized cross validation criterion.

HALSX-GAM

At each iteration of the algorithm, for $i = 1, ..., k$, we re-estimate the link function $f_{r,i}$ of the i-th column of $\mathbf{F}_r$ as a GAM.

The subproblem for i is the following

$$\begin{aligned}\arg\min_{\boldsymbol{\theta},\boldsymbol{\beta}^1,\boldsymbol{\beta}^2,...} &\|\mathbf{R}_i^t - (\mathbf{X}_r\boldsymbol{\theta} + \sum_{j=1} \mathbf{X}^j\boldsymbol{\beta}^j)(\mathbf{f}_{c,i}^t)^T\|_F^2 +\\ &\sum_j \lambda^j(\boldsymbol{\beta}^j)^T\mathbf{S}^j\boldsymbol{\beta}^j.\end{aligned} \tag{14}$$

With fixed penalization parameters λ_1, λ_2, ..., the optimization above can be solved by

$$\begin{pmatrix}\boldsymbol{\theta}\\ \boldsymbol{\beta^1}\\ \boldsymbol{\beta^2}\\ :\end{pmatrix}^{t+1} = \frac{1}{\|f_{c,i}^t\|^2} \times$$
$$\begin{pmatrix}\mathbf{X}_r^T\mathbf{X}_r & \mathbf{X}_r^T\mathbf{X}^1 & \mathbf{X}_r^T\mathbf{X}^2 & \cdots\\ (\mathbf{X}^1)^T\mathbf{X}_r & (\mathbf{X}^1)^T\mathbf{X}^1 + \frac{\lambda^1}{\|f_{c,i}^t\|^2}\mathbf{S}^j & (\mathbf{X}^1)^T\mathbf{X}^2 & \cdots\\ : & : & \ddots & \cdots\end{pmatrix}^{-1} \times$$
$$\begin{pmatrix}\mathbf{X}_r^T\\ (\mathbf{X}^1)^T\\ (\mathbf{X}^2)^T\\ :\end{pmatrix}\mathbf{R}_i^t\mathbf{f}_{c,i}^t.$$

In practice, we use the GAM estimation routines implemented in the *R* package *mgcv* [34] to choose the penalization parameter and estimate the model at the same time.

3.6 HALSX with other regression models

We can replicate the strategy above to work with other regression models. As with HALSX-GAM, the local convergence property is reserved, as long as the regression model estimation can be re-parameterized to verify the conditions of Proposition 4. Using this strategy, many off-the-shelf algorithms for regression model training can be plugged in. In the experiments described in the next section, we use the predictive model API provided in the *R* package *caret* [35].

When there are meta-parameters involved, we estimate them using cross validation as a part of the link function estimation step.

3.7 HALSX2 for an alternative problem

In the introduction, we formulated an alternative optimization problem (8) besides (3). In this section, we develop HALSX2 (Algorithm 3), an algorithm similar to HALSX, that solves (8), and compares its theoretical complexity to HALSX.

Before detailing Algorithm 3, we will first develop the elemental HALS iteration in the context of (8) where no supplemental information is supplied, namely $\mathbf{X}_r = \mathbf{I}_{n_1}, \mathbf{X}_c = \mathbf{I}_{n_2}$. Indeed, when updating one column of $\mathbf{F}_r$, the subproblem becomes: how to solve $\arg\min_{\mathbf{f}} \|\mathbf{b} - \mathcal{A}(\mathbf{f}(\mathbf{f}_c)^T)\|_2^2$?

We will use the fact that for all $\mathbf{M} \in \mathbb{R}^{n_1 \times n_2}$,

$$\mathcal{A}(\mathbf{M}) = (\langle \mathbf{A}_i, \mathbf{M} \rangle)_{1 \leq i \leq N},$$

and for all $\mathbf{b} \in \mathbb{R}^N$, $\mathcal{A}^*$, the transpose of $\mathcal{A}$ is defined by

$$\mathcal{A}^*(\mathbf{b}) = \sum_{i=1}^{N} b_i \mathbf{A}_i.$$

Since

$$\frac{\partial}{\partial \mathbf{f}} \|\mathbf{b} - \mathcal{A}(\mathbf{f}(\mathbf{f}_c)^T)\|_2^2 = \mathcal{A}^*[\mathcal{A}(\mathbf{f}(\mathbf{f}_c)^T) - \mathbf{b}]\mathbf{f}_c,$$

the first order optimality condition $\frac{\partial}{\partial \mathbf{f}} \|\mathbf{b} - \mathcal{A}(\mathbf{f}(\mathbf{f}_c)^T)\|_2^2 0$ is therefore equivalent to

$$\mathcal{A}^*[\mathcal{A}(\mathbf{f}(\mathbf{f}_c)^T)]\mathbf{f}_c = \mathcal{A}^*[\mathbf{b}]\mathbf{f}_c.$$

