NUMERICAL LINEAR ALGEBRA WITH APPLICATIONS
Numer. Linear Algebra Appl. 2010; **00**:1–20
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nla

Weighted norms in subspace-based methods for time series analysis

J. W. Gillard*and A. A. Zhigljavsky

Cardiff School of Mathematics, Cardiff University, Senghennydd Road, Cardiff, UK, CF24 4AG.

SUMMARY

Many modern approaches of time series analysis belong to the class of methods based on approximating high-dimensional spaces by low-dimensional subspaces. A typical method would embed a given time series into a structured matrix and find a low-dimensional approximation to this structured matrix. The purpose of this paper is two-fold: (i) to establish a correspondence between a class of SVD-compatible matrix norms on the space of Hankel matrices and weighted vector norms (and provide methods to construct this correspondence), and (ii) to motivate the importance of this for problems in time series analysis. Examples are provided to demonstrate the merits of judiciously selecting weights on imputing missing data and forecasting in time series.

Received . . .

KEY WORDS: Time series analysis; low-rank approximation; missing values and forecasting

1. INTRODUCTION

Subspace-based methods of time series analysis use the techniques of low-rank approximation of structured matrices. These methods assume that a matrix formed from the time series is of full rank, perhaps due to measurement error or unaccounted effects. This matrix is a perturbation of a 'true' matrix of low rank. The aim of low-rank approximation is to modify the data as little as possible so that the matrix constructed from the modified data has some specified rank. Typical applications are diverse and include dimensionality reduction, feature extraction, and classification, see [1].

In this paper we focus on the so-called Hankel structured low-rank approximation (HSLRA) problem. We form a Hankel structured matrix from a given vector of observations. The rank of this Hankel matrix constructed from the time series corresponds to the complexity of a linear recurrence that fits the time series exactly. The rank of an approximation to this matrix can be considered as a user-choice which allows the user to trade-off accuracy versus complexity. From another perspective we may consider an observed time series to be the sum of a deterministic series plus noise. When embedded into a structured matrix the deterministic series is of low-complexity (small rank), however the structured matrix of the observed series (which has been perturbed by noise) is typically of full rank. Consequently to attempt to separate the noise from the deterministic series we find a lower rank approximation of the observed structured matrix. HSLRA is an important problem and has been applied in a number of areas. In some applications the true rank may be known in advance. Examples include applications in time series analysis, systems and control, computer algebra, signal processing, machine learning and computer vision, see [2], [3], [4], [5], and [6].

*Correspondence to: J. W. Gillard, Cardiff School of Mathematics, Cardiff University, Senghennydd Road, Cardiff, UK, CF24 4AG. E-mail: GillardJW@Cardiff.ac.uk

There has been little work done which shows how to use weights in HSLRA. We wish to consider time series where each observation has an allocated weight. This weight can be used for example to denote an observation as 'missing', or to denote an observation as 'important' (representing 'confidence' in this observation) and warrants additional leverage upon the fitted model. We may even wish to classify an observation as 'exact'. For forecasting, we may wish to down-weigh points at the start of the time series, and up-weigh the more recent observations towards the end of a time series. The weighting may also be used as a tool for correcting an unintentional weighting which has arisen from embedding the time series into a structured matrix. For example, suppose we embed the time series of three observations (a,b,c) into the 2×2 Hankel matrix $\begin{pmatrix} a & b \\ b & c \end{pmatrix}$. Observations a and c appear once whilst observation b appears twice, see Section 2 for the general case. This issue has already been noted as a problem, the unintended consequences of which have been described in [7], [8], [9], and [10].

The paper has the following structure. We formally introduce the problem considered in this paper in Section 2 and also state the main result. Low-rank approximations of matrices are described in Section 3. In Section 4 we describe how to relate vector and matrix norms and we provide a numerical algorithm in Section 5. An example of how the material developed in Sections 4 and 5 can be used to develop methodology for modelling in time series, namely for imputing missing values and forecasting is given in Section 6. A number of examples are given in Section 7.

2. NOTATION, PROBLEM STATEMENT, AND THE MAIN RESULT

2.1. Notation

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

N, L, K, r	Positive integers with $1\leq r\leq L\leq K<N,\ N=L+K$
$\mathbb{R}^{N+1}$	Set of vectors $Y=(y_0,y_1,\ldots,y_N)^T$ of length $N+1$
$\mathbb{R}^{u\times v}$	Set of all real-valued $u\times v$ matrices, for some positive u and v
$\mathbb{M}_u^{>}$	Set of all symmetric positive definite matrices of size $u\times u$
$\mathcal{H}$	Set of all Hankel matrices of size $(L+1)\times(K+1)$
$\mathcal{M}_r$	Set of all $(L+1)\times(K+1)$ matrices of rank $\leq r$
$\mathbb{L}_{\leq r}\subset\mathbb{R}^{N+1}$	The set of vectors in $\mathbb{R}^{N+1}$ satisfying a linear recurrence relation of order $\leq r$
$\mathcal{A}=\mathcal{M}_r\cap\mathcal{H}$	Set of all $(L+1)\times(K+1)$ Hankel matrices of rank $\leq r$
$Y=(y_0,\ldots,y_N)^T$	Given vector in $\mathbb{R}^{N+1}$
$\mathbf{X}=\mathbb{H}(Y)\in\mathcal{H}$	Hankel matrix associated with Y

2.2. Problem statement

Assume we are given a vector $Y=(y_0,\ldots,y_N)^T\in\mathbb{R}^{N+1}$. The problem we consider is

$$\rho(S,Y)\to\min_{S\in\mathbb{L}_{\leq r}} \tag{1}$$

where $\rho(\cdot,\cdot)$ is a distance on $\mathbb{R}^{(N+1)}\times\mathbb{R}^{(N+1)}$ and $\mathbb{L}_{\leq r}\subset\mathbb{R}^{N+1}$ is the set of vectors in $\mathbb{R}^{N+1}$ which satisfy a linear recurrence relation (LRR) of order $\leq r$; we say that a vector $S=(s_0,\ldots,s_N)^T$ satisfies an LRR of order $\leq r$ if

$$s_n=a_1s_{n-1}+\ldots+a_rs_{n-r},\quad \text{for all}\ \ n=r,r+1,\ldots,N\,, \tag{2}$$

where $a_1,\ldots,a_r$ are some real numbers with $a_r\neq 0$. The model (2) includes, as a special case, the model of a sum of exponentially damped sinusoids:

$$s_n=\sum_{\ell=1}^{q}a_\ell\exp\left(d_\ell n\right)\sin\left(2\pi\omega_\ell n+\phi_\ell\right),\ n=0,\ldots,N\,, \tag{3}$$

DOI: 10.1002/nla

where typically $q = r/2$, see for example [11], [12] or [13] .

A solution to problem (1) would yield a representation of the observed data $Y = (y_0, \ldots, y_N)^T$ in the form

$$y_n = s_n + \varepsilon_n,\ n = 0, \ldots, N,$$

where $S = (s_0, s_1, \ldots, s_N)^T \in \mathbb{R}^{N+1}$ is an unobserved signal satisfying the model (2) and $(\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_N)^T$ is a vector of noise, where ε_n are not necessarily i.i.d.r.v.

The optimization problem (1) can be equivalently formulated as a matrix optimization problem, where vectors in (1) are represented by $(L+1)\times(K+1)$ Hankel matrices. With a vector $Z = (z_0, z_1, \ldots, z_N)^T$ of size $N+1$ and given $L < N$ we associate an $(L+1)\times(K+1)$ Hankel matrix

$$\mathbf{X}_Z = \begin{pmatrix} z_0 & z_1 & \cdots & z_K \\ z_1 & z_2 & \cdots & z_{K+1} \\ \vdots & \vdots & \vdots & \vdots \\ z_L & z_{L+1} & \cdots & z_N \end{pmatrix} \in \mathcal{H}\,,$$

where $K = N - L$. We also write this matrix as $\mathbf{X}_Z = \mathbb{H}(Z)$ and note that $\mathbb{H}$ makes a one-to-one correspondence between the spaces $\mathbb{R}^{N+1}$ and $\mathcal{H}$ so that for any matrix $\mathbf{X} \in \mathcal{H}$ we may uniquely define $Z = \mathbb{H}^{-1}(\mathbf{X})$ with $\mathbf{X} = \mathbb{H}(Z)$.

The matrix version of the optimization problem (1) can now be written as

$$d(\mathbf{X}, \mathbf{X}_Y) \to \min_{\mathbf{X}\in\mathcal{A}} \tag{4}$$

where $d(\cdot,\cdot)$ is a distance on $\mathbb{R}^{(L+1)\times(K+1)} \times \mathbb{R}^{(L+1)\times(K+1)}$ and $\mathcal{A}$ is the set of all $(L+1)\times(K+1)$ Hankel matrices of rank $\le r$. The optimization problem (4) is the general problem of Hankel structured low-rank approximation (HSLRA).

The optimization problems (1) and (4) are equivalent if the distance functions $\rho(\cdot,\cdot)$ in (1) and $d(\cdot,\cdot)$ in (4) are such that

$$\rho(Z, Z') = c \cdot d(\mathbb{H}(Z), \mathbb{H}(Z')) \tag{5}$$

for $Z, Z' \in \mathbb{R}^{N+1}$, where $c > 0$ is arbitrary. It can be assumed that $c = 1$ without loss of generality.

Natural approaches for solving the initial optimization problem (4) would use global optimization techniques for optimizing parameters in either representation (2) or (3). In the case of (2), the parameters are the coefficients of the LRR: $a_1, \ldots, a_r$, and the initial values $s_0, \ldots, s_{r-1}$. If we were to use (3) then the set of parameters is $\{(a_\ell, d_\ell, \omega_\ell, \phi_\ell), \ell = 1, \ldots, q\}$. In both cases, the parametric optimization problem is extremely difficult with multi-extremality and large Lipschitz constants of the objective functions [14]. The number of local minima is known to increase linearly with the number of observations. Many of the existing algorithms depend on local optimization based algorithms and do not move significantly from a starting point, see [12], [15], and [16]. The difficulty of solving parametric versions of (1) is well understood and that is the reason why HSLRA described by (4), the equivalent matrix formulation of (1), is almost always considered instead of (1). As already stated, there is little work in the literature describing how to use weights in HSLRA. However, some recent work has commented that even unstructured weighted low-rank approximation is difficult, see [9].

The standard choice of the distance $d(\cdot,\cdot)$ in the HSLRA problem (4) is $d(\mathbf{X}, \mathbf{X}') = \|\mathbf{X} - \mathbf{X}'\|_F$, where $\|\cdot\|_F$ is the Frobenius norm. The primary reason for this choice is the availability of the singular value decomposition (SVD) which is considered in Section 3 and constitutes the essential part of many algorithms attempting to solve the HSLRA problem (4). One of the most popular methods is given in Section 6 (see (24)).

