Trend of High Dimensional Time Series Estimation Using Low-Rank Matrix Factorization: Heuristics and Numerical Experiments via the TrendTM Package

Emilie Lebarbier, Nicolas Marie, Amélie Rosier

Trend of High Dimensional Time Series Estimation Using Low-Rank Matrix Factorization: Heuristics and Numerical Experiments via the `TrendTM` Package

Emilie LEBARBIER[1†], Nicolas MARIE[1†] and Amélie ROSIER[1,2†]

[1]MODAL'X, UPL, Univ. Paris Nanterre, CNRS, F92000 Nanterre, France.
[2]ESME Sudria, Ivry-sur-Seine, France.

Contributing authors: emilie.lebarbier@parisnanterre.fr; nmarie@parisnanterre.fr; amelie.rosier@esme.fr;
[†]These authors contributed equally to this work.

Abstract

This article focuses on the practical issue of a recent theoretical method proposed for trend estimation in high dimensional time series. This method falls within the scope of the low-rank matrix factorization methods in which the temporal structure is taken into account. It consists of minimizing a penalized criterion, theoretically efficient but which depends on two constants to be chosen in practice. We propose a two-step strategy to solve this question based on two different known heuristics. The performance and a comparison of the strategies are studied through an important simulation study in various scenarios. In order to make the estimation method with the best strategy available to the community, we implemented the method in an R package `TrendTM` which is presented and used here. Finally, we give a geometric interpretation of the results by linking it to PCA and use the results to solve a high-dimensional curve clustering problem. The package is available on CRAN.

Keywords: trend estimation; dimension reduction; high-dimensional data; penalized contrast; slope heuristic

1 Introduction

Since the 1970's, it is usual to model a one-dimensional time series by a process $(X_t)_{t\in\mathbb{Z}}$ satisfying

$$F_\theta(X_{t+q},\dots,X_{t-q},\eta_{t+p},\dots,\eta_{t-p}) = 0\ ;\ \ t\in\mathbb{Z}, \tag{1}$$

where $p,q\in\mathbb{N}$, $\eta=(\eta_t)_{t\in\mathbb{Z}}$ is a second order stationary process, often a white noise, and $\mathcal{F}=(F_\theta)_{\theta\in\Theta}$ is a family of continuous maps from $\mathbb{R}^{2(p+q+1)}$ into $\mathbb{R}$ indexed in a set Θ. For instance, ARMA models, GARCH models, and all their extensions are defined this way. An advantage of Model (1) is that $\mathcal{F}$ can be chosen in order to take into account properties known on the dynamics of the modeled phenomenon regardless to the data. However, except in simple cases, Model (1) is difficult to extend to the high-dimensional framework. For instance, the vector autoregressive (VAR) and vector autoregressive moving average (VARMA) models have been intensively investigated on the theoretical side and in applications (see Lütkepohl (2005)). However, in the high-dimensional framework, these models cannot be applied directly. Indeed, as mentioned in Gao and Tsay (2022), VARMA models often suffer the difficulties of over-parametrization and lack of identifiability. To bypass such difficulties, some authors have studied extensions of the VAR models: the LASSO regularization of the VAR models (see Shojaie and Michailidis (2010)), the sparse VAR models (see Davis et al. (2016)), the VAR models with low-rank transition matrix (see Alquier et al. (2020)), the factor models (see Lam and Yao (2012a), Gao and Tsay (2022), etc.). Note that the VAR models and their extensions are tailor-maid to take into account a (linear) relationship between the X_t's, but not a sophisticated high-dimensional trend component as in Model (2) presented below and considered throughout our paper. Finally, note that it is also difficult to bypass the stationary condition on η (for a good reference on the classic time series models, see Gourieroux and Monfort (1997)).

Independently, for almost two decades, in particular thanks to the Netflix challenge on movies recommendations, the low rank matrix factorization for the denoising (also for the completion) of high-dimensional matrices with i.i.d. entries has been deeply investigated on the theoretical side (see Cai and Zhang (2015); Klopp et al. (2017, 2019); Koltchinskii et al. (2011); Moridomi et al. (2018)). Indeed, high-dimensional time series often have strong correlation, and it is thus natural to assume that the matrix that contains such a series is low rank (exactly, or approximately). Let denote by $\mathbf{X}$ the observed $d\times n$ matrix which rows are d time series with length n and assume that both d and n are high. Matrix factorization consists in approximating $\mathbf{X}$ by a matrix $\mathbf{M}$ of low rank $k\in\mathbb{N}^*$ (i.e. $k\ll d\wedge n$), which can therefore be written as the product $\mathbf{UV}$ of two matrix $\mathbf{U}\in\mathcal{M}_{d,k}(\mathbb{R})$ and $\mathbf{V}\in\mathcal{M}_{k,n}(\mathbb{R})$ where $\mathcal{M}_{d,k}(\mathbb{R})$ is the set of the matrices of size $d\times k$ with coefficients in $\mathbb{R}$. Formally, let us

consider the model

$$\mathbf{X} = \mathbf{M} + \varepsilon \tag{2}$$

The matrix $\mathbf{M}$ is usually estimated by using a contrast minimization approach, the most popular being the least squares contrast associated to the Frobenius norm: the best rank-k approximation of $\mathbf{X}$ is

$$\widehat{\mathbf{M}}_k \in \arg \min_{\mathbf{U}\in\mathcal{M}_{d,k},\mathbf{V}\in\mathcal{M}_{k,n}(\mathbb{R})} \|\mathbf{X} - \mathbf{UV}\|_{\mathcal{F}}^2, \tag{3}$$

where $\|.\|_{\mathcal{F}}$ is the Frobenius norm (for a matrix $\mathbf{A}$, $\|\mathbf{A}\|_{\mathcal{F}} := \text{trace}(\mathbf{AA}^*)^{1/2}$). Then, the choice of the rank k can be viewed as a model selection issue. In the matrix factorization framework, several approaches have been proposed in the literature (see for instance Candes et al. (2013); Ulfarsson and Solo (2008); Lam and Yao (2012b) for criteria based on the estimated eigenvalue study, Seghouane and Cichocki (2007); Lopes and West (2004) for the classical BIC criterion or dos S. Dias and Krzanowski (2003) for a cross-validation strategy).

When dealing with time series, the matrix $\mathbf{X}$, besides being of low rank, can have a temporal structure, a structure in time, as periodicity, smoothness, etc. It is likely that the temporal properties of the data can be exploited to obtain an accurate factorization. Recently, Alquier and Marie (2019) has extended the latter factorization method in order to take into account the time series trends properties. To this aim, they assume that the matrix $\mathbf{M}$ is structured as follows:

$$\mathbf{M} = \underline{\mathbf{M}}\mathbf{\Lambda}, \tag{4}$$

where $\underline{\mathbf{M}}$ is a $d \times \tau$ matrix of low rank k (thus with $\tau > k$) and $\mathbf{\Lambda}$ is a known $\tau \times n$ full rank matrix reflecting the temporal structure of the data. To estimate $\mathbf{M}$, they developed a penalized least squares criterion (based on the Frobenius norm) and shown that, on the theoretical side, to take into account trends properties in the definition of the denoising estimator allowed to improve existing risk bounds. The penalization aims to choose two parameters: the rank k of the matrix and the parameter τ related to the temporal structure. This penalty function depends also on the noise structure and involves an unknown constant.

In practice, this joint model selection issue is not standard. In addition to a penalty constant to be chosen, parameters from the distribution of the noise need to be estimated in advance. In this paper, we propose an automatic way to deal with these two problems. First, the parameters of the noise distribution are combined with the penalty constant to get a penalty function involving a single constant. This avoids having to estimate the parameters beforehand. Then, we propose a two-stage strategy, as in Devijver et al. (2017); Collilieux et al. (2019), combined with the use of a heuristic for the constant calibration problem. Several heuristics have been considered here, now well-known for the penalty constant calibration in model selection frameworks Lavielle (2005); Birgé and Massart (2001). We demonstrate the performances of our procedure

and compare the considered heuristics in the case of independent Gaussian noises through simulation experiments. The robustness to an autoregressive noise and a nonnormality distribution are also studied.
The method has been implemented in the R package `TrendTM`, for Trend of High-Dimensional Time Series Matrix Estimation, which is available on the CRAN and presented here. When the factorization problem is solved using the Singular Value Decomposition (svd) method, we can make a link to the Principal Component Analysis (PCA) and give an geometrical interpretation of the factorization results. Moreover, we show that, based on this interpretation, a simple clustering method of multiple time series can be derived. It is illustrated on a benchmark dataset of such statistical purpose (see the review and the comparison in Jacques and Preda (2014)).

The paper is organized as follows. Section 2 recalls the estimation procedure proposed in Alquier and Marie (2019). Section 3 presents the proposed two-stage heuristic for the joint selection of the rank and the trend parameter whose performances are studied in Section 4 on simulated data. Section 5 gives some details and guidelines on the proposed method in the `TrendTM` package and shows an application on real data. In Section 6, we give a geometrical interpretation of our results and present the clustering method we proposed.

2 Recall of the trend estimation method proposed by Alquier and Marie (2019)

In this section, we present the method they proposed for estimating $\mathbf{M}$ in model (3) when $\mathbf{M} := \underline{\mathbf{M}}\mathbf{\Lambda}$ (see (4)) and when the noise ε has Gaussian i.i.d. rows of covariance matrix $\mathbf{\Sigma}_\varepsilon$. The general idea of the proposed inference is to estimate first $\underline{\mathbf{M}}$ from the data $\underline{\mathbf{X}}$ (with $\mathbf{X} := \underline{\mathbf{X}}\mathbf{\Lambda}$) by solving the optimization problem (3) on this new dataset and then to come back to the estimation of $\mathbf{M}$.

First, the two temporal structures they considered are the following:

- **periodicity:** if the trend of $\mathbf{X}$ is τ-periodic, then $\mathbf{\Lambda} = (\mathbf{I}_\tau \mid \cdots \mid \mathbf{I}_\tau)$ where $\mathbf{I}_\tau$ is the identity matrix in $\mathcal{M}_{\tau,\tau}(\mathbb{R})$,
- **smoothness:** if the form of the trend is $t \in \{1,\dots,n\} \mapsto f(t/n)$ with $f \in \mathbb{L}^2([0,1];\ \mathbb{R}^d)$, then

$$\mathbf{\Lambda} = \left(\varphi_\ell\left(\frac{t}{n}\right)\right)_{(\ell,t)\in\{1,\dots,\tau\}\times\{1,\dots,n\}},$$

where τ is odd and $(\varphi_1,\dots,\varphi_\tau)$ is the τ-dimensional trigonometric basis defined by

$$\varphi_\ell(x) := \begin{cases} 1 & \text{if } \ell = 1 \\ \sqrt{2}\cos(2\pi m x) & \text{if } \ell = 2m \\ \sqrt{2}\sin(2\pi m x) & \text{if } \ell = 2m+1 \end{cases}$$

for every $x \in [0,1]$ and $m \in \{1,\dots,(\tau-1)/2\}$.