The left-hand side of the equation can be written as

$$\begin{aligned}
\mathcal{A}^*[\mathcal{A}(\mathbf{f}(\mathbf{f}_c)^T)]\mathbf{f}_c &= (\sum_{i=1}^{N} \langle \mathbf{A}_i, \mathbf{f}(\mathbf{f}_c)^T \rangle \mathbf{A}_i)\mathbf{f}_c \\
&= \sum_{i=1}^{N} \mathrm{Tr}(\mathbf{f}(\mathbf{A}_i\mathbf{f}_c)^T)(\mathbf{A}_i\mathbf{f}_c) \\
&= \sum_{i=1}^{N} (\mathbf{A}_i\mathbf{f}_c)\mathrm{Tr}((\mathbf{A}_i\mathbf{f}_c)^T\mathbf{f}) \\
&= \sum_{i=1}^{N} (\mathbf{A}_i\mathbf{f}_c)(\mathbf{A}_i\mathbf{f}_c)^T\mathbf{f},
\end{aligned}$$

which leads to the following symmetric n_1-by-n_1 system on $\mathbf{f}$:

$$(\sum_{i=1}^{N} (\mathbf{A}_i\mathbf{f}_c)(\mathbf{A}_i\mathbf{f}_c)^T)\mathbf{f} = \sum_{i=1}^{N} b_i \mathbf{A}_i\mathbf{f}_c,$$

or

$$\mathbf{f} = (\sum_{i=1}^{N} (\mathbf{A}_i\mathbf{f}_c)(\mathbf{A}_i\mathbf{f}_c)^T)^{-1} \sum_{i=1}^{N} b_i \mathbf{A}_i\mathbf{f}_c$$

This computation generalizes to linear exogenous variables, with the optimality condition:

$$\boldsymbol{\beta} = (\sum_{i=1}^{N} (\mathbf{X}\mathbf{A}_i\mathbf{f}_c)(\mathbf{X}\mathbf{A}_i\mathbf{f}_c)^T)^{-1} \sum_{i=1}^{N} b_i \mathbf{A}_i\mathbf{f}_c.$$

When the matrices to be inversed in these equations are not invertible, we will use the generalized inverse instead.

Using these elementary steps, we propose Algorithm 3 to solve Problem (8). Compared to Algorithm 2, in Algorithm 3

- no update is need for $\mathbf{V}$;
- checks of the deviation with data are for frequent;
- each subproblem is more costly because of the presence of $\mathcal{A}$ in the subproblem. When the sample size (the dimension of image of $\mathcal{A}$) is large, each update involves rather costly computations.

Algorithm 3 Hierarchical Alternating Least Squares with eXogeneous variables for NMF (HALSX2)

Require: Measurement operator $\mathcal{A}$, measurements $\mathbf{b}$, rank $1 \leq k \leq \min\{n_1, n_2\}$, features $\mathbf{X}_r$ and $\mathbf{X}_c$, functional spaces F_r and F_c in which to search the link functions.
Initialize $\mathbf{F}_r^0, \mathbf{F}_c^0 \geq 0, t = 0$
while Stopping criterion is not satisfied **do**
  $\mathbf{R}^t = \mathbf{b} - \mathcal{A}(\mathbf{F}_r^t(\mathbf{F}_c^t)^T)$
  for $i = 1, 2, ..., k$ **do**
    $\mathbf{R}^t = \mathbf{R}^t + \mathcal{A}(\mathbf{f}_{r,i}^t(\mathbf{f}_{c,i}^t)^T)$
    $f_{r,i}^{t+1} = \arg\min_{f \in F_r} \|\mathbf{R}^t - \mathcal{A}(f(\mathbf{X}_r)(\mathbf{f}_{c,i}^t)^T)\|_2^2$
    $\mathbf{f}_{r,i}^{t+1} = \max(0, f_{r,i}^{t+1}(\mathbf{X}_r))$
    $\mathbf{R}^t = \mathbf{R}^t - \mathcal{A}(\mathbf{f}_{r,i}^{t+1}(\mathbf{f}_{c,i}^t)^T)$
  end for
  for $i = 1, 2, ..., k$ **do**
    $\mathbf{R}^t = \mathbf{R}^t + \mathcal{A}(\mathbf{f}_{r,i}^{t+1}(\mathbf{f}_{c,i}^t)^T)$
    $f_{c,i}^{t+1} = \arg\min_{f \in F_c} \|\mathbf{R}^t - \mathcal{A}(\mathbf{f}_{r,i}^{t+1} f(\mathbf{X}_c)^T)\|_2^2$
    $\mathbf{f}_{c,i}^{t+1} = \max(0, f_{c,i}^{t+1}(\mathbf{X}_c))$
    $\mathbf{R}^t = \mathbf{R}^t - \mathcal{A}(\mathbf{f}_{r,i}^{t+1}(\mathbf{f}_{c,i}^{t+1})^T)$
  end for
  $t = t + 1$
end while
return $\mathbf{F}_r^t \in \mathbb{R}_+^{n_1 \times k}, f_{r,1}^t, ..., f_{r,k}^t \in F_r$, $\mathbf{F}_c^t \in \mathbb{R}_+^{n_2 \times k}, f_{c,1}^t, ..., f_{c,k}^t \in F_c$.

Complexity comparison between HALSX and HALSX2

At each sub-iteration of Algorithm 3, we need to calculate N n_1-by-n_1 or n_2-by-n_2 matrices $((\mathbf{A}_i\mathbf{f}_c)(\mathbf{A}_i\mathbf{f}_c)^T)$, then inverse the sum of the these N matrices. While the computation of the sum is map-reducible, on a single-threaded machine, this can be computationally expensive when N is large. Each iteration of Algorithm 3 has a multiplicative complexity of $O(kN(n_1^2 + n_2^2))$, while each iteration of Algorithm 2 has a complexity of $O(k(n_1^2 + n_2^2) + Nn_1n_2)$ with general linear measurement operator. This difference means that Algorithm 3 can be much slower when N or k is large.