However, if the distance $d(\mathbf{X}, \mathbf{X}')$ in (4) is $d(\mathbf{X}, \mathbf{X}') = \|\mathbf{X} - \mathbf{X}'\|_F$ then the distance ρ in (1) takes a particular form. In this case,

$$d^2(\mathbb{H}(Z), \mathbb{H}(Z')) = \|\mathbf{X}_Z - \mathbf{X}_{Z'}\|_F^2 = \sum_{n=0}^{N} \kappa_n (z_n - z'_n)^2\,, \tag{6}$$

DOI: 10.1002/nla

where

$$\kappa_n = \begin{cases} n+1, & \text{if } 0 \le n < L, \\ L+1, & \text{if } L \le n \le N-L, \\ N+1-n, & \text{if } N-L < n \le N. \end{cases} \tag{7}$$

One would prefer to define the distance function $\rho(\cdot,\cdot)$ in (1) and acquire the distance $d(\cdot,\cdot)$ for (4) from (5), rather than vice-versa, which is a common practice. Different distances $\rho(\cdot,\cdot)$ can be used and may be desired. There is one serious problem, however, related to the complexity of the resulting HSLRA problem (4). If $d(\cdot,\cdot)$ is defined by the Frobenius norm then the HSLRA problem (4), despite being difficult, is still considered as solvable since there is a very special tool available at intermediate stages, the SVD. If $d(\cdot,\cdot)$ does not allow the use of SVD or similar tools then the HSLRA problem (4) becomes practically unsolvable (except, of course, for some very simple cases).

The purpose of this paper is to extend the choice of the norms that define distances in (4) and (1) preserving the availability of the SVD. These norms allow the construction of exactly the same algorithms that can be constructed for the Frobenius norm and, since the family of the norms considered is rather wide, we will be able to exactly or approximately match any given distance in (1). More precisely, we will consider the class of distances in (1) of the form $\rho(Z,Z') = \|Z-Z'\|_{\mathbf{W}}$ with

$$\|Z\|_{\mathbf{W}}^2 = Z^T\mathbf{W}Z = \sum_{n=0}^{N} w_n z_n^2\,, \tag{8}$$

where $Z = (z_0,\ldots,z_N)^T \in \mathbb{R}^{N+1}$ and $\mathbf{W} = \mathrm{diag}(W) \in \mathbb{M}_{N+1}^{>}$ with diagonal $W = (w_0,w_1,\ldots,w_N)^T$, $w_n > 0$ for all $n = 0,1,\ldots,N$. The class of distances in (4) considered in this paper has the form $d(\mathbf{X},\mathbf{X}') = ||\mathbf{X}-\mathbf{X}'||_{\mathbf{Q},\mathbf{R}}$ with

$$||\mathbf{X}||_{\mathbf{Q},\mathbf{R}}^2 = \mathrm{Trace}\,\mathbf{Q}\mathbf{X}\mathbf{R}\mathbf{X}^T\,, \tag{9}$$

where $\mathbf{Q} \in \mathbb{M}_{L+1}^{>}$ and $\mathbf{R} \in \mathbb{M}_{K+1}^{>}$ are diagonal positive definite matrices. That is $\mathbf{Q} = \mathrm{diag}(Q)$, $\mathbf{R} = \mathrm{diag}(R)$ where $Q = (q_0,q_1,\ldots,q_L)^T$, and $R = (r_0,r_1,\ldots,r_K)^T$. We will call this norm the $(\mathbf{Q},\mathbf{R})$-norm. Note that if $\mathbf{Q}$ and $\mathbf{R}$ are identity matrices then (9) defines the Frobenius norm.

In Section 3 we will show that low-rank approximation problem in the $(\mathbf{Q},\mathbf{R})$-norm can be reduced to the low-rank approximation problem in the Frobenius norm which is solved by applying the SVD.

2.3. *Relation between the weighted vector norm and the $(\mathbf{Q},\mathbf{R})$-norm*

Theorem 1. *Consider the matrix norm (9) for a Hankel matrix $\mathbf{X}_Z = \mathbb{H}(Z)$, where $Z \in \mathbb{R}^{N+1}$ is arbitrary. Assume that the matrices $\mathbf{Q}$ and $\mathbf{R}$ in (9) are diagonal; that is, $\mathbf{Q} = \mathrm{diag}(Q)$ and $\mathbf{R} = \mathrm{diag}(R)$, where $Q = (q_0,q_1,\ldots,q_L)^T \in \mathbb{R}^{L+1}$ and $R = (r_0,r_1,\ldots,r_K)^T \in \mathbb{R}^{K+1}$. Then*

$$||\mathbf{X}_Z||_{\mathbf{Q},\mathbf{R}}^2 = ||Z||_{\mathbf{W}}^2 = Z^T\mathbf{W}Z\,, \tag{10}$$

where $\mathbf{W} = \mathrm{diag}(W)$ and the vector $W = (w_0,\ldots,w_N)^T \in \mathbb{R}^{N+1}$ is the convolution of the vectors Q and R; that is,

$$W = Q \star R \tag{11}$$

or, equivalently,

$$w_n = \sum_{l=\max\{0,n-K\}}^{\min\{n,L\}} q_l r_{n-l} = \sum_{k=\max\{0,n-L\}}^{\min\{n,K\}} q_{n-k} r_k\,; \quad n = 0,1,\ldots,N\,. \tag{12}$$

DOI: 10.1002/nla

Proof. Consider the norm $||\mathbf{X}_Z||^2_{\mathbf{Q},\mathbf{R}}$ with $\mathbf{X}_Z = \mathbb{H}(Z)$ (so that $x_{l,k} = z_{l+k}$ for $l = 0, \ldots, L$ and $k = 0, \ldots, K$), $\mathbf{Q} = \mathrm{diag}(Q)$ and $\mathbf{R} = \mathrm{diag}(R)$. We have

$$||\mathbf{X}_Z||^2_{\mathbf{Q},\mathbf{R}} = \mathrm{Trace}\, \mathbf{Q}\mathbf{X}_Z\mathbf{R}\mathbf{X}_Z^T = \sum_{l=0}^{L}\sum_{k=0}^{K} q_l x_{l,k} r_k x_{l,k} = \sum_{l=0}^{L}\sum_{k=0}^{K} q_l r_k z^2_{l+k} \tag{13}$$

If in the rhs of (13) we set $n = k + l$, change the summation index k to n, change the order of summation (with respect to l and n) and equate the coefficients with z_n^2 in the rhs of (13) and the rhs of (8), then we obtain the first equality in (12). Similarly, we obtain the second equality in (12) if we change the summation index l to n in the rhs of (13). □

Equations (12) and (5) imply the following relation between the distances associated with the norms (8) and (9) which respectively appear in (1) and (4) for diagonal $\mathbf{Q}$, $\mathbf{R}$ and $\mathbf{W}$:

$$\rho_{\mathbf{W}}(S, Y) = ||S - Y||_{\mathbf{W}} = \|\mathbf{X}_S - \mathbf{X}_Y||_{\mathbf{Q},\mathbf{R}} = d_{\mathbf{Q},\mathbf{R}}(\mathbf{X}_S, \mathbf{X}_Y)\,.$$

2.4. Different forms for computing the convolution $Q \star R$

We now give two different forms for expressing the convolution $Q \star R$ in Theorem 1 and provide a generating function based expression for the convolution equation $W = Q \star R$.

- Let $\mathbf{C_Q} \in \mathbb{R}^{(N+1)\times(K+1)}$ and $\mathbf{C_R} \in \mathbb{R}^{(N+1)\times(L+1)}$ be two matrices defined by

$$\mathbf{C_Q} = \begin{pmatrix} q_0 & 0 & \cdots & 0 \\ \vdots & q_0 & \ddots & \vdots \\ q_L & \vdots & \ddots & 0 \\ 0 & q_L & & q_0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & q_L \end{pmatrix}, \quad \mathbf{C_R} = \begin{pmatrix} r_0 & 0 & \cdots & 0 \\ \vdots & r_0 & \ddots & \vdots \\ r_K & \vdots & \ddots & 0 \\ 0 & r_K & & r_0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & r_K \end{pmatrix}. \tag{14}$$

  Then it follows from (12) that

$$W = \mathbf{C_Q} R = \mathbf{C_R} Q\,. \tag{15}$$

- Define $q_l = 0$ for $l = -1, -2, \ldots$ and $l = L+1, L+2, \ldots$. Also, define $r_k = 0$ for $k = -1, -2, \ldots$ and $k = K+1, K+2, \ldots$. Then the convolution equations (12) can be written simpler as

$$w_n = \sum\nolimits_{l=0}^{L} q_l r_{n-l} = \sum\nolimits_{k=0}^{K} q_{n-k} r_k\,, \quad n = 0, 1, \ldots, N\,. \tag{16}$$

- Define the generating functions of the sequences Q, R and W:

$$Q(t) = \sum_{l=0}^{L} q_l t^l\,, \quad R(t) = \sum_{k=0}^{K} r_k t^k\,, \quad W(t) = \sum_{n=0}^{N} w_n t^n\,. \tag{17}$$

  Then the convolution equation $W = Q \star R$ is equivalent to the equality

$$W(t) = Q(t)R(t) \quad \text{for all} \quad t \in \mathbb{R}. \tag{18}$$

3. LOW-RANK APPROXIMATIONS

In Section 3.1 below we describe a classic result of finding optimal rank r approximations to given matrices in the Frobenius norm, which is the unstructured low-rank approximation problem. Then in Section 3.2 we show how to construct optimal rank r approximations in the $(\mathbf{Q}, \mathbf{R})$ norm.

DOI: 10.1002/nla

3.1. Low-rank approximation in the Frobenius norm.

Let $\mathbf{A}$ be a real matrix of size $(L+1)\times(K+1)$ with $L \leq K$. The singular value decomposition (SVD) of $\mathbf{A}$ is a factorization of the form $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where $\mathbf{U}$ and $\mathbf{V}$ are real matrices of sizes $(L+1)\times(L+1)$ and $(K+1)\times(K+1)$ respectively and $\mathbf{\Sigma}$ is a $(L+1)\times(K+1)$ matrix with real numbers on the diagonal. The numbers $\mathbf{\Sigma}_{i,i} = \sigma_i$ $(i = 0,\ldots,L)$ are called the singular values of $\mathbf{A}$ and are assumed to be arranged in the order of descent so that $\sigma_0 \geq \sigma_1 \geq \ldots \geq \sigma_L$.

