Extracting a Low-Dimensional Predictable Time Series

Yining Dong          S. Joe Qin         Stephen Boyd

December 22, 2019

Abstract

Large scale multi-dimensional time series can be found in many disciplines, including finance, biomedical engineering, and industrial engineering. In this paper we develop a method for projecting the time series onto a low-dimensional time-series that is predictable, in the sense that an auto-regressive model achieves low prediction error. Our formulation and method follow ideas from principle component analysis, so we refer to the extracted low-dimensional time series as principal time series. In a few cases we can compute the optimal projection exactly; in others, we give a heuristic method that seems to work well in practice. The effectiveness of the method is demonstrated on synthesized and real time series.

1 Introduction

High dimensional time series analysis and applications have become increasingly important in many different domains. In many cases, the high dimensional time series data are both cross-correlated and auto-correlated. Cross-correlations among different time series makes it possible to use a set of lower dimensional time series to represent the original, high dimensional time series. For example, PCA [CK86, BN08] and generalized PCA [Cho12] methods utilize the cross-correlations among different time series to extract lower dimensional factors that capture maximal variance. Although PCA has seen wide use as a dimension reduction method, it does not provide modeling of any auto-correlations or dynamics that can exist in time series data. Since auto-correlations make it possible to predict future values from the past values, it is desirable to provide such low dimensional modeling of the dynamics.

In this work, a linear projection method is proposed to extract a lower dimensional most predictable time series from high-dimensional time series. The entries of the low dimensional time series are uncorrelated so that they capture as much dynamics as possible. The advantage of the proposed method is that it focuses on extracting principal time series with highest dynamics. Therefore, the most dynamic features of the high dimensional data are concentrated in a set of lower dimensional time series, which makes it very useful for data prediction, dynamic feature extraction and visualization.

The proposed method has numerous potential applications, ranging from finance to industrial engineering. In finance, if the high dimensional time series consist of returns of
some assets, applying the proposed method gives the most predictable portfolio. In chemical processes, oscillations are usually undesirable \cite{TH97, THZ03} and applying the proposed method to the process measurements can help detect the unwanted oscillations. In biomedical engineering, electroencephalography (EEG) data can be characterized with waves with different frequencies \cite{Tep02, Tat14}. Applying the proposed method to EEG data has the potential to detect the different waves.

1.1 Related work

Here we review existing works that relate to extracting a lower-dimensional most predictable time series from high dimensional data. Forecastable component analysis (ForeCA) utilizes the differential entropy as the predictability (forecastability) measure, which is the lower bound of the expected squared loss of an estimator \cite{Goe13}. Methods such as \cite{BT77, RW15, Sto01, WS02} first make predictions of the original high-dimensional vector time series, and then find a linear projection based on the covariance matrix of the prediction error.

Optimal persistence analysis (OPA) maximizes the decorrelation time \cite{Del01}. Average predictability time decomposition (APTD) maximizes the average predictability time \cite{DT09a, DT09b}. Graph-based predictable feature analysis (GPFA) maximizes a predictability measure defined in terms of graph embedding \cite{WFW17}. Dynamical component analysis (DCA) maximizes the mutual information between the past and future \cite{CLB19}.

This work is also related to dynamic factor models \cite{SW06, SW11, FHLR00, BN07, AW07, PY08, LY12, LYB11, PB87}, dynamic latent variable models \cite{KSC95, LQZ11, LOZ14, DQ17b, DQ17a} and reduced rank auto-regressive models \cite{VRW86, AR88, WB04, BLM18}.

The novelty of our method is that the extraction of principal time series and the VAR modeling of the principal time series are achieved simultaneously by solving the optimization problem (\textit{i.e.}, there is no need to refit a VAR model for the principal time series after they are extracted). In addition, the extracted principal time series are best predictable from their past values and capture most of dynamics. This property makes the proposed method very useful for prediction, dynamic feature extraction, and visualization.