4 Experiments

4.1 Datasets

We use one Synthetic dataset and three real datasets to evaluate the proposed methods.

- **Synthetic data**[1]: a rank-20 150-by-180 nonnegative matrix simulated following the generative model (Section 1.1), with $\mathbf{X}_r \in \mathbb{R}^{150 \times 3}$ and $\mathbf{X}_c \in \mathbb{R}^{180 \times 4}$ matrices with independent Gaussian entries, $f_r : \mathbb{R}^3 \to \mathbb{R}^{20}$ ($f_c : \mathbb{R}^4 \to \mathbb{R}^{20}$) is a function formed with a dimension-33 (44 for f_c) spline basis with random weights, truncated at 0 ($T = 150, N = 180$). The simulated features matrices are used as side information.

1. This dataset and the code to generate a similar one can be downloaded at https://www.dropbox.com/s/ftq7602qx1x406v/simulation_data.zip?dl=0

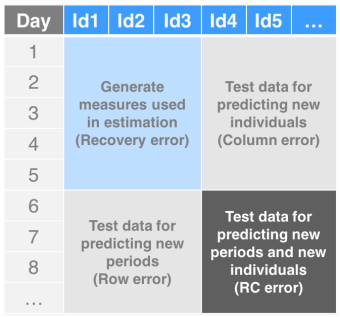

Figure 1. Dividing the data matrix into four sub-matrices: one for generating observations to estimate the model, the other three for prediction.

- **French electricity consumption** (proprietary dataset): daily consumption of 473 medium-voltage feeders gathering each around 1,500 consumers based near Lyon in France from 2010 to 2012. The first two years are used as training data ($T = 1096, N = 473$). The daily temperature of a weather station in this area, calendar variables such as weekday/weekend, position of the year, bank holidays, and percentage of four types of clients (residential, professional, industrial, high-voltage clients) are used as side information.
- **Portuguese electricity consumption** [36] daily consumption of 370 Portuguese clients from 2010 to 2014 ($T = 1461, N = 369$). The daily temperature in Portugal and calendar variables are used as side information.
- **MovieLens 100k** [37] an anonymized public dataset with 100,000 movie scores for 1682 movies from 943 users ($T = 943, N = 1682$). This is a standard public dataset for matrix completion. Note that the data matrix is not complete. Error rates are calculated on the vector of available scores. The genres of the movies, and gender, age and profession of the users are used as side information.

4.2 Validation procedure

For each data matrix, we apply a linear measurement operator on an upper-left submatrix to obtain the measures (Figure 1). More precisely, the upper-left submatrix is of dimension 100-by-130 for **Synthetic** data, 730-by-270 for **French** data, 731-by-369 for **Portuguese** data, and 666-by-1189 for **MovieLens** data. A random shuffle between columns is done before dividing the matrix into sub-matrices. On time series datasets (the first three), temporal aggregates are generated by choosing a number of observation dates (more details in Section 4.5). On the **MovieLens** dataset, observations are random entries sampled uniformly on the upper-left matrix, as in matrix completion.

We use a reference method or the proposed method to estimate the whole data matrix. We then report the matrix recovery error on the sampled upper-left submatrix, and the prediction errors on the rest of the data matrix. The error rates on the lower-left, upper-right, and lower-right submatrices are called respectively row prediction error, column prediction error, and row-column prediction error (Figure 1).

We report the relative root-mean-squared error metric:

- for electricity datasets, $\mathrm{RRMSE}(\mathbf{V}, \mathbf{V}^*) = \frac{\|\mathbf{V}-\mathbf{V}^*\|_F}{\|\mathbf{V}^*\|_F}$,
- for **MovieLens**, we calculate the RRMSE on the vector of all available movie scores.

In preliminary tests, other error metrics were also evaluated, and were not qualitatively different from this metric.

4.3 Compared methods

Several methods that can be used matrix recovery and time series prediction are compared to the HALSX algorithm.

Among these methods, the following methods are used to compare the matrix recovery performance.

- **interpolation** For temporal aggregates only: to recover the target matrix, temporal aggregates are interpolated equally on the covered periods.
- **HALS [30], NeNMF [38], softImpute [39]** State of art nonnegative matrix factorization and matrix completion methods.

The following methods use side information, and can be used for predicting new columns and/or rows from incomplete data:

- **individual_gam** Estimating separate GAMs on each column or row, on the matrix obtained from **interpolation** or on the whole data matrix when it is available.
- **factor_gam** Estimating GAMs on the factors obtained by **HALS** or **NeNMF**.
- **rrr** [40] Applying reduced-rank regression on the matrix obtained from **interpolation** or on the whole data matrix when it is available.
- **grmf** [2] A matrix completion algorithm using graph-based side information to enhance collaborative filtering performance.
- **trmf** [41] A matrix completion algorithm tailored to time series, by adding three penalization terms to the matrix factorization quadratic error. When only temporal aggregate measurements are available, we apply this method on the matrix obtained from **interpolation**.
- **HALSX** and **HALSX2** Algorithm 2 and Algorithm 3.