Lemma 1. (Low-rank approximation of $\mathbf{A}$ in the Frobenius norm [17]) *Let $\mathbf{A} \in \mathbb{R}^{(L+1)\times(K+1)}$ be a given matrix and let $\mathbf{\Sigma}^{(r)}$ be a matrix of size $(L+1)\times(K+1)$ with elements*

$$\mathbf{\Sigma}^{(r)}_{i,j} = \begin{cases} \sigma_i & \text{if } 0 \leq i \leq r \text{ and } j = i \\ 0 & \text{otherwise.} \end{cases}$$

Then the solution to the low-rank approximation problem

$$||\mathbf{A} - \mathbf{B}||^2_F \to \min_{\mathbf{B}:rank(\mathbf{B})\leq r}$$

is given by the matrix

$$\mathbf{A}^{(r)} = \mathbf{U}\mathbf{\Sigma}^{(r)}\mathbf{V}^T = \operatorname*{argmin}_{\mathbf{B}:rank(\mathbf{B})\leq r} ||\mathbf{A} - \mathbf{B}||^2_F .$$

3.2. Low-rank approximation in the $(\mathbf{Q},\mathbf{R})$-norm.

Lemma 2. (Low-rank approximation of $\mathbf{A}$ in the $(\mathbf{Q},\mathbf{R})$-norm.) *Consider the squared matrix norm (9) for some given matrices $\mathbf{A} \in \mathbb{R}^{(L+1)\times(K+1)}$, $\mathbf{Q} \in \mathbb{M}^{>}_{L+1}$ and $\mathbf{R} \in \mathbb{M}^{>}_{K+1}$.*

Then the low-rank approximation problem

$$||\mathbf{A} - \mathbf{B}||^2_{\mathbf{Q},\mathbf{R}} \to \min_{\mathbf{B}:rank(\mathbf{B})=r} \tag{19}$$

is equivalent to the low-rank approximation problem

$$||\mathbf{Q}^{1/2}(\mathbf{A} - \mathbf{B})\mathbf{R}^{1/2}||^2_F \to \min_{\mathbf{B}:rank(\mathbf{B})=r} . \tag{20}$$

Proof. By the definition of the (**Q**,**R**) norm (9),

$$\begin{aligned} ||\mathbf{A} - \mathbf{B}||^2_{\mathbf{Q},\mathbf{R}} = \operatorname{Trace} \mathbf{Q}(\mathbf{A} - \mathbf{B})\mathbf{R}(\mathbf{A} - \mathbf{B})^T &= \operatorname{Trace} \mathbf{Q}^{1/2}(\mathbf{A} - \mathbf{B})\mathbf{R}^{1/2}(\mathbf{Q}^{1/2}(\mathbf{A} - \mathbf{B})\mathbf{R}^{1/2})^T \\ &= ||\mathbf{Q}^{1/2}(\mathbf{A} - \mathbf{B})\mathbf{R}^{1/2}||^2_F . \end{aligned}$$

□

Theorem 2. (Solution of the low-rank approximation problem (19)) *Consider the squared matrix norm (9) for some given matrices $\mathbf{A} \in \mathbb{R}^{(L+1)\times(K+1)}$, $\mathbf{Q} \in \mathbb{M}^{>}_{L+1}$ and $\mathbf{R} \in \mathbb{M}^{>}_{K+1}$. Let $\tilde{\mathbf{A}} = \mathbf{Q}^{1/2}\mathbf{A}\mathbf{R}^{1/2}$ and let $\tilde{\mathbf{A}} = \tilde{\mathbf{U}}\tilde{\Sigma}^{(r)}\tilde{\mathbf{V}}^T$ be the rank r approximation of $\tilde{\mathbf{A}}$ as given by Lemma 1. Then the (globally optimal) solution to (20) is given by $\hat{\mathbf{A}} = \hat{\mathbf{U}}\hat{\Sigma}^{(r)}\hat{\mathbf{V}}^T$ where $\hat{\mathbf{U}} = (\mathbf{Q}^{1/2})^{-1}\tilde{\mathbf{U}}$, $\hat{\mathbf{V}} = (\mathbf{R}^{1/2})^{-1}\tilde{\mathbf{V}}$, and $\hat{\mathbf{\Sigma}}^{(r)} = \tilde{\mathbf{\Sigma}}^{(r)}$.*

Proof. Follows directly from Lemmas 1 and 2, see also [18]. □

Note that as $\mathbf{Q}$ and $\mathbf{R}$ are assumed to be symmetric positive definite matrices then $\mathbf{Q}^{1/2}$ and $\mathbf{R}^{1/2}$ always exist, as do their inverses.

4. CONSTRUCTING Q AND R SUCH THAT $W = Q \star R$

In view of the equivalence of (10) and (11) and the fact that the weight vector W of the vector norm $\sqrt{Y^T\mathbf{W}Y}$ is assumed to be given, the main problem now is to find vectors Q and R such that the convolution $W = Q \star R$ holds either exactly or approximately.

DOI: 10.1002/nla

4.1. Solving $W = Q \star R$ exactly

In this section, we assume that the vector $W = (w_0, w_1, \ldots, w_N)^T$ with non-negative entries w_n is given and we need to find conditions on W, L and K so that we can decompose W as a convolution $W = Q \star R$, where $Q \in \mathbb{R}^{L+1}$ and $R \in \mathbb{R}^{K+1}$. Ideally, both vectors, Q and R, should have positive entries in which case the expression (9) defines a proper norm.

Our key reference point will be equation (18) for the generating functions $W(t)$, $Q(t)$ and $R(t)$. These generating functions defined in (17) are polynomials of degrees N, L and $K = N - L$ correspondingly.

If we would allow negative entries for the vectors Q and R, then the answer to the main question of the existence of Q and R so that $W = Q \star R$ is very simple for any W. If the function $W(t) = \prod_{n=0}^{N}(t - t_n)$ has at least one real root, then the decomposition $W(t) = Q(t)R(t)$ can be done in a number of different ways for any $L < N$. If, however, N is even and all the roots of $W(t)$ are complex then L (and hence K) must be even (in this case, for odd L the decomposition $W = Q \star R$ is impossible).

To extend this simple observation, the answer to the main question of the existence of Q and R so that $W = Q \star R$, remains to be generally positive if we only allow one of the two vectors (either Q or R) to have negative entries. The answer depends on how many real roots the generating function $W(t)$ has. More precisely, according to [1], if a polynomial has non-negative coefficients and some complex roots, then one can always find a pair of conjugate complex roots so that after factoring the corresponding second-degree polynomial out of the original one, the coefficients of the remaining polynomial stay non-negative. In our notation this reads as follows. If $W(t)$ has non-negative coefficients and p pairs of conjugate complex roots, then we can always find a representation $W(t) = Q(t)R(t)$, where $Q(t)$ has even degree $L \le 2p$ and all the coefficients of $R(t)$ are non-negative.

4.2. Assumption of non-negativity for all components of Q and R

It is more difficult to answer the question of whether Q and R exist such that $W = Q \star R$ if we require the non-negativity for all components of Q and R.

4.2.1. $W = (1, 1, \ldots, 1)^T$ In this section we assume $W = (1, 1, \ldots, 1)^T$ so that

$$W(t) = 1 + t + t^2 + \ldots + t^N = (1 - t^{N+1})/(1 - t)\,.$$

The n-th cyclotomic polynomial, for any positive integer n, is the unique irreducible polynomial with integer coefficients, which is a divisor of $1 - x^n$ and is not a divisor of $1 - x^k$ for any $k < n$. Its roots are the n-th primitive roots of unity $e^{2i\pi k/n}$, where k runs over the integers smaller than n and co-prime to n. In other words, the n-th cyclotomic polynomial is equal to

$$\Phi_n(x) = \prod_{1 \le k \le n,\, \gcd(k,n)=1} \left(x - e^{2i\pi k/n}\right)$$

If $W(t) = Q(t)R(t)$, where $Q(t)$ and $R(t)$ are polynomials with non-negative coefficients of degree L and K correspondingly, then we can always assume that $L \le K$,

$$Q(t) = 1 + a_1 t + a_2 t^2 + \ldots + a_{L-1} t^{L-1} + t^L \text{ and } R(t) = 1 + b_1 t + \ldots + b_{K-1} t^{K-1} + t^K\,,$$

where $a_l, b_k \in \{0, 1\}$, for $l = 1, \ldots, L$ and $b_k \in \{0, 1\}$ for $k = 1, \ldots, K$.

Since $a_l, b_k \in \{0, 1\}$ and for $1 < L \le K$ not all these coefficients are 1, we deduce that there are no solutions of the equation $W(t) = Q(t)R(t)$ where all coefficients are positive. There are, however, many combinations of N and L when there are solutions of the equation $W(t) = Q(t)R(t)$ where all coefficients are non-negative.

All cyclotomic polynomials are palindromic and hence the polynomials $Q(t)$ and $R(t)$ are palindromic too; this means $a_l = a_{L-l}$ for all $l = 1, \ldots, L$ and $b_k = b_{K-k}$ for all $k = 1, \ldots, K$.

DOI: 10.1002/nla

If N is not very large then with the help of these relations we can enumerate all solutions of the equation $W(t) = Q(t)R(t)$, if these solutions exist.

In general, it does not seem to be possible to enumerate all combinations of N and L where solutions to the equation $W(t) = Q(t)R(t)$ exist, where all coefficients of the polynomials are non-negative. In the examples below we consider some particular cases where the existence of solutions of the equation $W(t) = Q(t)R(t)$ can be established easily. At the end of this section we provide a table of all combinations of L and N (with $N < 50$) such that this solution exists.

Example 1. Assume that $N+1$ is divisible by $L+1$. In this case, $A = (N+1)/(L+1) - 1$ is an integer which gives us $K = N - L = A(L+1)$ and

$$1 + t + t^2 + \ldots + t^N = (1 + t + t^2 + \ldots + t^L) \cdot (1 + t^{L+1} + t^{2(L+1)} + \ldots + t^{A(L+1)})$$

This corresponds to the equality $W = Q \star R$ with $Q = (1, 1, \ldots, 1)^T \in \mathbb{R}^{L+1}$ and $R = (r_1, \ldots, r_K)^T$ such that

$$r_k = \begin{cases} 1, & \text{if } k = j(L+1); j = 0, 1, \ldots, A, \\ 0, & \text{otherwise}. \end{cases}$$

Similarly, let $N+1$ be divisible by $K+1$ and set $B = (N+1)/(K+1) - 1$. In this case, $L = B(K+1)$ and

$$1 + t + t^2 + \ldots + t^N = (1 + t + t^2 + \ldots + t^K) \cdot (1 + t^{K+1} + t^{2(K+1)} + \ldots + t^{B(K+1)}).$$

This corresponds to the equality $W = Q \star R$ with $R = (1, 1, \ldots, 1)^T \in \mathbb{R}^{K+1}$ and $Q = (q_1, \ldots, q_L)^T$ such that

$$q_l = \begin{cases} 1, & \text{if } l = j(K+1); j = 0, 1, \ldots, B, \\ 0, & \text{otherwise}. \end{cases}$$

Example 2. For some particular values of N, L and K there are several choices of Q and R with non-negative coefficients so that $W = Q \star R$. Indeed, let $N = 11$, $L = 3$, $K = 8$, $W = (1, 1, \ldots, 1)^T$. We have

$$1 + t + t^2 + \ldots + t^{11} = (1 + t + t^2 + t^3)(1 + t^4 + t^8) = (1 + t^3)(1 + t + t^2 + t^6 + t^7 + t^8).$$

This implies that $W = Q \star R$ with

$$\{Q^T, R^T\} = \begin{cases} \{(1,1,1,1), (1,0,0,0,1,0,0,0,1)\} \\ \quad \text{or} \\ \{(1,0,0,1), (1,1,1,0,0,0,1,1,1)\} \end{cases}$$

From Example 1 we observe that if $N+1$ is composite then we can always find vectors Q and R with non-negative coefficients providing the convolution $W = Q \star R$. On the contrary, if $N+1$ is prime then, as we see in the next example, there are no vectors Q and R providing $W = Q \star R$.