1.2 Outline

In \S\ 2 we define the predictability measure for stationary time series. Based on the predictability measure, an optimization problems for extracting a lower-dimensional most predictable \textit{principal time series} is proposed. In \S\ 2.3 we give exact solutions to the optimization problem when the memory of the VAR model is one and the factor number is one. In \S\ 3 a heuristic algorithm for solving the general optimization problem is developed. In \S\ 4 we demonstrate the effectiveness of the proposed method on a simulation dataset and a quarterly GDP dataset. Finally, extensions and variations these methods are discussed in \S\ 5.
2 The most predictable projected time series

2.1 Predictability of a time series

Consider a wide-sense stationary \( n \)-dimensional vector time series process \( z_t \in \mathbb{R}^n, t \in \mathbb{Z} \), with

\[
E z_t = 0, \quad E z_t z_{t+\tau}^T = \Sigma_\tau, \quad \tau \in \mathbb{Z}.
\]

(1)

Here \( \Sigma_\tau \) is the auto-covariance matrix for lag \( \tau \). The zero mean assumption is without loss of generality, since this can be arranged by subtracting the mean from the original process, if it is not zero. We refer to a time series with \( \Sigma_0 = I \) as standardized.

Predictability measure. An \( M \)-memory auto-regressive (AR) predictor for \( z_t \) has the form

\[
\hat{z}_t = A_1 z_{t-1} + A_2 z_{t-2} + \cdots + A_M z_{t-M},
\]

where \( A_i \in \mathbb{R}^{n \times n}, i = 1, 2, \ldots, M \) are the AR (matrix) coefficients. We define the (\( M \)-memory) (un-)predictability measure for the time series as the smallest possible mean square AR prediction error,

\[
\alpha = \min_{A_1, \ldots, A_M} E \| z_t - \hat{z}_t \|_2^2,
\]

(2)

which has the same value for all \( t \).

We can easily evaluate the predictability measure \( \alpha \). The objective can be expressed as

\[
E \| z_t - \hat{z}_t \|_2^2 = \text{Tr} \left( \Sigma_0 - 2 \left[ \begin{array}{c} \Sigma_1 \\ \Sigma_2 \\ \vdots \\ \Sigma_M \end{array} \right]^T \left[ \begin{array}{c} A_1^T \\ A_2^T \\ \vdots \\ A_M^T \end{array} \right] \left[ \begin{array}{cccc} \Sigma_0 & \Sigma_1^T & \cdots & \Sigma_{M-1}^T \\ \Sigma_1 & \Sigma_0 & \cdots & \Sigma_{M-2}^T \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{M-1} & \Sigma_{M-2} & \cdots & \Sigma_0 \end{array} \right] \left[ \begin{array}{c} A_1^T \\ A_2^T \\ \vdots \\ A_M^T \end{array} \right] \right). \]

The optimal AR coefficients are readily found to be

\[
\begin{bmatrix} A_1^T \\ A_2^T \\ \vdots \\ A_M^T \end{bmatrix} = \left[ \begin{array}{cccc} \Sigma_0 & \Sigma_1^T & \cdots & \Sigma_{M-1}^T \\ \Sigma_1 & \Sigma_0 & \cdots & \Sigma_{M-2}^T \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{M-1} & \Sigma_{M-2} & \cdots & \Sigma_0 \end{array} \right]^{-1} \begin{bmatrix} \Sigma_1 \\ \Sigma_2 \\ \vdots \\ \Sigma_M \end{bmatrix},
\]

assuming the inverse of the symmetric semidefinite block Toeplitz matrix above exists. It follows that

\[
\alpha = \text{Tr} \left( \Sigma_0 - \left[ \begin{array}{c} \Sigma_1 \\ \Sigma_2 \\ \vdots \\ \Sigma_M \end{array} \right]^T \left[ \begin{array}{cccc} \Sigma_0 & \Sigma_1^T & \cdots & \Sigma_{M-1}^T \\ \Sigma_1 & \Sigma_0 & \cdots & \Sigma_{M-2}^T \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{M-1} & \Sigma_{M-2} & \cdots & \Sigma_0 \end{array} \right]^{-1} \left[ \begin{array}{c} \Sigma_1 \\ \Sigma_2 \\ \vdots \\ \Sigma_M \end{array} \right] \right). \]