So, the estimation procedure consists in two steps:

Step 1: Estimation of $\mathbf{M}$ *for* k *and* τ *being fixed.* They define the following auxiliary model

$$\underline{\mathbf{X}} = \underline{\mathbf{M}} + \underline{\varepsilon}, \tag{5}$$

where $\underline{\mathbf{X}} := \mathbf{X}\mathbf{\Lambda}^+$, $\underline{\varepsilon} := \varepsilon\mathbf{\Lambda}^+$ and $\mathbf{\Lambda}^+ = \mathbf{\Lambda}^*(\mathbf{\Lambda}\mathbf{\Lambda}^*)^{-1}$ is the Moore-Penrose inverse of $\mathbf{\Lambda}$. This model doesn't embed some trend's property anymore. The least squares estimator of the matrix $\underline{\mathbf{M}}$ is thus classical:

$$\widehat{\underline{\mathbf{M}}}_{k,\tau} \in \arg\min_{\mathbf{A}\in\mathcal{S}_{k,\tau}} \|\underline{\mathbf{X}} - \mathbf{A}\|_{\mathcal{F}}^2, \tag{6}$$

where $\mathcal{S}_{k,\tau} = \{\mathbf{U}\mathbf{V}\ ;\ \mathbf{U} \in \mathcal{M}_{d,k}(\mathbb{R}) \text{ and } \mathbf{V} \in \mathcal{M}_{k,\tau}(\mathbb{R})\}$. So, a natural estimator of $\mathbf{M}$ is given by

$$\widehat{\mathbf{M}}_{k,\tau} := \widehat{\underline{\mathbf{M}}}_{k,\tau}\mathbf{\Lambda}.$$

Step 2: Choice of k *and* τ. For a fixed $s > 0$, the final estimator of $\mathbf{M}$ is $\widehat{\mathbf{M}}_s := \widehat{\mathbf{M}}_{\widehat{k}(s),\widehat{\tau}(s)}$ where

$$(\widehat{k}(s),\widehat{\tau}(s)) \in \arg\min_{(k,\tau)\in\mathcal{K}\times\mathcal{T}} \{\|\mathbf{X} - \widehat{\mathbf{M}}_{k,\tau}\|_{\mathcal{F}}^2 + \mathrm{pen}_s(k,\tau)\}$$

with $\mathcal{K} = \{1,\dots,d\wedge n\}$, $\mathcal{T} = \{1,\dots,n\}$, and

$$\mathrm{pen}_s(k,\tau) := \mathfrak{c}_{\mathrm{pen}}\|\mathbf{\Sigma}_\varepsilon\|_{\mathrm{op}} k(d+\tau+s)\ ;\ \forall (k,\tau)\in\mathcal{K}\times\mathcal{T}, \tag{7}$$

where $\mathfrak{c}_{\mathrm{pen}} > 0$ is a deterministic constant and $\|.\|_{\mathrm{op}}$ is the operator norm on $\mathcal{M}_{n,n}(\mathbb{R})$ ($\|\mathbf{A}\|_{\mathrm{op}} := \sup_{\|x\|=1}\|\mathbf{A}x\|$ with $\|.\|$ the Euclidean norm on $\mathbb{R}^n$). They establish an oracle-type inequality on the resulting estimator (see Alquier and Marie (2019) (Theorem 4.1)): for every $\theta \in (0,1)$, with probability larger than $1-2e^{-s}$,

$$\|\widehat{\mathbf{M}}_s - \mathbf{M}\|_{\mathcal{F}}^2 \leqslant \min_{(k,\tau)\in\mathcal{K}\times\mathcal{T}} \min_{\mathbf{A}\in\mathcal{S}_{k,\tau}} \left\{ \left(\frac{1+\theta}{1-\theta}\right)^2 \|\mathbf{A}\mathbf{\Lambda} - \mathbf{M}\|_{\mathcal{F}}^2 + \frac{4}{\theta(1-\theta)^2}\mathrm{pen}_s(k,\tau) \right\}. \tag{8}$$

The parameter s in the penalty is linked to the confidence level in the risk bound for a fixed k and τ. Making the penalty depend on s is necessary to establish a risk bound on the adaptive estimator. This is a technical condition in fact, which we could also get rid of if we only selected τ for a fixed k (see Theorem 4.1 and Remark 4.2 in Alquier and Marie (2019)). The penalty is

also proportional to the number of series d as in multiple serie estimation framework (see for example Collilieux et al. (2019)) since the estimation cost increases naturally with d.

Finally, note that through the penalty defined by (7), the right-hand side of inequality (8) depends on $\|\Sigma_\varepsilon\|_{\text{op}}$ because of the concentration inequality for random matrices with i.i.d. (sub-)Gaussian rows (see Vershynin (2012), Theorem 5.39 and Remark 5.40.(2)) used to control the variance term in the proof of Alquier and Marie (2019), Theorem 3.2 (and then Theorem 4.1). This is one of the reasons why we consider the quadratic loss and why the second order moment Σ_ε of the rows of ε appears in the risk bound (8).

3 The proposed two-stage heuristic for the model selection issue in practice

Discussion on the penalty function. The penalty function given by (7) depends of some constants s and $\mathfrak{c}_{\text{pen}}$ that must be chosen or calibrated in practice. It also depends on the parameters of the noise distribution through $\|\mathbf{\Sigma}_\varepsilon\|_{\text{op}}$ that must be estimated thus in advance: we explicit just below this norm in two cases that are considered in the simulation study (Section 4):

- when the errors are uncorrelated ($\text{cov}(\varepsilon_{1,t}, \varepsilon_{1,t'}) = \sigma^2 \mathbb{1}_{t=t'}$), then

$$\|\mathbf{\Sigma}_\varepsilon\|_{\text{op}} = \sigma^2,$$

- when $(\varepsilon_{1,t})_t$ is a zero-mean stationary AR(1) Gaussian process (defined as the solution of $\varepsilon_{1,t} = \rho\varepsilon_{1,t-1} + \eta_{1,t}$ where $\rho \in (-1, 1)$ and $(\eta_{1,t})_t$ is a white noise of standard deviation σ), we can show that

$$\|\mathbf{\Sigma}_\varepsilon\|_{\text{op}} \geqslant \sigma^2(1+\rho) =: f(\rho). \tag{9}$$

Indeed, the covariance matrix of a row noise $(\varepsilon_0, \ldots, \varepsilon_{n-1})$ is

$$\mathbf{\Sigma}_\varepsilon := (\sigma^2 \rho^{|i-j|})_{i,j}.$$

Then, for every $x \in \mathbb{R}^n$ such that $\|x\| = 1$,

$$\begin{aligned} x^* \mathbf{\Sigma}_\varepsilon x &= \sum_{i,j=1}^{n} x_i x_j [\mathbf{\Sigma}_\varepsilon]_{i,j} = \sigma^2 \left(\|x\|^2 + \sum_{i \neq j} x_i x_j \rho^{|i-j|} \right) \\ &= \sigma^2 \left(1 + 2 \sum_{i>j} x_i x_j \rho^{i-j} \right). \end{aligned}$$

Since $\mathbf{\Sigma}_\varepsilon$ is a symmetric matrix,

$$\|\mathbf{\Sigma}_\varepsilon\|_{\text{op}} = \sup_{\|x\|=1} |x^* \mathbf{\Sigma}_\varepsilon x| \geqslant |\mathbf{x}^* \mathbf{\Sigma}_\varepsilon \mathbf{x}| \quad \text{with} \quad \mathbf{x} = \frac{1}{\sqrt{2}}(1, 1, 0, \ldots, 0)$$
$$\geqslant \mathbf{x}^* \mathbf{\Sigma}_\varepsilon \mathbf{x} = \sigma^2 \left(1 + 2 \cdot \frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{2}} \cdot \rho^{2-1}\right) = \sigma^2(1+\rho).$$

Two-stage heursitic. We set $\mathfrak{c}_{\text{cal}} = \mathfrak{c}_{\text{pen}} \|\mathbf{\Sigma}_\varepsilon\|_{\text{op}}$, representing a global penalty constant that we propose to calibrate using the data. So, this allows us to avoid the estimation of the noise distribution parameters, which turns out to be a difficult task. The penalty function is thus reduced to

$$\text{pen}(k, \tau) := \mathfrak{c}_{\text{cal}} k(d + \tau + s) \; ; \forall (k, \tau) \in \mathcal{K} \times \mathcal{T},$$

and the resulting adaptive estimator is denoted by $\widehat{\mathbf{M}}_s := \widehat{\mathbf{M}}_{\widehat{k},\widehat{\tau}}$.

If the constant s can be easily chosen, this is not the case for the penalty constant $\mathfrak{c}_{\text{cal}}$. Several heuristics have been proposed in the literature for this purpose in model selection frameworks, but for a one-dimensional parameter only (see Lavielle (2005); Birgé and Massart (2001)). First, in practice, we could take $s = -\log((1-\alpha)/2)$ with α fixed to 99%, 95% or 90%. Here we choose to fix $s = 4$. Then, for the selection of both k and τ, we follow the same strategy than in Devijver et al. (2017), that is a two-stage heuristic. We first recall some heuristics for the selection of one parameter, and then we present the two-stage heuristic for the joint selection of (k, τ).

Up to our knowledge, there exit the three following heuristics dedicated to the constant calibration question in the model selection frameworks of one parameter:

- the one proposed in Lavielle (2005), denoted here ML, that involves a threshold S which is fixed to $S = 0.75$ as suggested by the author, and
- the two proposed in Birgé and Massart (2001) (see the more recent version of Arlot and Massart (2009) and the huge survey paper of Arlot (2019)) that are two versions of the well-known slope heuristic: the 'dimension jump' and the 'slope', denoted here BJ and Slope respectively. The both heuristics have been implemented in the R package `capushe` described in Baudry et al. (2012).

A brief description of these heuristics is given in Appendix 8. For the joint selection of (k, τ), the two-stage heuristic is the following: first, we choose the best τ for each $k \in \mathcal{K}$ via the criterion

$$\widehat{\tau}(k) \in \arg\min_{\tau \in \mathcal{T}} \{\|\mathbf{X} - \widehat{\mathbf{M}}_{k,\tau}\|_{\mathcal{F}}^2 + \mathfrak{c}_{\text{cal},\tau} k(d + \tau + s)\},$$

where the penalty constant $\mathfrak{c}_{\text{cal},\tau}$ is calibrated using one of the previous heuristics, and then we select the best k among them via the criterion

$$\widehat{k} \in \arg\min_{k\in\mathcal{K}}\{\|\mathbf{X} - \widehat{\mathbf{M}}_{k,\widehat{\tau}(k)}\|_{\mathcal{F}}^2 + \mathfrak{c}_{\text{cal},k}k(d + \widehat{\tau}(k) + s)\},$$

where the penalty constant $\mathfrak{c}_{\text{cal},k}$ is calibrated using the same heuristic to be consistant, and $\widehat{\tau} = \widehat{\tau}(\widehat{k})$.

Note that in practice $\mathcal{K} = \{1, \ldots, k_{\max}\}$ and $\mathcal{T} = \{k+1, \ldots, \tau_{\max}\}$, where $k_{\max}$ is the maximal rank and $\tau_{\max}$ is the maximal value of τ. These two quantities need to be specified. Moreover, we propose this strategy and not the opposite because on the one hand $k < \tau$ theoretically and on the other hand the slope heuristic requires having a minimum point. Using the proposed strategy allows to visit clearly more dimensions (k, τ).