Example 3. If $L > 1$, $K > 1$ and $N + 1 = p$ is prime then there are no vectors Q and R with non-negative coefficients providing the convolution $W = Q \star R$.

The roots of $W(t)$ are

$$t_{m,N} = \exp\{2\pi m I/p\} = \cos(2\pi m/p) + I \sin(2\pi m/p), \quad m = 1, 2, \ldots, N.$$

All roots are complex and there are $N/2$ pairs of conjugate roots $\{t_{m,N}, t_{p-m,N}\}$ $(m = 1, \ldots, N/2)$ so that $W(t) = \prod_{m=1}^{N/2} P_m(t)$, where

$$P_m(t) = (t - t_{m,N})(t - t_{p-m,N}) = 1 - 2t\cos(2\pi m/p) + t^2.$$

DOI: 10.1002/nla

The polynomial $Q(t)$, therefore, is a product of $L/2$ polynomials $P_m(t)$, where m belongs to $I_L = \{m_1, \ldots, m_{L/2}\}$, a subset of $\{1, \ldots, N/2\}$ containing $L/2$ different numbers $\leq N/2$.

It is known that $Q(t)$ has the form $Q(t) = 1 + a_1 t + \ldots + t^L$, where $a_1 \in \{0, 1\}$. The term t in the product $Q(t) = \prod_{m \in I_L} P_m(t)$ has the coefficient

$$A_1 = -2 \sum_{m_i \in I_L} \cos(2\pi m_i / p)\,.$$

Since p is a prime number, all numbers $\cos(2\pi m/p)$ $(m = 1, \ldots, N/2)$ are irrational and are linearly independent over the set of rational numbers. This implies that A_1 cannot possibly be any rational number, including 0 and $1/2$. This makes a contradiction with the existence of the vectors Q and R with non-negative coefficients providing the convolution $W = Q \star R$.

The pairs $[L, N]$ such that the solution to $W = Q \star R$ exists Below we provide all the pairs $[L+1, N+1]$ with $N < 50$ so that the solution to $W = Q \star R$ exists for some Q and R. We provide the pairs $[L+1, N+1]$ rather than $[L, N]$ as it is easier to see the divisibility properties of $N+1$. The list of these pairs (ordered with respect to the value of L) is:

$[2,4]$, $[2,6]$, $[2,8]$, $[2,10]$, $[2,12]$, $[2,14]$, $[2,16]$, $[2,18]$, $[2,20]$, $[2,22]$, $[2,24]$, $[2,26]$, $[2,28]$, $[2,30]$, $[2,32]$, $[2,34]$, $[2,36]$, $[2,38]$, $[2,40]$, $[2,42]$, $[2,44]$, $[2,46]$, $[2,48]$, $[2,50]$, $[3,6]$, $[3,8]$, $[3,9]$, $[3,12]$, $[3,15]$, $[3,16]$, $[3,18]$, $[3,20]$, $[3,21]$, $[3,24]$, $[3,27]$, $[3,28]$, $[3,30]$, $[3,32]$, $[3,33]$, $[3,36]$, $[3,39]$, $[3,40]$, $[3,42]$, $[3,44]$, $[3,45]$, $[3,48]$, $[4,8]$, $[4,12]$, $[4,16]$, $[4,18]$, $[4,20]$, $[4,24]$, $[4,28]$, $[4,30]$, $[4,32]$, $[4,36]$, $[4,40]$, $[4,42]$, $[4,44]$, $[4,48]$, $[5,10]$, $[5,12]$, $[5,15]$, $[5,16]$, $[5,18]$, $[5,20]$, $[5,24]$, $[5,25]$, $[5,30]$, $[5,32]$, $[5,35]$, $[5,36]$, $[5,40]$, $[5,42]$, $[5,45]$, $[5,48]$, $[5,50]$, $[6,12]$, $[6,16]$, $[6,18]$, $[6,20]$, $[6,24]$, $[6,30]$, $[6,32]$, $[6,36]$, $[6,40]$, $[6,42]$, $[6,48]$, $[6,50]$, $[7,14]$, $[7,16]$, $[7,18]$, $[7,21]$, $[7,24]$, $[7,27]$, $[7,28]$, $[7,32]$, $[7,35]$, $[7,36]$, $[7,40]$, $[7,42]$, $[7,45]$, $[7,48]$, $[7,49]$, $[8,16]$, $[8,24]$, $[8,28]$, $[8,32]$, $[8,36]$, $[8,40]$, $[8,42]$, $[8,48]$, $[9,18]$, $[9,20]$, $[9,24]$, $[9,27]$, $[9,30]$, $[9,32]$, $[9,36]$, $[9,40]$, $[9,45]$, $[9,48]$, $[9,50]$, $[10,20]$, $[10,24]$, $[10,30]$, $[10,32]$, $[10,36]$, $[10,40]$, $[10,48]$, $[10,50]$, $[11,22]$, $[11,24]$, $[11,30]$, $[11,32]$, $[11,33]$, $[11,36]$, $[11,40]$, $[11,44]$, $[11,45]$, $[11,48]$, $[12,24]$, $[12,32]$, $[12,36]$, $[12,40]$, $[12,44]$, $[12,48]$, $[13,26]$, $[13,28]$, $[13,30]$, $[13,32]$, $[13,36]$, $[13,39]$, $[13,42]$, $[13,45]$, $[13,48]$, $[14,28]$, $[14,32]$, $[14,36]$, $[14,42]$, $[14,48]$, $[15,30]$, $[15,32]$, $[15,36]$, $[15,40]$, $[15,42]$, $[15,45]$, $[15,48]$, $[16,32]$, $[16,36]$, $[16,40]$, $[16,48]$, $[17,34]$, $[17,36]$, $[17,40]$, $[17,48]$, $[18,36]$, $[18,40]$, $[18,48]$, $[19,38]$, $[19,40]$, $[19,42]$, $[19,48]$, $[20,40]$, $[20,48]$, $[21,42]$, $[21,44]$, $[21,48]$, $[21,50]$, $[22,44]$, $[22,48]$, $[23,46]$, $[23,48]$, $[24,48]$, $[25,50]$.

4.2.2. $W = (1, \beta, \ldots, \beta^N)^T$ *with* $\beta \neq 0$ This case is very similar to the case when W is the vector $(1, 1, \ldots, 1)^T$; the case considered in the previous subsection. In this case, the generating function of $W = (1, \beta, \ldots, \beta^N)^T$ is $W(t) = \sum_{n=0}^N \beta^n t^n = \widetilde{W}(\beta t)$, where $\widetilde{W}(t) = \sum_{n=0}^N t^n$ is the generating function of $\widetilde{W} = (1, 1, \ldots, 1)^T \in \mathbb{R}^{N+1}$. Consequently, if $\widetilde{W} = (1, 1, \ldots, 1)^T = \widetilde{Q} \star \widetilde{R}$ for some $\widetilde{Q} = (\tilde{q}_0, \tilde{q}_1, \ldots, \tilde{q}_L)^T$ and $\widetilde{R} = (\tilde{r}_0, \tilde{r}_1, \ldots, \tilde{r}_K)^T$ then $W = Q \star R$ for $Q = (\tilde{q}_0, \beta\tilde{q}_1, \ldots, \beta^L \tilde{q}_L)^T$ and $R = (\tilde{r}_0, \beta\tilde{r}_1, \ldots, \beta^K \tilde{r}_K)^T$. The practical application of the weight vector $W = (1, \beta, \ldots, \beta^N)^T$ can be related to long time series which we wish to forecast, putting more weight on the latest observations.

5. NUMERICAL METHODS FOR SOLVING $W \simeq Q \star R$

As mentioned above, the equation $W \simeq Q \star R$ with respect to Q and R can easily be solved if there are no constraints on the elements of Q and R. However, as Q and R have to define the norm (9), all elements of Q and R have to be positive. Let α (the lower bound on the values of elements of Q and R) be fixed, $0 < \alpha < 1$, and we will seek for $\hat{Q}$ and $\hat{R}$ such that

$$(\hat{Q}_\alpha, \hat{R}_\alpha) = \underset{Q \in \mathcal{Q}_\alpha,\ R \in \mathcal{R}_\alpha}{\operatorname{argmin}} \|W - Q \star R\|_2^2 = \underset{Q \in \mathcal{Q}_\alpha,\ R \in \mathcal{R}_\alpha}{\operatorname{argmin}} \sum_{n=0}^{N} \left(w_n - \sum_{l=\max\{0, n-K\}}^{\min\{n, L\}} q_l r_{n-l} \right)^2 . \tag{21}$$

DOI: 10.1002/nla

where

$$\mathcal{Q}_\alpha = \left\{ Q = (q_0, \ldots, q_L)^T \in \mathbb{R}^{L+1} \text{ such that } q_i \geq \alpha \text{ for all } i = 0, \ldots, L \text{ and } q_0 = 1 \right\},$$

$$\mathcal{R}_\alpha = \left\{ R = (r_0, \ldots, r_K)^T \in \mathbb{R}^{K+1} \text{ such that } r_i \geq \alpha \text{ for all } i = 0, \ldots, K \right\}.$$

For fixed Q (or R) the optimization problem (21) is an example of a quadratic program with linear constraints. Hence to solve (21) we suggest the following iterative algorithm:

$$\left\{ \begin{array}{ll} R_s = & \operatorname{argmin}_{R \in \mathcal{R}_\alpha} \|W - Q_{s-1} \star R\|_2^2 \\ Q_s = & \operatorname{argmin}_{Q \in \mathcal{Q}_\alpha} \|W - Q \star R_s\|_2^2 \end{array} \right. \tag{22}$$

where $s = 1, 2, \ldots$ and $Q_0 \in \mathcal{Q}_\alpha$ is an arbitrary starting vector. This algorithm is monotonic, but the limiting point is not necessarily the global minimizer of (21). Repeated application of (22) from a number of different starting points is recommended. Algorithm (22) stops if either $\|Q_{n-1} - Q_n\|_2 \leq \varepsilon$ for some n or the total number of iterations reaches some prescribed value. Here $\varepsilon > 0$ is a small given tolerance. There are a number of possible solution methods for each minimization in (22), and solvers exist in many popular programs. To solve (21) we use CVX, a package for specifying and solving convex problems [19, 20].