(3)
It can be shown that $0 \leq \alpha \leq \text{Tr} \Sigma_0$. A low value of $\alpha$ indicates high predictability; a high value of $\alpha$ indicates low predictability. The extreme case $\alpha = 0$ means that the AR model has zero residual, and the extreme case $\alpha = \text{Tr} \Sigma_0$ occurs when $z_t$ and $z_s$ are uncorrelated for $t \neq s$, so $\Sigma_\tau = 0$ for $\tau \neq 0$, and the optimal AR coefficients are all zero.

To simplify the notation for the rest of paper, we define $A = \begin{bmatrix} A_1 & \cdots & A_M \end{bmatrix}$.

### 2.2 The most predictable projected time series

We can obtain a lower-dimensional time series $x_t \in \mathbb{R}^m$ as a linear function of the original time series $z_t \in \mathbb{R}^n$, as $x_t = W^T z_t$, where $W \in \mathbb{R}^{n \times m}$, with $m < n$. We denote the auto-covariance matrices of $x_t$ as $S_\tau = E x_t x_{t+\tau}^T = W^T \Sigma_\tau W$, $\tau \in \mathbb{Z}$.

Our goal is to choose $W$ so that the series $x_t$ is predictable, i.e., has a low value of $\alpha$. We evidently need to normalize $W$ to rule out the solution $W = 0$; we do this with the constraint

$$W^T \Sigma_0 W = S_0 = I.$$ 

This ensures that $E x_t x_t^T = I$, i.e., the low-dimensional time series $x_t$ is standardized.

The most predictable projected time series is found by solving the optimization problem

$$\begin{align*}
\text{maximize} & \quad f(W) \\
\text{subject to} & \quad S_0 = I,
\end{align*}$$

with variables $W$ and $A$, where

$$f(W) = \text{Tr} \begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_M \end{bmatrix}^T \begin{bmatrix} S_0 & S_0^T & \cdots & S_{M-1}^T \\ S_1 & S_0 & \cdots & S_{M-2}^T \\ \vdots & \vdots & \ddots & \vdots \\ S_{M-1} & S_{M-2} & \cdots & S_0 \end{bmatrix}^{-1} \begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_M \end{bmatrix}.$$ 

The solution is evidently not unique; if $Q$ is an orthonormal $m \times m$ matrix, then $f(WQ) = f(W)$.

### 2.3 Special case with exact solutions

Here we observe that when $M = 1$ and $m = 1$, the problem can be solved exactly. For projection, the problem is

$$\begin{align*}
\text{maximize} & \quad \|W^T \Sigma_1 W\|_F^2 \\
\text{subject to} & \quad W^T \Sigma_0 W = I,
\end{align*}$$

using $S_0 = I$ to simplify the objective. This is readily solved. Define $Z = \Sigma_0^{1/2} W$ and $Y = \Sigma_0^{-1/2} \Sigma_1 \Sigma_0^{-1/2}$, to the problem is to maximize $\|Z^T Y Z\|_F^2$ subject to $Z^T Z = I$. 

4
Let $V$ denote the eigenvector of $Y + Y^T$ corresponding to the eigenvalue with the maximum magnitude. Then $V$ satisfies the constraint and maximizes the objective value. Therefore, the optimal $W$ is $W^* = \Sigma_0^{-1/2} V$.

Once an optimal $W^*$ is obtained, the optimal $A^*$ can be easily calculated as

$$A^* = S_1^T S_0^{-1} = S_1^T = (W^*)^T \Sigma_1^T W^*.$$ 

3 Algorithm

When $M \neq 1$, there is no exact solution to (4), to the best of our knowledge. Problem (4) is essentially an optimization problem over the Grassmannian manifold. There has been research on how to address such optimization problems, including [AMS09, UM14, EAS98]. In this section, we give a heuristic method that seems to work well in practice.