For all methods relying on matrix factorization (**HALS**, **NeNMF**, **rrr**, **factor_gam**, **grmf**, **trmf**, **HALSX**), we use the method with several ranks, then choose the best rank ($k \in \{2, 3, ..., 20\}$ for **Synthetic** and **Portuguese** data, $k \in \{2, 3, ..., 10\}$ for **French** and **MovieLens** data). For **trmf**, we do a grid search on the three penalization parameters, and choose the best combination.

As explained in the previous section, a regression method needs to be specified in order to use **HALSX**. This regression method specifies the functional spaces F_r and F_c in which the link functions are to be searched (line 7 and 13 of Algorithm 2). In our experiments, we use four different regression methods with HALSX: linear model (**lm** in standard `R`), GAM (**gam** in the `R` package `mgcv` [34]), support vector regression with linear kernel, and Gaussian process regression with radial basis kernel (**svmLinear** and **gaussprRadial** of the `R` package `kernlab` [42], through the `caret` API [43]).

4.4 Comparison between HALSX and HALSX2

To show-case the computational complexity difference discussed in Section 3.7, we run **HALSX** and **HALSX2** on random temporal aggregate masks, with no side information.

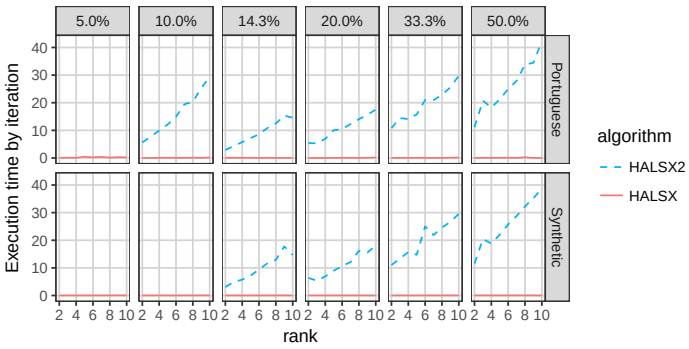

Figure 2. Execution time per iteration of **HALSX** and **HALSX2** without exogenous variables

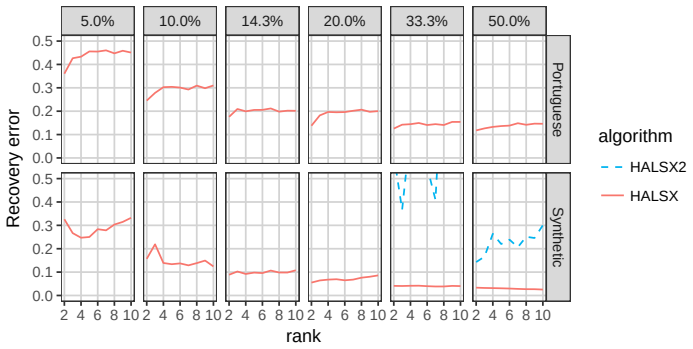

Figure 4. Reconstitution precision of **HALSX** and **HALSX2** without exogenous variables

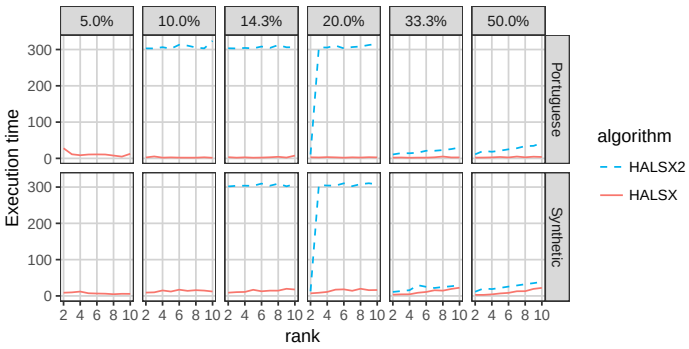

Figure 3. Execution time of **HALSX** and **HALSX2** without exogenous variables

The per iteration execution time is shown in Figure 2. The difference in complexity discussed in Section 3.7 is fairly clear here. In **HALSX2**, the execution time per iteration increases both with the rank and the number of samples in data, at least for sampling rate from 14.3% on. For low sampling rates, **HALSX2** often diverges, where as **HALSX** is always stable.

Although the execution time per iteration is greater, if it had less iterations, **HALSX2** could still be more efficient than **HALSX**. In Figure 3, we can see that this is not the case: for problems with high sampling rates, **HALSX2** indeed does less iterations to converge. However, the total execution time is still larger than **HALSX**. For lower sampling rates, **HALSX2** has troubles converging, and only stops when the maximal execution time allowed (300 seconds) is reached. This is also confirmed in Figure 4, where **HALSX2** has much worse recovery error than **HALSX**.

Given these results, we will only use the **HALSX** (Algorithm 2)in the following tests.

4.5 Performance on temporal aggregate measurements

On the time series datasets, that is **Synthetic**, **French** and **Portuguese**, we use **HALSX** to perform matrix recovery and prediction on temporal aggregate measurements. Every method except for **grmf** is used in this setting.

We use two types of temporal aggregate measurements: periodic and random. In periodic measurements, each scalar measure covers a fixed number of periods of one individual. In random measurements, the number of periods covered by a measure is random (see [5] for more details). In electricity consumption data, periodic measurements are closer to the actual meter reading schedules of utility companies, while matrix recovery with random measurements is an easier problem. For both sampling types, we sample 10%, 20%, ..., 50% of the data to show-case the matrix recovery performance of the proposed method.