5.1. Example A

We now consider a number of examples considering the performance of (22) and the complexity of the optimization problem (21). In these examples considering (22) we will often report the quantities $Dist = \|W - \hat{Q}_\alpha \star \hat{R}_\alpha\|_2^2$ and $Dev = \max |W - \hat{Q}_\alpha \star \hat{R}_\alpha|$ at the point of convergence $(\hat{Q}_\alpha, \hat{R}_\alpha)$. In all the examples that follow we run the algorithm (22) to solve (21) starting from 1000 random Q_0 until convergence. Each Q_0 is independently sampled from a uniform distribution on $[0, 1]^{L+1}$ normalized so that $q_0 = 1$.

5.1.1. Examples where exact solutions to $W = Q \star R$ with non-negative Q and R exist

Example A1. Here we revisit Example 2 considered in Section 4.2. Let $N = 11$, $L = 3$, $K = 8$, $W = (1, 1, \ldots, 1)^T$ and $\alpha = 0$. Table I contains the two solutions that were converged to from 1000 random starting points. Half of the runs of the algorithm converged to the first solution given in Example 2 whilst the other half converged to the second solution. The problem becomes much more difficult for larger N. This is demonstrated in the next example.

Q^T	R^T	%
$(1, 1, 1, 1)$	$(1, 0, 0, 0, 1, 0, 0, 0, 1)$	50%
$(1, 0, 0, 1)$	$(1, 1, 1, 0, 0, 0, 1, 1, 1)$	50%

Table I. Two solutions converged to using (22) for example described with $N = 11$, $L = 3$, $K = 8$, $W = (1, 1, \ldots, 1)^T$

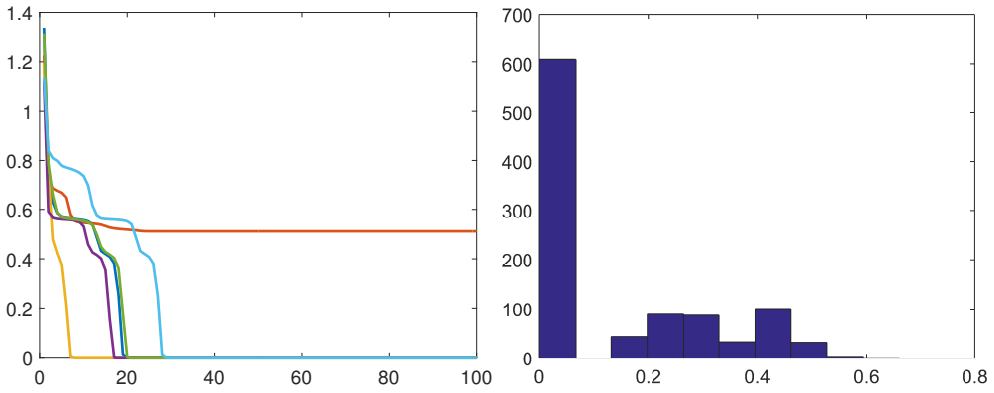
Example A2. Let $N = 143$, $L = 15$, $K = 128$, $W = (1, 1, \ldots, 1)^T$ and $\alpha = 0$. There are four possible (Q, R)-pairs providing the equality $Q \star R = W$ with non-negative Q and R. For brevity, we only report the possible solutions for Q:

$$\begin{aligned} Q_{(1)}^T &= (1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1), \\ Q_{(2)}^T &= (1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1), \\ Q_{(3)}^T &= (1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1), \\ Q_{(4)}^T &= (1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1). \end{aligned}$$

DOI: 10.1002/nla

In this instance, the algorithm converges to $Q_{(1)}^T$ for approximately 20% of the 1000 starting values. It does not converge to any of the other solutions presented above, unless the algorithm is started very 'close' to one of them. This suggests that the optimization problem given by (21) may be multi-extremal and of complex structure. This issue is further discussed in the next example.

Figure 1 contains a plot of the distances $\|W - Q_s \star R_s\|_2^2$ against iteration s for six chosen starting values. A histogram of $Dist$ at the point of convergence for each of the 1000 random starting values is also given. It can be seen that for the six randomly chosen starting values algorithm (22) converges monotonically. Likewise, the majority of the solutions of (22) provide 'small' values of $Dist$.



(a) Plot of distances by iteration for six starting values (b) Histogram

Figure 1. Plot of $\|W - Q_s \star R_s\|_2$ against iteration s for six randomly chosen starting values and histogram of $Dist$ at the point of convergence.

5.1.2. Minimization of $\|W - Q \star R\|_2$ In the next three examples we assume $\alpha = 10^{-6}$. The solution to $W = Q \star R$ with non-negative $Q \in \mathcal{Q}_\alpha$ and $R \in \mathcal{R}_\alpha$ does not exist and we are seeking for a solution to the minimization problem (21) which minimizes $\|W - Q \star R\|_2$.

Example A3. Let $N = 5$, $L = 1$ and $W = (1, 1, 1, 1, 1)^T$. We use this simple example to consider the complexity of the optimization problem (21). Figure 2 contains a three-dimensional plot and contour plot of $\|W - Q \star R\|_2$ against $Q = (q_0, q_1)^T$ for fixed R. The optimization problem (21) can be seen to have many local minima with narrow region of attraction. Consequently we recommend starting the algorithm (22) from a number of starting points. From experiments conducted, as L, K and N increase, then so do the number of local minima of (21).

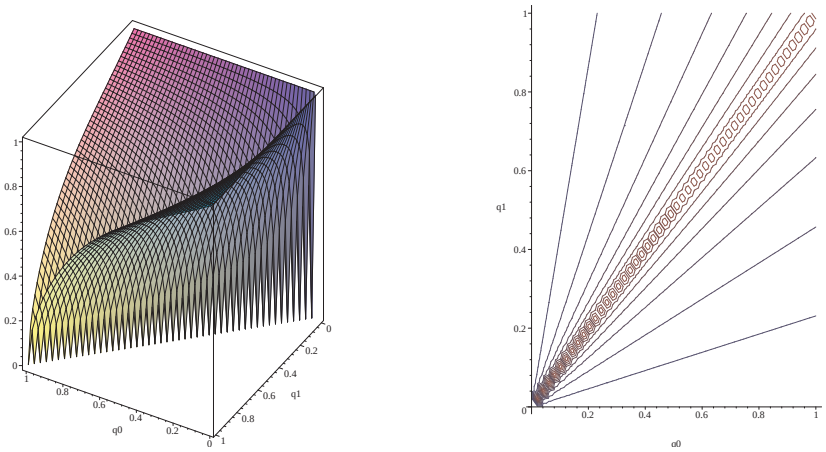
Example A4. Suppose $N = 12$, $L = 3$ and $W = (1, 1, \ldots, 1)^T$. Table II contains the two solutions that were converged to from the 1000 random starting points. For most starting points used (95%) of them, the algorithm converged to the first solution given. For the remaining starting values the algorithm converged to the second solution. Also given in the table is the value $Dist$ at the point of convergence.

Q^T	R^T	%	Dist
$(1, 0.7795, 0.7795, 1)$	$(0.9233, 0.3571, \alpha, \alpha, 0.6763, 0.6763, \alpha, \alpha, 0.3571, 0.9233)$	95%	0.1930
$(1, 1.5481, 1.5481, 1)$	$(0.9212, \alpha, 0.2575, 0.5844, \alpha, \alpha, 0.5844, 0.2575, \alpha, 0.9212)$	5%	3.2110

Table II. Two solutions converged to for example with $N = 12$, $L = 3$, $K = 9$, $W = (1, 1, \ldots, 1)^T$

Example A5. Assume $N = 59$, $L = 9$. In this example we consider two weighting schemes (Scheme 1 and Scheme 2), by which we mean we consider two possible selections of the vector W.

DOI: 10.1002/nla

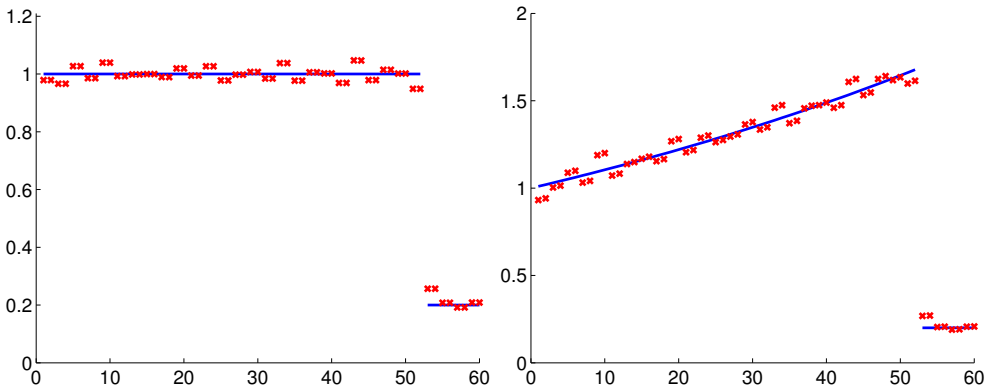


(a) 3d-plot (b) Contour plot

Figure 2. Complexity of the optimization problem (21) with $N = 5$, $L = 1$, $K = 4$, $W = (1, 1, \ldots, 1)^T$.

Scheme 1: $W = W_1 = (w_{1,0}, w_{1,1}, \ldots, w_{1,N})^T$ such that $w_{1,i} = 1$, $i = 0, \ldots, 51$, and $w_{1,i} = 0.2$, $i = 52, \ldots, 59$,
Scheme 2: $W = W_2 = (w_{2,0}, w_{2,1}, \ldots, w_{2,N})^T$ such that $w_{2,i} = 1.01^i$, $i = 0, \ldots, 51$, and $w_{2,i} = 0.2$, $i = 52, \ldots, 59$.
Similar weighting schemes will be used in subsequent sections.

Figure 3 plots $\hat{Q}_\alpha \star \hat{R}_\alpha$ obtained from solving (21) using (22) as described. Also plotted are the weights given by Scheme 1 and Scheme 2. Given under each figure is the value $Dist$ at the point of convergence. For this example it can be seen that the algorithm (22) is able to adapt to non-standard forms of the vector W. Similar forms will be investigated in examples that follow.



(a) Scheme 1, $W = W_1$. $Dist = 0.0361$. (b) Scheme 2, $W = W_2$. $Dist = 0.1106$.

Figure 3. Plot of $\hat{Q}_\alpha \star \hat{R}_\alpha$ obtained from solving (21) using (22). Approximation given by cross ($\times$) and original weights W given in solid line.