We will construct the columns of $W \in \mathbb{R}^{n \times m}$ sequentially. Each column is chosen satisfy the constraint $S_0 = I$, while maximizing the predictability of the projected time series. In this section, we explain how to achieve this.

Assume that we have already constructed $k$ columns of $W$, with $W^k \in \mathbb{R}^{n \times k}$ and $A^k \in \mathbb{R}^{Mk \times k}$. (We initialize $k = 0$, $W^k = 0$, and $A^0 = 0$ to construct the first column). Then, our goal is to choose $W^{k+1} = [W^k \ w]$, where $w \in \mathbb{R}^n$, such that the $(k+1)$-dimensional projected time series is most predictable.

This is equivalent to the optimization problem

$$\begin{align*}
\text{maximize} & \quad f(W^{k+1}) \\
\text{subject to} & \quad W^{k+1} = [W^k \ w] \\
& \quad S_0^{k+1} = I,
\end{align*}$$

where $W^k$ is fixed and $S_0^{k+1} = (W^{k+1})^T \Sigma_0 W^{k+1}$. We cannot find exact solutions to this problem. However, we know how to solve for $A^{k+1}$ exactly when $W^{k+1}$ is fixed, and how to solve for $W^{k+1}$ exactly when $A^{k+1}$ is fixed. We can iterate these two steps until convergence to obtain an approximate solution to (5).

3.1 Solving for $A^{k+1}$ with fixed $w$

When $w$ is fixed, we can solve for $A^{k+1}$ exactly. According to §2.1 when $W^{k+1}$ is known, the solution of $A^{k+1}$ is

$$A^{k+1} = \begin{bmatrix}
S_0^{k+1} & S_1^{k+1, T} & \cdots & S_{M-1}^{k+1, T} \\
S_1^{k+1} & S_0^{k+1} & \cdots & S_{M-2}^{k+1} \\
\vdots & \vdots & \ddots & \vdots \\
S_{M-1}^{k+1} & S_{M-2}^{k+1} & \cdots & S_0^{k+1}
\end{bmatrix}^{-1}
\begin{bmatrix}
S_1^{k+1} \\
S_2^{k+1} \\
\vdots \\
S_{M}^{k+1}
\end{bmatrix},$$

where $S_\tau^{k+1} = W^{k+1, T} \Sigma_\tau W^{k+1}$, $\tau \in \mathbb{Z}$. 

5
The time complexity for forming $S^{k+1}_1, \ldots, S^{k+1}_M$ is $O(M(k+1)n^2)$. Once $S^{k+1}_\tau$, $\tau = 1, \ldots, M$ are calculated, the time complexity for updating $A^{k+1}$ is dominated by the inversion step, which is $O(M^3(k+1)^3)$. Therefore, the overall time complexity for updating $A^{k+1}$ is max\{$O(M(k+1)n^2), O(M^3(k+1)^3)$\}.

### 3.2 Solving for $w$ with fixed $A^{k+1}$

When $A^{k+1}$ is fixed, we can solve for $W^{k+1}$ exactly. With some derivations, the optimization problem for $w$ can be expressed as

\[
\begin{align*}
\text{minimize} & \quad f(w) = w^T B w - 2c^T w \\
\text{subject to} & \quad W^{k,T} \Sigma_0 w = 0 \\
& \quad w^T \Sigma_0 w = 1,
\end{align*}
\]

where $B > 0$, and $B, c$ are known. The derivation and expressions for $B$ and $c$ can be found in Appendix A. The solution of this problem can be obtained explicitly as follows.