4 Simulation study

In this study, we conduct different simulations studies to both evaluate the performance of the proposed method and compare the three different heuristics:

- Study 1: we consider the model selection issue for k and τ separatly,
- Study 2: we illustrate the importance to take into account the trend in the estimation procedure when it exits,
- Study 3: we consider the model selection issue for both k and τ.

We also performed additional separate simulations:

- Study 4: we assume that for each series there exists a temporal dependency which is modelled through an AR process,
- Study 5: we study the robustness of the proposed method to nonnormality errors.

4.1 Simulation design and quality criteria

4.1.1 Simulation design

We've simulated datasets with $d = 100$ and $n = 600$ as follows:

(1) we generate a matrix $\underline{\mathbf{M}} = \mathbf{UV}$ by simulating $\mathbf{U} \in \mathcal{M}_{d,k}(\mathbb{R})$ and $\mathbf{V} \in \mathcal{M}_{k,\tau}(\mathbb{R})$ for which the entries of $\mathbf{U}$ and $\mathbf{V}$ are assumed to be i.i.d. and follows a centered Gaussian distribution with same standard deviation σ_{uv} fixed to 0.5;

(2) two cases are considered according to the presence or not of a trend in the simulated series: if there is no trend, then $\tau = n$ and $\mathbf{M} = \underline{\mathbf{M}}$, and otherwise $\mathbf{M} = \underline{\mathbf{M}}\mathbf{\Lambda}$ with the matrix $\mathbf{\Lambda}$ of the smooth case. To distinguish between these two cases in the sequel, we call them $\text{dataset}_{\text{NoTrend}}$ and $\text{dataset}_{\text{Trend}}$ respectively;

(3) the rows of the error matrix ε are assumed to be i.i.d. and follow a centered Gaussian distribution of variance σ^2 (i.e. $\mathbf{\Sigma}_\varepsilon = \sigma^2 \mathbf{I}_n$) for Studies 1, 2 and 3; the rows of the error matrix ε are assumed to be i.i.d. stationary AR(1) Gaussian processes with a white noise of standard deviation $\sigma > 0$ and an autocorrelation parameter $\rho \in (-1, 1)$ for Study 4.

We take $k = 3$ and $\tau = 25$. We consider different values for the residual standard deviation σ in order to have different levels of difficulty for the estimation problem. First, according to the previous considerations, $\mathrm{var}(\mathbf{M}_{ij}) = k\sigma_{uv}^4$ for $\mathrm{dataset}_{\mathrm{NoTrend}}$ and $\mathrm{var}(\mathbf{M}_{ij}) = \tau k \sigma_{uv}^4$ for $\mathrm{dataset}_{\mathrm{Trend}}$. For Studies 1, 2 and 3, let us consider $s_v \in \{0.1, 0.5, 1.5, 2\}$. In order to have the same estimation difficulty (same ratio between σ and the standard deviation of $\mathbf{M}_{ij}$) for the two datasets, we set $\sigma = s_v$ for $\mathrm{dataset}_{\mathrm{NoTrend}}$ and $\sigma = \sqrt{\tau} s_v$ for $\mathrm{dataset}_{\mathrm{Trend}}$. The obtained four cases are judged as 'Easy', 'Medium', 'Difficult' and 'Hard' respectively. Study 4 is the same as Study 3 but with a noise modeled by an autoregressive process. More precisely, we consider two values for the standard deviation of the noise $s_v \in \{0.1, 1.5\}$ and an autocorrelation parameter $\rho \in \{-0.8, -0.3, 0, 0.3, 0.8\}$. For each combination of parameters, we've simulated 200 datasets.

Let us precise that when the trend is not considered in the estimation procedure, the resulting estimator is

$$\widehat{\mathbf{M}}_{k \text{ or } \widehat{k},n} \text{ (if } k \text{ is selected or not)},$$

and when it is considered the resulting estimator is

$$\widehat{\mathbf{M}}_{k \text{ or } \widehat{k},\tau \text{ or } \widehat{\tau}} \text{ (if both } k \text{ and } \tau \text{ are selected or one of them or none)}.$$

For Study 5, we consider the same simulation design as in Study 3 but by considering a heavy-tailed distribution for the errors $\{\varepsilon_t\}_t$, namely, a Student distribution with degrees of freedom $\nu = 50, 10, 3$ ($\nu = 50$ being the closest Gaussian case).

4.1.2 Quality criteria

The performance of our procedure is assessed via:

- the estimated k and/or τ; and
- the squared Frobenius distance between $\mathbf{M}$ and its estimate $\widehat{\mathbf{M}}_{\widehat{k},\widehat{\tau}}$.

Moreover, we also consider the Frobenius distance between $\mathbf{M}$ and

- the estimator of $\mathbf{M}$ for the true k and/or τ, that is $\widehat{\mathbf{M}}_{k,\tau}$; and
- the trajectorial oracle, that is $\widehat{\mathbf{M}}_{\widetilde{k},\widetilde{\tau}}$ where

$$(\widetilde{k}, \widetilde{\tau}) = \arg \min_{(k,\tau) \in \mathcal{K} \times \mathcal{T}} \|\mathbf{M} - \widehat{\mathbf{M}}_{k,\tau}\|_{\mathcal{F}}^2$$

when both k and τ are selected, $\widehat{\mathbf{M}}_{\widetilde{k},\widetilde{\tau}}$ where

$$\widetilde{\tau} = \arg\min_{\tau \in \mathcal{T}} \|\mathbf{M} - \widehat{\mathbf{M}}_{k,\tau}\|_{\mathcal{F}}^2$$

when k is fixed, and $\widehat{\mathbf{M}}_{\widetilde{k},n}$ where

$$\widetilde{k} = \arg\min_{k \in \mathcal{K}} \|\mathbf{M} - \widehat{\mathbf{M}}_{k,n}\|_{\mathcal{F}}^2$$

when no trend is considered.

4.2 Study 1: behavior of the three heuristics for the selection of k or τ

We first study the selection of k for $\text{dataset}_{\text{NoTrend}}$ when no trend is considered in the estimation procedure. We consider two different values of the maximal rank $k_{\max} \in \{15, 35\}$. The results are presented in Figure 1. When the noise is small, i.e. the estimation problem is easy (cases 'Easy' and 'Medium'), all the heuristics recover the true rank, and therefore the obtained estimators perform as well as $\widehat{\mathbf{M}}_{k,n}$ (the estimator of $\mathbf{M}$ for k fixed to its true value). When the estimation problem gets more difficult (cases 'Difficult' and 'Hard'), the heuristics tend to underestimate the rank. This underestimation behavior seems to be logical and even desirable in the particular 'Hard' case. Indeed, we observe that in terms of Frobenius norm, the obtained estimators perform better compared to the one with the true rank. Moreover, they have performance close to the oracle. Comparing the three heuristics, the Slope heuristic shows better performances compared to the two other heuristics. This is particularly marked for the 'Medium' case and $k_{\max} = 15$. We can note that the behavior of the three heuristics can be affected by the choice of $k_{\max}$. This problem is well-known for both the BJ and Slope heuristics (see Arlot (2019) for more explanations in the case univariate series analysis).

Then, we study the selection of τ for $\text{dataset}_{\text{Trend}}$ for k fixed to the true value. We fix $\tau_{\max} = 55$. The results are presented in Figure 2. Except with BJ that is more unstable, the heuristics retrieve the true value of τ whatever the estimation difficulty with same performance as the oracle.

From this study, we choose the Slope heuristic for the model selection issue for both k and τ in the sequel and in the developed package.

4.3 Study 2: accounting for the smooth structure in the trend

We compare the performance of the procedure on the $\text{dataset}_{\text{Trend}}$ when the trend is considered ($\tau = \widehat{\tau}$) or not ($\tau = n$) for k fixed to the true value. We choose $\tau_{\max} = 55$. The results are represented in Figure 3. Whatever the

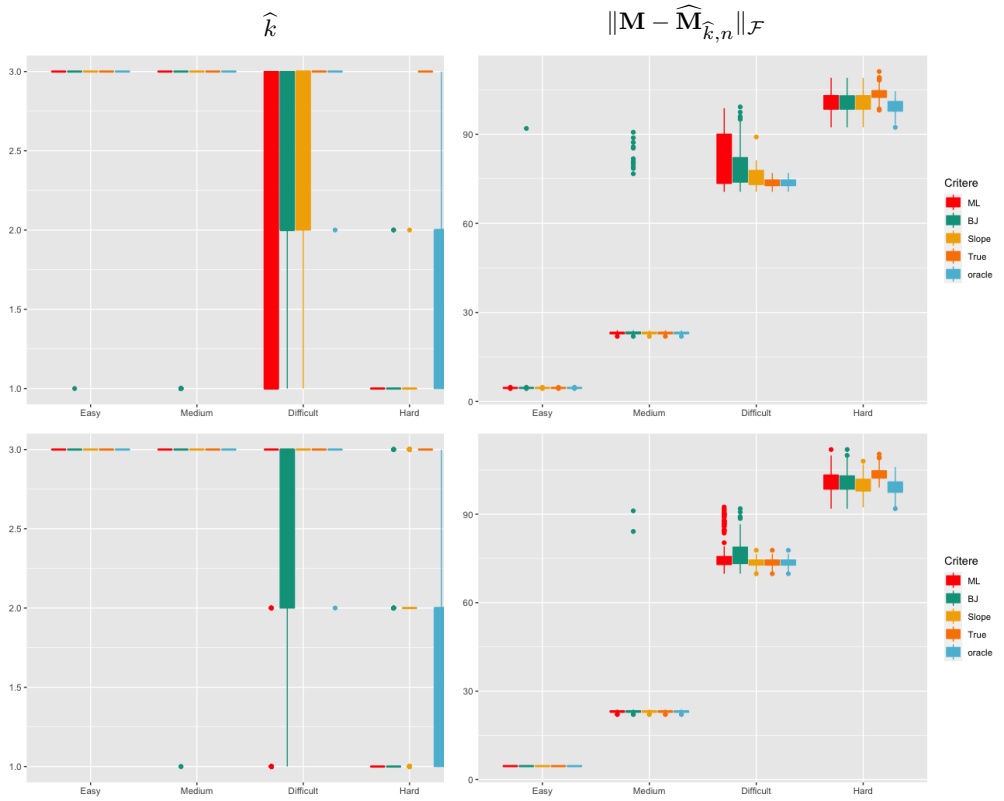

Fig. 1 Comparison of the three heuristics for the selection of k for $\text{dataset}_{\text{NoTrend}}$ (Study 1). Left: boxplot of estimated number of the rank k. Right: boxplot of $\|\mathbf{M} - \widehat{\mathbf{M}}_{\widehat{k},n}\|_{\mathcal{F}}$ for two values of $k_{\max} = 15$ (first line) and $k_{\max} = 35$ (second line), and different values of σ. On each graph and for each value of σ, from left to right, we have the result from ML, BJ, Slope ($\widehat{k}$), the true rank (k) and the oracle ($\widetilde{k}$).

difficulty of the estimation problem (different values of σ), accounting for the trend increases the precision of the estimation. This is more marked for high values of σ. Note that, similarly as Study 1, the estimation naturally degrades with the increasing of σ.

4.4 Study 3: selection of k and τ

Table 1 shows that the joint heuristic retrieves the true values of k and τ whatever the difficulty of the estimation problem, except very few times. Thus, the performance of the estimator of $\mathbf{M}$ is comparable to the one of the estimator $\mathbf{M}_{k,\tau}$ and moreover it has performance close to the oracle (see Figure 4). Compared to Study 1 where $\tau = n$, here for difficult estimation problems, k is not underestimated.