Example A6. In this example we fix $W = (1, 1, \ldots, 1)^T$, $\alpha = 0$ and consider a number of different L and K. Table III gives the values $Dist$, and Dev at the point of convergence from running algorithm (22) as already described. Here it can be seen that if $W = Q \star R$ can be solved exactly, then running the algorithm (22) from a number of starting points will routinely find the exact solution. There are a number of cases where exact solutions do not exist, but the algorithm (22) appears to provide a robust method of finding good approximations.

DOI: 10.1002/nla

L	K	$Dist$	Dev
3	3	0.3333	0.3333
3	4	0.0000	0.0000
3	5	0.1295	0.1710
3	6	0.1691	0.2184
3	7	0.1149	0.1429
3	8	0.0000	0.0000
3	9	0.0960	0.1395
3	14	0.0000	0.0000
3	19	0.0542	0.0739
3	49	0.0231	0.0332
3	99	0.0148	0.0254

L	K	$Dist$	Dev
4	4	0.3333	0.3333
4	9	0.1280	0.1609
4	19	0.0000	0.0000
4	49	0.0000	0.0000
4	99	0.0180	0.0318
9	9	0.3333	0.3333
9	49	0.0656	0.0903
9	99	0.0363	0.0589
19	49	0.1253	0.1169
19	99	0.0723	0.1071
49	99	0.1762	0.2160

Table III. Distances and maximum deviances of solutions to (21) obtained using (22) for different L and K

6. METHODOLOGY FOR TIME SERIES ANALYSIS

As an example of how we may use our results for the analysis of time series, consider the following example of an algorithm which may be used for modelling time series. We expect that there are other algorithms and application areas which could use the result of Theorem 1.

Let $Y = (y_0, \ldots, y_N)^T \in \mathbb{R}^{N+1}$ be a given time series. We do not make any formal distinctions between the problems of imputing missing data or forecasting. Informally we consider the problem of forecasting to be imputing missing data located at the end of the time series. Hence we allow for the possibility of Y to contain missing values which we would like to impute. In this case we set the missing values to some initial values (such as the mean of the time series) and account for our uncertainty of our initial values in our vector of weights $W = (w_0, w_1, \ldots, w_N)^T \in \mathbb{R}^{N+1}$.

The methodology we use in this paper consists of the following components:

Approximating the vector norm by the matrix norm Suppose we are given a vector of weights $W = (w_0, w_1, \ldots, w_N)^T \in \mathbb{R}^{N+1}$. In order to find $\mathbf{Q} = \mathrm{diag}(q_0, q_1, \ldots, q_L)^T$ and $\mathbf{R} = \mathrm{diag}(r_0, r_1, \ldots, r_K)^T$, so that we may approximate the vector norm (8) by (9), we solve the optimization problem (21) as described in the previous section.

Low-rank approximation in the $(\mathbf{Q}, \mathbf{R})$ norm Suppose we have a given $(L+1) \times (K+1)$ matrix $\mathbf{X} = \mathbf{X}_Y = \mathbb{H}(Y)$ and an integer $r < L+1$. Suppose that the matrices $\mathbf{Q}$, $\mathbf{R}$ corresponding to the given vector of weights W are known, as described in the previous paragraph.

Denote

$$\pi^{(r)}_{(\mathbf{Q},\mathbf{R})}(\mathbf{X}) = \operatorname*{argmin}_{\mathbf{B}:rank(\mathbf{B})=r} ||\mathbf{X} - \mathbf{B}||^2_{\mathbf{Q},\mathbf{R}}\,.$$

The projection $\pi^{(r)}_{(\mathbf{Q},\mathbf{R})}(\mathbf{X})$ may be computed by the use of Theorem 2 as given in Section 3.

Projection to the space of Hankel matrices in the $(\mathbf{Q}, \mathbf{R})$ norm Denote

$$\pi^{\mathcal{H}}_{(\mathbf{Q},\mathbf{R})}(\mathbf{X}) = \operatorname*{argmin}_{\mathbf{B}:\mathbf{B}\in\mathcal{H}} ||\mathbf{X} - \mathbf{B}||^2_{\mathbf{Q},\mathbf{R}}\,. \tag{23}$$

We can numerically solve the optimization problem (23) using standard convex optimization routines. For some $(\mathbf{Q}, \mathbf{R})$ it is possible to write explicit solutions to (23).

For example, the closest Hankel matrix in Frobenius norm to any given matrix is obtained by using the diagonal averaging procedure, see [21, Sect. 6.2]. Recall that every $(L+1) \times (K+1)$ Hankel matrix $\mathbf{X} \in \mathcal{H}$ is in a one-to-one correspondence with some vector $Y = (y_0, \ldots, y_N)^T$. We associate this one-to-one correspondence with the function $\mathbb{H} : \mathbb{R}^{(N+1)} \to \mathcal{H}^{(L+1)\times(K+1)}$. Each element of the vector Y is repeated in $\mathbf{X} = \mathbb{H}(Y)$ several times. From (7) it can be seen that the value κ_n is the

DOI: 10.1002/nla

number of times the element y_n of the vector Y is repeated in the Hankel matrix $\mathbf{X} = \mathbf{X}_Y = \mathbb{H}(Y)$. Let $\pi^{\mathcal{H}}_{(\mathbf{Q},\mathbf{R})}(\mathbf{X}) = \pi^{\mathcal{H}}(\mathbf{X}) =$ denote the projection in Frobenius norm of $\mathbf{X} \in \mathbb{R}^{L\times K}$ onto the space $\mathcal{H}$. The element $\tilde{x}_{ij}$ of $\pi_{\mathcal{H}}(\mathbf{X})$ is given by

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

Other choices of $(\mathbf{Q},\mathbf{R})$ which yield explicit solutions to (23) are described in [22] and [23]. These papers also consider the use of weights in similar problems to that considered in this paper.

We are now able to state our algorithm which we use in the next section.

Algorithm Set $\mathbf{X}_0 = \mathbf{X} = \mathbb{H}(Y)$. For $n = 1, 2, \ldots$

$$\mathbf{X}_n = \pi^{\mathcal{H}}_{(\mathbf{Q},\mathbf{R})}(\pi^{(r)}_{(\mathbf{Q},\mathbf{R})}(\mathbf{X}_{n-1}))\,. \tag{24}$$

We make the following remarks. Iterations of (24) are known as the so-called Cadzow iterations [24] if $\mathbf{Q}$ and $\mathbf{R}$ are identity matrices of dimension $(L+1)$ and $(K+1)$ respectively. If these iterations are performed until the matrix $\mathbf{X}_n$ is of rank r, then they are known to linearly converge to a solution which can be renormalized to become a local optimum of (4), see [16].

By starting these iterations from a number of different starting points (akin to the multistart approach used in global optimization, see for example [25]) then under some conditions these iterations converge to the global optimum of (4), see [16]. One Cadzow iteration corresponds to the basic version of the technique known as singular spectrum analysis (SSA), see [13, 21, 26] for further discussion. These techniques (as well as some other classical techniques) are considered in the next section.

7. EXAMPLES

In this section we first solely consider the optimization problem (21) described in Section 5. We then consider two 'real-life' applications of algorithm (24) and compare our results to those which have been published previously.

7.1. Example B: Missing data and fortified wine

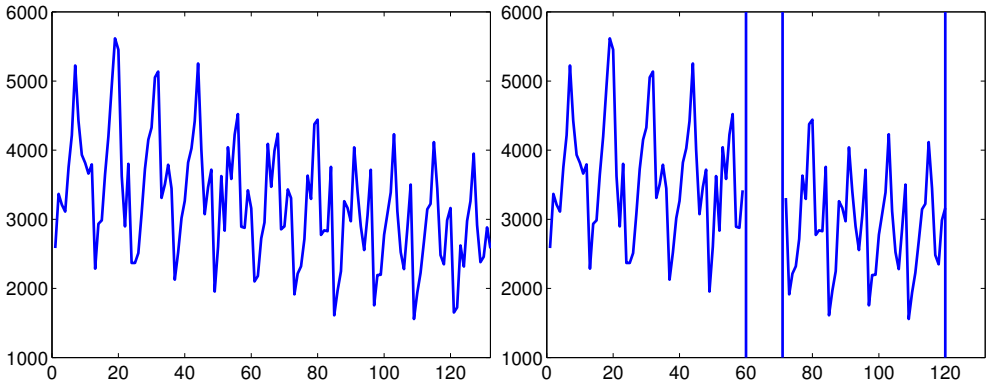
To demonstrate the methods of filling in missing data, we consider a real-life time series which is the monthly volumes of fortified wine sales observed in Australia from January 1980 until January 1990. The data can be found in a number of popular time series repositories, as well as [27]. A plot of the time series is given in Figure 4(a). In this example we removed 12 known values, starting at the 61st point, that is, we assume that the values for one year (January 1985 - December 1985) are unknown. Also, to perform a forecast we add 12 missing data points at the end of the series.

The complete data, and the data with the missing sections is shown in Figure 4(b). This time series was also studied in [21] where the authors recommend the selection of the parameters $L = 36$ and $r = 11$. Hence these are the parameters that will be used in this example.

We will investigate the following selection of the vector W. Let the vector of weights W be constructed so that the missing values received a weight of 0.3, and all other values received a weight of 1. We estimate $\mathbf{Q}$ and $\mathbf{R}$ using (22) with $\varepsilon = 10^{-6}$, and then proceed to impute missing data (and construct the forecast) using (24). We run algorithm (22) starting from 1000 random Q_0 and set $\alpha = 0.1$. We need an initial imputation for the missing values, and this is described later.

Table IV contains the square root of the mean square error (MSE) of our approximations from the true values of the time series. The table gives these values separately for the missing middle section of the data, the forecasted end section of the data, and for the entire time series for weighing schemes W. Our algorithm requires the missing data to be initially imputed using some starting values. Table IV contains the square root of the MSEs when the starting values are (i) the mean of

DOI: 10.1002/nla



(a) Complete series (b) Series with section missing

Figure 4. Monthly volumes of fortified wine sales in Australia from January 1980 until January 1990

the entire observed series, (ii) the output from one iteration of our algorithm (22) obtained using the mean as the starting values and so on. Here we repeatedly update the starting values based on the previous approximation. We stop after five updates of the starting values as no significant subsequent improvement was found. These results are an improvement over those reported in [21], who give the values 221.2, 333.0 and 282.7 for the middle section of the data, the forecasted end section of the data and for the entire time series, respectively, using the same choice of L and r. Figure 5 contains

Starting values	Middle	End	Total
Mean	416.0	550.5	303.7
One iteration	281.4	377.4	242.3
Two iterations	236.7	301.7	222.9
Five iterations	220.3	287.4	217.7

Table IV. Square root of the MSEs of our approximations from the true values of the time series, using different starting values.

plots of the observed data with approximations found from different starting values, corresponding to those described in Table IV for weighting scheme W.