Let $U \in \mathbb{R}^{n \times (n-k)}$ be the orthogonal complement of $\Sigma_0^{1/2} W^k$ and $U^T U = I$. Denote the SVD decomposition of $U^T \Sigma_0^{-1/2} B \Sigma_0^{-1/2} U$ as $U^T \Sigma_0^{-1/2} B \Sigma_0^{-1/2} U = V \Lambda V^T$ with $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_{n-k})$, $(\lambda_1 \leq \lambda_2 \cdots \leq \lambda_{n-k})$. Then, problem (6) can be transformed as

\[
\begin{align*}
\text{minimize} & \quad f(z) = y^T \Lambda y - 2\beta^T y \\
\text{subject to} & \quad y^T y = 1,
\end{align*}
\]

where $\beta = [\beta_1 \beta_2 \cdots \beta_{n-k}] = V^T U^T \Sigma_0^{-1/2} c$ and $w = \Sigma_0^{-1/2} U V y$. The solution to problem (7) has the form $y = (\Lambda + \mu I)^{-1} \beta$ where $\mu$ can be obtained as the root of

\[
\sum_{i=1}^{n-k} \frac{\beta_i^2}{(\lambda_i + \mu)^2} = 1, \quad \text{where } \mu > -\lambda_i.
\]

After $\mu$ is solved, the optimal $w$ of problem (6) can be obtained as

\[
w^* = \Sigma_0^{-1/2} U V (\Lambda + \mu I)^{-1} \beta.
\]

According to the expressions of $B$ and $c$ in Appendix A, the time complexity for forming $B$ and $c$ is $O(M^2n^2)$. Once $B$ and $c$ are calculated, the time complexity for updating $w$ is dominated by the SVD step, which is $O(n^3)$. Therefore, the overall time complexity for updating $w$ is max \{$O(M^2n^2), O(n^3)$\}.

### 3.3 Complete algorithm

We have discussed algorithms to solve for $A^{k+1}$ when $W^{k+1}$ is fixed and to solve for $W^{k+1} = [W^k \ w]$ when $A^{k+1}$ is fixed. The complete procedure to construct the $(k+1)$th column of $W$ is to iterate these two steps until convergence. As discussed in §3.1 and §3.2, the time complexity for updating $A^{k+1}$ is max\{$O(M(k+1)n^2), O(M^3(k+1)^3)$\}, and the time
Algorithm 1 Complete algorithm for solution to (4).

1: Set initial values of $W^0$ as a vector of zeros of proper size, and $A^0 = 0$.
2: for $k = 1, 2, \ldots$ do
3:   repeat
4:     Set $w = 0$
5:     Solve $A^{k+1}$ with fixed $W^{k+1} = [W^k \, w]$ according to §3.1
6:     Solve $W^{k+1} = [W^k \, w]$ with fixed $A^{k+1}$, according to §3.2
7:   until termination condition is satisfied
8: end for

complexity for updating $w$ is max \{O($M^2n^2$), O($n^3$)\}. In practice, it is often the case that $m \ll n$ and $M \ll n$. Therefore, the overall time complexity at each iteration step is O($n^3$).

Once $W^{k+1}$, $A^{k+1}$ are obtained, the same procedure can be applied to construct the next column of $W$. The complete algorithm is given in Algorithm 1.

The heuristic method proposed has two advantages over directly solving the original problem (4). First, there is no uniqueness issue in this recursive method, because starting from the first column of $W$, each column of $W$ is deterministic according to the optimization problem (6). Second, the iterative procedure gives an indication of the dimension of the predictable vector time series.

Let $\alpha_k$ denote the predictability measure for the $k$ dimensional projected time series, or the value of the objective function in (4) with optimal $W^k$ and $A^k$. Then, when extracting the $(k + 1)$th scalar time series, we have the following upper bound for $\alpha_{k+1}$ of the $(k + 1)$-dimensional time series,

$$\alpha_{k+1} \leq \alpha_k + w^T \Sigma_0 w = \alpha_k + 1.$$

When the optimal $\alpha_{k+1}$ is very close to the upper bound $\alpha_k + 1$, we can draw two conclusions. First, adding another scalar time series does not improve the prediction of the $k$ dimensional vector time series. Second, no other self-predictable scalar time series can be extracted. These two facts suggest that all the predictable components in the time series data are already extracted and hence, we can stop the iteration procedure.
4 Examples

In this section, we test our method by applying it to a synthesized high-dimensional time series dataset and a quarterly GDP growth dataset. In both examples, we demonstrate advantages of our proposed method over scalar time series AR fitting.