For **Synthetic**, we use the true row and column features used to produce the simulations. For the **French** electricity data, the row features are variables known to have an influence on electricity consumption: the temperature, the day type (weekday, weekend, or holiday), the position of the year. The column features are the percentage of residential, professional, or industrial usages in the group of users for each column. For the **Portuguese** electricity data, as no individual features are available, we only use the same row features as for the French dataset (temperature, day type, position of the year).

Figure 5 shows the matrix recovery error. For most of the scenarios, **HALSX** (red lines with symbols) are comparable or better than the other methods without side information. The only case where **HALSX** is a little worse is when compared with **HALS** and **NeNMF** (two NMF methods without side information) in **Synthetic** data with random measurements, which is the least close to the real application. The **softImpute** [39] method is not well adapted to temporal aggregate measurements, and has much higher error (higher than the maximal value in these graphics). When comparing the four regression methods used in **HALSX**, we see that **gam** and **gaussprRadial** (GAM and Gaussian process regression) are the best for **Synthetic** data, linear model (**lm**) is the best for **Portuguese** data, and Gaussian process regression (**gaussprRadial**) is the best for **French** data.

Figures 6, 7, and 8 show the prediction error on the three datasets. We can see that **trmf** [41] and **rrr** [40] are not well adapted to temporal aggregate measurements. When they are applicable (**trmf** is only applicable to row prediction, **rrr** only applicable to row or column predictions, not RC predictions), they have much worse performance than the other methods, except in **Synthetic** data with complete observations.

In most cases, **HALSX** are comparable to or better than **factor_gam** and **individual_gam**, which shows that

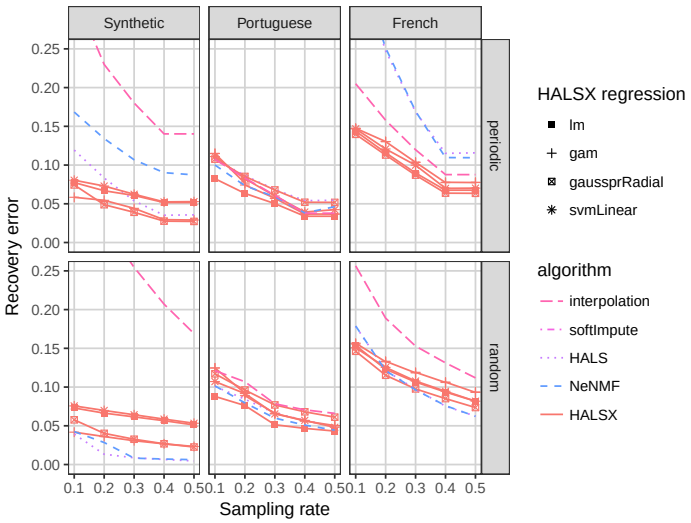

Figure 5. Recovery performance on time series datasets. The line for **softImpute** is sometimes missing because of much higher error rates.

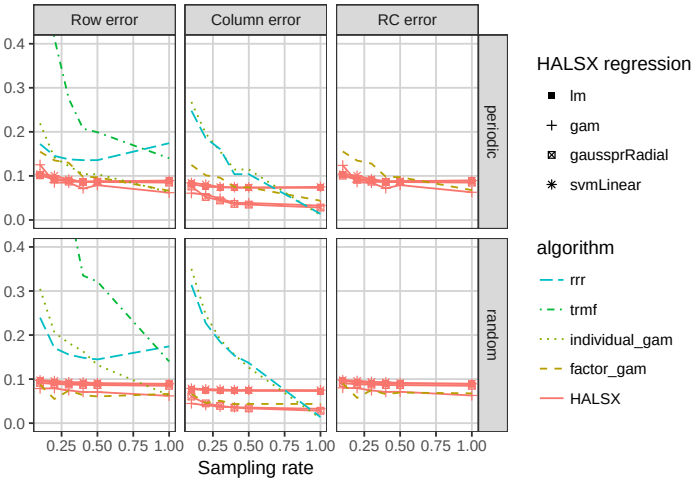

Figure 6. Prediction performance on **Synthetic** data. **trmf** is only available for row predictions. Only **factor_gam** and **HALSX** are available for RC predictions.

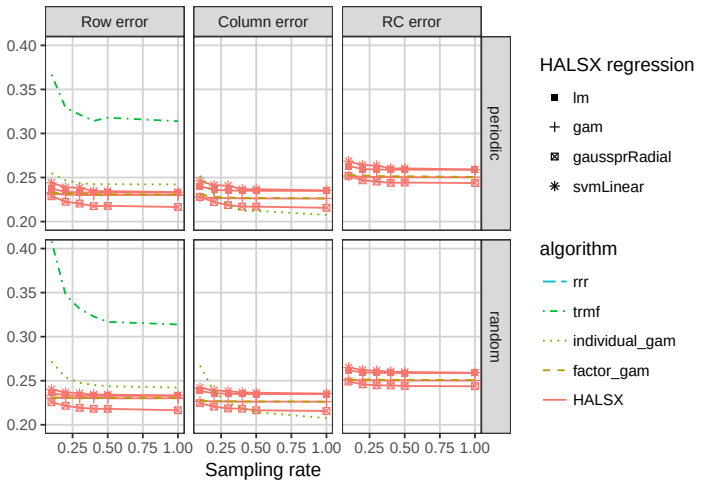

Figure 7. Prediction performance on real **French** electricity data. **trmf** is only available for row predictions. Only **factor_gam** and **HALSX** are available for RC predictions.

