Stochastic algorithms for solving structured low-rank matrix approximation problems

J. W. Gillard and A. A. Zhigljavsky

Cardiff School of Mathematics
Cardiff University
{GillardJW,ZhigljavskyAA}@Cardiff.ac.uk

Abstract

In this paper, we investigate the complexity of the numerical construction of the Hankel structured low-rank approximation (HSLRA) problem, and develop a family of algorithms to solve this problem. Briefly, HSLRA is the problem of finding the closest (in some pre-defined norm) rank r approximation of a given Hankel matrix, which is also of Hankel structure. We demonstrate that finding optimal solutions of this problem is very hard. For example, we argue that if HSLRA is considered as a problem of estimating parameters of damped sinusoids, then the associated optimization problem is basically unsolvable. We discuss what is known as the orthogonality condition, which solutions to the HSLRA problem should satisfy, and describe how any approximation may be corrected to achieve this orthogonality. Unlike many other methods described in the literature the family of algorithms we propose has the property of guaranteed convergence.

Keywords:
Structured low rank approximation, Hankel matrix, global optimization

1. Introduction

1.1. Statement of the problem

Let L, K and r be given positive integers such that $1 \le r < L \le K$. Denote the set of all real-valued $L \times K$ matrices by $\mathbb{R}^{L\times K}$. Let $\mathcal{M}_r = \mathcal{M}_r^{L\times K} \subset \mathbb{R}^{L\times K}$ be the subset of $\mathbb{R}^{L\times K}$ containing all matrices with rank $\le r$, and $\mathcal{H} = \mathcal{H}^{L\times K} \subset \mathbb{R}^{L\times K}$ be the subset of $\mathbb{R}^{L\times K}$ containing matrices of some known structure. The set of structured $L \times K$ matrices of rank $\le r$ is $\mathcal{A} = \mathcal{M}_r \cap \mathcal{H}$.

Assume we are given a matrix $\mathbf{X}_* \in \mathcal{H}$. The problem of structured low rank approximation (SLRA) is:

$$f(\mathbf{X}) \to \min_{\mathbf{X}\in\mathcal{A}} \tag{1}$$

where $f(\mathbf{X}) = \rho^2(\mathbf{X}, \mathbf{X}_*)$ is a squared distance on $\mathbb{R}^{L\times K} \times \mathbb{R}^{L\times K}$.

In this paper we only consider the case where $\mathcal{H}$ is the set of Hankel matrices and thus refer to (1) as HSLRA. Recall that a matrix $\mathbf{X} = (x_{lk}) \in \mathbb{R}^{L\times K}$ is called Hankel if $x_{lk} = \text{const}$ for all pairs (l, k) such that $l + k = \text{const}$; that is, all elements on the anti-diagonals of $\mathbf{X}$ are equal. There is a one-to-one correspondence between $L \times K$ Hankel matrices and vectors of size $N = L + K - 1$. For a vector $Y = (y_1, \dots, y_N)^T$, the matrix $\mathbf{X} = \mathbb{H}(Y) = (x_{lk}) \in \mathbb{R}^{L\times K}$ with elements $x_{lk} = y_{l+k-1}$ is Hankel and vise-versa: for any matrix $\mathbf{X} \in \mathcal{H}$, we may define $Y = \mathbb{H}^{-1}(\mathbf{X})$ so that $\mathbf{X} = \mathbb{H}(Y)$.

We consider the distances ρ defined by the semi-norms

$$\|\mathbf{A}\|_w^2 = \operatorname{tr} \mathbf{A}\mathbf{W}\mathbf{A}^T \quad (\text{so that } f(\mathbf{X}) = \operatorname{tr}(\mathbf{X} - \mathbf{X}_*)\mathbf{W}(\mathbf{X} - \mathbf{X}_*)^T), \tag{2}$$

where $\mathbf{A} \in \mathbb{R}^{L\times K}$ and $\mathbf{W}$ is a symmetric non-negative definite negative matrix of size $K \times K$. Moreover, in our main application the weight matrix $\mathbf{W}$ is diagonal:

$$\mathbf{W} = \operatorname{diag}(w_1, \dots, w_K), \tag{3}$$

where $w_1, \dots, w_K$ are some positive numbers.

1.2. Background

The aim of low-rank approximation methods is to approximate a matrix containing observed data, by a matrix of pre-specified lower rank r. The rank of the matrix containing the original data can be viewed as the order of complexity required to fit to the data exactly, and a matrix of lower complexity (lower rank) 'close' to the original matrix is often required. A further requirement is that if the original matrix of the observed data is of a particular structure, then the approximation should also have this structure. An example is the HSLRA problem as defined in the previous section.

HSLRA is a very important problem with applications in a number of different areas. In addition to the clear connection with time series analysis and signal processing, HSLRA has been extensively used in system identification (modeling dynamical systems) [1], in speech and audio processing [2], in modal and spectral analysis [3] and image processing [4]. Some discussion on the relationship of HSLRA with some well known subspace-based methods of time series analysis and signal processing is given in [5]. Similar structures used in (1) include Toeplitz, block Hankel and block Toeplitz structures. In image processing, there is much use of Hankel-block Hankel structures. Further details, references and specific applications of SLRA are provided in [6, 7, 8].

1.3. Notation

The following list contains the main notation used in this paper.

N, L, K, r	Positive integers with $1 \le r < L \le K < N$, $N = L + K - 1$
$\mathbb{R}^{L\times K}$	Set of $L \times K$ matrices
$\mathbb{R}^N$	Set of vectors of length N
$\mathcal{H}^{L\times K}$ or $\mathcal{H}$	Set of $L \times K$ Hankel matrices
$\mathcal{M}_r$	Set of $L \times K$ matrices of rank r
$\mathcal{A} = \mathcal{M}_r \cap \mathcal{H}$	Set of $L \times K$ Hankel matrices of rank r
$Y = (y_1, \ldots, y_N)^T$	Vector in $\mathbb{R}^N$
$\mathbb{H}(Y)$	Hankel matrix in $\mathcal{H}^{L\times K}$ associated with vector $Y \in \mathbb{R}^N$
$\mathbf{X}_* \in \mathcal{H}^{L\times K}$	Given matrix
$Y_* = (y_{1*}, \ldots, y_{N*})^T$	Vector in $\mathbb{R}^N$ such that $\mathbb{H}(Y_*) = \mathbf{X}_*$ (vector of observed values)
$\pi_{\mathcal{H}}(\mathbf{X})$	Projection of the matrix $\mathbf{X} \in \mathbb{R}^{L\times K}$ onto the set $\mathcal{H}$
$\pi^{(r)}(\mathbf{X})$	Projection of a matrix $\mathbf{X} \in \mathbb{R}^{L\times K}$ onto the set $\mathcal{M}_r$
$\mathbf{I}_p$	Identity matrix of size $p \times p$.

1.4. Structure of the paper and the main results

The structure of the paper is as follows. In Section 2 we formally define the HSLRA problem (1) as an optimization problem in the space of matrices and introduce a generic norm defining the objective function $f(\cdot)$. In Section 2 we also describe projection operators to $\mathcal{H}$ and especially $\mathcal{M}_r$ that are used throughout the majority of the algorithms introduced in this paper, in the process of solving the HSLRA problem. In Section 3 we study the relations between different norms which define the objective function in two different setups. In Section 3, we also discuss some computational aspects for dealing with infinite and infinitesimal numbers. In Section 4 we study the so-called orthogonality condition which optimal solutions of (1) should satisfy, and describe how an approximation may be corrected to achieve this orthogonality. Section 5 considers some algorithms for solving the HSLRA problem represented as optimization problems in the set of Hankel matrices $\mathcal{H}$. We start with formulating a well-known algorithm based on alternating projections to the spaces $\mathcal{M}_r$ and $\mathcal{H}$, and call this AP. This is followed by an introduction of an improved version of this algorithm which we call 'Orthogonalized Alternating Projections' (OAP). In Section 5.2 we introduce a family of algorithms which incorporate randomization, backtracking, evolution and selection. The algorithms described in this section have guaranteed convergence to the optimum, unlike all other methods described in the literature. The main algorithm introduced and studied in the paper is called APBR (which in an abbreviation for 'Alternating Projections with Backtracking and Randomization'). Examples provided in Section 6 show that APBR significantly outperforms AP, as well as some other methods. In Appendix A, we consider the HSLRA problem (1) by associating matrices $\mathbf{X} \in \mathcal{A}$ with vectors whose elements can be represented as sums of damped sinusoids; this approach is popular in the signal processing literature. We demonstrate that the resulting objective function can be very complex which means that the associated optimization problems are basically unsolvable. Section 7 concludes the paper.

2. HSLRA as an optimization problem

2.1. HSLRA as a problem of parameter estimation in non-linear regression

In the signal processing literature, it is customary to represent the HSLRA as a problem of estimating the parameters of a nonlinear regression model with terms given by damped sinusoids, see for example [9] and [10]. This can be formulated as follows.

Consider a general non-linear regression model where it is assumed that each element of the observed vector Y_* may be written as

$$y_{j*} = y_n(\theta) + \varepsilon_j \quad (j = 1, \ldots, N)\,, \tag{4}$$

where θ is a vector of unknown parameters, $y_j(\theta)$ is a function nonlinear in θ and $\varepsilon_1, \ldots, \varepsilon_N$ is a series of noise terms so that $E\varepsilon_j = 0$ and $E\varepsilon_i\varepsilon_j = 0$ for $i \neq j$.

Parameter estimation in a general (weighted) non-linear regression model is usually reduced to solving the minimization problem

$$F(\theta) = \sum_{j=1}^{N} s_j (y_{j*} - y_j(\theta))^2 \to \min_{\theta}\,. \tag{5}$$

Assuming the variances $\sigma_j^2 = E\varepsilon_j^2$ of ε_j's are known, the weights s_j can naturally be chosen as $s_j = 1/\sigma_j^2$.

The estimator $\hat{\theta}$ is defined as $\hat{\theta} = \arg\min_\theta F(\theta)$. In the case of damped sinusoids, the function $y_n(\theta)$ has the form

$$y_n(\theta) = \sum_{i=1}^{q} a_i \exp(d_i n) \sin(2\pi\omega_i n + \phi_i), \quad n = 1, \ldots, N\,, \tag{6}$$

where $\theta = (a, d, \omega, \phi)$ with $a = (a_1, \ldots, a_q)$, $d = (d_1, \ldots, d_q)$, $\omega = (\omega_1, \ldots, \omega_q)$, and $\phi = (\phi_1, \ldots, \phi_q)$.

The correspondence between q in (6) and r in (1) is as follows: if $\omega_i \neq 0$ then the term $a_i \exp(d_i n) \sin(2\pi\omega_i n + \phi_i)$ in (6) adds 2 to the rank of the corresponding matrix $\mathbf{X} \in \mathcal{A}$ while if $\omega_i = 0$ (so that the term is simply $a_i \exp(d_i n)$) then this term only adds 1 to the rank of this $\mathbf{X} \in \mathcal{A}$. Each vector Y of the form (6) generates a low-rank Hankel matrix $\mathbf{X} = \mathbb{H}(Y)$. Note, however, that the set of vectors $\{Y = \mathbb{H}^{-1}(\mathbf{X}), \mathbf{X} \in \mathcal{A}\}$ is slightly richer than the set of vectors of the form (6) with appropriate values of q and the forms of the terms in this formula; see [8] or [11].

Let $Y_* = (y_{1*}, \ldots, y_{N*})^T$, $Y(\theta) = (y_1(\theta), \ldots, y_N(\theta))^T$ and let $\mathbf{S}$ be the diagonal matrix $\mathbf{S} = \mathrm{diag}(s_1, \ldots, s_N) \in \mathbb{R}^{N\times N}$. Then the objective function in (5) can be written as

$$F(\theta) = (Y_* - Y(\theta))^T \mathbf{S} (Y_* - Y(\theta))\,. \tag{7}$$

The papers [10] and [12] contain discussions about the behaviour of the objective function (5), with $y_j(\theta)$ defined through (6). In [12] the fact that the objective function F is multiextremal has been observed; the function F was decomposed into three different components and it was numerically demonstrated that the part of the objective function with the observation noise removed dominates the shape of the objective function. An extension of this analysis is given in Appendix A.1. Note that up until now, only the weights $s_j = 1$ have been considered in the literature devoted to the optimization problem defined by (5) and (6). Given the form of the objective function (5) there is likely to be much potential for the methodology described in [13] but this is the subject of further work.

This optimization problem is very difficult with the objective function possessing many local minima. The objective function has very large Lipschitz constants which increase with N, the number of observations. Additionally, the number of local minima in the neighbourhood of the global minimum increases linearly in N. Adding noise to the observed data increases the complexity of the objective function and moves the global minimizer away from the true value; for more details see Sections A.1 and A.2 in Appendix A.

2.2. Matrix optimal solution and its approximation

Consider the HSLRA problem (1). Since $0 \le f(\mathbf{X}) < \infty$ for any $\mathbf{X} \in \mathcal{A}$, $f(\mathbf{X})$ is a continuous function on $\mathcal{A}$ and $f(\mathbf{X}) \to \infty$ as $\|\mathbf{X}\| \to \infty$, a solution to (1) always exists. However, the solution is not necessarily unique. Set

$$\mathfrak{X}^* = \{\mathbf{X}^* = \arg\min_{\mathbf{X}\in\mathcal{A}} f(\mathbf{X})\} \quad \text{and} \quad f^* = f(\mathbf{X}^*) = \min_{\mathbf{X}\in\mathcal{A}} f(\mathbf{X})\,.$$

Any algorithm designed to solve the optimization problem (1) should return a matrix $\mathbf{X}_{appr}$ which can be considered as an approximation to one of the solutions $\mathbf{X}^* \in \mathfrak{X}^*$; approximations to the value f^* alone (without approximations to $\mathfrak{X}^*$) are not sufficient. Ideally, the matrix $\mathbf{X}_{appr}$ should belong to the set of matrices $\mathcal{A}$.