4.1 Simulation dataset

The proposed method is first tested on a synthesized dataset generated from the model

\[
\begin{align*}
x_t &= B_1 x_{t-1} + B_2 x_{t-2} + v_t, \\
y_t &= P x_t + e_t, \\
z_t &= \Sigma^{-1/2} y_t,
\end{align*}
\]

where

\[
B_1 = \begin{bmatrix} 1.1241 & 0.3045 & 0.3806 \\
0.3902 & -0.8169 & -0.3114 \\
-0.7166 & -0.8630 & 1.0115 \end{bmatrix}, \quad B_2 = \begin{bmatrix} -0.2482 & 0.3676 & 0.0328 \\
-0.4240 & 0.1101 & 0.0267 \\
0.6011 & -0.5975 & -0.3224 \end{bmatrix}.
\]

The matrix \( P \in \mathbb{R}^{1000 \times 3} \) is random with orthonormal columns, \( v_t \) is i.i.d. \( \mathcal{N}(0, I) \) and \( e_t \) is i.i.d \( \mathcal{N}(0, 0.02^2 I) \). \( \Sigma_y \) is the empirical covariance matrix of \( y_t \), calculated as

\[
\Sigma_y = \frac{1}{T} \sum_{t=1}^{T} y_t y_t^T,
\]

where \( T \) is the number of samples. In this example, 50,000 consecutive samples are generated from the model. Hence, \( T = 50,000 \).

It is clear from the model assumption that the empirical covariance of \( z_t \) is approximately an identity matrix. Therefore, static analysis methods like PCA are not able to successfully identify the 3-dimensional most predictable factors.

Next, we fit a 2-memory AR model to each scalar time series in \( z_t \). We use mean squared error (MSE) to evaluate the prediction performance of the fitted AR predictors. We find that, of all 1000 scalar time series, the minimal MSE is 0.9995 and the maximum MSE is 1.0000. Since each scalar time series has approximately mean 0 and variance 1, this indicates that 2-memory scalar AR fitting fails to extract any significant predictability information from the data.

Finally, we apply the proposed method to this simulation dataset with 2-memory and \( m = 3 \). The auto-covariance matrices of \( z_t \) are estimated as

\[
\Sigma_\tau = \frac{1}{T - \tau} \sum_{t=1}^{T-\tau} z_t z_{t+\tau}^T, \quad \tau \in \mathbb{Z}_+.
\]

and

\[
\Sigma_{-\tau} = \Sigma_{\tau}^T, \quad \tau \in \mathbb{Z}_+.
\]
Using the extracted VAR predictor, we find that the MSE of the extracted 3-dimensional principal time series are 0.0594, 0.0916 and 0.0704. Since each of the principal time series have approximately mean 0 and variance 1, the low MSEs indicate that the extracted VAR predictor is able to make highly accurate predictions. This is a significant improvement over scalar AR predictors.

In addition to predictability, we also want to check the similarity between the extracted VAR model and the true model. According to (2.2) and the relationships in (8), the recovered VAR model parameters $A_1$ and $A_2$ can be equivalent to the true model parameters $B_1$, $B_2$ up to a similarity transformation. Therefore, we check how similar the two models are by comparing the eigenvalues of $\begin{bmatrix} A_1 & A_2 \\ I & 0 \end{bmatrix}$ and $\begin{bmatrix} B_1 & B_2 \\ I & 0 \end{bmatrix}$. The closer the two sets of eigenvalues are, the more similar the two models are. Table 1 lists the two sets of eigenvalues. We can see that the two sets of eigenvalues are quite close to each other, which implies that the recovered VAR model is close to the true model. The high predictability of the extracted principal time series and the successfully recovered VAR model demonstrate the effectiveness of the proposed method.