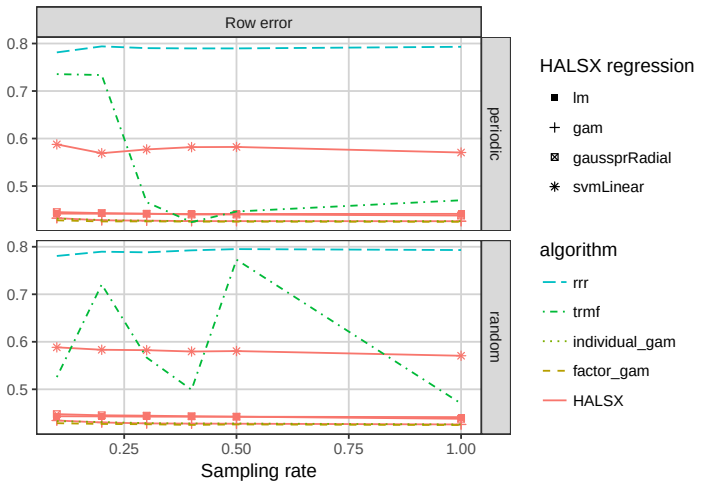

Figure 8. Prediction performance on real **Portuguese** electricity data. **trmf** is only available for row predictions. Only **factor_gam** and **HALSX** are available for RC predictions.

using side information while estimating the factorization model produces factors more adapted for prediction. It is interesting to note that in some cases, using **HALSX** with incomplete data (sampling rate less than 100%) is actually better **individual_gam** with complete data, which models each individual separately with an independent GAM. This means that compared to traditional regression methods, the proposed method achieves better prediction models using much less data, by exploiting the low-rank structure of the problem. When comparing the regression methods used with **HALSX** in prediction, **gam** is consistently the best for **Synthetic** and **Portuguese** data, **gaussprRadial** best for the French dataset.

Moreover, the performance **HALSX** is the least sensitive to sampling rates: it is mostly constant from the sampling rate of 30%. This shows that using side information is supplementary to observing more data.

4.6 Performance on matrix completion mask

On the MovieLens dataset, we use **HALSX** to perform matrix recovery and prediction with uniformly sampled matrix entries. The sampling rates are $10\%, 20\%, 30\%, 40\%, 50\%, 90\%$. As is the case for time series datasets, we use samples from the upper-left submatrix to estimate a model, evaluate matrix recovery on that submatrix, and evaluate row and/or column prediction errors. Every method except for **trmf** is used in this setting.

As side information, we use the gender, the age, and the occupation of the users and the genre (a dimension-19 binary variable) of the movies, which are often used in studies on these datasets [2].

For **grmf**, we produce a graph where each individual is connected to its ten nearest neighbors with euclidean distance with the features, as suggested in the reference [2]. As the parameters estimated in **grmf** is per user/movie, it can not be used to predict new individuals, even though it

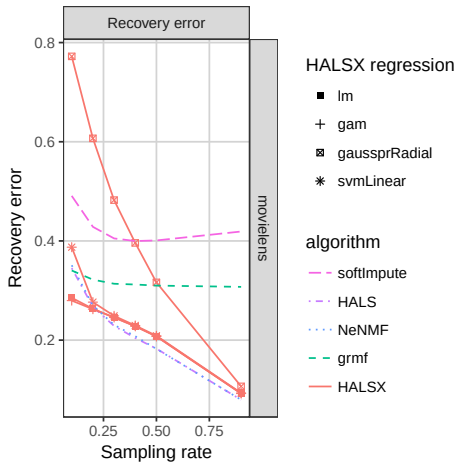

Figure 9. Matrix completion performance on **MovieLens** 100K data.

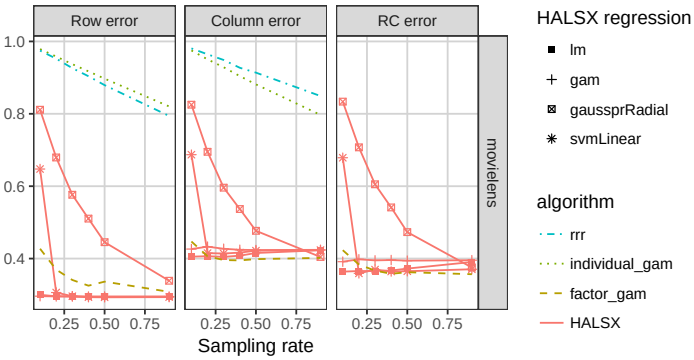

Figure 10. Matrix completion performance for new rows and new columns on **MovieLens** 100K data. Only **factor_gam** and **HALSX** are available for RC predictions.

uses side information.

Figure 9 shows the recovery error on MovieLens data. We can see that in the low sampling rate case (10%), **HALSX** with **lm** and **gam** works the best. In higher sampling rate cases, the NMF methods without side information (**HALS** and **NeNMF**) work the best, while **HALSX** with **lm**, **gam** or **svmLinear** are second to best. **HALSX** with **gaussprRadial** does not work very in this case, as is the case with the other comparison methods (**grmf** and **softImpute**)

Figure 10 shows the prediction error on **MovieLens** data for new users and/or new movies. The order of the variants of the HALSX is conserved: **gam** and **lm** are the best, **svmLinear** is not very good for 10%, but better with higher sampling rates, and **gaussprRadial** does not work well in this problem. Otherwise, the **factor_gam** method is slightly better for column predictions (new movies) in higher sampling rate cases, but worse in other cases. Both **individual_gam** and **rrr** are much worse than **HALSX**.