A typical optimization algorithm designed for solving the problem (1) could be represented as a procedure which generates a sequence of matrices $\mathbf{X}_0, \mathbf{X}_1, \ldots$ such that some of the matrices $\mathbf{X}_n$ for large n can be considered as approximations to $\mathbf{X}^*$, a solution of (1). This matrix sequence must have at least one limiting point (matrix) in the set $\mathcal{A}$. Denote by $\mathbf{X}_\infty$ the limiting point of the algorithm which has the smallest value of f among all its limiting points belonging to $\mathcal{A}$. In general, $f(\mathbf{X}_\infty) \geq f^*$. The algorithm (theoretically) converges to the optimal solution if $f(\mathbf{X}_\infty) = f^*$; that is, if $\mathbf{X}_\infty \in \mathfrak{X}^*$.

In many optimization algorithms attempting to solve the SLRA problem (1) and operating in matrix spaces, the projection to the spaces $\mathcal{H}$ and $\mathcal{M}_r$ is of prime importance. Let us consider these two projections.

2.3. Projection to $\mathcal{H}$

The space $\mathcal{H} = \mathcal{H}^{L\times K}$ of $L \times K$ Hankel matrices is a linear subspace of $\mathbb{R}^{L\times K}$. The closest Hankel matrix (for a variety of norms and semi-norms including (2)) to any given matrix is obtained by using the diagonal averaging procedure.

Recall that every $L\times K$ Hankel matrix $\mathbf{X} \in \mathcal{H}$ is in a one-to-one correspondence with some vector $Y = (y_1, \ldots, y_N)^T$, with $N = L + K - 1$. This correspondence is described by the function $\mathbb{H} : \mathbb{R}^N \to \mathcal{H}^{L\times K}$ which is defined by $\mathbb{H}(Y) = \|y_{l+k-1}\|_{l,k=1}^{L,K}$ for $Y = (y_1, \ldots, y_N)^T$. Each element of the vector Y is repeated in $\mathbf{X} = \mathbb{H}(Y)$ several times. Let $\mathbf{E} = (e_{lk}) \in \mathbb{R}^{L\times K}$ be the matrix consisting entirely of ones. We can compute the sum of each anti-diagonal of $\mathbf{E}$, denoted t_n, as

$$t_n = \sum_{l+k=n+1} e_{lk} = \begin{cases} n & \text{for } n = 1, \ldots, L-1\,, \\ L & \text{for } n = L, \ldots, K-1, \\ N-n+1 & \text{for } n = K, \ldots, N. \end{cases} \tag{8}$$

The value t_n is the number of times the element y_n of the vector Y is repeated in the Hankel matrix $\mathbb{H}(Y)$.

Let $\pi_{\mathcal{H}}(\mathbf{X})$ denote the projection of $\mathbf{X} \in \mathbb{R}^{L\times K}$ onto the space $\mathcal{H}$. Then the element $\tilde{x}_{ij}$ of $\pi_{\mathcal{H}}(\mathbf{X})$ is given by

$$\tilde{x}_{ij} = t_{i+j-1}^{-1} \sum_{l+k=i+j} x_{lk}\,.$$

The squared distance between matrix $\mathbf{X}$ and the space $\mathcal{H}$ is

$$\rho^2(\mathbf{X}, \mathcal{H}) = \min_{\mathbf{X}'\in\mathcal{H}} \rho^2(\mathbf{X}, \mathbf{X}') = \rho^2(\mathbf{X}, \pi_{\mathcal{H}}(\mathbf{X})) = \|\mathbf{X} - \pi_{\mathcal{H}}(\mathbf{X})\|^2\,.$$

Since projecting to $\mathcal{H}$ is very easy, using this subspace as the feasible domain for the HSLRA problem (1) is more natural than using the original space $\mathbb{R}^{L\times K}$. Based on results of Appendix A we also argue that this leads to more tractable optimization problems than the approach based on the use of the damped sinusoids model.

2.4. Projection to $\mathcal{M}_r$

Unlike $\mathcal{H}$, the set $\mathcal{M}_r$ is clearly not convex. However, the projections from $\mathbb{R}^{L\times K}$ to $\mathcal{M}_r$ for the semi-norms (2) can be easily computed using the singular value decomposition (SVD) of an appropriate matrix.

Let $\mathbf{A}$ be some matrix in $\mathbb{R}^{L\times K}$ and suppose that we need to compute the projection of this matrix onto $\mathcal{M}_r$.

2.4.1. Frobenius norm

Assume first that $\mathbf{W}$ is the $K \times K$ identity matrix so that the semi-norm in (2) is the usual Frobenius norm

$$\|\mathbf{A}\|_F^2 = \sum_{l=1}^{L}\sum_{k=1}^{K} a_{lk}^2 \quad \text{for } \mathbf{A} = (a_{lk})_{l,k=1}^{L,K} \in \mathbb{R}^{L\times K}\,. \tag{9}$$

Then for any r, a projection to $\mathcal{M}_r$ can be computed with the help of the SVD of $\mathbf{A}$ as follows. Let $\sigma_i = \sigma_i(\mathbf{A})$, the singular values of $\mathbf{A}$, be ordered so that $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_L$. Denote $\mathbf{\Sigma}_0 = \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_L)$ and $\mathbf{\Sigma} =$

$\mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r, 0, \ldots, 0)$. Then the SVD of $\mathbf{A}$ can be written as $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}_0\mathbf{V}^T$, where columns U_l of the matrix $\mathbf{U} \in \mathbb{R}^{L\times L}$ are the left singular vectors of $\mathbf{A}$ and columns V_l of the matrix $\mathbf{V} \in \mathbb{R}^{K\times L}$ are the right singular vectors $V_l = \mathbf{A}^T U_l/\sigma_l$ (if for some l the singular value $\sigma_l = 0$ then $V_l \in \mathbb{R}^K$ can be chosen arbitrarily). The matrix

$$\pi^{(r)}(\mathbf{A}) = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \sum_{i=1}^{r} U_i Z_i^T \;\text{ with }\; Z_i^T = U_i^T\mathbf{A}$$

belongs to $\mathcal{M}_r$ and minimizes the squared distance $\|\mathbf{A} - \tilde{\mathbf{A}}\|_F^2 = \mathrm{tr}(\mathbf{A} - \tilde{\mathbf{A}})(\mathbf{A} - \tilde{\mathbf{A}})^T$ over $\tilde{\mathbf{A}} \in \mathcal{M}_r$, see [14] or [11], Sect. 1.2.2. The projection $\pi^{(r)}(\mathbf{A})$ of $\mathbf{A}$ onto $\mathcal{M}_r$ is uniquely defined if and only if $\sigma_r > \sigma_{r+1}$. The squared distance between matrix $\mathbf{A}$ and $\mathcal{M}_r$ is

$$\rho^2(\mathbf{A}, \mathcal{M}_r) = \min_{\tilde{\mathbf{A}}\in\mathcal{M}_r} \rho^2(\mathbf{A}, \tilde{\mathbf{A}}) = \rho^2(\mathbf{A}, \pi^{(r)}(\mathbf{A})) = \|\mathbf{A} - \pi^{(r)}(\mathbf{A})\|_F^2 = \sum_{i=r+1}^{L} \sigma_i^2(\mathbf{A})\,.$$

2.4.2. The weighted semi-norm (2)

In the more general case of the semi-norm (2), one needs to compute SVD of the matrix $\mathbf{B} = \mathbf{A}\mathbf{W}^{1/2}$ rather than of $\mathbf{A}$. Note that for non-singular weight matrix $\mathbf{W}^{1/2}$ the considered problem is a special case of Theorem 3.18 in [15]. Let $\sigma'_i = \sigma_i(\mathbf{B})$, be the ordered singular values of $\mathbf{B}$ and let $\mathbf{\Sigma}'_0 = \mathrm{diag}(\sigma'_1, \sigma'_2, \ldots, \sigma'_L)$ and $\mathbf{\Sigma}' = \mathrm{diag}(\sigma'_1, \sigma'_2, \ldots, \sigma'_r, 0, \ldots, 0)$. Then the SVD of $\mathbf{B}$ is $\mathbf{B} = \mathbf{U}'\mathbf{\Sigma}'_0(\mathbf{V}')^T$ and the matrix

$$\pi^{(r)}(\mathbf{B}) = \mathbf{U}'\mathbf{\Sigma}'(\mathbf{V}')^T = \sum_{i=1}^{r} U'_i(Z'_i)^T \;\text{ with }\; (Z'_i)^T = (U'_i)^T\mathbf{B}$$

belongs to $\mathcal{M}_r$ and minimizes the squared distance $\|\mathbf{B} - \tilde{\mathbf{B}}\|_F^2 = \mathrm{tr}(\mathbf{B} - \tilde{\mathbf{B}})(\mathbf{B} - \tilde{\mathbf{B}})^T$ over $\tilde{\mathbf{B}} \in \mathcal{M}_r$.

Define

$$\pi_{\mathbf{w}}^{(r)}(\mathbf{A}) = \pi^{(r)}(\mathbf{B})\mathbf{W}^{-1/2} = \sum_{i=1}^{r} U'_i(T'_i)^T \in \mathcal{M}_r \;\text{ with }\; (T'_i)^T = (U'_i)^T\mathbf{B}\mathbf{W}^{-1/2}\,. \tag{10}$$

Then

$$\|\mathbf{A} - \pi_{\mathbf{w}}^{(r)}(\mathbf{A})\|_w^2 = \mathrm{tr}(\mathbf{A} - \pi_{\mathbf{w}}^{(r)}(\mathbf{A}))\mathbf{W}(\mathbf{A} - \pi_{\mathbf{w}}^{(r)}(\mathbf{A}))^T = \mathrm{tr}(\mathbf{A}\mathbf{W}^{1/2} - \pi_{\mathbf{w}}^{(r)}(\mathbf{A})\mathbf{W}^{1/2})(\mathbf{A}\mathbf{W}^{1/2} - \pi_{\mathbf{w}}^{(r)}(\mathbf{A})\mathbf{W}^{1/2})^T$$

$$= \mathrm{tr}(\mathbf{B} - \pi^{(r)}(\mathbf{B}))(\mathbf{B} - \pi^{(r)}(\mathbf{B}))^T = \|\mathbf{B} - \pi^{(r)}(\mathbf{B})\|_F^2\,.$$

Note that this equality holds even for singular $\mathbf{W}$. Note also that one can show $(\mathbf{A} - \pi_{\mathbf{w}}^{(r)}(\mathbf{A})) = (\mathbf{B} - \pi^{(r)}(\mathbf{B}))(\mathbf{W}^{1/2})^\dagger$ where $(\cdot)^\dagger$ denotes the pseudoinverse. This identity may help circumnavigate some of the problems to be described at the end of Section 3.2.

The squared weighted distance between matrix $\mathbf{A}$ and $\mathcal{M}_r$ is

$$\rho^2(\mathbf{A}, \mathcal{M}_r) = \min_{\tilde{\mathbf{A}}\in\mathcal{M}_r} \|\mathbf{A} - \tilde{\mathbf{A}}\|_{\mathbf{w}}^2 = \|\mathbf{A} - \pi^{(r)}(\mathbf{A})\|_{\mathbf{w}}^2 = \|\mathbf{B} - \pi^{(r)}(\mathbf{B})\|_F^2 = \sum_{i=r+1}^{L} \sigma_i^2(\mathbf{B})\,.$$

3. Distances defining the objective function f in (1)

In accordance with (2) the objective function f in (1) is

$$f(\mathbf{X}) = \mathrm{tr}(\mathbf{X} - \mathbf{X}_*)\mathbf{W}(\mathbf{X} - \mathbf{X}_*)^T\,, \tag{11}$$

where $\mathbf{W}$ is a diagonal matrix of weights, $\mathbf{W} = \mathrm{diag}(w_1, \ldots, w_K)$. In this section, we discuss the choice of the matrix $\mathbf{W}$ in (11) by making associations between the squared distances (7) and (11).

Note that Y_* in (7) is in the one-to-one correspondence with $\mathbf{X}_* = \mathbb{H}(Y_*)$ in (11) and $Y(\theta)$ in (7) defines a matrix $\mathbf{X} = \mathbb{H}(Y(\theta)) \in \mathcal{A}$ conditionally the values of q in (7) and r in (1) are in agreement, as discussed in Sect. 2.1. Given this, the main difference between the optimization problem (1) with the objective function (11) and problem (5) is that in (5) all the constraints (Hankel structure and rank of the matrices) are incorporated into the objective function.

3.1. Frobenius norm

The Frobenius norm (9) is the most commonly used norm for defining the distance in (1); see for example [11, 16, 15]. This frequent occurrence of the Frobenius norm can be explained by the following two reasons:

(i) the Frobenius norm is very natural in the non-structured low-rank approximation (when $\mathcal{A} = \mathcal{M}_r$) and the structured low-rank approximation problems are often considered simply as extensions of the non-structured approximation problems;

(ii) the very important operation of projecting a matrix on the set of matrices of a given rank (the set $\mathcal{M}_r$) is simplest when the chosen norm is Frobenius, see Sect. 2.4.

Consideration of the Frobenius norm in (1) corresponds to defining the objective function (7) with $\mathbf{S} = \mathbf{U}$, where $\mathbf{U} = \mathrm{diag}(u_1, \ldots, u_N)$ is the diagonal matrix with elements $u_1, \ldots, u_N$ defined in (8). This does not look natural if we look at the HSLRA problem as a problem of time series analysis or signal processing.

3.2. Uniform weights for each observation

If the uncertainty for all observations y_{j*} is approximately the same, then from the view-point of a time series analyst the most natural definition of the HSLRA objective function would be (5) with $s_j = 1$ for all j; that is, (7) with $\mathbf{S} = \mathbf{I}_N$, the $N \times N$ identity matrix. An important question thus arises: 'can we find a matrix $\mathbf{W}$ such that the semi-norm in (11) coincides with the norm in (7) with $\mathbf{S} = \mathbf{I}_N$ (conditionally r in (1) and q in (5) match)?' The answer is given in the following lemma.