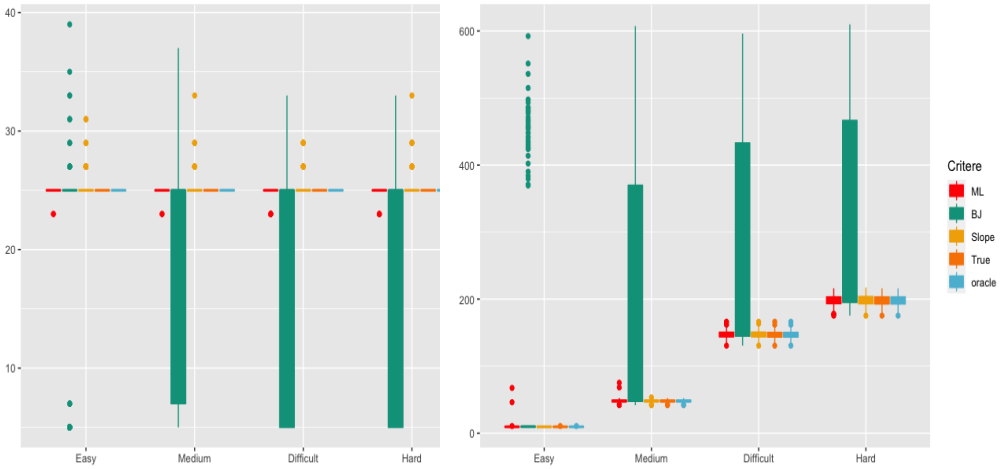

Fig. 2 Comparison of the three heuristics for the selection of τ for dataset$_{\text{Trend}}$ when k is fixed to the truth ($k = 3$) for different values of σ (Study 1). Left: boxplot of estimated τ. Right: boxplot of $\|\mathbf{M} - \widehat{\mathbf{M}}_{k,\widehat{\tau}}\|_{\mathcal{F}}$. In each graph and each value of σ, from left to right, we have the result from ML, BJ, Slope ($\widehat{\tau}$), the true value (τ) and the oracle ($\widetilde{\tau}$).

$(\hat{k}, \hat{\tau})$	Easy	Medium	Difficult	Hard
Mean	(3.01, 25.26)	(3.035, 25.18)	(3.045, 25.14)	(3.05, 25.29)
Sd	(0.099, 0.926)	(0.209, 0.728)	(0.231, 0.618)	(0.267, 1.159)

$(\tilde{k}, \tilde{\tau})$	Easy	Medium	Difficult	Hard
Mean	(3, 25)	(3,25)	(3,25)	(3,25)
Sd	(0, 0)	(0, 0)	(0, 0)	(0, 0)

Table 1 Estimated k and τ, $(\hat{k}, \hat{\tau})$, and the oracle $(\tilde{k}, \tilde{\tau})$ for different values of σ (Study 3). The true values are $(k, \tau) = (3, 25)$.

4.5 Study 4: robustness to autocorrelated noise

Whatever the dependence and the noise variance, the joint heuristic retrieves the true values of k and τ (see Tables 2 and 3), except for a large variance ($s_v = 1.5$) and a high positive autocorrelation ($\rho = 0.8$) where it underestimates k and the selection of τ is more variable. For all noise cases, the method leads to estimators that have close performance compared to the oracle (see Figures 5 and 6) and with better performance than the one with the true values for the excepted case.

Moreover, we can observe that the more the autocorrelation parameter ρ increases (from -1 to 1), the more $\|\mathbf{M} - \widehat{\mathbf{M}}_{k,\tau}\|_{\mathcal{F}}$ increases also with a noticeable gap between $\rho = 0.3$ and $\rho = 0.8$ for both values of s_v. First, the estimation

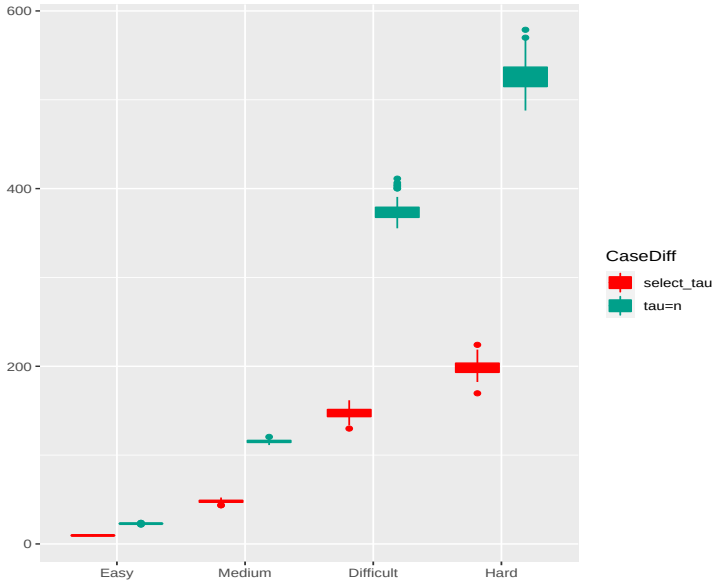

Fig. 3 Boxplot of $\|\mathbf{M} - \widehat{\mathbf{M}}_{k,\tau}\|_{\mathcal{F}}$ with $\tau = \hat{\tau}$ (select_tau) and $\tau = n$ (tau = n) for different values of σ (Study 2).

$(\hat{k}, \hat{\tau})$	-0.8	-0.3	0	0.3	0.8
Mean	(3, 25.24)	(3.015, 25.3)	(3.025, 25.25)	(3.005, 25.16)	(3.01, 26.01)
Sd	(0, 0.973)	(0.157, 1.075)	(0.186, 0.895)	(0.071, 0.760)	(0.099, 2.242)

$(\tilde{k}, \tilde{\tau})$	-0.8	-0.3	0	0.3	0.8
Mean	(3, 25)	(3, 25)	(3, 25)	(3, 25)	(3, 25)
Sd	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)

Table 2 Estimated k and τ, $(\hat{k}, \hat{\tau})$, and the oracle $(\tilde{k}, \tilde{\tau})$ for different values of ρ and for the standard deviation $s_v = 0.1$ (Study 4). The true values are $(k, \tau) = (3, 25)$.

is better with high and negative autocorrelation. Then, the observed phenomenon on the norm can be explained. The *variance term* in the risk bound of the estimator $\widehat{\mathbf{M}}_{\hat{k},\hat{\tau}}$ (i.e. the penalty, see (8)) depends on $\|\mathbf{\Sigma}_\varepsilon\|_{\text{op}}$ (see (7)). Using (9), we can show that this term is lower-bounded by

$$f(\rho)\frac{k(d+\tau)}{dT}$$

up to a multiplicative constant where f is increasing and nonnegative on $[-1, 1]$.

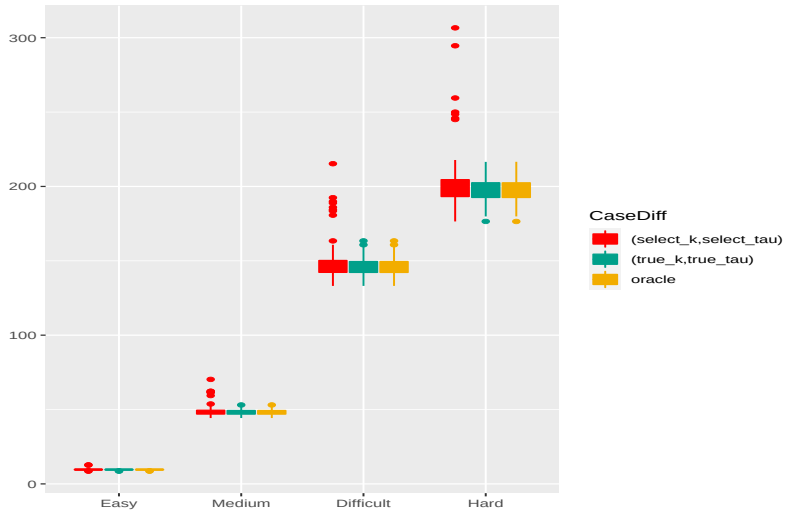

Fig. 4 Boxplot of $\|\mathbf{M} - \widehat{\mathbf{M}}_{k,\tau}\|_{\mathcal{F}}$ with $(k,\tau) = (\widehat{k},\widehat{\tau})$ the selected k and τ, $(k,\tau) = (k,\tau)$ the true values and $(k,\tau) = (\widetilde{k},\widetilde{\tau})$ the oracle for different values of σ (Study 3).

$(\hat{k},\hat{\tau})$	-0.8	-0.3	0	0.3	0.8
Mean	(3.05, 25.29)	(3.025, 25.28)	(3.015, 25.25)	(3.025, 25.27)	(1.4, 25.67)
Sd	(0.279, 1.030)	(0.157, 0.962)	(0.158, 0.825)	(0.157, 0.936)	(0.783, 5.081)

$(\tilde{k},\tilde{\tau})$	-0.8	-0.3	0	0.3	0.8
Mean	(3, 25)	(3, 25)	(3, 25)	(3, 25)	(1, 17.5)
Sd	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 9.152)

Table 3 Estimated k and τ, $(\hat{k},\hat{\tau})$, and the oracle $(\tilde{k},\tilde{\tau})$ for different values of ρ and for the standard deviation $s_v = 1.5$ (Study 4). The true values are $(k,\tau) = (3,25)$.

4.6 Study 5: robustness to nonnormality of the errors

Figure 7 and Table 4 display the results of the Student simulation. The joint heuristic retrieves the true values of k and τ in average, but with a slight overestimation of k for the extreme case ($\nu = 3$). However, we observe a slight degradation in the quality of the estimation of $\mathbf{M}$, which is more marked as we move away from the Gaussian hypothesis.

5 Using the `TrendTM` package

The version of the package is 2.0.19.

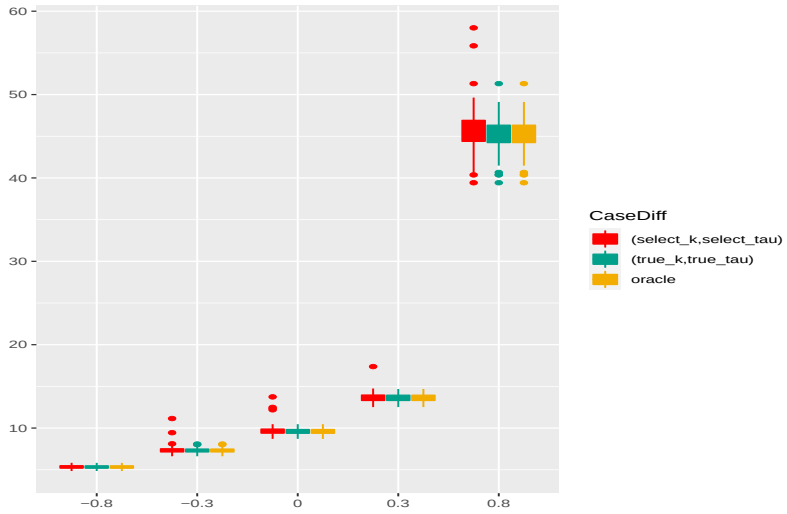

Fig. 5 Boxplot of $\|\mathbf{M} - \widehat{\mathbf{M}}_{k,\tau}\|_{\mathcal{F}}$ with $(k,\tau) = (\widehat{k},\widehat{\tau})$ the selected k and τ, $(k,\tau) = (k,\tau)$ the true values and $(k,\tau) = (\widetilde{k},\widetilde{\tau})$ the oracle for different values of ρ and for the standard deviation $s_v = 0.1$ (Study 4).