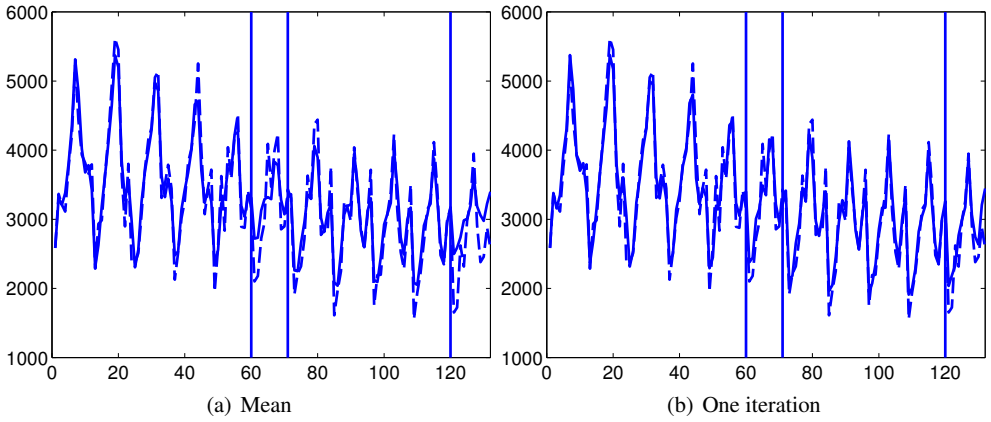
7.2. *Example C: Simulated example*

We now consider a simulated example. Suppose we observe $Y = (y_0, \ldots, y_N)^T$ in the form $y_n = s_n + \varepsilon_n$, $n = 0, \ldots, N$ with $N = 99$ (that is, we consider 100 observations) where elements of $S = (s_0, \ldots, s_N)^T$ are such that $s_n = \sin(0.04n)\sin(0.4\pi n)$. We take $\{\varepsilon_n\}$ to be a series of normally distributed i.i.d. random variables with standard deviation selected to be 0.25. Figure 6 contains a plot of S and Y for one realization of $\{\varepsilon_n\}$.

In this example we suppose that we wish to forecast the final ten observations of Y. We do this by truncating the series to the first 90 observations and forecast the remaining ten observations. The aim of this study is to improve the result of an initial forecast (in a sense defined below). Specifically we conduct the following exercise with $L + 1 = 20$ and $r = 4$:

- We generate 100 realizations of the time series Y. For each realization we produce an initial forecast of the remaining ten observations from the truncated series $(y_0, \ldots, y_{N-10})^T$, using one iteration of (24) with $\mathbf{Q}$ and $\mathbf{R}$ taken to be identity matrices. Specifically, we construct the forecast using SSA which was described in Section 6. Denote the forecast by $(\tilde{y}_{N-9}, \ldots, \tilde{y}_N)^T$

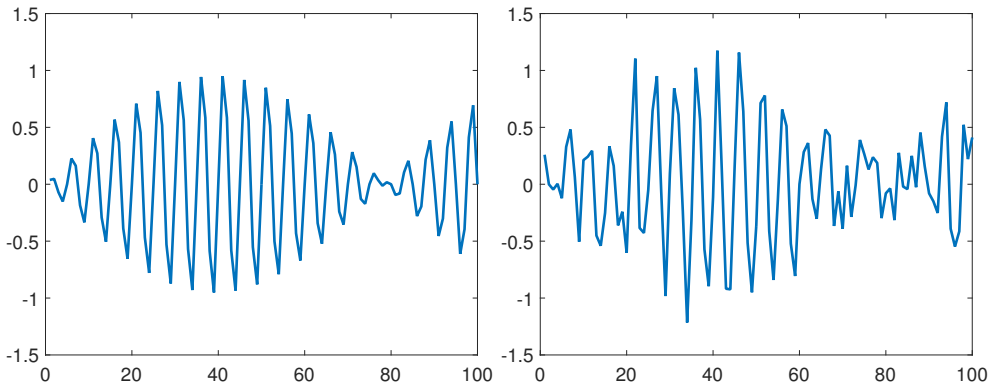
Prepared using nlaauth.cls
DOI: 10.1002/nla



(a) Mean (b) One iteration

(c) Two iterations (d) Five iterations

Figure 5. Monthly volumes of fortified wine sales in Australia from January 1980 until January 1990, observed data in dashed line, approximation in solid line for weighting scheme W.



(a) S (b) Y

Figure 6. Plot of S and Y for one realization of $\{\varepsilon_n\}$ with $s_n = \sin(0.04n)\sin(0.4\pi n)$.

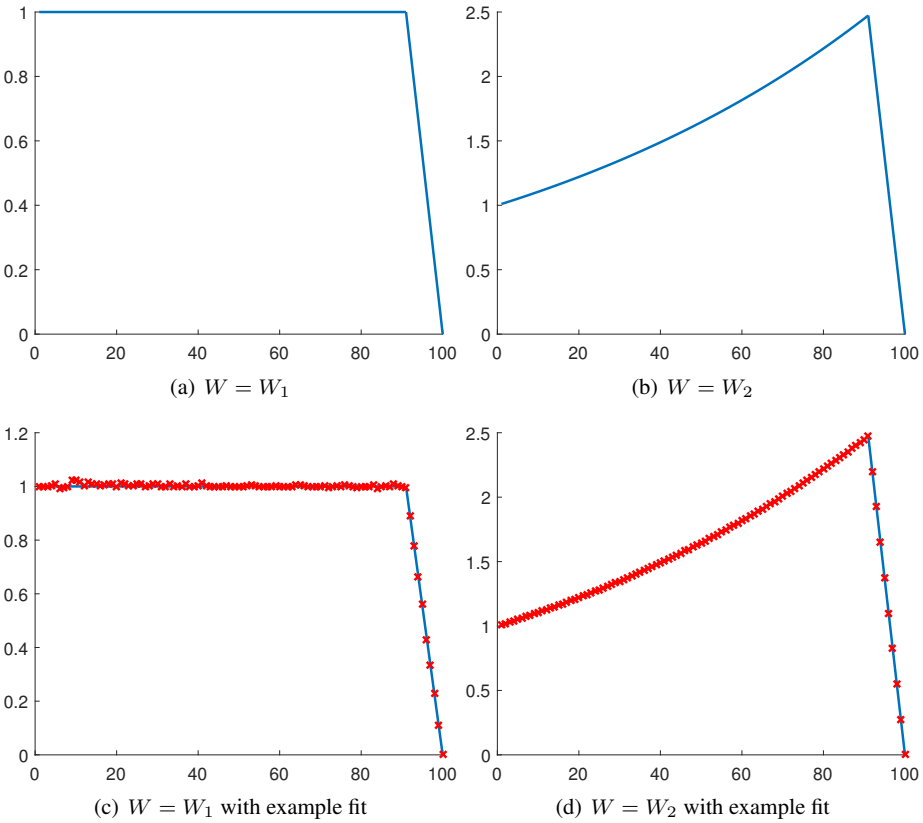
- For each realization we form $\mathbf{X} = \mathbb{H}(y_0, \ldots, y_{N-10}, \tilde{y}_{N-9}, \ldots, \tilde{y}_N)$ and use algorithm (24) with $\mathbf{Q}$ and $\mathbf{R}$ computed using (22) corresponding to a given W (described below) with $\varepsilon = 10^{-6}$, starting from 1000 random Q_0 and set $\alpha = 0.1$.

We consider two versions of the vector W:

DOI: 10.1002/nla

W_1: Take $W = W_1 = (w_{1,0}, w_{1,1}, \ldots, w_{1,N})^T$ such that $w_{1,i} = 1$ for $i = 0, \ldots, N-10$, and $w_{1,i} = a + bi$ for $i = N-9, \ldots, N$ where a and b define the line such that $w_{1,N-10} = 1$ and $w_{1,N+1} = 0$. A figure of this weighting scheme is given in Figure 7(a).

W_2: Take $W = W_2 = (w_{2,0}, w_{2,1}, \ldots, w_{2,N})^T$ such that $w_{2,i} = \beta^i$, $i = 0, \ldots, N-10$ with $\beta = 1.01$, and $w_{2,i} = a + bi$ for $i = N-9, \ldots, N$ where a and b define the line such that $w_{2,N-10} = \beta^{N-10}$ and $w_{2,N+1} = 0$. A figure of this weighting scheme is given in Figure 7(b).



(a) $W = W_1$ (b) $W = W_2$

(c) $W = W_1$ with example fit (d) $W = W_2$ with example fit

Figure 7. Plot of W_1 and W_2 with example fits.

Table V contains the mean absolute errors (MAE) and mean square errors (MSE) of the ten forecasted observations averaged over the 100 simulated realizations of Y, for the two possible vectors W_1 and W_2. Results are given for the forecasts generated by (i) taking the initial forecasts from SSA (as described above), (ii) performing one iteration of algorithm (24) with $\mathbf{Q}$ and $\mathbf{R}$ computed using (22) using the initial forecast from SSA, and (iii) performing iterations of algorithm (24) with $\mathbf{Q}$ and $\mathbf{R}$ computed using (22) using the initial forecast from SSA until the solution is of rank r.

In summary, we note the following. Use of weights always improved the initial forecasts generated. It appears that weights given by $W = W_2$ has resulted in better forecasts (in the sense of smaller MAE and MSE). Additionally for this example the forecast from the rank r solution (found by performing iterations of algorithm (24) until the solution is of rank r) has resulted in better forecasts. This is likely to be due to the fact that the rank of the matrix obtained from the noise-free vector S is of rank 4, and in this case, where there is a clear model of this rank, methods which yield matrices which approximate this rank more closely are likely to be more accurate. In most real, practical examples however, the rank is unknown. This was the case in the previous example as well as in the next example that follows. To give an idea of the size of the MAE and MSE values

DOI: 10.1002/nla

	MAE (i)	MSE (i)	MAE (ii)	MSE (ii)	MAE (iii)	MSE (iii)
W_1	0.245 (0.133)	0.282 (0.152)	0.233 (0.106)	0.270 (0.121)	0.242 (0.075)	0.262 (0.085)
W_2	0.245 (0.133)	0.282 (0.152)	0.222 (0.128)	0.256 (0.122)	0.190 (0.038)	0.215 (0.044)

Table V. Mean absolute errors (MAE) and mean square errors (MSE) of the ten forecasted observations averaged over the 100 simulated realizations of Y. Standard deviations given in brackets. Results for forecasts generated by (i) taking the initial forecasts from SSA, (ii) performing one iteration of algorithm (24) with $\mathbf{Q}$ and $\mathbf{R}$ computed using (22) corresponding to the given W using the initial forecast from (i), and (iii) performing iterations of algorithm (24) with $\mathbf{Q}$ and $\mathbf{R}$ computed using (22) corresponding to the given W using the initial forecast from (i) until the solution is of rank r.

reported, they are 0.26011 and 0.28914 averaged over 100 simulations, if the forecast was set to be the mean of the series. A dependent samples t-test suggests that the forecasts obtained by taking $W = W_2$ are superior to those obtained by taking $W = W_1$, with p-value $p < 0.01$.