Lemma 1. *Let $Y \in \mathbb{R}^N$ and $\mathbf{X} = \mathbb{H}(Y) \in \mathbb{R}^{L\times K}$. If $h = N/L$ is integer, then $\mathrm{tr}\,\mathbf{X}\,\mathbf{W}^{(0)}\,\mathbf{X}^T = Y^T Y$ where $\mathbf{W}^{(0)}$ is a diagonal matrix with diagonal elements*

$$w_{kk}^{(0)} = \begin{cases} 1 & \text{if } k = jL + 1 \text{ for some } \quad j = 0, \ldots, h-1, \\ 0 & \text{otherwise.} \end{cases}$$

Proof. Assume that $h = N/L$ is integer so that $N = hL$ and $K = N - L + 1 = (h-1)L + 1$.

By the definition, the elements of the $L \times K$ matrix $\mathbf{X}$ are $x_{lk} = y_{l+k-1}$. This gives

$$\mathrm{tr}\,\mathbf{X}\,\mathbf{W}^{(0)}\,\mathbf{X}^T = \sum_{l=1}^{L}\sum_{k=1}^{K}\sum_{k'=1}^{K} x_{lk} w_{kk'}^{(0)} x_{lk'} = \sum_{l=1}^{L}\sum_{k=1}^{K} w_{kk}^{(0)} x_{lk}^2 = \sum_{l=1}^{L}\sum_{j=0}^{h-1} x_{l,jL+1}^2 = \sum_{l=1}^{L}\sum_{j=0}^{h-1} y_{l+jL}^2 = \sum_{n=1}^{N} y_n^2 .$$

□

Thus, the answer to the question raised above is positive conditionally N is divisible by L. Very often this is not a serious restriction as typically there is some freedom in the choice of L and, if needed, the first few values of Y can be ignored (to alter the value of N).

Note that the matrix $\mathbf{W}^{(0)}$ has zeros at the diagonal and therefore (10) cannot be used as such. To be able to use the technique of Sect. 2.4.2 we either need to replace 0 with a small number or use the methodology outlined in Sect. 3.5.

3.3. Arbitrary observation weights

Consider now the general case of the HSLRA objective function (5) with arbitrary $s_j > 0$. Now, the answer to the question: 'can we find a matrix $\mathbf{W}$ such that the semi-norm in (11) coincides with the norm in (7) with arbitrary diagonal matrix $\mathbf{S}$?' is generally negative. The reason is as follows. First, it is easy to see that allowing the matrix $\mathbf{W}$ to be non-diagonal will not increase our freedom in improving the quality of approximation of the sum $Y^T\mathbf{S}Y$ by $\mathrm{tr}\,\mathbf{X}\,\mathbf{W}\,\mathbf{X}^T$, where $Y \in \mathbb{R}^N$ is arbitrary and $\mathbf{X} = \mathbb{H}(Y)$. And second, for a diagonal $\mathbf{W}$, $\mathrm{tr}\,\mathbf{X}\,\mathbf{W}\,\mathbf{X}^T$ is a weighted sum of squares of y_j's but the number of degrees of freedom (that is, diagonal elements in $\mathbf{W}$) is K which is less than the number of diagonal elements in $\mathbf{S}$.

The next question is: 'how can we approximate the sum of squares $Y^T\mathbf{S}Y$ by $\mathrm{tr}\,\mathbf{X}\,\mathbf{W}\,\mathbf{X}^T$ for arbitrary Y in the best way?' To answer this question, we shall use Least Squares (LS) approximation.

Let $\mathbf{S} \in \mathbb{R}^{N\times N}$ and $\mathbf{W} \in \mathbb{R}^{K\times K}$ be diagonal matrices with diagonals determined by the vectors $S = (s_1, \ldots, s_N)^T$ and $W = (w_1, \ldots, w_K)^T$, respectively. The vector S is a given and chosen, for example, as discussed in Sect. 2.1.

The vector W is unknown and has to be determined by trying to match the sum of squares $SS_1(Y) = \operatorname{tr} \mathbf{X}\mathbf{W}\mathbf{X}^T$ to $SS_2(Y) = Y^T\mathbf{S}Y$ for all $Y \in \mathbb{R}^N$.

Lemma 2. *Let $Y \in \mathbb{R}^N$, $\mathbf{X} = \mathbb{H}(Y)$ and $\mathbf{W}$ be a diagonal matrix with diagonal $W = (w_1, \dots, w_K)^T$. Then we have $SS_1(Y) = \sum_{n=1}^N \tilde{s}_n y_n^2$ where $\tilde{s}_n = \sum_{k=1}^K c_{nk} w_k$ and the coefficients c_{nk} are given by*

$$c_{nk} = \begin{cases} 1 & \text{if } \ 1 \le n \le K \ \text{ and } \ \max\{1, n-L+1\} \le k \le n, \\ 1 & \text{if } \ K+1 \le n \le N \ \text{ and } \ n-L+1 \le k \le K, \\ 0 & \text{otherwise.} \end{cases}$$

Proof. Using the substitution of indices $l \to n = l + k - 1$ (so that $1 \le n \le N$) we obtain

$$SS_1(Y) = \operatorname{tr} \mathbf{X}\mathbf{W}\mathbf{X}^T = \sum_{l=1}^{L}\sum_{k=1}^{K} w_k x_{lk}^2 = \sum_{l=1}^{L}\sum_{k=1}^{K} w_k y_{l+k-1}^2 = \sum_{n=1}^{N} y_n^2 \sum_{\substack{1 \le k \le K \\ 1 \le n-k+1 \le L}} w_k = \sum_{n=1}^{N} y_n^2 \sum_{k=1}^{K} c_{nk} w_k = \sum_{n=1}^{N} \tilde{s}_n y_n^2$$

where the coefficients c_{nk} are as above. □

Set $\tilde{S} = (\tilde{s}_1, \dots, \tilde{s}_N)^T \in \mathbb{R}^N$ and $\mathbf{C} = (c_{nk}) \in \mathbb{R}^{N \times K}$, where c_{nk} are the coefficients defined above. Then $\tilde{S} = \mathbf{C}W$ and thus we have the approximation problem $S \simeq \mathbf{C}W$. We write this problem in the form of a linear regression $S = \mathbf{C}W + \varepsilon$. Note that $\mathbf{C}$ has the following structure:

$$\mathbf{C} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 1 & & \ddots & 0 \\ 0 & \ddots & & 1 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 \end{pmatrix}.$$

Here S plays the role of the vector of observations and vector W is the vector of unknown parameters. We use a general least squares estimator (LSE) for W:

$$\hat{W} = \left(\mathbf{C}^T\mathbf{\Sigma}^{-1}\mathbf{C}\right)^{-1}\mathbf{C}^T\mathbf{\Sigma}^{-1}S\,, \tag{12}$$

where $\mathbf{\Sigma}$ is an arbitrary positive definite $N \times N$ matrix. The LSE of the vector S is $\hat{S} = \mathbf{C}\hat{W}$. The matrix Σ in (12) plays the role of a covariance matrix determining the precision of estimates of elements of the vector S. In accordance with results of Sect. 3.2, if $\mathbf{\Sigma} = \mathbf{I}_N$, the vector C is the vector of 1's and N is divisible by L, then we achieve the equality $\hat{S} = S$.

The vectors $\hat{W} = (\hat{w}_1, \dots, \hat{w}_K)^T$ and $\hat{S} = (\hat{s}_1, \dots, \hat{s}_N)^T$ are the vectors which define the norms (or semi-norms) we use instead of the norm we would have liked to use: for any $Y \in \mathbb{R}^N$,

$$Y^T\mathbf{S}Y \simeq Y^T\hat{\mathbf{S}}Y = \operatorname{tr} \mathbf{X}\hat{\mathbf{W}}\mathbf{X}^T$$

where $\mathbf{X} = \mathbb{H}(Y)$, $\hat{\mathbf{S}} = \operatorname{diag}(\hat{s}_1, \dots, \hat{s}_N)$ and $\hat{\mathbf{W}} = \operatorname{diag}(\hat{w}_1, \dots, \hat{w}_K)$.

3.4. Dealing with missing values

We now explain how one can formulate the HSLRA problem when there are missing observations among y_{j*}. Denote by $J \subset \{1, 2 \dots, N\}$ be the set of indices j such that the observations y_{j*} are missing.

Let us return to the discussion in Sect. 2.1 just after formula (5). In that discussion, we mentioned that a natural choice of s_j is $s_j = 1/\sigma_j^2$, where σ_j^2 is the variance of the observation error of y_j. If $j \in J$ then we can use any number as y_{j*} but assume $\sigma_j^2 = \infty$ for this j. This would yield $s_j = 0$ for all $j \in J$. This is, however, not sufficient as we also want to achieve $\hat{s}_j = 0$ for all $j \in J$. Note that in formula (12), we may need to multiply large numbers by s_j and do

not want the results to vanish. We therefore assume that s_j are very small numbers (infinitesimal) for all $j \in J$ rather than zeros.

Assume $\mathbf{\Sigma}$ is diagonal: $\mathbf{\Sigma} = \mathrm{diag}(\Sigma_{11}, \ldots, \Sigma_{NN})$. As we want to achieve $\hat{s}_j = 0$ for all $j \in J$, we need to choose $\Sigma_{jj} = 0$ for all $j \in J$ implying $\Sigma_{jj}^{-1} = \infty$ for $j \in J$. This would guarantee $\hat{s}_j = 0$ for all $j \in J$. The main problem here is the organization of calculations as in the process of computing the estimator (12) we many times need to multiply and divide by infinity. This problem is discussed in the next section (Sect. 3.5). Note that it is also possible to achieve $\hat{s}_j = 0$ for all $j \in J$ by using the method known as constrained least squares [17].

Note that in a special case when missing observations are at the end of the series, the problem of estimation of missing observations is known as the problem of forecasting. To forecast using low-rank approximations we can proceed in two ways: (i) solve the HSLRA problem and consequently build the model (6) using the available observations and use the model (6) for forecasting, and (ii) solve the HSLRA problem with a larger value of N treating the observations at the end of the series as missing. An important question arises of whether the two HSRA models and associated models (6) are the same. The answer to this question is positive conditionally that we deal with infinity as explained in Sect. 3.5 (that is, avoiding any approximate computations), see Example 1′ for the illustration of this phenomena.

3.5. Computations involving infinite and infinitesimal numbers

As explained in Sect. 3.4, if there are missing values in the series Y_*, then to compute the weighting matrix $\mathbf{W}$ defining the objective function in the HSLRA problem (1) we need to perform many operations with infinity (including division by 0). The usual way would be to replace infinity with a very large number and 0 with a very small number, see an example below. This creates difficulties as we do not know in advance how large or small should be the numbers that replace infinity and zero and we thus need to try several numbers to be sure that we have reached an acceptable accuracy. Also, in this way we would never be able to get an exact answer as the one derived in Sect. 3.2.

There is, however, a novel methodology for dealing with numerical infinity; that is, with infinite and infinitesimal quantities. This methodology has been developed by Ya. Sergeyev and published in a small book [18] and a series of papers; see for example, [19, 20, 21]. The place of Sergeyev's methodology for dealing with infinitesimals and infinites among other mathematical approaches is discussed in a historical survey [22].

Following Ya. Sergeyev, we denote a numerical infinity by ① and a numerical infinitesimal quantity by $①^{-1}$. ① satisfies some axioms; see the above cited papers of Ya. Sergeyev and [23] for a discussion of Sergeyev's axioms and their modifications.

If the 'Infinity computer' of Ya. Sergeyev existed then we would have been able to perform all operations with infinite and infinitesimal numbers numerically (rather than symbolically, which is computationally very demanding).

Let us consider a simple example of constructing the norm (2) (to be used for defining the objective function (11)) in the case of missing data.

Example 1. Assume $N = 10$ and $L = 3$ so that $K = 8$. Assume we have 2 missing observations which are y_{3*} and y_{5*} (the results are very similar with respect to the location of missing values and even with respect to the number of missing values). The vector S, defining the 'ideal' norm (7), that we have to approximate is $S_{ideal} = (1, 1, 0, 1, 0, 1, 1, 1, 1, 1)^T$. In view of the discussion in Sect. 3.4, we set $S = (1, 1, \alpha, 1, \alpha, 1, 1, 1, 1, 1)^T$, where $\alpha > 0$ is a very small (infinitesimal) number.

According to the recommendation of Sect. 3.4 we may choose the matrix $\mathbf{\Sigma} = \mathrm{diag}(\Sigma_{11}, \ldots, \Sigma_{NN})$ so that $\Sigma_{33} = \Sigma_{55} = 0$ and all other diagonal elements $\Sigma_{jj} = 1$ for $j \neq 3, 5$. Since we need to invert $\mathbf{\Sigma}$, we cannot set $\Sigma_{33} = \Sigma_{55} = 0$ and therefore we set $\Sigma_{33} = \Sigma_{55} = \beta$, where $\beta > 0$ is a very small positive number (it may and generally should differ from α). Straightforward calculations using (12) give

$$\hat{W} = \frac{1}{\beta + 10} \left[\gamma_1, 6 - 2\alpha, \beta + 11\alpha - 12, 12 - 6\alpha, \beta + 5\alpha, 0, 6 - 2\alpha, \gamma_1\right]^T ,$$

$$\hat{S} = \mathbf{C}\hat{W} = \frac{1}{\beta + 10} \left[\gamma_1, \gamma_2, 2\beta + 10\alpha, \gamma_1, 2\beta + 10\alpha, \gamma_2, \gamma_1, \gamma_2, \gamma_2, \gamma_1\right]^T$$

where $\gamma_1 = 6 + 2\alpha + \beta$, $\gamma_2 = 12 - \alpha + \beta$. An important observation here is that $\hat{S}[3]$ and $\hat{S}[5]$ tend to 0 as $\alpha \to 0$ and

$\beta \to 0$. In this particular case, we may set $\alpha = \beta = ①^{-1}$. Then we have

$$\hat{S} = \frac{3}{5}\,[1,2,0,1,0,2,1,2,2,1]^T + \frac{3}{25\,①}\,[2,-1,10,2,10,-1,2,-1,-1,2]^T + O\left(\frac{1}{①^2}\right), \tag{13}$$

where $O\left(①^{-2}\right)$ indicates the terms or order $①^{-2}$ or less. In addition to the limiting vector of weights, the expansion (13) gives the exact rate of convergence to this limiting vector (as α and β tend to 0 with the same speed).