$(\hat{k},\hat{\tau})$	50	10	3
Mean	(3.02, 25.2)	(3.02, 25.3)	(3.28, 25.4)
Sd	(0.157, 0.807)	(0.122, 1.29)	(0.539, 0.998)

$(\tilde{k},\tilde{\tau})$	50	10	3
Mean	(3, 25)	(3, 25)	(3, 25)
Sd	(0, 0)	(0, 0)	(0, 0)

Table 4 Estimated k and τ, $(\hat{k},\hat{\tau})$, and the oracle $(\tilde{k},\tilde{\tau})$ for different values of ν (Study 5). The true values are $(k,\tau) = (3, 25)$.

5.1 Comments on the package

The package is organized around the main function `TrendTM`. In this section, we present the arguments used in a call of this function to a dataset named Data_Series

```
TrendTM(Data_Series, k_select = FALSE,
                    k_max = 20, struct_temp = "none", tau_select = FALSE,
                    tau_max = floor(n / 2), type_soft = "als")
```

This function returns a list containing six elements:

- `k_est`, the estimated k or the true k when no selection is chosen;
- `tau_est`, the estimated τ or the true τ when no selection is chosen;

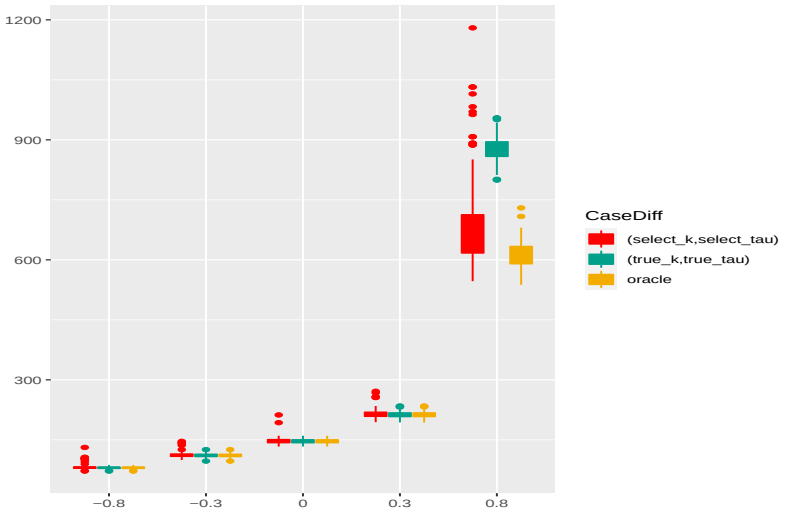

Fig. 6 Boxplot of $\|\mathbf{M} - \widehat{\mathbf{M}}_{k,\tau}\|_{\mathcal{F}}$ with $(k,\tau) = (\widehat{k},\widehat{\tau})$ the selected k and τ, $(k,\tau) = (k,\tau)$ the true values and $(k,\tau) = (\widetilde{k},\widetilde{\tau})$ the oracle for different values of ρ and for the standard deviation $s_v = 1.5$ (Study 4).

- `M_est`, the estimation of $\mathbf{M}$ (`M_est` = `U_estV_est` if no temporal structure is considered and `M_est` = `U_estV_est` $\mathbf{\Lambda}$ if a temporal structure is considered);
- `U_est`, the component U of the decomposition of $\widehat{\mathbf{M}}$;
- `V_est`, the component V of the decomposition of $\widehat{\mathbf{M}}$;
- `contrast`, the squared Frobenius norm of `Data_Series - M_est`. If k and τ are fixed, the contrast is an unique value; if k is selected and τ is fixed or if τ is selected and k is fixed, the contrast is a vector containing the norms for each visiting values of k or τ respectively; and if k and τ are selected, the contrast is a matrix with $k_{\max}$ rows and $\tau_{\max}$ columns such that $\text{contrast}_{k,\tau} = \|\text{Data\_Series} - \widehat{\mathbf{M}}_{k,\tau}\|_{\mathcal{F}}^2$.

5.1.1 Model selection

The selection of k or/and τ is requested using the options `k_select` or/and `tau_select` that are booleans. When there is no selection, the option is set to `FALSE` and `k_max` $= k$ or/and `tau_max` $= \tau$. Note that if no trend is considered in the estimation procedure, $\tau = n$, otherwise `tau_max` must be a smaller than n and larger than `k_max` $+ 2$ in order to ensure that the rank of $\mathbf{M}$ is k.

5.1.2 Taking the trend into account

Let us give more details about the different arguments of `TrendTM` that need to be specified when accounting for a temporal structure in the estimation procedure.

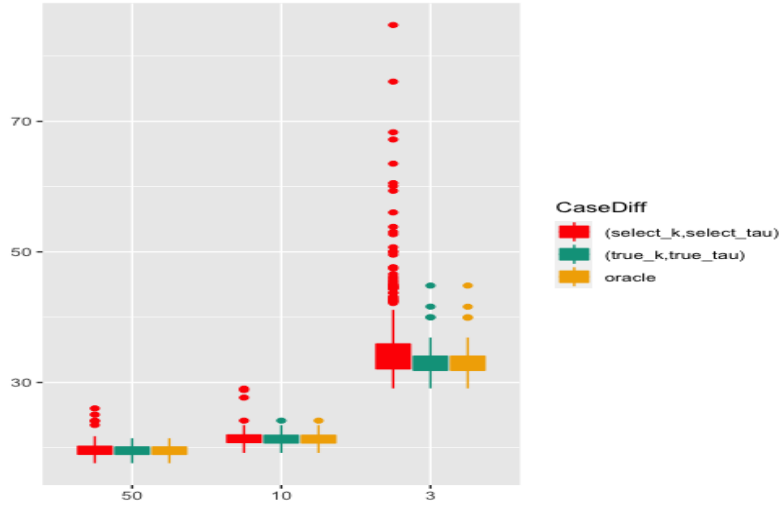

Fig. 7 Boxplot of $\|\mathbf{M} - \widehat{\mathbf{M}}_{k,\tau}\|_{\mathcal{F}}$ with $(k,\tau) = (\widehat{k},\widehat{\tau})$ the selected k and τ, $(k,\tau) = (k,\tau)$ the true values and $(k,\tau) = (\widetilde{k},\widetilde{\tau})$ the oracle for different values of ν (Study 5).

Two temporal structures are considered: periodic trend and smooth trend. This can be specified using the option `struct_temp`, `struct_temp="periodic"` or `struct_temp="smooth"` respectively. Recall that the selection of τ is only possible when a smooth trend is considered. Thus, when

- `struct_temp="periodic"`, then `tau_select=FALSE` and `tau_max`$= \tau$. In this case, τ must be such that n is a multiple of τ;
- `struct_temp="smooth"`, then `tau_select` is either `FALSE` or `TRUE`. Whatever this choice, `tau_max` must be an odd number.

When no trend is taken into account, `struct_temp="none"` and `tau_max` $= n$.

5.1.3 Estimation of M, k and τ being fixed (Step 1)

The least squares estimator $\widehat{\mathbf{M}}_{k,\tau}$ of $\mathbf{M}$, given by (6), is obtained by using the `softImpute` function from the R package of the same name developed for matrix completion by Hastie and Mazumder (2015). In this package, two algorithms are implemented: 'svd' and 'als'. In a simulation study, we observed that they have both provided the same accuracy of the estimator (results not shown). We decide to use the 'als' algorithm (als for Alternating Least Squares) by default but the choice is left free to the user in our package `TrendTM`. In this package, this choice is specified using the option `type_soft`.

5.1.4 The Slope heuristic

Let us now focus on the selection problem of the rank k, and write the penalty as $\text{pen}(k) = \mathfrak{c}_{\text{cal},k}\varphi(k)$. The Slope heuristic, proposed by Birgé and Massart (2001), consists in estimating the slope $\widehat{s}$ of the contrast $\|\mathbf{X} - \widehat{\mathbf{M}}_{k,n}\|_{\mathcal{F}}^2$ as a function of $\varphi(k)$ with k 'large enough' and defining $\mathfrak{c}_{\text{cal},k} = -2\widehat{s}$. The implementation of this heuristic requires the choice of the dimensions on which to perform the regression, that can be difficult in practice. To deal with this problem, Baudry et al. (2012) proposed to make robust regressions for dimensions between k and $k_{\max}$ for $k = 1, 2, \ldots,$ resulting in different selected $\widehat{k}$. The choice of the final dimension is the maximal value $\widehat{k}$ such that the length of successive same $\widehat{k}$ is greater than the option point of the function DDSE of their R package `capushe`. In order to avoid some implementation problems as such condition is not reached and no k is selected, we decide to take the value $\widehat{k}$ associated to the maximal length of successive same $\widehat{k}$.

5.2 Application to pollution dataset

Let us use the package on a real dataset [1]. The dataset contains :

- The date in the (DD/MM/YYYY) format,
- The time in the (HH.MM.SS) format,
- The hourly average concentration of 10 toxic gases in the air : CO, PT08.S1, NMHC, C6H6, PT08.S2, NOx, PT08.S3, NO2, PT08.S4 and PT08.S5,
- The temperature in °C,
- The relative humidity (RH) in %,
- The absolute humidity (AH).

The concentration of the 10 toxic gases, the temperature and the relative and absolute humidity have been recorded $n = 9357$ times during one year. We do a first step of data imputation using the function `complete` of the package `softImpute` since missing values (coded with -200) exist in this dataset (see Alquier et al. (2022) for more details on high-dimensional time series completion).

Our procedure selects $\widehat{k} = 7$ and $\widehat{\tau} = 13$. Figure 8 shows the obtained trend estimation for the 13 toxic gases. The denoising process seems to have been well applied to the data.

6 Link to PCA and clustering

In this section, we give a geometrical interpretation of the results when the factorization problem is solved via a svd using the link to PCA. We also propose a simple method for the clustering of high-dimensional time series based on their projections on the resulting subspace. This method is compared to different existing methods described and compared in Jacques and Preda (2014) on the benchmark ECG dataset.