5 Conclusion

Motivated by electricity consumption estimation, we proposed a general approach for including side information on the columns and row in nonnegative matrix factorization methods, with general linear measurements. Based on a generative model, the framework we propose generalizes many prior works in multivariate regression and matrix factorization.

In order to explore the identifiability of the model, we established a sufficient condition on the features for the factorization to be unique. We deduced the algorithm HALSX to estimate the generative model, and showed that the limiting points of HALSX are are guaranteed to be stationary points with rather mild conditions.

The proposed algorithm was tested on Synthetic and real datasets, both in electricity consumption and recommendation systems. In various sampling scenarios, HALSX produced better or equivalent performance both in matrix recovery and in prediction, compared to a number of reference methods.

Acknowledgments

The authors would like to thank Enedis for their help with the proprietary datasets.

References

[1] P. Jain and I. S. Dhillon, "Provable inductive matrix completion," *arXiv preprint arXiv:1306.0626*, 2013.

[2] N. Rao, H.-F. Yu, P. K. Ravikumar, and I. S. Dhillon, "Collaborative Filtering with Graph Information: Consistency and Scalable Methods," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 2107–2115.

[3] S. Si, K.-Y. Chiang, C.-J. Hsieh, N. Rao, and I. S. Dhillon, "Goal-Directed Inductive Matrix Completion," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, Aug. 2016.

[4] D. Donoho and V. Stodden, "When does non-negative matrix factorization give a correct decomposition into parts?" in *Advances in Neural Information Processing Systems*, 2003, p. None.

[5] 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*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, pp. 2382–2390. [Online]. Available: http://proceedings.mlr.press/v70/mei17a.html

[6] P. Gaillard, Y. Goude, and R. Nedellec, "Additive models and robust aggregation for GEFCom2014 probabilistic electric load and electricity price forecasting," *International Journal of Forecasting*, vol. 32, no. 3, pp. 1038–1050, 2016.

[7] T. Hong, P. Pinson, S. Fan, H. Zareipour, A. Troccoli, and R. J. Hyndman, "Probabilistic energy forecasting: Global Energy Forecasting Competition 2014 and beyond," *International Journal of Forecasting*, vol. 32, no. 3, pp. 896–913, Jul. 2016.

[8] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM review*, vol. 52, no. 3, pp. 471–501, 2010.

[9] M. Roughan, Y. Zhang, W. Willinger, and L. Qiu, "Spatio-Temporal Compressive Sensing and Internet Traffic Matrices (Extended Version)," *IEEE/ACM Transactions on Networking*, vol. 20, no. 3, pp. 662–676, Jun. 2012.

[10] E. J. Candès and B. Recht, "Exact Matrix Completion via Convex Optimization," *Foundations of Computational Mathematics*, vol. 9, no. 6, pp. 717–772, 2009.

[11] T. T. Cai, A. Zhang, and others, "ROP: Matrix recovery via rank-one projections," *The Annals of Statistics*, vol. 43, no. 1, pp. 102–138, 2015.

[12] O. Zuk and A. Wagner, "Low-Rank Matrix Recovery from Row-and-Column Affine Measurements," in *Proceedings of The 32nd International Conference on Machine Learning*, 2015, pp. 2012–2020.

[13] V. Kekatos, Y. Zhang, and G. B. Giannakis, "Low-rank kernel learning for electricity market inference," in *2013 Asilomar Conference on Signals, Systems and Computers*, Nov. 2013, pp. 1768–1772.

[14] J. Abernethy, F. Bach, T. Evgeniou, and J.-P. Vert, "A new approach to collaborative filtering: Operator estimation with spectral regularization," *The Journal of Machine Learning Research*, vol. 10, pp. 803–826, 2009.

[15] R. Velu and G. C. Reinsel, *Multivariate Reduced-Rank Regression: Theory and Applications*. Springer Science & Business Media, Apr. 2013, google-Books-ID: dsfSBwAAQBAJ.

[16] F. Bunea, Y. She, and M. H. Wegkamp, "Joint variable and rank selection for parsimonious estimation of high-dimensional matrices," *The Annals of Statistics*, vol. 40, no. 5, pp. 2359–2388, Oct. 2012.

[17] L. Chen and J. Z. Huang, "Sparse reduced-rank regression for simultaneous dimension reduction and variable selection," *Journal of the American Statistical Association*, vol. 107, no. 500, pp. 1533–1545, 2012.

[18] V. Kekatos, Y. Zhang, and G. B. Giannakis, "Electricity Market Forecasting via Low-Rank Multi-Kernel Learning," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 6, pp. 1182–1193, Dec. 2014.

[19] R. Foygel, M. Horrell, M. Drton, and J. D. Lafferty, "Nonparametric reduced rank regression," in *Advances in Neural Information Processing Systems*, 2012, pp. 1628–1636.

[20] D. Agarwal and B.-C. Chen, "Regression-based latent factor models," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2009, pp. 19–28.

[21] M. Xu, R. Jin, and Z.-H. Zhou, "Speedup Matrix Completion with Side Information: Application to Multi-Label Learning," in *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 2301–2309.