Example 1′. As in Example 1, assume $N = 10$, $L = 3$ but assume that we place two missing observations at the end of the series; that is, we assume that y_{9*} and $y_{10\,*}$ are missing. Similar to (13) we obtain $\hat{S} = S_{as} + O\left(①^{-1}\right)$, where $S_{as} = \frac{3}{7}\,[2,2,3,2,2,3,2,2,0,0]^T$. If we set $N = 8$, $L = 3$, $\mathbf{\Sigma} = \mathbf{I}_8$ and no observations are missed, then we obtain $\hat{S} = \frac{3}{7}\,[2,2,3,2,2,3,2,2]^T$, with no infinitesimals involved. This vector gives the first eight components of $S_{as} \in \mathbb{R}^{10}$ above. This illustrates the statement we have made at the end of Sect. 3.4.

4. Orthogonality condition and associated algorithms

4.1. Orthogonality condition

In this section, we consider the so-called orthogonality condition which any locally optimal solution to (1) should satisfy.

Theorem 1. *Let $Y = \mathbb{H}^{-1}(\mathbf{X})$ and $Y_* = \mathbb{H}^{-1}(\mathbf{X}_*)$ be non-zero vectors in $\mathbb{R}^N$, $\mathbf{W} \in \mathbb{R}^{K\times K}$ be a diagonal weight matrix (with diagonal given by a vector W) defining the norm* (2)*, $S = \mathbf{C}W \in \mathbb{R}^N$ be the associated vector (computed as $\tilde{S}$ in Lemma 2) defining the squared norm $Y^T\mathbf{S}Y = \sum_{n=1}^N s_n y_n^2 = \operatorname{tr}\mathbf{X}\mathbf{W}\mathbf{X}^T$, where $\mathbf{S} \in \mathbb{R}^{N\times N}$ is a diagonal matrix with vector S on the diagonal. Set*

$$\beta_* = Y^T\mathbf{S}Y_*/Y^T\mathbf{S}Y = \operatorname{tr}\mathbf{X}\mathbf{W}\mathbf{X}_*^T/\operatorname{tr}\mathbf{X}\mathbf{W}\mathbf{X}^T. \tag{14}$$

Then we have

$$(\beta_*Y - Y_*)^T\mathbf{S}Y = \operatorname{tr}(\beta_*\mathbf{X} - \mathbf{X}_*)\mathbf{W}\mathbf{X}^T = 0\,. \tag{15}$$

Proof. Following from the statement of the Theorem it is straightforward to show that $\operatorname{tr}(\beta_*\mathbf{X} - \mathbf{X}_*)\mathbf{W}\mathbf{X}^T = (\beta_*Y - Y_*)^T\mathbf{S}Y$.

We have

$$\begin{aligned}(\beta_*Y - Y_*)^T\mathbf{S}Y &= \left(\frac{Y^T\mathbf{S}Y_*}{Y^T\mathbf{S}Y}Y - Y_*\right)^T\mathbf{S}Y\\ &= \frac{1}{Y^T\mathbf{S}Y}[(Y^T\mathbf{S}Y_*)Y - (Y^T\mathbf{S}Y)Y_*]^T\mathbf{S}Y\\ &= \frac{1}{Y^T\mathbf{S}Y}[(Y^T\mathbf{S}Y_*)Y^T\mathbf{S}Y - (Y^T\mathbf{S}Y)Y_*^T\mathbf{S}Y] = 0\,.\end{aligned}$$

□

In a particular case $\mathbf{W} = \mathbf{I}_K$ we have $S = T$, where $T = (t_1,\ldots,t_N)^T$ is the vector with elements given by (8). In this particular case, the equality

$$(\tilde{Y} - Y_*)^T\mathbf{S}\tilde{Y} = 0 \tag{16}$$

is called orthogonality condition for $\tilde{Y}$, see [16]. It was proved in [16] that if $\tilde{Y}$ is a local minima of $SS(Y) = (Y - Y_*)^T\mathbf{S}(Y - Y_*)$, then $\tilde{Y}$ must satisfy the orthogonality condition. By analogy, we shall call (16) the orthogonality condition even if $S \neq T$. In Theorem 1, we have shown that we can achieve orthogonality simply by multiplying all elements of either the Hankel matrix, or its associated vector, by a suitable constant. Note that the space $\mathcal{A}$ is closed with respect to multiplication by a non-zero constant.

4.2. Associated algorithms with $\mathbf{W} = \mathbf{I}_K$

In this section we consider two existing algorithms described in the literature with $\mathbf{W} = \mathbf{I}_K$. Algorithms for general $\mathbf{W}$ follow similarly.

4.2.1. De Moor's method

Following De Moor [16], introduce the Lagrangean function as follows. Let $c_1 \in \mathbb{R}^{L-r}$, $c_2 \in \mathbb{R}$, and $\mathbf{R} \in \mathbb{R}^{K\times(K-r)}$. The Lagrangean function corresponding to the HSLRA with the unweighted distance (9) is given by

$$F_1(\mathbf{X}, \mathbf{R}, c_1, c_2) = \|\mathbf{X} - \mathbf{X}_*\|_F^2 + c_1^T \mathbf{X}_* \mathbf{R} + c_2(\mathbf{R}^T\mathbf{R} - \mathbf{I}_{K-r}) \,. \tag{17}$$

The constraint $\mathrm{rank}(\mathbf{X}_*) = r$ is defined through its kernel representation $\mathbf{X}_*\mathbf{R} = 0$, where $\mathbf{R} = (\mathbf{R}', -\mathbf{I}_{K-r})^T$ for some matrix $\mathbf{R}' \in \mathbb{R}^{r\times(K-r)}$. The constraint $\mathbf{R}^T\mathbf{R} = \mathbf{I}_{(K-r)}$ is introduced to ensure the identifiability of $\mathbf{R}$. By setting all derivatives of the Lagrangean to zero, De Moor derived the condition (14) for the solution to the HSLRA problem. Moreover, by manipulating with the Lagrangean, De Moor has represented the solution of (17) as a non-linear generalized SVD. Subsequently, De Moor has developed an algorithm for numerical approximation of the non-linear generalized SVD problem which approximates the solution to the HSLRA problem; the convergence to the optimal solution is however not guaranteed. For full details we refer to De Moor [16]. For our comparative examples discussed in Section 6 we shall refer to this method as the 'DM' method.

4.2.2. I. Markovsky's method

The body of work by I.Markovsky and some of his co-authors (see [7], [8] and [15]) has concentrated on developing computationally efficient methods of approximating the solution of (1), where the structure is not necessarily Hankel. In the case of HSLRA, Markovsky defines the problem (1) through the minimization of the objective function

$$F_2(\mathbf{X}, \mathbf{R}) = \|\mathbf{X} - \mathbf{X}_*\|_F^2 \;\text{ such that }\; \mathbf{X}_*\mathbf{R} = 0,\; \mathbf{R}^T\mathbf{R} = \mathbf{I}_{(K-r)}, \tag{18}$$

using again the unweighted distance (9). $\mathbf{R}$ is as in (17). It can be seen this objective function has a direct analogy with (17).

The objective function (18) is non-convex, and in [8] the function (18) is locally optimized from an initial starting matrix (or vector). The latest software implementation to optimize (18) is given and described in [24]. In this software, by default, the initial starting point is the unstructured rank r approximation of $\mathbf{X}_*$, which is computed by $\pi^{(r)}(\mathbf{X}_*)$. Consequently there is no guarantee that a global minimum of (18) is found. We shall use this software for our comparative examples discussed in Section 6; note that this software can only be used when $r = L - 1$. We will refer to the minimization of (18), via the software described in [24], as the 'IM method'.

Some other methods are known; see [10, 25, 26]. As far as the authors are aware, none of the methods known in the literature on HSLRA have the theoretical property of convergence and hence the construction of reliable methods for solving the HSLRA problem remains an open problem. Moreover, most of the known algorithms require the condition $r = L - 1$. Except for the IM method, we failed to find reliable software implementing these other methods.

5. Algorithms based on the use of alternating projections

In this section we consider algorithms for solving the HSLRA problem represented as optimization problems using alternating projections between the spaces $\mathcal{H}$ and $\mathcal{M}_r$. We restrict our attention to the distance function associated with the matrix Frobenuis norm (9), that is, we take $\mathbf{W} = \mathbf{I}$ in (11).

5.1. Classical algorithms and their modifications

5.1.1. Alternating projections (AP)

The algorithm (19) below is the direct implementation of the alternating projections. For brevity we will refer to this algorithm as AP.

$$\mathbf{X}_0 = \mathbf{X}_*, \quad \mathbf{X}_{n+1} = \pi_{\mathcal{H}}\left[\pi^{(r)}(\mathbf{X}_n)\right] \quad \text{for } n = 0, 1, \ldots \tag{19}$$

These projections have also been studied in [27] and are sometimes known as Cadzow iterations [28]. Here we simply alternate projections to the space $\mathcal{M}_r$ with projections to the space $\mathcal{H}$. In this form of alternating projections, we have $\mathbf{X}_n \in \mathcal{H}$ for all $n = 0, 1, \ldots$.

General information about algorithms that use alternating projections (where the sets are not necessarily convex) is provided in Andersson and Carlsson [29] and Andersson et al. [30]. AP can be considered as a particular instance of the Alternating Least Squares method used in signal processing, see Section 3.3 in [31]. Note also that one iteration of AP for HSLRA corresponds to the basic version of the technique of time series analysis known as singular spectrum analysis (SSA), see [32]; for further details regarding the link between AP and SSA; see, for example, Gillard [28].

5.1.2. Orthogonalized Alternating Projections (OAP)

The following algorithm is a slight improvement over AP (19):

$$\mathbf{X}_0 = \mathbf{X}_*, \quad \mathbf{X}_{n+1} = \frac{\mathrm{tr}\mathbf{X}_n\mathbf{X}_*^T}{\mathrm{tr}\mathbf{X}_n\mathbf{X}_n^T}\,\pi_{\mathcal{H}}\left[\pi^{(r)}(\mathbf{X}_n)\right] \quad \text{for } n = 0, 1, \ldots \tag{20}$$

The algorithm (20) uses the coefficients (14) to improve at each iteration. We shall refer to the algorithm (20) as 'Orthogonalized Alternating Projections' (abbreviated OAP in the discussion and examples below).

5.1.3. Discussion on the convergence of AP and OAP

Despite AP often appearing to be myopic and too greedy by only aiming at minimizing the distance $\rho^2(\mathbf{X}, \mathcal{M}_r)$, it is very popular in practice. The popularity of AP is explained by the simplicity of the algorithm and by the fact that convergence to the space $\mathcal{A}$ is guaranteed, see [33]. However, as seen in examples provided in Section 6, AP often converges to a matrix which is far away from the set of optimal solutions $\mathfrak{X}^*$.

As shown in [29, Th. 6.1], AP converges linearly; that is, there exist constants $c < 1$ and $A > 0$ such that $\rho^2(\mathbf{X}_\infty, \mathbf{X}_n) < Ac^n$, $\forall n$, where $\mathbf{X}_\infty$ is some matrix in $\mathcal{A}$. Moreover, it is easy to prove monotonicity of AP iterations. As derived by Chu et al. [33], we have

$$\|\mathbf{X}_{n+1} - \pi^{(r)}(\mathbf{X}_{n+1})\|_F^2 \le \|\mathbf{X}_{n+1} - \pi^{(r)}(\mathbf{X}_n)\|_F^2 \le \|\mathbf{X}_n - \pi^{(r)}(\mathbf{X}_n)\|_F^2 .$$

Similar to AP (19), the algorithm OAP (20) converges to some matrix in $\mathcal{A}$. Numerical results show that the resulting approximation is never worse than the approximation obtained by (19) (usually it is slightly better than the approximation obtained by AP); it also converges faster to the set $\mathcal{A}$. Examples of Section 6 show that typically both AP and OAP do not converge to the set of optimal solutions $\mathfrak{X}^*$.

5.2. Alternating Projections with Backtracking and Randomization

In this section, we describe a family of algorithms which can be run as a random multistart-type algorithm, as a multistage algorithm and also as an evolutionary method. The main steps of this algorithm are summarized by its title 'Alternating Projections with Backtracking and Randomization' and we abbreviate this algorithm APBR. Here we describe two versions of this algorithm, Multistart APBR and APBR with selection. APBR with selection significantly reduces the number of computations by terminating non-prospective trajectories at early stages.

The underpinning idea for the family of the APBR algorithms has been suggested by the authors in [12] where the potential of the multistart APBR has been demonstrated on a number of examples.

5.2.1. Multistart APBR

The multistart version of APBR is described as follows. Let U denote a realization of a random number with uniform distribution in $[0, 1]$ and let $\tilde{\mathbf{X}}$ denote a random Hankel matrix which corresponds to a realization of a white noise Gaussian process $\tilde{Y} = (\xi_1, \ldots, \xi_N)$ with ξ_i, $i = 1, \ldots N$, independent Gaussian random variables with mean 0 and variance $s^2 \ge 0$.

In Multistart APBR, we run M independent trajectories in the space $\mathcal{H}$ starting at random Hankel matrices

$$\mathbf{X}_{0,j} = (1 - s_0)\mathbf{X}_* + s_0\tilde{\mathbf{X}}, \tag{21}$$

with some s_0 ($0 \le s_0 \le 1$), and use the updating formula

$$\mathbf{X}_{n+1,j} = \left(\mathrm{tr}\mathbf{Z}_{n,j}\mathbf{X}_*^T / \mathrm{tr}\mathbf{Z}_{n,j}\mathbf{Z}_{n,j}^T\right)\mathbf{Z}_{n,j} \tag{22}$$

where $j = 1, \ldots, M$,

$$\mathbf{Z}_{n,j} = (1 - \delta_n)\,\pi_{\mathcal{H}}\left[\pi^{(r)}(\mathbf{X}_{n,j})\right] + \delta_n\mathbf{X}_* + \sigma_n\tilde{\mathbf{X}} \tag{23}$$

and

$$\begin{cases} \delta_n = U/(n+1)^p, \quad \sigma_n = c/(n+1)^q, & \text{if } \rho^2(\mathbf{X}_{n,j}, \mathcal{M}_r) \ge \varepsilon\,, \\ \delta_n = 0, \quad \sigma_n = 0, & \text{otherwise}\,. \end{cases} \tag{24}$$

Each trajectory is either run until convergence or for a pre-specified number of iterations. U could be either random or simply set to 1, $c \in \{0, 1\}$ and positive numbers p, q and ε can be chosen arbitrarily, see Section 6 concerning the choice of U, c, p and q. A MATLAB implementation of this version of APBR, developed by the authors, is available at [34].