[1] available at https://archive.ics.uci.edu/ml/datasets/Air+quality

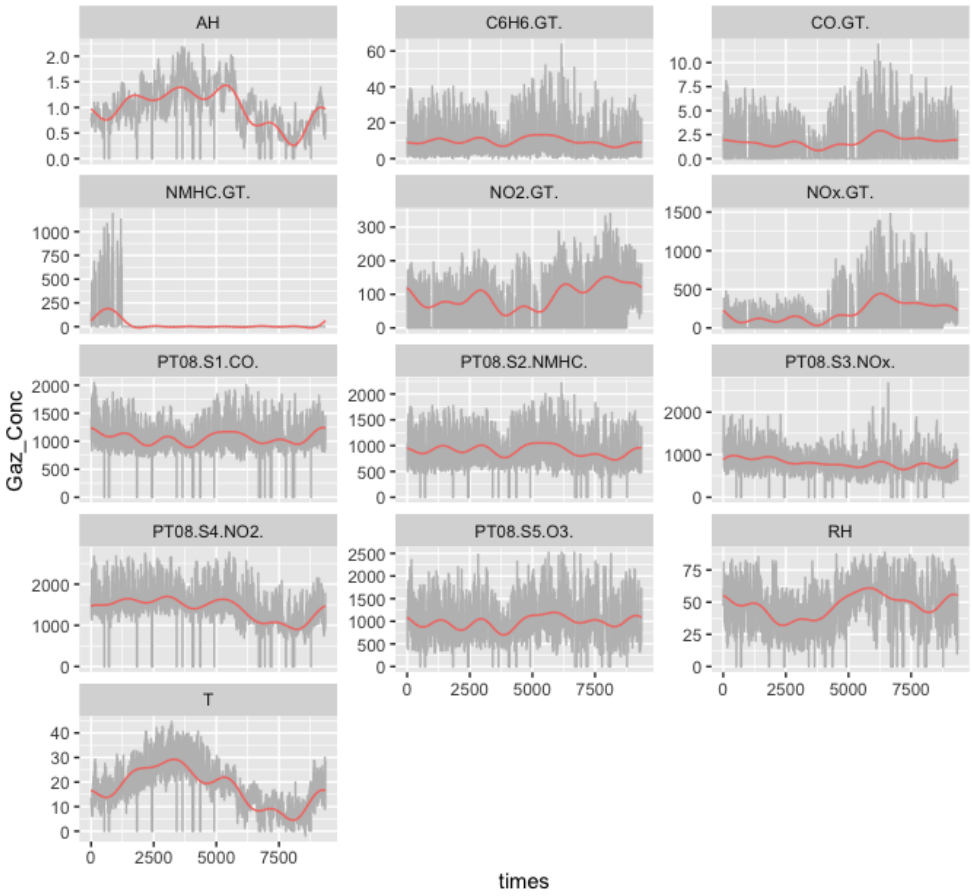

Fig. 8 Data (in grey) and trend estimation (in red) for the 13 toxic gases (red).

The ECG dataset.

The ECG dataset (taken from the UCR Time Series Classification and Clustering website) consists in 200 electrocardiograms from 2 groups of patients sampled at 96 time instants in which 133 are classified as normal and 67 as abnormal based on medical considerations. The time series are plotted in Figure 9.

Geometrical interpretation of the factorization results.

Recall that the PCA problem is solved using a svd (Singular Value Decomposition) leading to a matrix factorization. The svd solution is unique and generates orthogonal factors allowing graphical representations. Our framework without considering the temporal trend ($\mathbf{\Lambda} = \mathbf{I}$) and by solving the optimization problem (3) with svd is thus equivalent to a PCA, up to a centering of the matrix $\mathbf{X}$. The rank-k svd solution provides $\widehat{\mathbf{U}}_k$ and $\widehat{\mathbf{V}}_k$ such

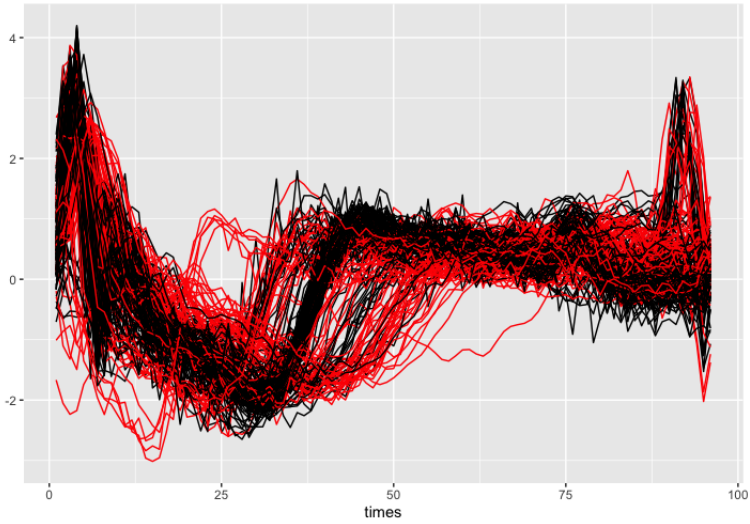

Fig. 9 The ECG dataset (black: normal, red: abnormal).

that $\widehat{\mathbf{U}}_k = \mathbf{X}\widehat{\mathbf{V}}_k^*$ where $\widehat{\mathbf{V}}_k^*$ is an othogonal matrix and the estimator of $\mathbf{M}$ with rank k, i.e. the solution of (3), is $\widehat{\mathbf{M}}_k = \widehat{\mathbf{U}}_k\widehat{\mathbf{V}}_k$. Consequently, the d lines of the matrix $\widehat{\mathbf{U}}_k$ contains the coordinates of the projection of the d series on the first k axes, i.e. in the basis given by the k lines of the matrix $\widehat{\mathbf{V}}_k$. When a temporal structure is taken into account, as for example a τ-periodic, $\mathbf{M}$ is estimated via the estimation of $\underline{\mathbf{M}}$ resulting from a PCA on the transformed data $\underline{\mathbf{X}} = \mathbf{X}\mathbf{\Lambda}^+$. Indeed, the estimator of $\mathbf{M}$ is $\widehat{\mathbf{M}}_{k,\tau} = \widehat{\underline{\mathbf{M}}}_{k,\tau}\mathbf{\Lambda}$ where $\widehat{\underline{\mathbf{M}}}_{k,\tau} = \widehat{\mathbf{U}}_k\widehat{\mathbf{V}}_{k,\tau}\mathbf{\Lambda}$ is the solution of (6) and thus the lines of $\widehat{\mathbf{U}}_k$ contains the coordinates of the transformed time series ($\underline{\mathbf{X}}$) in the basis defined by the lines of $\widehat{\mathbf{V}}_{k,\tau}$.

Some remarks:

- this interpretation is only possible if the factorization is solved by svd. For example, this is no longer the case when the other classical NMF (Non Negativ Matrix Factorization) method is used;
- our work provides a criterion, theoretically performant, for the choice of the rank k, i.e. for the number of axes in PCA, usually chosen using empirical criteria;
- the svd requires the calculation of eigenvalues of a matrix which is numerically tedious when the dimension of the problem is very large as in our framework. In this case, the svd is performed using the R package `softImpute`;
- in a simple PCA, two projected time series are close if they share globally the same trend's property. However, in a high-dimensional space (n high), the euclidean distance used in PCA can lose its meaning and a local trend

similarity could be preferred as by using the temporal structure of the series.

To illustrate the effect of the trend reduction (using $\mathbf{\Lambda}^+$ on a period with length τ), the PCA on the raw ECGs series and on the transformed ECGs series are represented in Figure 10, respectively. The representation is given only on the two first axes (thus with $k = 2$) and the series are colored according to the normal (black) and abnormal (red) status. For the transformed data, we considered that the temporal trend is periodic with period $\tau = 32$. The PCA on the raw data, called the raw PCA, is very structured with respect to the time. Let us consider the 28th and the 121th time series. These series are represented in Figure 11 on the left in their raw version and after the temporal transformation on the right (called the transformed PCA). As we can observed, the raw series differ quite strongly at the beginning and at the end of time, which explains why they are quite far away on the principal components of the raw PCA. This difference is largely attenuated by the smoothing carried out by the transformation and they are close in the transformed PCA. We also compared to a Sparse PCA using the function SPC of the R package `PMA` (see Figure 10). The projection is different from the transformed PCA and in particular the two previous series are no longer close or as far away as in the raw PCA.

Clustering of the d times series.

When dealing with high-dimensional data, a large category of methods proposed in the literature proceed in two steps: first we reduce the dimension of the data and second we perform the clustering (see Jacques and Preda (2014) for a review). Following this line, we propose here a very simple method which consists in applying a classical clustering method (that is here the well-known Hierarchical Agglomerative Clustering with the Ward's linkage) to the projected series, i.e. on the rows of $\widehat{\mathbf{U}}_k$. We apply this strategy on the ECG dataset for a fixed number of groups to 2 since we want to compare our results to those given in Jacques and Preda (2014) with different considerations: (1) with and without taking into account for a trend (here a periodic trend with $\tau = 32$) and (2) for $k = 2$ and for a selected number k.

The Correct Classification Rates (CCR), that is the quality criterion used by the authors, according to the known partitions are given in Table 5. We also report the CCR obtained for the best method among those tested in Jacques and Preda (2014). In addition, we indicate the time taken by the different methods on a laptop 1.6 GHz CPU. We observe that accounting for the trend improves significantly the clustering performances, which are the same compared to the best clustering method. However, our proposed strategy is clearly much more faster. We apply also the clustering on the sparse PCA result and obtained a CCR of 71 that is lower compared to both the raw and the transformed PCA.

Fig. 10 Top: PCA on $\mathbf{X}$. Middle: PCA on $\underline{\mathbf{X}}$ with a periodic trend ($\tau = 32$). Bottom: Sparse PCA for the ECGs series.

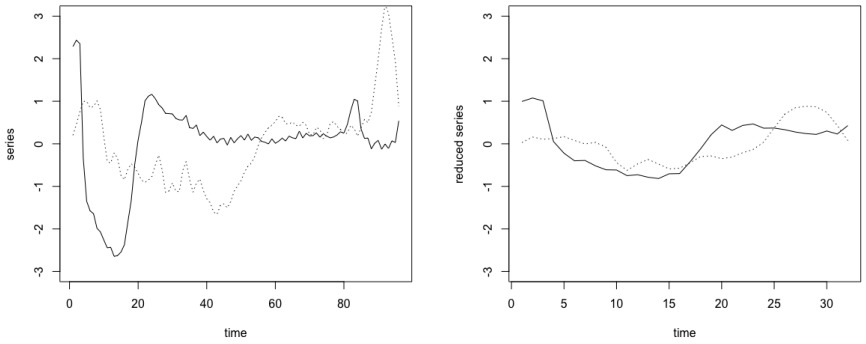

Fig. 11 The 28th (dotted line) and the 121th (solid line) time series. Left: the raw series. Right: the transformed series with a periodic trend ($\tau = 32$).

	$\text{proc}_{\text{NoTrend}}$	$\text{proc}_{\text{Trend}}$ periodic with $\tau = 32$	Best method in Jacques and Preda (2014): Funclust
CCR	74.5 ($k = 2$)	83.5 ($k = 2$)	84 (see Jacques and Preda (2014))
mean times in second (on 30 runs)	0.012	0.0033	19.2
CCR	75.5 ($\widehat{k} = 4$)	80.5 ($\widehat{k} = 6$)	
mean times in second (on 30 runs)	0.16	0.0392	

Table 5 Correct classification rates (CCR) in percentage accounting or not for a trend on the ECG dataset and with a selection or not for k. Mean times in second obtained on 30 runs.

Note that for the ECG dataset, when a trend is considered with $\tau = 32$ and $\widehat{k} = 6$, 3 groups is prefered according to the NbClust R package (chosen by the majority of model selection methods included in this package). Note that, in Figure 12, the series of the ECG dataset are plotted on the left and their projections, colored according to the 3 groups, are provided on the right.