[22] K.-Y. Chiang, C.-J. Hsieh, and I. S. Dhillon, "Matrix Completion with Noisy Side Information," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 3447–3455.

[23] E. J. Candès and Y. Plan, "Tight Oracle Inequalities for Low-Rank Matrix Recovery From a Minimal Number of Noisy Random Measurements," *IEEE Transactions on Information Theory*, vol. 57, no. 4, pp. 2342–2359, 2011.

[24] A. Rohde and A. B. Tsybakov, "Estimation of high-dimensional low-rank matrices," *The Annals of Statistics*, vol. 39, no. 2, pp. 887–930, 2011.

[25] S. Bhojanapalli, B. Neyshabur, and N. Srebro, "Global Optimality of Local Search for Low Rank Matrix Recovery," *arXiv:1605.07221 [cs, math, stat]*, May 2016.

[26] E. A. Pnevmatikakis and L. Paninski, "Sparse nonnegative deconvolution for compressive calcium imaging: Algorithms and phase transitions," in *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 1250–1258.

[27] K. Chen, H. Dong, and K.-S. Chan, "Reduced rank regression via adaptive nuclear norm penalization," *Biometrika*, vol. 100, no. 4, pp. 901–920, Dec. 2013.

[28] K. Huang, N. D. Sidiropoulos, and A. Swami, "Non-Negative Matrix Factorization Revisited: Uniqueness and Algorithm for Symmetric Decomposition," *IEEE Transactions on Signal Processing*, vol. 62, no. 1, pp. 211–224, Jan. 2014.

[29] 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*, vol. 2008, pp. 1–9, 2008.

[30] A. Cichocki, R. Zdunek, and S.-i. Amari, "Hierarchical ALS algorithms for nonnegative matrix and 3D tensor factorization," in *Independent Component Analysis and Signal Separation*. Springer, 2007, pp. 169–176.

[31] J. Kim, Y. He, and H. Park, "Algorithms for nonnegative matrix and tensor factorizations: A unified view based on block coordinate descent framework," *Journal of Global Optimization*, vol. 58, no. 2, pp. 285–319, 2014.

[32] L. Grippo and M. Sciandrone, "On the convergence of the block nonlinear Gauss–Seidel method under convex constraints," *Operations Research Letters*, vol. 26, no. 3, pp. 127–136, 2000.

[33] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Prentice hall Englewood Cliffs, NJ, 1989, vol. 23.

[34] S. Wood, *Generalized Additive Models: An Introduction with R*. CRC press, 2006.

[35] M. Kuhn, "Building predictive models in R using the caret package," *Journal of Statistical Software*, vol. 28, no. 5, pp. 1–26, 2008.

[36] A. Trindade, "UCI Maching Learning Repository - ElectricityLoadDiagrams20112014 Data Set," http://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014, 2016.

[37] "MovieLens 100K Dataset," https://grouplens.org/datasets/movielens/100k/, 2015-09-23T15:02:16+00:00.

[38] N. Guan, D. Tao, Z. Luo, and B. Yuan, "NeNMF: An Optimal Gradient Method for Nonnegative Matrix Factorization," *IEEE Transactions on Signal Processing*, vol. 60, no. 6, pp. 2882–2898, 2012.

[39] R. Mazumder, T. Hastie, and R. Tibshirani, "Spectral regularization algorithms for learning large incomplete matrices," *Journal of machine learning research*, vol. 11, no. Aug, pp. 2287–2322, 2010.

[40] C. Addy, "Reduced-Rank Regression [R package rrr version 1.0.0]."

[41] H.-F. Yu, N. Rao, and I. S. Dhillon, "High-dimensional Time Series Prediction with Missing Values," *arXiv preprint arXiv:1509.08333*, 2015.

[42] A. Karatzoglou, A. Smola, K. Hornik, and A. Zeileis, "Kernlab-an S4 package for kernel methods in R," 2004.

[43] M. Kuhn, "Caret: Classification and regression training," *Astrophysics Source Code Library*, 2015.

Jiali Mei is a PhD student in statistics from 2015 at Université Paris-Sud Orsay and EDF R&D. Her main research interests are statistics and machine learning techniques applied to electricity consumption modeling.

Yohann De Castro received the Ph.D. degree from the Instut de Mathématiques de Toulouse in 2011. Since 2012, he has been an Assistant Professor at Laboratoire de Mathématiques d'Orsay, Université Paris-Sud.

Yannig Goude is a research-engineer/project manager at EDF R&D and associate professor at Université Paris-Sud Orsay, France. He obtained his PhD in statistics and probability in 2008 at Université Paris-Sud Orsay. His research interests are electricity load forecasting, time series analysis, non-parametric models and expert aggregation.

Jean-Marc Azaïs is a Professor of Statistics ans Probability at Institut de Mathématiques de Toulouse. His research themes range from extremes and level sets of stochastic processes and random fields to sparse models, and the application of statistics to biology and global change.

Georges Hébrail is a senior researcher at EDF R&D. His background is in Business Intelligence covering many aspects from data storage and querying to data analytics. From 2002 to 2010, he was a professor of computer science at Telecom ParisTech, teaching and doing research in the field of information systems and business intelligence, with a focus on time series management, stream processing and mining. His current research interest is on distributed and privacy-preserving data mining on electric power related data.