If $s_0 = \delta_n = \sigma_n = 0$ then the iterations in (22) coincide with iterations of OAP (20). If $s_0 > 0$ then the j-th trajectory of the algorithm starts at a random matrix in the neighbourhood of $\mathbf{X}_*$ (the width of this neighbourhood is controlled by the parameter s_0). If $\sigma_n > 0$ then there is a 'random mutation' at the n-th iteration (22). When $\delta_n > 0$, the current approximation 'backtracks' towards $\mathbf{X}_*$ conditionally that the backtracking does not worsen the distance $\rho^2(\mathbf{X}_{n,j}, \mathbf{X}_*)$. If $\rho^2(\mathbf{X}_{n,j}, \mathcal{M}_r) < \varepsilon$, we set $\delta_n = 0$ and $\sigma_n = 0$. That is, in the final stage for any trajectory of the APBR we perform OAP iterations (20) to achieve faster convergence to $\mathcal{A}$.

Note that in APBR the initial value of $\rho^2(\mathbf{X}_{0,j}, \mathbf{X}_*)$ could be large but the resulting j-th trajectory may be very good, see Figure 3(b).

5.2.2. APBR with selection

Step I (initialization). Run OAP until convergence and record the distance $F_* = \rho^2(\mathbf{X}_*, \mathbf{X}_{OAP})$, where $\mathbf{X}_{OAP}$ is the approximation of $\mathbf{X}_*$ obtained by OAP (20).

Step II (main iterations). For some pre-specified numbers M and Q, we compute the first Q terms of M independently run trajectories in the space $\mathcal{H}$ starting at random Hankel matrices $\mathbf{X}_{0,j} = \tilde{\mathbf{X}}$, so that we use (21) with $s_0 = 1$, and apply the updating formula (22).

Having reached the matrix $\mathbf{X}_{Q,j}$ of the j-th trajectory ($j = 1, \dots, M$), we test the prospectiveness of this j-th trajectory. Any trajectory with $\rho^2(\mathbf{X}_*, \mathbf{X}_{Q,j}) \geq F_*$ is considered non-prospective and terminated. We then choose k trajectories corresponding to the k smallest distances $\rho^2(\mathbf{X}_*, \mathbf{X}_{Q,j})$, $j = 1, \dots, M$, and perform OAP iterations until convergence (here k is a predefined small number). Let $\mathbf{X}_{\infty,i}$, $i = 1, \dots, k$, be the k HSLRA approximations obtained. Update the record value $F_* \leftarrow \min_{i=1,\dots,k}\{F_*;\, \rho^2(\mathbf{X}_*, \mathbf{X}_{\infty,i})\}$.

The trajectories which are not run until convergence and not terminated are halted and kept in a buffer. Step II is repeated with the updated record value F_*.

Step III (buffer check). Consider all trajectories halted at Step II and held in the buffer. First, remove all matrices $\mathbf{X}_{Q,j}$ with $\rho^2(\mathbf{X}_*, \mathbf{X}_{Q,j}) \geq F_*$ for the updated value of F_*. We then compare all the halted approximations $\mathbf{X}_{Q,j}$ in terms of their distances to $\mathbf{X}_*$ and to the space $\mathcal{M}_r$. If for some $i \neq j$ we have $\rho^2(\mathbf{X}_{Q,i}, \mathbf{X}_*) < \rho^2(\mathbf{X}_{Q,j}, \mathbf{X}_*)$ and $\rho^2(\mathbf{X}_{Q,i}, \mathcal{M}_r) < \rho^2(\mathbf{X}_{Q,j}, \mathcal{M}_r)$ then the matrix $\mathbf{X}_{Q,i}$ dominates the matrix $\mathbf{X}_{Q,j}$ and the j-th trajectory could be discontinued. All trajectories left after this sieving are run to convergence using OAP.

Remark 1. In Step II, prior to running AP for the k most prospective trajectories, we may 'split' them into further trajectories by adding small random Hankel matrices to the matrices $\mathbf{X}_{Q,j}$ from these prospective trajectories.

Remark 2. Using several matrices (with $n \geq Q$) from any trajectory $\{\mathbf{X}_{n,j}\}_n$ created by APBR with selection, we can estimate the exact value of the limit $F_{\infty,j} = \lim_{n\to\infty} \rho^2(\mathbf{X}_{n,j}, \mathbf{X}_*)$ in the following way. Consider a Hankel matrix $\mathbf{X}_{n,j}$ with some j and $n > Q$. Let $\mathbf{X}_{\infty,j}$ be the (unknown) limiting matrix for the trajectory $\{\mathbf{X}_{i,j}\}_i$. Assuming that $\mathbf{X}_{n,j}$ is close enough to the set $\mathcal{A}$, we have the geometrically fast convergence of $\mathbf{X}_{i,j}$ to $\mathbf{X}_{\infty,j}$ for $i = n, n+1, \dots$. Indeed, at all iterations with $n > Q$ the APBR with selection only makes alternative projections (20) to the spaces $\mathcal{H}$ and $\mathcal{M}_r$. Therefore, as follows from results stated in Section 5.1.3, we have the geometrically fast convergence of $\pi^{(r)}(\mathbf{X}_{i,j})$ to $\mathbf{X}_{\infty,j}$ with the sequence $\{\rho^2(\mathbf{X}_{i,j}, \mathbf{X}_*)\}_i$ increasing and the sequence $\{\rho^2(\pi^{(r)}(\mathbf{X}_{i,j}), \mathbf{X}_*)\}_i$ decreasing. (It is straightforward to check that $\rho^2(\mathbf{X}_{i,j}, \mathbf{X}_*) \leq \rho^2(\pi^{(r)}(\mathbf{X}_{i,j}), \mathbf{X}_*)$ for any i and j.) Then $F_{\infty,j}$ is the uniquely defined constant α such that the sequence of ratios $[\rho^2(\mathbf{X}_{i,j}, \mathbf{X}_*) - \alpha]/[\rho^2(\mathbf{X}_{i+1,j}, \mathbf{X}_*) - \alpha]$ tends to a constant $c < 1$ as $i \to \infty$. Similarly, $F_{\infty,j}$ is the uniquely defined constant β such that the sequence of ratios $[\rho^2(\pi^{(r)}(\mathbf{X}_{i,j}), \mathbf{X}_*) - \beta]/[\rho^2(\pi^{(r)}(\mathbf{X}_{i+1,j}), \mathbf{X}_*) - \beta]$ tends to a constant $c' < 1$ as $i \to \infty$.

Using the estimators $\hat{F}_j$ of $F_{\infty,j}$ we can terminate non-prospective trajectories as soon as $\hat{F}_j \geq F_*$; this may happen earlier than the event $\rho^2(\mathbf{X}_*, \mathbf{X}_{Q,j}) \geq F_*$ which has been used in Steps II and III for termination of non-prospective trajectories.

5.3. Convergence of algorithms

Consider the APBR defined by the formulas (21)–(24); these formulas underpin both versions of APBR defined above. These algorithms are typical stochastic global optimization algorithms and hence their theoretical properties

of convergence can be studied using the common tools readily available in the literature on global random search; see, for example, [35, Sect. 2.1.3].

The APBR algorithms, considered as optimization algorithms for the objective function (1), benefit from both globality and locality of search. Local convergence to the set $\mathcal{A}$ is a consequence of the fact that OAP iterations always locally converge (linearly) to $\mathcal{A}$, see [29, 33]. Global convergence is guaranteed if we assume that $s_0 = 1$ and $M \to \infty$. In this case, the set of starting matrices for APBR becomes everywhere dense. This is a necessary condition of convergence for any global optimization algorithm employing local descent iterations which does not use any properties of the objective function other than its continuity. It also becomes a sufficient condition if the iterations are monotonic so that the value of the objective function is non-increasing. For the algorithms considered, this is a consequence of the condition used in (22).

6. Examples

In examples below we shall use the following additional notation:

$\mathbf{X}_{AP} = \mathbb{H}(Y_{AP})$	Approximation obtained by AP (19)
$\mathbf{X}_{OAP} = \mathbb{H}(Y_{OAP})$	Approximation obtained by OAP (20)
$\mathbf{X}_{IM} = \mathbb{H}(Y_{IM})$	Approximation obtained by software of I.Markovsky and K.Usevich [24]
$\mathbf{X}_{DM} = \mathbb{H}(Y_{DM})$	Approximation obtained by De Moor in [16]
$\mathbf{X}_{APBR} = \mathbb{H}(Y_{APBR})$	Approximation obtained by Multistage APBR
Min(APBR)	Minimal distance to $\mathbf{X}_*$ obtained by Multistage APBR
Med(APBR)	Median distance to $\mathbf{X}_*$ obtained by Multistage APBR

In all examples we have used Multistage APBR, with ρ^2 defined by the Frobenius norm (9) and the number of trajectories was chosen as $M = 1000$. Unless otherwise stated, U is a realization of a random number with uniform distribution in $[0, 1]$. For ease of exposition, (24) is defined as

$$\begin{cases} \delta_n = U/(n+1)^p, \quad \sigma_n = c/(n+1)^q, & \text{for } n = 0, 1, \ldots, P, \\ \delta_n = 0, \quad \sigma_n = 0, & \text{for } n > P. \end{cases}$$

6.1. Numerical results for the example of De Moor

In this section we consider the data given by De Moor [16] to demonstrate the sub-optimality of AP (19). De Moor's data and parameter settings are as follows: $Y_* = (3, 4, 2, 1, 5, 6, 7, 1, 2)^T$, $N = 9$, $L = 4$, and $r = 3$. Let $\mathbf{X}_* = \mathbb{H}(Y_*)$. Table 1 contains the Frobenius distances to $\mathbf{X}_*$ obtained using AP (19), OAP (20), the IM method, for different values of the parameters L and r. Table 1 also contains the result obtained by De Moor. The approximation achieved by AP and by De Moor, Y_{AP} and Y_{DM} are provided in [16] (four decimal places only). One of the best Multistage APBR approximations for the De Moor's data is $Y_{APBR} = (3.451346, 3.533941, 2.002535, 1.487395, 4.039565, 7.078974, 5.995627, 1.720123, 1.613392)^T$. This Multistage APBR solution is slightly different from Y_{DM} while the values of the Frobenius distances to $\mathbf{X}_*$ coincide (with the precision provided by De Moor). We have no access to the software realizing the method of De Moor and therefore we cannot perform a comparative study involving this method. I.Markovsky's latest software implementation [24] can only be applied for the case $r = L - 1$ unless one reshapes the Hankel matrix. Note that if a $L \times (N - L + 1)$ Hankel matrix is of rank r then the reshaped $(r + 1) \times (N - r)$ Hankel matrix is also of rank r, see for example [36]. Also note that this reshaping approach can only be applied to Hankel matrices.

In this example, the parameters of Multistage APBR are selected to be $M = 1000$, $P = 500$, $c = 1$, $s_0 = 0.25$, $s = 1$, $p = 0.5$ and $q = 1.5$. The total number of iterations was set at 600. OAP (20) was marginally better than AP (19). Multistage APBR yields solutions similar to that obtained by the method of De Moor. For all values of L and r Multistage APBR provides better (or, in one case, similar) solutions than the other methods considered.

Table 2 contains the minimum Frobenius distance to $\mathbf{X}_*$ using Multistage APBR, varying the parameters p and q in (22). With $s_0 = 0.25$ and $s = 1$ it can be seen that there are many (p, q) parameter pairs which yield similar solutions, with Frobenius distances to $\mathbf{X}_*$ comparable to that achieved by De Moor's method. Summarizing the numerical results obtained in this example (and similar ones), we can give the following recommendations concerning the choice of parameters p and q of the Multistage APBR algorithm (22).

	$L = 4$					
	AP	OAP	IM	Med (APBR)	Min (APBR)	DM
$r = 1$	110.3142	110.3141	110.0095	110.0101	110.0095	-
$r = 2$	73.6980	73.6955	72.8526	72.8550	72.8530	-
$r = 3$	14.8251	14.8218	14.1482	14.1481	14.1478	14.1478

	$L = 5$				
	AP	OAP	IM	Med (APBR)	Min (APBR)
$r = 1$	111.8552	111.8552	111.5625	111.5690	111.5625
$r = 2$	73.3795	73.3786	73.1739	73.1790	73.1740
$r = 3$	15.6168	15.6160	14.9518	14.9597	14.9519
$r = 4$	3.4535	3.4535	3.4535	3.4535	3.4509

Table 1: Frobenius distances of the approximations to $\mathbf{X}_*$ using Multistage APBR, AP (19), OAP (20), IM method (18) and De Moor (DM) results.

$p \diagdown q$	0.25	0.5	0.75	1	1.5	2
0.25	14.2678	14.1528	14.1483	14.1478	14.1478	14.1478
0.5	14.3230	14.1540	14.1486	14.1478	14.1478	14.1478
0.75	16.6083	14.2294	14.1535	14.1479	14.1478	14.1478
1	17.1857	14.4427	14.1721	14.1498	14.1478	14.1478
1.5	36.1126	16.3141	14.3893	14.4292	14.1976	14.2249
2	38.4927	18.0746	14.9541	14.9674	14.4582	14.7119

Table 2: Minimum Frobenius distances to $\mathbf{X}_*$ using Multistage APBR, varying the parameters p and q; $L = 4$, $r = 3$.