7 Conclusion

The penalized criterion developed in Alquier and Marie (2019) for high-dimensional time series analysis consists in selecting both the rank k of the matrix and the parameter τ related to the temporal structure. The penalty function involves a constant to be calibrated and depends on the temporal structure through its associated parameters to be estimated. For such minimization contrast estimation context, it is well-known in the literature that

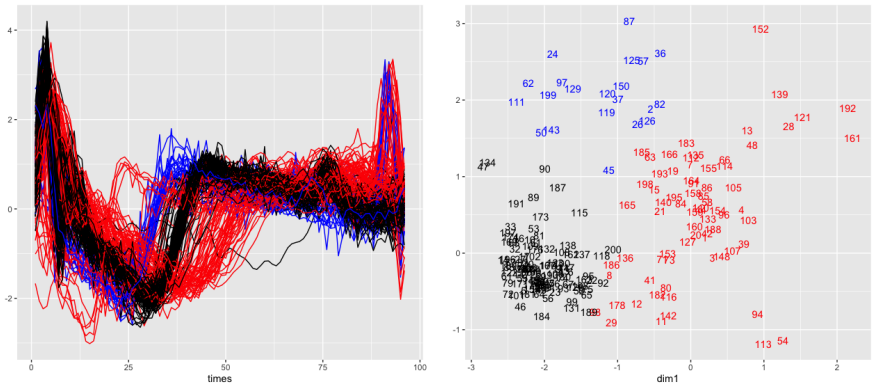

Fig. 12 Clustering of the ECG series with a periodic trend with $\tau = 32$, $\widehat{k} = 6$ and 3 groups. Left: the series and right: the PCA result, colored according to the groups.

despite the selection of both parameters issue that is not standard, the calibration of penalty constant is not an easy task and many heuristics have been proposed to this aim. We proposed in this paper a two-stage strategy based on a popular heuristic: the slope heuristic proposed by Birgé and Massart (2001) and used in many statistical problems. We conducted a large simulation study to compare different heuristics as well as to study the performance of the method. In particular, through these simulations, we show that the joint heuristic performs well: the true values of k and τ are retrieved or underestimated when the estimation problem is more difficult, but with good reasons (the estimation is better than with the true values in this case). Moreover, whatever all the tested cases, the performance of the final estimator is comparable to that of the oracle. The method has been implemented in the R package `TrendTM` yet available on the CRAN and which is detailed in this paper. We also give a geometrical interpretation of the factorization results in the case of using a svd method for solving the optimization problem and propose a simple clustering method of multiple curves. On a benchmark dataset, we observed that this simple method works as well as the best ones proposed in the literature but is computationally much faster. Moreover, we show that accounting for the tendency in this dataset improve the clustering.
Our model assumes that the time series are independent. In some applications, this assumption is not realistic and a perpective of this work will be to take into account a between-series dependence.

8 Appendix

We give here details on the heuristics for the calibration of the penalty constant. Let us consider the model selection problem of selecting a parameter k in the set $\{1, \ldots, k_{\max}\}$ by minimizing a general penalized contrast:

$$\hat{k}(\beta) \in \arg\min_{k} C(k) + \beta \ \mathrm{pen(k)},$$

where β is the unknown penalty constant, C and pen are the contrast and penalty function, respectively. The two well known heuristics for the calibration of one penalty constant are the following:

- the one proposed by Lavielle (2005), called **ML** in our paper. The idea of this heuristic is to select the dimension for which the curve $C(k)$ w.r.t. pen(k) ceases to decrease significantly, i.e. to look of a break in the slope of this curve. The author thus proposed the following automatic procedure:

$$\hat{k} = \max_k \{k \in \{1, \dots, k_{\max}\} \mid D(k) < S\}$$

  where $D(k)$ is the second derivative of the previous curve and S is a threshold to be fixed.
- the 'slope heuristic' proposed by Birgé and Massart (2001). The idea of this heuristic is based on two theoretical facts: first there exists a minimal penalty such that when the penalty is smaller then $\hat{k}$ is huge leading to overfitting, but when the penalty is larger then $\hat{k}$ is reasonable. Two data-driven algorithms have been proposed to search for this minimal penalty (see the huge paper dedicated to this heuristic Arlot and Massart (2009)):
  - **Slope:** this heuristic consists in estimating the slope β_s of $C(k)$ as a function of pen(k) for 'large' k and to consider $\hat{k}(\hat{\beta})$ where $\hat{\beta} = -2\beta_s$.
  - **Biggest Jump (BJ):** this heursitic consists in taking the constant β_{bg} associated to the biggest jump in the curve $\beta \mapsto \hat{k}(\beta)$ and to consider $\hat{k}(\hat{\beta})$ where $\hat{\beta} = 2\beta_{bg}$.

Statements and Declarations

- **Funding:**
  - The authors did not receive support from any organization for the submitted work.
  - No funding was received to assist with the preparation of this manuscript.
  - No funding was received for conducting this study.
  - No funds, grants, or other support was received.

- **Conflict of interest/Competing interests (check journal-specific guidelines for which heading to use):**
  - The authors have no relevant financial or non-financial interests to disclose.
  - The authors have no competing interests to declare that are relevant to the content of this article.
  - All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.
  - The authors have no financial or proprietary interests in any material discussed in this article.

- **Ethics approval:** The authors have respected the ethical standards.
- **Consent for publication:** The authors consent.
- **Availability of data and materials:** All the data used in this paper are available either on websites (the adresses are given in the paper) or in a R package (the name is given in the paper).
- **Code availability:** The developed R package TrendTM is available in the CRAN and the codes for some applications are given in the paper.
- **Authors contributions:** All the authors contributed equally to this work.

Acknowledgments. This research has been conducted within the FP2M federation (CNRS FR 2036).

References

Alquier, P., Bertin, K., Doukhan, P., and Garnier, R. (2020). High-dimensional var with low-rank transition. *Statistics and Computing*, 30(4):1139–1153.

Alquier, P. and Marie, N. (2019). Matrix factorization for multivariate time series analysis. *Electronic journal of statistics*, 13(2):4346–4366.

Alquier, P., Marie, N., and Rosier, A. (2022). Tight risk bound for high dimensional time series completion. *Electronic Journal of Statistics*, 16(1):3001–3035.

Arlot, S. (2019). Minimal penalties and the slope heuristics: a survey. *Journal de la société française de statistique*, 160(3):1–106.

Arlot, S. and Massart, P. (2009). Data-driven calibration of penalties for least-squares regression. *Journal of Machine learning research*, 10(2).

Baudry, J.-P., Maugis, C., and Michel, B. (2012). Slope heuristics: overview and implementation. *Statistics and Computing*, 22(2):455–470.

Birgé, L. and Massart, P. (2001). Gaussian model selection. *Journal of the European Mathematical Society*, 3(3):203–268.

Cai, T. T. and Zhang, A. (2015). Rop: Matrix recovery via rank-one projections. *The Annals of Statistics*, 43(1):102–138.

Candes, E. J., Sing-Long, C. A., and Trzasko, J. D. (2013). Unbiased risk estimates for singular value thresholding and spectral estimators. *IEEE transactions on signal processing*, 61(19):4643–4657.

Collilieux, X., Lebarbier, E., and Robin, S. (2019). A factor model approach for the joint segmentation with between-series correlation. *Scandinavian Journal of Statistics*, 46(3):686–705.

Davis, R. A., Zang, P., and Zheng, T. (2016). Sparse vector autoregressive modeling. *Journal of Computational and Graphical Statistics*, 25(4):1077–1096.

Devijver, E., Gallopin, M., and Perthame, E. (2017). Nonlinear network-based quantitative trait prediction from transcriptomic data. *arXiv preprint arXiv:1701.07899*.

dos S. Dias, C. T. and Krzanowski, W. J. (2003). Model selection and cross validation in additive main effect and multiplicative interaction models. *Crop Science*, 43(3):865–873.

Gao, Z. and Tsay, R. S. (2022). Modeling high-dimensional time series: A factor model with dynamically dependent factors and diverging eigenvalues. *Journal of the American Statistical Association*, 117(539):1398–1414.

Gourieroux, C. and Monfort, A. (1997). *Time Series and Dynamic Models*. Cambridge University Press.

Hastie, T. and Mazumder, R. (2015). softimpute: Matrix completion via iterative soft-thresholded svd. *R package version*, 1:p1.

Jacques, J. and Preda, C. (2014). Functional data clustering: a survey. *Advances in Data Analysis and Classification*, 8(3):231–255.

Klopp, O., Lounici, K., and Tsybakov, A. B. (2017). Robust matrix completion. *Probability Theory and Related Fields*, 169(1):523–564.

Klopp, O., Lu, Y., Tsybakov, A. B., and Zhou, H. H. (2019). Structured matrix estimation and completion. *Bernoulli*, 25(4B):3883–3911.

Koltchinskii, V., Lounici, K., and Tsybakov, A. B. (2011). Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion. *The Annals of Statistics*, 39(5):2302–2329.

Lam, C. and Yao, Q. (2012a). Factor modeling for high-dimensional time series: inference for the number of factors. *The Annals of Statistics*, pages 694–726.

Lam, C. and Yao, Q. (2012b). Factor modeling for high-dimensional time series: inference for the number of factors. *The Annals of Statistics*, pages 694–726.

Lavielle, M. (2005). Using penalized contrasts for the change-point problem. *Signal processing*, 85(8):1501–1510.

Lopes, H. F. and West, M. (2004). Bayesian model assessment in factor analysis. *Statist. Sinica*, 14(1):41–68.

Lütkepohl, H. (2005). *New introduction to multiple time series analysis*. Springer Science & Business Media.

Moridomi, K.-i., Hatano, K., and Takimoto, E. (2018). Tighter generalization bounds for matrix completion via factorization into constrained matrices. *IEICE TRANSACTIONS on Information and Systems*, 101(8):1997–2004.

Seghouane, A.-K. and Cichocki, A. (2007). Bayesian estimation of the number of principal components. *Signal Processing*, 87(3):562–568.

Shojaie, A. and Michailidis, G. (2010). Discovering graphical granger causality using the truncating lasso penalty. *Bioinformatics*, 26(18):i517–i523.

Ulfarsson, M. O. and Solo, V. (2008). Dimension estimation in noisy pca with sure and random matrix theory. *IEEE transactions on signal processing*, 56(12):5804–5816.

Vershynin, R. (2012). Introduction to the non-asymptotic analysis of random matrices. *Compressed Sensing Theory and Applications*, pages 210–268.

Supplemental materiel

We give here a large majority of our codes.