7.3. Example D: Forecasting deaths

In this section we consider forecasting the famous 'death' series recording the monthly accidental deaths in the USA between 1973 and 1978. This data has been studied by many authors (such as [21]) and can be found in a number of time series data libraries. We use the same parameters as those given by [28]. We wish to replicate the exercise given in [29] which aimed to forecast the final six values of this series. The time series contains a total of $N = 78$ observations. We truncate the series to the first 72 observations and will forecast the remaining six observations. In a similar manner to the previous example, we will take an initial forecast and aim to improve it.

Table VI contains forecasts of the final six data points of the data series by several methods along with the square root of the mean square error (MSE) and mean absolute error (MAE). These results are taken from [29] and full details of the fitted models can be found within. In summary Model I and Model II are examples of SARIMA models as described by [30]. Model I is given by

$$y_n - y_{n-12} = 28.831 + (1 - 0.478B)(1 - 0.588B^{12})Z_n\,,$$

and Model II is given by

$$(1 - B)(1 - B^{12})y_n = 28.831 + Z_n - 0.596Z_{n-1} - 0.407Z_{n-6} - 0.685Z_{n-12} + 0.460Z_{n-13}\,,$$

where Z_n is a realisation of white noise with zero mean and variance 0.9439 and B is the backward shift operator defined as: $B^j Z_n = Z_{n-j}$. HWS represents the model as fitted by the Holt-Winter seasonal algorithm. ARAR represents the model as fitted by transforming the data prior to fitting an autoregressive model. Additionally from [28] is the SSA forecast with $L = 24$ and $r = 12$. These are the parameter values we also select for (24), further details are given later. SSA has been described briefly in Section 6.

	1	2	3	4	5	6	$\sqrt{MSE}$	MAE
Original data	7798	7406	8363	8460	9217	9316		
Model I	8441	7704	8549	8885	9843	10279	582.63	524
Model II	8345	7619	8356	8742	9795	10179	500.50	415
HWS	8039	7077	7750	7941	8824	9329	401.26	351
ARAR	8168	7196	7982	8284	9144	9465	253.20	227
SSA	7782	7428	7804	8081	9302	9333	278.20	180

Table VI. Forecasted data using four different models, along with the square root of the mean square error (MSE) and mean absolute error (MAE) of the forecast.

As a demonstration of the potential of the methodology described in previous sections, we consider the following set-up. Let us suppose that we wish to forecast ahead 6 data points akin to the exercise described in [29]. Specifically we would like to improve the forecasts given in Table VI. We consider the following two weighting schemes, that is, values for W:

DOI: 10.1002/nla

W_1: Take $W = W_1 = (w_{1,0}, w_{1,1}, \ldots, w_{1,N})^T$ such that $w_{1,i} = 1$ for $i = 0, \ldots, N-6$, and $w_{1,i} = a + bi$ for $i = N-6, \ldots, N$ where a and b define the line such that $w_{1,N-6} = 1$ and $w_{1,N+1} = 0$.

W_2: Take $W = W_2 = (w_{2,0}, w_{2,1}, \ldots, w_{2,N})^T$ such that $w_{2,i} = \beta^i$, $i = 0, \ldots, N-6$ with $\beta = 1.01$, and $w_{2,i} = a + bi$ for $i = N-9, \ldots, N$ where a and b define the line such that $w_{2,N-6} = \beta^{N-6}$ and $w_{1,N+1} = 0$.

Tables VII and VIII contain the forecasted data using different sets of starting values provided by Table VI (in an identical manner to that as described in the previous example) for each of the two weighting schemes considered. We run algorithm (22) starting from 1000 random Q_0 with $\varepsilon = 10^{-6}$ and set $\alpha = 0.1$. It can be seen that in all cases, we arrive at an improved forecast from the initial starting value. As the rank of the true underlying model is not clear, we perform one iteration of (24).

	1	2	3	4	5	6	$\sqrt{MSE}$	MAE
Original data	7798	7406	8363	8460	9217	9316		
Model I	8410	7698	8467	8781	9940	10341	582.20	512.92
Model II	8300	7632	8310	8657	9823	10156	486.03	403.85
HWS	8021	7101	7729	7914	9005	9387	385.81	331.82
ARAR	8132	7228	7965	8280	9135	9479	247.56	222.63
SSA	7778	7422	7809	8080	9284	9349	276.28	178.62

Table VII. Forecasted data using five different starting values, along with the square root of the MSE and MAE of the forecast for $W = W_1$.

	1	2	3	4	5	6	$\sqrt{MSE}$	MAE
Original data	7798	7406	8363	8460	9217	9316		
Model I	8395	7686	8469	8828	9849	10264	559.55	488.46
Model II	8308	7606	8302	8692	9798	10152	481.91	403.19
HWS	8012	7105	7749	7955	8898	9326	380.79	327.10
ARAR	8125	7228	7965	8288	9130	9471	244.61	219.78
SSA	7789	7427	7804	8094	9301	9348	275.45	178.68

Table VIII. Forecasted data using five different starting values, along with the square root of the MSE and MAE of the forecast for $W = W_2$.

Tables VII and VIII show that the use of weights has led to significantly improved forecasts. It appears that taking $W = W_2$ leads to marginally better forecasts than taking $W = W_1$. In this example, the underlying model was unclear, and so we used only one iteration of (24). This is a popular approach in many subspace-based methods where r is used to control the precision of the approximation to the observed data rather than as a 'hard constraint' for the rank of the approximation (see [13]).

8. CONCLUSION

When fitting a time series with a vector satisfying a LRR of order r, a standard approach is to consider an equivalent but larger problem of fitting a Hankel matrix of rank r. Traditional approaches to fitting the Hankel matrix use the Frobenius norm out of convenience. This however can be a bad choice, as this yields a particular form for the corresponding norm on the time series.

In this paper we have shown how a matrix norm on Hankel matrices can be selected to either match exactly, or approximate, a weighted vector norm. We have shown that the accuracy of time series forecasting and imputing missing values can be significantly improved if the weights are chosen appropriately.

DOI: 10.1002/nla

REFERENCES

1. Chu MT, Funderlic RE, Plemmons RJ. Structured low rank approximation. *Linear Algebra and its Applications* 2003; **366**:157–172.
2. Willems JC. From time series to linear systems - Part II. Exact modelling. *Automatica* 1986; **22**(6):675–694.
3. De Moor B. Structured total least squares and L_2 approximation problems. *Linear Algebra and its Applications* 1993; **188**:163–205.
4. Roorda B. Algorithms for global total least squares modelling of finite multivariable time series. *Automatica* 1995; **31**(3):391–404.
5. Wentzell PD, Andrews DT, Hamilton DC, Faber K, Kowalski BR. Maximum likelihood principal component analysis. *Journal of Chemometrics* 1997; **11**(4):339–366.
6. Markovsky I. Structured low-rank approximation and its applications. *Automatica* 2008; **44**(4):891–909.
7. Gabriel KR, Zamir S. Lower rank approximation of matrices by least squares with any choice of weights. *Technometrics* 1979; **21**(4):489–498.
8. ten Berge J, Kiers HA. An alternating least squares method for the weighted approximation of a symmetric matrix. *Psychometrika* 1993; **58**(1):115–118.
9. Gillis N, Glineur F. Low-rank matrix approximation with weights or missing data is NP-hard. *SIAM Journal on Matrix Analysis and Applications* 2011; **32**(4):1149–1165.
10. Gillard J, Zhigljavsky A. Application of structured low-rank approximation methods for imputing missing values in time series. *Statistics and its Interface* 2015; **8**(3):321–330.
11. Tufts DW, Kumaresan R. Estimation of frequencies of multiple sinusoids: Making linear prediction perform like maximum likelihood. *Proceedings of the IEEE* 1982; **70**(9):975–989.
12. Gillard J, Zhigljavsky A. Optimization challenges in the structured low rank approximation problem. *Journal of Global Optimization* 2013; **57**(3):733–751.
13. Golyandina N, Zhigljavsky A. *Singular Spectrum Analysis for time series*. Springer Science & Business Media, 2013.
14. Gillard J, Kvasov D. Lipschitz optimization methods for fitting a sum of damped sinusoids to a series of observations. *Statistics and its Interface* 2016; (to appear).
15. Gillard J, Zhigljavsky A. Analysis of structured low rank approximation as an optimization problem. *Informatica* 2011; **22**(4):489–505.
16. Gillard J, Zhigljavsky A. Stochastic algorithms for solving structured low-rank matrix approximation problems. *Communications in Nonlinear Science and Numerical Simulation* 2015; **21**(1):70–88.
17. Eckart C, Young G. The approximation of one matrix by another of lower rank. *Psychometrika* 1936; **1**(3):211–218.
18. Allen GI, Grosenick L, Taylor J. A generalized least-square matrix decomposition. *Journal of the American Statistical Association* 2014; **109**(505):145–159.
19. CVX Research, Inc. CVX: Matlab software for disciplined convex programming, version 2.0. http://cvxr.com/cvx Aug 2012.
20. Grant M, Boyd S. Graph implementations for nonsmooth convex programs. *Recent Advances in Learning and Control*, Blondel V, Boyd S, Kimura H (eds.). Lecture Notes in Control and Information Sciences, Springer-Verlag Limited, 2008; 95–110.
21. Golyandina N, Nekrutkin V, Zhigljavsky AA. *Analysis of time series structure: SSA and related techniques*. CRC press, 2001.
22. Golyandina N, Shlemov A. Variations of singular spectrum analysis for seperability improvement: non-orthogonal decompositions of time series. *Statistics and its Interface* 2015; **8**(3):277–294.
23. Zvonarev N, Golyandina N. Iterative algorithms for weighted and unweighted finite-rank time-series approximations. *Statistics and its Interface* ; (to appear).
24. Cadzow JA. Signal enhancement-a composite property mapping algorithm. *IEEE Transactions on Acoustics, Speech and Signal Processing* 1988; **36**(1):49–62.
25. Ugray Z, Lasdon L, Plummer J, Glover F, Kelly J, Martí R. Scatter search and local nlp solvers: A multistart framework for global optimization. *INFORMS Journal on Computing* 2007; **19**(3):328–340.
26. Gillard J. Cadzow's basic algorithm, alternating projections and singular spectrum analysis. *Statistics and its interface* 2010; **3**(3):335–343.
27. Golyandina N, Korobeynikov A. Basic singular spectrum analysis and forecasting with R. *Computational Statistics and Data Analysis* 2014; **71**:934–954.
28. Hassani H. Singular spectrum analysis: Methodology and comparison. *Journal of Data Science* 2007; **5**(2):239–257.
29. Brockwell PJ, Davis RA. *Time series: theory and methods*. Springer Science & Business Media, 2009.
30. Box GE, Jenkins GM, Reinsel GC. *Time series analysis: forecasting and control*. John Wiley & Sons, 2013.

DOI: 10.1002/nla