Backtracking (regulated by the parameter p), is extremely important. It is worth noting that numerical results show that in many examples the use of the random variable U in the formula for δ_n in (22) often works marginally better than a constant. For the data considered in this example, Multistage APBR was best performing with values of p in between 0.25 and 1, implying that the rate at which backtracking decreases should be slow.

Randomization (regulated by the parameter q) could be useful too. Note also that the mechanism of this stochastic mutation in Multistage APBR resembles the mechanism of regularization of the Alternating Least Squares methods used in signal processing and in particular tensor decompositions, see [31]. Randomization appears to be beneficial both at the start of the iterations and throughout the running of the algorithm, but as illustrated in Table 2, the rate at which randomization decreases should be slightly faster than for backtracking (that is, we recommend choosing $p < q$). For the data considered in this example, Multistage APBR was best performing with values of q in between 1 and 2.

6.2. Numerical results for other examples

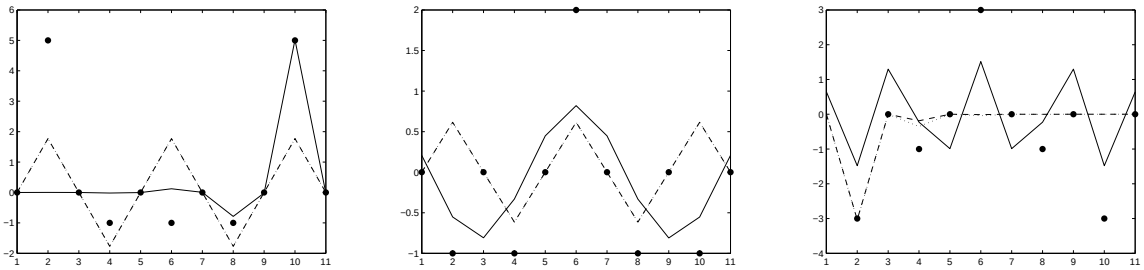
In this example we introduce a parametric family based on the data originally studied in [37]. Let $N = 11$ and $Y_*^{(m)} = (0, 3-2m, 0, -1, 0, m, 0, -1, 0, 3-2m, 0)^T$, where $m = -1, 0, 1, 2, 3$. We fix $L = 3$ and $r = 2$. Set $\mathbf{X}_*^{(m)} = \mathbb{H}\left(Y_*^{(m)}\right)$. We have rank $\left(\mathbf{X}_*^{(1)}\right) = 2$ while the rank of other matrices $\mathbf{X}_*^{(m)}$ (for $m = -1, 0, 2, 3$) is equal to 3.

We compare the results obtained from performing AP (19), OAP (20), Multistage APBR (22) and the IM method from [24]. The parameters of (22) are $M = 1000$, $c = 1$, $s_0 = 0.25$, $s = 1$, $p = 0.5$ and $q = 1.5$. The total number of iterations was fixed at 250 with $P = 200$. Table 3 contains the Frobenius distances to $\mathbf{X}_*^{(m)}$ using AP (19), OAP, (20), Multistage APBR (22) and the IM method. The results show that Multistage APBR gives better results than the other methods.

Figure 1 contains plots of the original data $Y_*^{(m)}$, for $m = -1, 2, 3$, and approximations obtained from performing Cadzow iterations (19), Multistart APBR (22) and the IM method. The difference between the results obtained using the IM method and AP is not seen for $m = -1$ and $m = 2$. Results obtained by different methods for $m = 0$ and $m = 1$ are indistinguishable in the figures and so are not included.

m	AP	OAP	IM	Med (APBR)	Min (APBR)
-1	68.3077	68.1548	68.3077	56.8699	56.7487
0	17.0769	17.0769	17.0769	17.0769	17.0769
1	0.0000	0.0000	0.0000	0.0000	0.0000
2	17.0769	17.0769	17.0769	12.9900	12.8791
3	50.1888	50.1873	49.9663	36.2506	36.2357

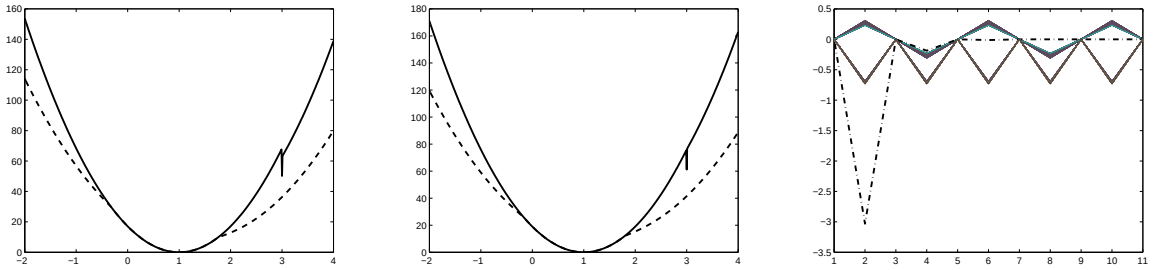
Table 3: Frobenius distances to $\mathbf{X}_*^{(m)}$ using AP (19), OAP (20) and IM method. The minimal and median Frobenius distances to $\mathbf{X}_*^{(m)}$ using Multistage APBR (22) with $M = 1000$ are also provided.



(a) $m = -1$ (b) $m = 2$ (c) $m = 3$

Figure 1: Plots of $Y_*^{(m)}$ for $m = -1, 2, 3$ (points) with approximations using AP (dashed line), IM method (dotted line) and the best approximation achieved using Multistart APBR (solid line).

Figure 2 contains the Frobenius distances from $\mathbf{X}_*$ for AP and Multistart APBR as a function of m where $-2 \leq m \leq 4$, for $L = 3$ and $L = 4$. Note a slight improvement in the overall trend for AP at $m = 3$. Figure 2 shows, in particular, that AP provides consistently poor results in our example. Figure 6.2 contains a plot of the solutions obtained by AP in the region $2.8 \leq m \leq 3.2$. The solution at $m = 3$ is different from the other solutions obtained in the regions $2.8 \leq m < 3$ and $3 < m \leq 3.2$. Solutions in the regions $2.8 \leq m < 3$ and $3 < m \leq 3.2$ are very similar.

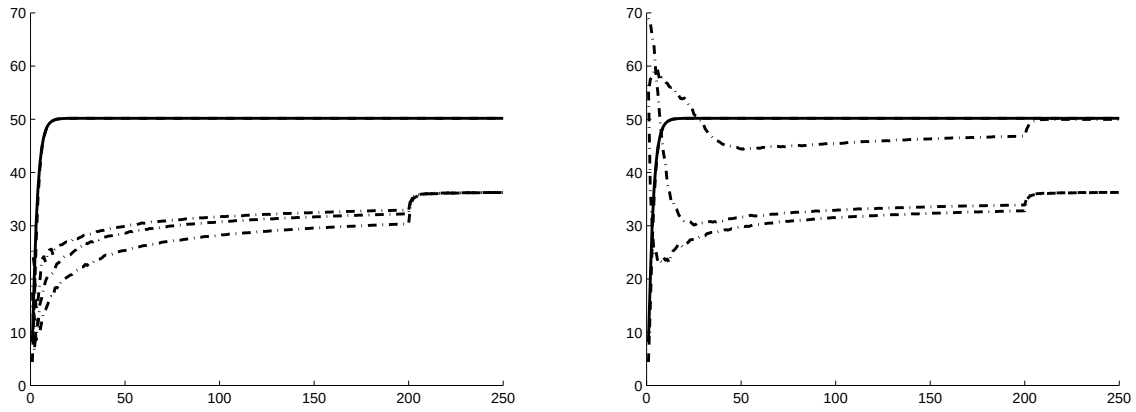


(a) Frobenius distances from $\mathbf{X}_*$, $L = 3$ (b) Frobenius distances from $\mathbf{X}_*$, $L = 4$ (c) Solutions for the case $L = 3$ in the region of $2.8 \leq m \leq 3.2$

Figure 2: Frobenius distances from $\mathbf{X}_*$ for AP (solid line) and the best approximation achieved using Multistart APBR (dashed line) as a function of m evaluated at increments of 0.01, with solutions for the case $L = 3$ in the region of $2.8 \leq m \leq 3.2$. The solution corresponding to $m = 3$ is in the dashed line.

In this example we can see that AP (19) is the poorest. OAP (20) and the IM method give marginal improvement on AP. For the cases $m = 0$ and $m = 1$, all algorithms give identical solutions. In these cases, the SLRA problem is quite simple. For example, for $m = 1$ the first AP iteration yields the optimal SLRA approximation.

Consider some results for the case $m = 3$. Figure 3 contains plots of the Frobenius distances $\|\mathbf{X}_n - \mathbf{X}_*\|_F^2$ as



(a) $s_0 = 0.25$ (b) $s_0 = 0.9$

Figure 3: Plots of the Frobenius distances $\|\mathbf{X}_n - \mathbf{X}_*\|_F^2$ as functions of n for AP (bold line) and three randomly selected Multistart APBR iterations (dot-dash line), for different s_0.

functions of n (for AP and three randomly selected Multistart APBR iterations), for different s_0. For small s_0 the distances are initially small, but then grow as the algorithm iterates towards a solution. For large s_0 it is likely that the distances are large initially, but the inclusion of backtracking in the Multistart APBR algorithm eventually makes these distances smaller. In Multistart APBR, after performing $P = 200$ iterations with randomization and backtracking, we perform 50 iterations of the OAP algorithm. The effect of using OAP in the Multistart APBR is very clear: the distances from $\mathbf{X}_*$ increase while the algorithm converges to some matrix in $\mathcal{A}$.

Table 4 contains the minimum and median Frobenius distances to $\mathbf{X}_*^{(3)}$ ($m = 3$) using Multistart APBR, for varying parameters p and q in (22). With $s_0 = 0.25$ and $s = 1$ it can be seen that there are a number of (p, q) parameter pairs which yield similar solutions. As is consistent with the previous example we advise to choose $0 < p < q$. We can also see that the results are very stable with respect to the values of p and q.

Min (APBR)						
$p \diagdown q$	0.25	0.5	0.75	1	1.5	2
0.25	36.8456	37.1360	37.1899	36.2055	37.2088	37.2088
0.5	36.1380	37.1232	36.2359	36.2359	36.2357	36.2358
0.75	36.1970	37.1932	36.2620	36.2661	36.2744	36.2749
1	36.4415	36.4320	36.2829	36.2912	36.2998	36.2999
2	36.4580	36.4960	36.3614	36.3178	36.3672	36.3253

Median (APBR)						
$p \diagdown q$	0.25	0.5	0.75	1	1.5	2
0.25	37.5396	37.2330	37.2111	36.2178	37.2129	37.2132
0.5	37.7568	37.3393	36.2489	36.2448	36.2408	36.2410
0.75	40.1898	37.6974	36.3167	36.2876	36.2791	36.2794
1	44.0458	37.8351	36.5178	36.3345	36.3069	36.3055
2	47.2857	42.0455	39.1298	38.1299	37.7234	37.5386

Table 4: Minimum and median Frobenius distances to $\mathbf{X}_*^{(3)}$ (in Frobenius norm) using Multistart APBR respectively, varying the parameters p and q.

6.3. Application to 'Air Passenger' data

In this example we consider the celebrated 'Air Passenger' data (available from [38]) which consists of monthly counts of airline passengers, measured in thousands, for the period January 1949 through December 1960. We denote this data by Y_*, and let $\mathbf{X}_* = \mathbb{H}(Y_*)$. We include this example to demonstrate that Multistart APBR may be used for the time series analysis of real data. Note that for general guidance concerning choice of the parameters L and r, see for example [5]. We wish to find an $r = 2$ approximation to the log-transformed data using AP (19), OAP (20) and Multistart APBR (22). Table 5 contains the Frobenius distances to $\mathbf{X}_*$ obtained using these methods with $L = 24$. The parameters of Multistart APBR (22) are selected to be $M = 1000$, $c = 1$, $s_0 = 0.25$, $s = 1$, $p = 0.5$, $q = 1.5$, and $P = 600$. The total number iterations was 800. Figure 4 contains a plot of the log-transformed data and the rank 2 AP and Multistart APBR approximations respectively.

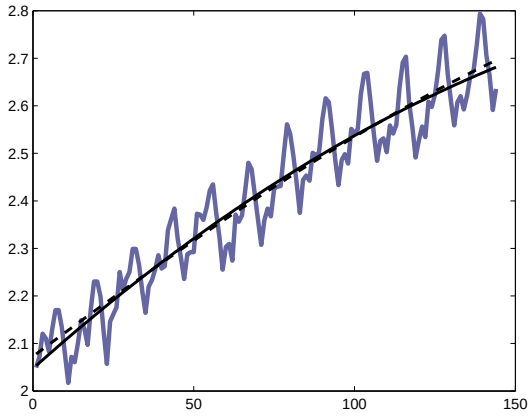


Figure 4: Plot of the log-transformed Air Passenger data (grey) and the rank 2 AP (dashed line) and Multistart APBR approximations (solid line) respectively.

AP	OAP	Med (APBR)	Min (APBR)
9.9652	9.9652	9.8606	9.8578

Table 5: Frobenius distances to $\mathbf{X}_*$ obtained using AP (19), OAP (20), and Multistart APBR (22).

7. Conclusion

This paper is devoted to the construction of numerical methods for solving the HSLRA problem. Finding optimal solutions to the HSLRA problem is very difficult. If HSLRA is considered as a problem of estimating parameters of damped sinusoids, then the associated objective function becomes extremely complex so that the associated optimization problem is basically unsolvable. This leads us to a conclusion that constructing algorithms for solving the HSLRA problem with trajectories in the space $\mathcal{H}$ of Hankel matrices lead to more tractable algorithms with higher chances of success. This is the approach which we undertook in the main body of the paper.

In Section 4 we discussed the so-called orthogonality condition which optimal solutions should satisfy, and described how any approximation may be corrected to achieve this orthogonality. In Section 5 we introduced our family of algorithms called APBR, which can be viewed as a global random search extension of AP. Examples provided in Section 6 show that Multistart APBR significantly outperforms AP and some other methods. APBR with selection is much more efficient than Multistart APBR. Many other, possibly more efficient, techniques could be adapted for solving the HSLRA problem but this is a theme for future research.