Code for the simulations

The code of Study 3 is the following:

```
# Used packages  ----------------------------

library("TrendTM")
library("tidyverse")
library("fda")
library("wesanderson")


# Used function  ----------------------------

Simulated_Series <- function(d, n, k, tau, sUV, s0) {
  U <- matrix(rnorm(d * k, 0, sUV), d, k)
  V <- matrix(rnorm(k * tau, 0, sUV), k, tau)
  times_eval <- seq(1 / n, 1, by = 1 / n)
  fbasis_obj <- create.fourier.basis(rangeval = c(0, 1), nbasis = tau, period = 1)
  fbasis_evals <- eval.basis(times_eval, fbasis_obj)
  Lambda <- t(fbasis_evals)
  M <- U %*% V %*% Lambda
  s2 <- sqrt(tau) * s0
  E <- matrix(rnorm(d * n, 0, s2), d, n)
  X <- M + E
  return(X)
}



# Used parameters  ----------------------------

d <- 100
n <- 600
sUV <- 0.5
k_true <- 3
tau_true <- 25
k_select <- TRUE
k_max <- 15
tau_select <- TRUE
tau_max <- 55
type_soft <- "als"
struct_temp <- "smooth"

seq_s0 <- c(0.1, 0.5, 1.5, 2)
NbSim <- 10

# Simulation and Estimaiton  ----------------------------

TrendTM_For_Study3 <- purrr::map(seq_s0, ~ {
  Res_Total_Sim <- matrix(0, ncol = 6, nrow = NbSim)

  for (i in 1:NbSim) {
    Data_Series <- Simulated_Series(d, n, k_true, tau_true, sUV, .x)
    # truth
    NormeF_true <- TrendTM(Data_Series,
      k_max = k_true,
      struct_temp = struct_temp, tau_max = tau_true
    )$contrast
    # estimated
    Res_TrendTM_k_tau <- TrendTM(Data_Series,
```

```
      k_max = k_max, k_select = k_select,
      struct_temp = struct_temp, tau_select = tau_select,
      tau_max = tau_max
    )
    k_est <- Res_TrendTM_k_tau$k_est
    tau_est <- Res_TrendTM_k_tau$tau_est

    Res_Total_Sim[i, ] <- c(
     k_true, tau_true, NormeF_true, k_est,
     tau_est, Res_TrendTM_k_tau$contrast[k_est, which(colnames(Res_TrendTM_k_tau$contrast) %in% tau_est)]
    )
  }

  Res_Total_Sim <- Res_Total_Sim %>%
    as.data.frame()


  return(Res_Total_Sim)
})

TrendTM_For_Study3_Org <- TrendTM_For_Study3 %>%
  bind_rows()
colnames(TrendTM_For_Study3_Org) <- c(rep(c("k", "tau", "norme"), 2))

Study3 <- bind_rows(TrendTM_For_Study3_Org[, 1:3], TrendTM_For_Study3_Org[, 4:6]) %>%
  mutate(CaseDiff = rep(c("(true_k,true_tau)", "(select_k,select_tau)"), each = 4 * NbSim)) %>%
  mutate(Case = rep(rep(c("Easy", "Medium", "Difficult", "Hard"), each = NbSim), 2))
Study3$Case <- fct_relevel(Study3$Case, c("Easy", "Medium", "Difficult", "Hard"))


# Plot result  ----------------------------

ggplot(Study3, aes(x = Case, y = sqrt(norme), col = CaseDiff, fill = CaseDiff)) +
  geom_boxplot() +
  scale_fill_manual(values = wes_palette("Darjeeling1", n = 3)) +
  scale_color_manual(values = wes_palette("Darjeeling1", n = 3)) +
  xlab("") +
  ylab("")

# Summarise results  ----------------------------


Selected_k_tau_Mean_SD <- Study3 %>%
  filter(CaseDiff == "(select_k,select_tau)") %>%
  group_by(Case) %>%
  summarise(
    moy_selected_k = mean(k), sd_selected_k = sd(k),
    moy_selected_tau = mean(tau), sd_selected_tau = sd(tau)
  )
```

Code for the application

The code for the application on the pollution dataset presented in Section 5.2 is the following:

```
# Used packages  ----------------------------

library("softImpute")
library("TrendTM")
library("tidyverse")

# Load data ----------------------------

AirPollution <- read.table("AirQualityUCI.csv", sep = ";", header = TRUE, dec = ",") %>%
  dplyr::select(-Date, -Time) %>%
  mutate_all(., ~ replace(., which(. == -200), 0))
```

```
Names_Pol <- colnames(AirPollution)

AirPollution_Data <- AirPollution %>%
  as.matrix() %>%
  t()
AirPollution_Imp <- softImpute::complete(
  AirPollution_Data,
  softImpute(AirPollution_Data, rank = 1, lambda = 0)
)

# Trend estimation ---------------------------

AirPollution_Trend <- TrendTM(AirPollution_Imp,
  k_select = TRUE, k_max = 13,
  struct_temp = "smooth",
  tau_select = TRUE, tau_max = 101
)

# Plot data and results ----------------------------

times <- rep(1:dim(AirPollution)[1], dim(AirPollution)[2])
AirPollution_Struc <- AirPollution %>%
  gather(key = "Gaz", value = "Gaz_Conc") %>%
  mutate(times = times)
Tendency_Struc <- t(AirPollution_Trend$M_est) %>%
  as.data.frame() %>%
  rename_all(funs(c(Names_Pol))) %>%
  gather(key = "Gaz", value = "Gaz_Conc_Tendency") %>%
  mutate(times = times)
Data_Res <- left_join(AirPollution_Struc, Tendency_Struc)


ggplot(Data_Res, aes(x = times, y = Gaz_Conc)) +
  geom_line(color = "grey") +
  facet_wrap(facets = ~ as.factor(Gaz), scales = "free_y", ncol = 3) +
  geom_line(aes(x = times, y = Gaz_Conc_Tendency, color = "red")) +
  theme(legend.position = "none")
```

Code for the PCA and clustering

The code of the study presented in Section 6 is the following:

```
# Used packages  ----------------------------

library("TrendTM")
library("tidyverse")
library("FactoMineR")
library("factoextra")
library("PMA")
library("reshape2")
library("NbClust")


# Load data ----------------------------

TRAIN <- read.delim("ECG200_TRAIN", sep = ",", header = FALSE) %>%
  rename_all(funs(c("label", paste("T", 1:96, sep = ""))))
TEST <- read.delim("ECG200_TEST", sep = ",", header = FALSE) %>%
  rename_all(funs(c("label", paste("T", 1:96, sep = ""))))

ECG <- bind_rows(TRAIN, TEST)
Group <- ECG %>%
  mutate(Group = replace(label, which(label == -1), 0)) %>%
  dplyr::select(Group) %>%
  as.matrix()

ECG <- ECG %>%
```

```
  dplyr::select(-label) %>%
  as.matrix()
n <- ncol(ECG)
d <- nrow(ECG)


# Plot data  ---------------------------

ECG_For_Plot <- ECG %>%
  t() %>%
  melt(.) %>%
  mutate(times = rep(c(1:n), d), Group = rep(Group, each = n))

ggplot(ECG_For_Plot, aes(x = times, y = value, group = Var2, col = as.factor(Group))) +
  geom_line() +
  theme(legend.position = "none") +
  ylab("") +
  scale_color_manual(values = c("red", "black"))

# Geometrical interpretation ---------------------------

Raw_ECG_k2_SVD <- TrendTM(ECG, k_max = 2, type_soft = "svd")$U_est %>%
  as.data.frame() %>%
  rename(dim1 = V1, dim2 = V2)
ggplot(Raw_ECG_k2_SVD, aes(x = dim1, y = dim2, col = as.factor(Group))) +
  geom_text(label = as.character(c(1:d))) +
  theme(legend.position = "none") +
  ylab("") +
  scale_color_manual(values = c("red", "black"))

Transf_ECG_k2_SVD <- TrendTM(ECG,
  k_max = 2,
  tau_max = 32, struct_temp = "periodic",
  type_soft = "svd"
)$U_est %>%
  as.data.frame() %>%
  rename(dim1 = V1, dim2 = V2)
ggplot(Transf_ECG_k2_SVD, aes(x = dim1, y = dim2, col = as.factor(Group))) +
  geom_text(label = as.character(c(1:d))) +
  theme(legend.position = "none") +
  ylab("") +
  scale_color_manual(values = c("red", "black"))

Sparse_ECG_k2_SVD_out <- SPC(ECG, sumabsv = 1, K = 2, orth = TRUE, niter = 20, trace = FALSE)
Sparse_ECG_k2_SVD <- Sparse_ECG_k2_SVD_out$u %*% (diag(Sparse_ECG_k2_SVD_out$d)) %>%
  as.data.frame() %>%
  rename(dim1 = V1, dim2 = V2)
ggplot(Sparse_ECG_k2_SVD, aes(x = dim1, y = dim2, col = as.factor(Group))) +
  geom_text(label = as.character(c(1:d))) +
  theme(legend.position = "none") +
  ylab("") +
  scale_color_manual(values = c("red", "black"))


# Clustering ---------------------------

Clustering_funct <- function(Coord, P) {
  CAH_Ward <- Coord %>%
    dist(., method = "euclidean") %>%
    .^2 %>%
    hclust(., method = "ward.D")
  return(cutree(CAH_Ward, k = P))
}

Group <- Group + 1
# Fixed P
P <- 2
```

```
ECG_Clust_k2_WO_Temp <- TrendTM(ECG, k_max = 2, tau_max = n, type_soft = "svd")
cluster_k2_WO_Temp_P2 <- Clustering_funct(ECG_Clust_k2_WO_Temp$U_est, P)
sum(cluster_k2_WO_Temp_P2 == Group) / d * 100

ECG_Clust_k2_WITH_Temp <- TrendTM(ECG,
  k_max = 2,
  tau_max = 32, struct_temp = "periodic",
  type_soft = "svd"
)
cluster_k2_WITH_Temp_P2 <- Clustering_funct(ECG_Clust_k2_WITH_Temp$U_est, P)
sum(cluster_k2_WITH_Temp_P2 == Group) / d * 100



ECG_Clust_kselect_WO_Temp <- TrendTM(ECG,
  k_max = 10, k_select = TRUE,
  tau_max = n, type_soft = "svd"
)
cluster_kselect_WO_Temp_P2 <- Clustering_funct(ECG_Clust_kselect_WO_Temp$U_est, P)
sum(cluster_kselect_WO_Temp_P2 == Group) / d * 100

ECG_Clust_kselect_WITH_Temp <- TrendTM(ECG,
  k_max = 10, k_select = TRUE,
  tau_max = 32, struct_temp = "periodic",
  type_soft = "svd"
)
cluster_kselect_WITH_Temp_P2 <- Clustering_funct(ECG_Clust_kselect_WITH_Temp$U_est, P)
sum(cluster_kselect_WITH_Temp_P2 == Group) / d * 100

# Unknown P
Choice_Nbclust <- NbClust(
  data = ECG, distance = "euclidean",
  method = "ward.D2", index = "all",
  min.nc = 2, max.nc = 16
)$Best.nc %>%
  t() %>%
  as.data.frame()
ggplot(Choice_Nbclust, aes(x = as.factor(Number_clusters))) +
  geom_bar()

cluster_kselect_WITH_Temp_P3 <- Clustering_funct(ECG_Clust_kselect_WITH_Temp$U_est, P = 3)

ECG_For_Plot <- ECG_For_Plot %>% mutate(ClusterP3 = rep(cluster_kselect_WITH_Temp_P3, each = n))
ggplot(ECG_For_Plot, aes(x = times, y = value, group = Var2, col = as.factor(ClusterP3))) +
  geom_line() +
  theme(legend.position = "none") +
  ylab("") +
  scale_color_manual(values = c("red", "blue", "black"))

ggplot(Transf_ECG_k2_SVD, aes(x = dim1, y = dim2, col = as.factor(cluster_kselect_WITH_Temp_P3))) +
  geom_text(label = as.character(c(1:d))) +
  theme(legend.position = "none") +
  ylab("") +
  scale_color_manual(values = c("red", "blue", "black"))
```