Appendix A. Complexity of the HSLRA optimization problem when using parametric forms of the solution

Appendix A.1. Objective function and its split into components

In this Appendix, we consider the HSLRA problem (1) as an optimization problem after a parametric representation of the solution has been set. The most popular parameterization of the set $\mathcal{A} = \mathcal{M}_r \cap \mathcal{H}$, often found in the signal processing literature (see for example [26]), is obtained by associating matrices $\mathbf{X} \in \mathcal{A}$ with vectors $Y(\theta) = (y_1(\theta), \ldots, y_N(\theta))^T$ whose elements can be represented as sums of damped sinusoids (other parametric forms are known but they lead to even more difficult optimization problems).

The definition of the objective function was given in (5), with elements of the vector $Y(\theta) = (y_1(\theta), \ldots, y_N(\theta))^T$ given by (6).

Let Θ be a parameter space so that $\theta \in \Theta$. The ranges for parameters a_i and d_i is $(-\infty, \infty)$, whilst the ranges for ω_i and ϕ_i are $[0, 1)$ and $[0, \pi/2)$ respectively. Denote the true value of parameters by $\theta^{(0)} = (a^{(0)}, d^{(0)}, \omega^{(0)}, \phi^{(0)})$.

If we assume that there is a true signal represented in the form (6), such as in the standard 'signal plus noise' model (4), then we denote the true values of parameters by $\theta^{(0)} = (a^{(0)}, d^{(0)}, \omega^{(0)}, \phi^{(0)})$ where $a^{(0)} = (a_1^{(0)}, \ldots, a_q^{(0)})$, $d^{(0)} = (d_1^{(0)}, \ldots, d_q^{(0)})$, $\omega^{(0)} = (\omega_1^{(0)}, \ldots, \omega_q^{(0)})$ and $\phi^{(0)} = (\phi_1^{(0)}, \ldots, \phi_q^{(0)})$. The associated true signal values will be $y_n^{(0)}$, $n = 1, \ldots, N$. If the observations are noise-free, then the vector of observations $Y = (y_1, \ldots, y_N)^T$ coincides with the signal vector $Y^{(0)} = (y_1^{(0)}, \ldots, y_N^{(0)})^T$. Otherwise Y is different from $Y^{(0)}$.

Given an observed vector $Y = (y_1, \ldots, y_N)^T$ we define the objective function $f(\theta)$ as follows:

$$f(\theta) = \sum_{n=1}^{N} s_n \varepsilon_n^2(\theta) = \sum_{n=1}^{N} s_n (y_n - y_n(\theta))^2 , \tag{A.1}$$

where $0 \le s_n \le \infty$, $n = 1, \ldots, N$ are a series of weights and $\varepsilon_n(\theta) = y_n - y_n(\theta)$.

A comprehensive study of the objective function (A.1) with the parameterization (6) has been undertaken in [37]. Some analysis of the objective function (A.1) has also been reported in [10], but we show below that it is possible to extend the analysis of [10].

Let $\hat{\theta} = (\hat{a}, \hat{d}, \hat{\omega}, \hat{\phi})$ denote either an estimated value of $\theta^{(0)}$ or simply any value in Θ. Then we may write:

$$\varepsilon_n(\theta) = y_n - y_n(\theta) = y_n - y_n(\hat{\theta}) + y_n(\hat{\theta}) - y_n(\theta) = \varepsilon_n(\hat{\theta}) + (y_n(\hat{\theta}) - y_n(\theta)) . \tag{A.2}$$

Hence we may write (A.1) as

$$f(\theta) = f(\hat{\theta}) + f_1(\theta, \hat{\theta}) + f_2(\theta, \hat{\theta}), \tag{A.3}$$

where

$$f_1(\theta, \hat{\theta}) = \sum_{n=1}^{N} s_n \left(y_n(\theta) - y_n(\hat{\theta})\right)^2 \quad \text{and} \quad f_2(\theta, \hat{\theta}) = 2 \sum_{n=1}^{N} s_n \varepsilon_n(\hat{\theta}) \left(y_n(\hat{\theta}) - y_n(\theta)\right).$$

Here we consider each component of (A.3) in turn. The component $f(\hat{\theta})$ is a constant representing the sum of squares between the observed vector $Y = (y_1, \ldots, y_N)^T$ and the vector $Y(\hat{\theta}) = (y_1(\hat{\theta}), \ldots, y_N(\hat{\theta}))^T$. The function $f_1(\theta, \hat{\theta})$ is a sum of squares but $f_2(\theta, \hat{\theta})$ is a sum of terms which may have alternating signs. This observation has led the authors of [10] to the suggestion that the following could be a common phenomena: if $\hat{\theta}$ is not a good approximation to the true parameter $\theta^{(0)}$ then $f_1(\theta, \hat{\theta})$ dominates the shape of the objective function $f(\theta)$, and the contribution of $f_2(\theta, \hat{\theta})$ is almost negligible. However, this seems to be true only if the vector Y is observed either without noise or when the noise is very small; that is, when $Y \simeq Y^{(0)}$. In the next paragraph we will argue and then demonstrate in the examples that follow that $f_2(\theta, \hat{\theta})$ often has a considerable and important contribution to the objective function $f(\hat{\theta})$ for all $\hat{\theta} \in \Theta$, especially if $\hat{\theta}$ is not a good approximation of $\theta^{(0)}$.

Consider the function $f_2(\theta, \hat{\theta})$. Represent it as $f_2(\theta, \hat{\theta}) = C(\hat{\theta}) - f_3(\theta, \hat{\theta})$, where

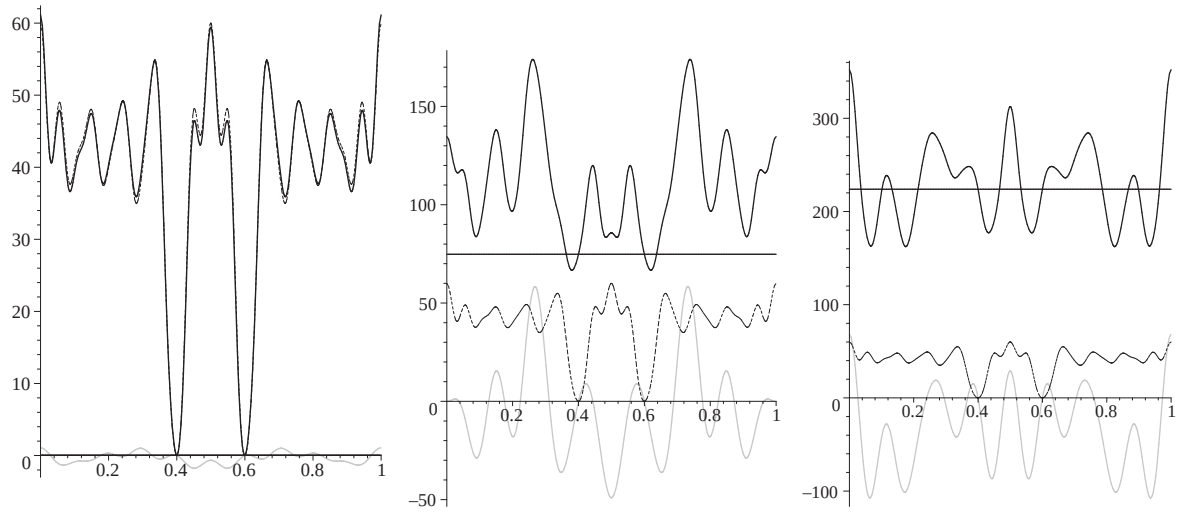
$$C(\hat{\theta}) = 2 \sum_{n=1}^{N} s_n \left(y_n - y_n(\hat{\theta})\right) y_n(\hat{\theta}) \quad \text{and} \quad f_3(\theta, \hat{\theta}) = 2 \sum_{n=1}^{N} s_n \left(y_n - y_n(\hat{\theta})\right) y_n(\theta).$$

Appealing to the standard results concerning the orthogonality of residuals [39], if $\hat{\theta}$ has been obtained as a least squares estimator of $\theta^{(0)}$ then $f_3(\theta^{(0)}, \hat{\theta}) \approx 0$ and $f_3(\theta^{(0)}, \theta)$ does not make a large contribution to the shape of the

objective function $f(\theta)$, at least in the region where $\theta \simeq \hat{\theta}$. However, if $\hat{\theta}$ is not a good approximation to $\theta^{(0)}$ then $f_3(\theta^{(0)}, \hat{\theta})$ may be significantly different from 0, and we can expect $f_3(\theta^{(0)}, \theta)$ to be significantly contributing to the shape of $f(\theta)$. Moreover, due to the autocorrelation inherent in Y, there are likely to be some oscillatory or seasonal patterns to be observed in $f_2(\theta, \hat{\theta})$. This phenomenon is confirmed by the results in the example below.

Appendix A.2. Example

Similar to [10] let us consider the parametrization (6) with $q = 1$ and the objective function defined by (A.1). Assume we generate a vector of $N = 10$ observations $Y^{(0)} = (y_1^{(0)}, \ldots, y_N^{(0)})^T$ from (6) with $\omega^{(0)} = 0.4$, $\phi^{(0)} = \frac{\pi}{2}$, $d^{(0)} = 0$, $a^{(0)} = 2$ and create a vector of observations $Y = (y_1, \ldots, y_N)^T$ such that $y_n = y_n^{(0)} + \epsilon_n$. The noise terms ϵ_n are assumed to be normally distributed with mean 0 and variance σ^2. We take $w_1 = \cdots = w_N = 1$. Similar phenomenon as demonstrated in this example can be observed for different weights.



(a) $\sigma^2 = 0.01$, $\hat{\omega} = 0.4004$, $f(\hat{\omega}) = 0.0613$, $f(\omega^{(0)}) = 0.0653$.

(b) $\sigma^2 = 9$, $\hat{\omega} = 0.3801$, $f(\hat{\omega}) = 66.6674$, $f(\omega^{(0)}) = 74.7927$.

(c) $\sigma^2 = 25$, $\hat{\omega} = 0.1752$, $f(\hat{\omega}) = 162.3507$, $f(\omega^{(0)}) = 223.8836$.

Figure A.5: Function $f(\omega)$ in solid line, $f(\omega^{(0)})$ in solid line (horizontal), $f_1(\omega, \omega^{(0)})$ in dashed line and $f_2(\omega, \omega^{(0)})$ in grey; for different values of σ. Estimated global minimum $\hat{\omega}$ and corresponding value of the objective function $f(\hat{\omega})$ also given.

In this example we assume that $d^{(0)}$, $\phi^{(0)}$ and $a^{(0)}$ are known but $\omega^{(0)}$ is not (a similar case was considered in [10]). This yields $\theta = \omega$ and the objective function becomes

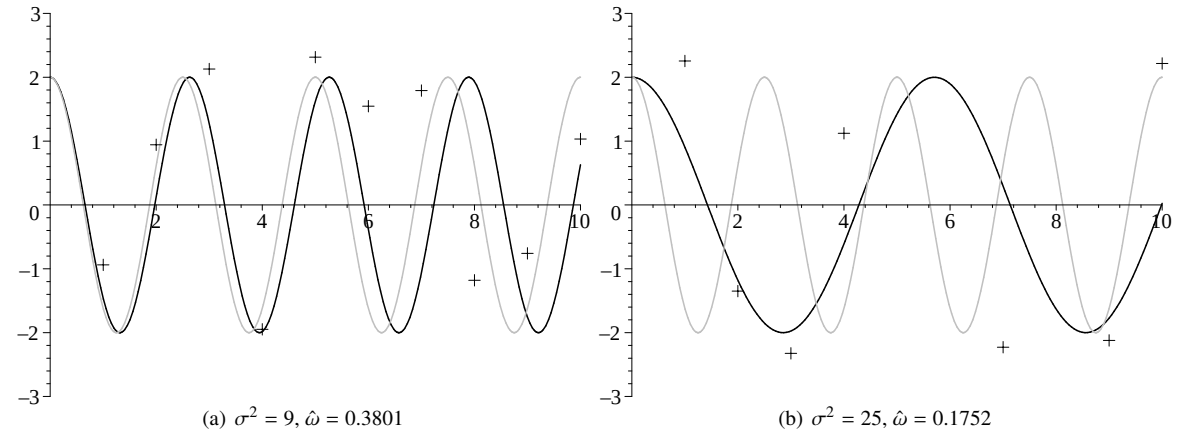
$$f(\omega) = \sum_{n=1}^{N} \left(y_n - a^{(0)} \exp(d^{(0)} n) \sin(2\pi\omega n + \phi^{(0)}) \right)^2 .$$

The feasible domain for ω can be chosen as $\Theta = [0, 1)$; in this interval, the function f many local minimizers (in addition to the global minimizer). As mentioned in [37] the number of local minima of $f(\omega)$ is linear in N and therefore the complexity of the objective function $f(\cdot)$ increases with N.

Fig. A.5 contains plots of $f(\omega)$, $f(\omega^{(0)})$, $f_1(\omega, \omega^{(0)})$ and $f_2(\omega, \omega^{(0)})$ for particular realizations of noise for varying values of σ^2. Adding noise to the observed data increases the complexity of the function $f(\omega)$ and moves the global minimizer of $f(\omega)$ away from the true value $\omega^{(0)} = 0.4$. As $\hat{\omega}$ we use the global minimizer of $f(\omega)$.

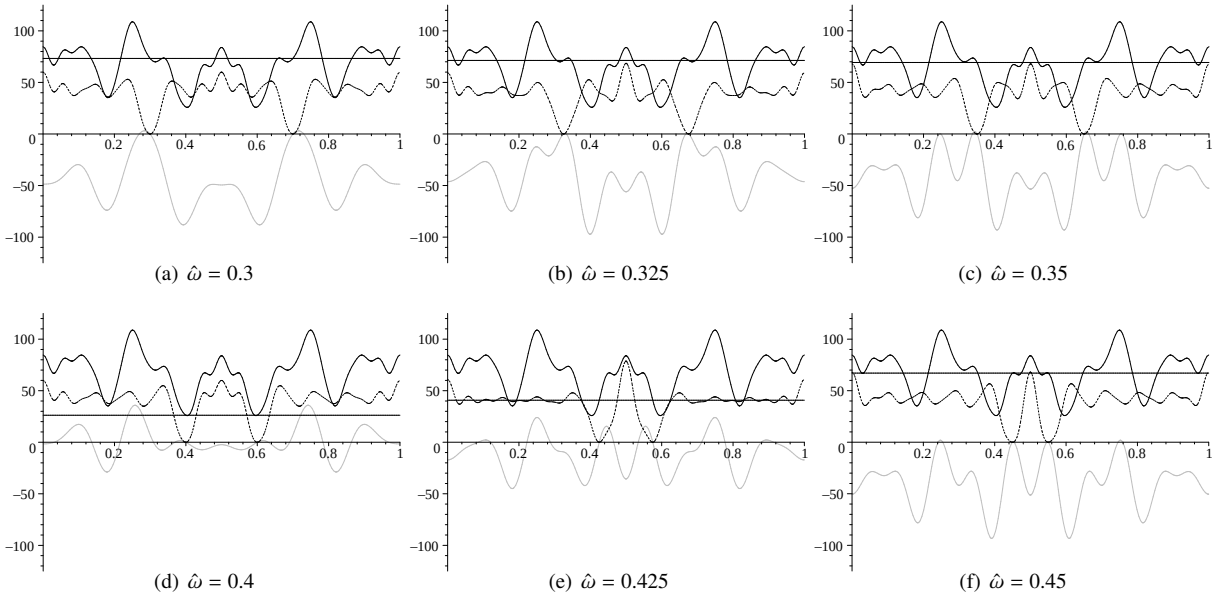
Fig. A.6 contains plots of Y and the estimated reconstructed signal. The effect of adding noise is to increase the complexity of the objective function f, and as such it it possible to obtain estimates of $\omega^{(0)}$ that are far away from the true value. The consequences of measurement error are clearly seen in this figure.

Now set $\sigma^2 = 2.5$. Fig. A.7 contains plots of $f(\omega)$, $f(\hat{\omega})$, $f_1(\omega, \hat{\omega})$ and $f_2(\omega, \hat{\omega})$ with $\hat{\theta} = \hat{\omega}$ perturbed away from the true value $\omega^{(0)} = 0.4$. Theoretical work by Lemmerling and Van Huffel [10] suggests that it is sufficient to study



(a) $\sigma^2 = 9$, $\hat{\omega} = 0.3801$ (b) $\sigma^2 = 25$, $\hat{\omega} = 0.1752$

Figure A.6: Plots of Y and the estimated reconstructed signal $a^{(0)} \exp(d^{(0)}n) \sin(2\pi\omega^{(0)}n + \phi^{(0)})$ in black, with true signal $a^{(0)} \exp(d^{(0)}n) \sin(2\pi\hat{\omega})n + \phi^{(0)}$ in grey, corresponding to the estimated global minimizer shown in Fig. A.5.



(a) $\hat{\omega} = 0.3$ (b) $\hat{\omega} = 0.325$ (c) $\hat{\omega} = 0.35$

(d) $\hat{\omega} = 0.4$ (e) $\hat{\omega} = 0.425$ (f) $\hat{\omega} = 0.45$

Figure A.7: $\omega^{(0)} = 0.4$, function $f(\omega)$ in solid line, $f(\omega^{(0)})$ in solid line (horizontal), $f_1(\omega, \omega^{(0)})$ in dashed line and $f_2(\omega, \omega^{(0)})$ in grey; with $\sigma^2 = 2.5$.

$f_1(\omega, \hat{\omega})$, as this component dominates the shape of the objective function $f(\omega)$, and the contribution of $f_2(\omega, \hat{\omega})$ is negligible particularly when $\hat{\omega}$ is not close to the true value of θ. However, it can be seen in Fig. A.7 that $f_2(\omega, \hat{\omega})$ is a very significant component of the objective function $f(\cdot)$, whatever the choice of $\hat{\omega}$. We can also see that the inclusion of noise in the vector Y always moves the global minimizer of $f(\cdot)$ away from the true value $\omega^{(0)}$ (which is the minimizer of $f(\cdot)$ in the noise-free situation).

Appendix A.3. Derivatives and simplification of the objective function (5)*, parameterized by* (6) *with* $q = 1$

Let $q = 1$ and consider the optimization problem (5) defined by the model (6). For brevity let $x_n = \exp(dn)\sin(2\pi\omega n + \phi)$ and $w_1 = \cdots = w_N = 1$. Equation (5) may be written

$$f(a, d, \omega, \phi) = \sum_{n=1}^{N} (y_n - a x_n)^2 \,. \tag{A.4}$$

Since

$$\frac{\partial f(a, d, \omega, \phi)}{\partial a} = -2 \sum_{n=1}^{N} (y_n - a x_n) x_n \,,$$

then we may obtain an explicit estimator for a, which we denote $\hat{a}$. This estimator is a function of the remaining parameters d, ω and ϕ;

$$\hat{a} = \frac{\sum_{n=1}^{N} y_n x_n}{\sum_{n=1}^{N} x_n^2} \,.$$

Substituting $\hat{a}$ into (A.4) gives a new objective function, which we denote $g(d, \omega, \phi)$:

$$g(d, \omega, \phi) = \sum_{n=1}^{N} \left(y_n - x_n \frac{\sum_{k=1}^{N} y_k x_k}{\sum_{k=1}^{N} x_k^2} \right)^2 = \sum_{n=1}^{N} \epsilon_{n,k}^2 \,.$$

where $\epsilon_{n,k} = y_n - \dfrac{\sum_{k=1}^{N} y_k x_k}{\sum_{k=1}^{N} x_k^2}$.

It is possible to compute the derivative of the objective function with respect to each of the unknown parameters. However, even for simple cases the derivatives cannot be written in a neat form. Here we state the first derivatives of $g(d, \omega, \phi)$ with respect to each of the unknown parameters:

$$\frac{\partial g}{\partial d} = -2 \sum_{n=1}^{N} \left\{ \epsilon_{n,k} \left[\frac{x_n \sum_{k=1}^{N} k y_k x_k}{\sum_{k=1}^{N} x_k^2} + \frac{x_n \sum_{k=1}^{N} y_k x_k}{\left(\sum_{k=1}^{N} x_k^2 \right)^2} \left(n \sum_{k=1}^{N} x_k^2 - 2 \sum_{k=1}^{N} k x_k^2 \right) \right] \right\} .$$

Let $c_1^{(n)} = n e^{dn} \cos(2\pi\omega n + \phi) \sum_{k=1}^{N} x_k^2 - x_n \sum_{k=1}^{N} k x_k e^{dk} \cos(2\pi\omega k + \phi)$, then

$$\frac{\partial g}{\partial \omega} = -4\pi \sum_{n=1}^{N} \left\{ \epsilon_{n,k} \left[\frac{x_n \sum_{k=1}^{N} k e^{dk} \cos(2\pi\omega k + \phi)}{\sum_{k=1}^{N} x_k^2} + \frac{\sum_{k=1}^{N} y_k x_k}{\left(\sum_{k=1}^{N} x_k^2 \right)^2} c_1^{(n)} \right] \right\} .$$

Let $c_2^{(n)} = e^{dn} \cos(2\pi\omega n + \phi) \sum_{k=1}^{N} x_k^2 - 2 x_n \sum_{k=1}^{N} x_k e^{dk} \cos(2\pi\omega n + \phi)$, then

$$\frac{\partial g}{\partial \phi} = -2 \sum_{n=1}^{N} \left\{ \epsilon_{n,k} \left[\frac{x_n \sum_{k=1}^{N} y_k e^{dk} \cos(2\pi\omega k + \phi)}{\sum_{k=1}^{N} x_k^2} + \frac{\sum_{k=1}^{N} y_k x_k}{\left(\sum_{k=1}^{N} x_k^2 \right)^2} c_2^{(n)} \right] \right\} .$$

References

[1] I. Markovsky, J. C. Willems, S. Van Huffel, B. De Moor, R. Pintelon, Application of structured total least squares for system identification and model reduction, IEEE Trans. Automat. Control 50 (10) (2005) 1490–1500.

[2] P. Lemmerling, N. Mastronardi, S. Van Huffel, Efficient implementation of a structured total least squares based speech compression method, Linear Algebra Appl. 366 (2003) 295–315.

[3] A. Yeredor, Multiple delays estimation for chirp signals using structured total least squares, Linear Algebra Appl. 391 (2004) 261–286.

[4] A. Pruessner, D. P. O'Leary, Blind deconvolution using a regularized structured total least norm algorithm, SIAM J. Matrix Anal. Appl. 24 (4) (2003) 1018–1037.

[5] N. Golyandina, On the choice of parameters in Singular Spectrum Analysis and related subspace-based methods, Statistics and Its Interface 3 (2010) 259–279.

[6] I. Markovsky, Structured low-rank approximation and its applications, Automatica 44 (4) (2008) 891–909.
[7] I. Markovsky, Bibliography on total least squares and related methods, Statistics and Its Interface 3 (3) (2010) 329–334.
[8] I. Markovsky, Low rank approximation: Algorithms, implementation, applications, Springer, 2012.
[9] W. B. Bishop, P. M. Djuric, Model order selection of damped sinusoids in noise by predictive densities, IEEE Trans. Signal. Process. 44 (3) (1996) 611–619.
[10] P. Lemmerling, S. Van Huffel, Analysis of the structured total least squares problem for Hankel/Toeplitz matrices, Numerical Algorithms 27 (1) (2001) 89–114.
[11] N. Golyandina, V. Nekrutkin, A. Zhigljavsky, Analysis of Time Series Structure: SSA and related techniques, Chapman & Hall/CRC, New York - London, 2001.
[12] J. Gillard, A. Zhigljavsky, Optimization challenges in the structured low rank approximation problem, Journal of Global Optimization (2012) 1–19.
[13] A. Žilinskas, J. Žilinskas, Interval arithmetic based optimization in nonlinear regression, Informatica 21 (1) (2010) 149–158.
[14] C. Eckart, G. Young, The approximation of one matrix by another of lower rank, Psychometrika 1 (3) (1936) 211–218.
[15] I. Markovsky, J. C. Willems, S. Van Huffel, B. De Moor, Exact and Approximate Modeling of Linear Systems, SIAM, Philadelphia, 2006.
[16] B. De Moor, Structured total least squares and L2 approximation problems, Linear Algebra and its Applications 188-189 (1036) (1993) 163–205.
[17] C. L. Lawson, R. J. Hanson, Solving least squares problems, Vol. 161, SIAM, 1974.
[18] Y. D. Sergeyev, Arithmetic of infinity, Edizioni Orizzonti Meridionali CS, 2003.
[19] Y. D. Sergeyev, A new applied approach for executing computations with infinite and infinitesimal quantities, Informatica 19 (4) (2008) 567–596.
[20] Y. D. Sergeyev, Numerical point of view on calculus for functions assuming finite, infinite, and infinitesimal values over finite, infinite, and infinitesimal domains, Nonlinear Analysis: Theory, Methods & Applications 71 (12) (2009) 1688–1707.
[21] Y. D. Sergeyev, Numerical computations and mathematical modelling with infinite and infinitesimal numbers, Journal of Applied Mathematics and Computing 29 (1-2) (2009) 177–195.
[22] G. Lolli, Infinitesimals and infinites in the history of mathematics: A brief survey, Applied Mathematics and Computation 218 (16) (2012) 7979–7988.
[23] A. Zhigljavsky, Computing sums of conditionally convergent and divergent series using the concept of grossone, Applied Mathematics and Computation 218 (16) (2012) 8064–8076.
[24] I. Markovsky, K. Usevich, Software for weighted structured low-rank approximation, J. Comput. Appl. Math. 256 (2014) 278–292.
[25] H. Park, L. Zhang, J. B. Rosen, Low rank approximation of a Hankel matrix by structured total least norm, BIT Numerical Mathematics 39 (4) (1999) 757–779.
[26] S. Van Huffel, Enhanced resolution based on minimum variance estimation and exponential data modeling, Signal Processing 33 (3) (1993) 333–355.
[27] J. A. Cadzow, Signal enhancement: A composite property mapping algorithm, IEEE Trans. on Acoust., Speech, Signal Processing 36 (1988) 1070–1087.
[28] J. Gillard, Cadzow's basic algorithm, alternating projections and singular spectrum analysis, Statistics and Its Interface 3 (3) (2010) 335–343.
[29] F. Andersson, M. Carlsson, Alternating projections on non-tangential manifolds, arXiv:1107.4055.
[30] F. Andersson, M. Carlsson, P.-A. Ivert, A fast alternating projection method for complex frequency estimation, arXiv:1107.2028.
[31] P. Comon, X. Luciani, A. L. F. de Almeida, Tensor decompositions, alternating least squares and other tales, Journal of Chemometrics 23 (2009) 393–405.
[32] N. Golyandina, A. A. Zhigljavsky, Singular Spectrum Analysis for time series, Springer Briefs in Statistics, Springer, 2013.
[33] M. T. Chu, R. E. Funderlic, R. J. Plemmons, Structured low rank approximation, Linear algebra and its applications 366 (2003) 157–172.
[34] J. W. Gillard, A. A. Zhigljavsky, Software for alternating projections with backtracking and randomization, *http://www.jonathangillard.co.uk*.
[35] A. Zhigljavsky, A. Žilinskas, Stochastic global optimization, Springer, New York, 2008.
[36] G. Heinig, K. Rost, Algebraic methods for Toeplitz-like matrices and operators, Springer, 1984.
[37] J. Gillard, A. A. Zhigljavsky, Analysis of Structured Low Rank Approximation as an Optimization Problem, Informatica 22 (4) (2011) 489–505.
[38] R. J. Hyndman, Time series data library, http://data.is/TSDLdemo.
[39] W. A. Fuller, Introduction to statistical time series, 2nd Edition, Wiley & Sons, N.Y., 1996.