### Table 1: Two sets of eigenvalues.

<table>
<thead>
<tr>
<th>eigenvalue</th>
<th>$A_1$</th>
<th>$A_2$</th>
<th>$B_1$</th>
<th>$B_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.9527</td>
<td>-0.9560</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.9244</td>
<td>0.9230</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>$0.7128 + 0.5548i$</td>
<td>$0.7117 + 0.5542i$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>$0.7128 - 0.5548i$</td>
<td>$0.7117 - 0.5542i$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>-0.2571</td>
<td>-0.2544</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.1954</td>
<td>0.1827</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 2: (Un-)predictabilities of Mexico and Belgium using single variable AR fitting, (un-)
predictabilities of principal time series extracted by the proposed method and (un-)predictabilities
of naïve zero predictor.

<table>
<thead>
<tr>
<th></th>
<th>(Un-)predictability</th>
<th>(Un-)predictability of naïve zero predictor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mexico, $m = 1$</td>
<td>0.8289</td>
<td>1</td>
</tr>
<tr>
<td>Belgium, $m = 1$</td>
<td>0.7939</td>
<td>1</td>
</tr>
<tr>
<td>$M = 1, m = 1$</td>
<td>0.4452</td>
<td>1</td>
</tr>
<tr>
<td>$M = 2, m = 1$</td>
<td>1.0579</td>
<td>2</td>
</tr>
</tbody>
</table>

4.2 GDP dataset

This dataset is composed of seasonally adjusted quarterly GDP growth data of 17 countries
from 1961-Q2 to 2017-Q3. The 17 countries are selected based on the largest GDP countries
according to the world bank data in 2016 with complete GDP growth records from
1961-Q2 to 2017-Q3, and this data is downloaded from https://stats.oecd.org/index.aspx?queryid=350#.
The 17 countries are United States, Japan, United Kingdom, France, Italy, Canada, South Korea, Australia, Spain, Mexico, Netherlands, Switzerland, Germany, Sweden, Belgium, Austria, Norway.

Two approaches are applied to the first 135 samples from 1961-Q2 to 1994-Q4. The
first approach is to fit a single variable AR model to each country’s GDP data. The second
approach is to use the proposed method in this work to extract the most predictable principal
time series. For each approach, the 135 samples are preprocessed such that each variable has
zero mean and unit variance, and an AR model with 1-memory is fitted to the preprocessed
data.

Table 2 summarizes the (un-)predictabilities of fitting single variable AR model to two
representative countries, Mexico and Belgium, the (un-)predictabilities of the one- and two-
dimensional most predictable time series extracted by the proposed method and the (un-)
predictabilities of naïve zero predictor. The naïve zero predictor is defined to always predict
zero. In this case, since all the variables are preprocessed to have zero mean, the naïve zero
predictor is the essentially the same as the mean predictor, and the corresponding (un-)
predictabilities can serve as a reference to evaluate the performance of other predictors.
The reason we pick Mexico and Belgium is that they are the two countries with the best
predictabilities (lowest (un-)predictability values) using single variable AR predictor.

We can see from Table 2 that both approaches result in lower (un-)predictabilities than the
naïve zero predictor, indicating that both approaches give more efficient predictors that
naïve zero predictor. In addition, the improvements in the (un-)predictabilities of principal
time series extracted by the proposed method are much more obvious than the improvements
in the (un-)predictabilities of single variable AR predictor. This implies that the principal
time series extracted by the proposed method are significantly more predictable than any
individual scalar time series.
Figure 1 shows the prediction results of the two approaches. We can see that the predictions of the principal time series extracted by our method are closer to their true values, compared to the predictions of Mexico and Belgium’s scaled GDP using single variable AR predictor. This again demonstrates the effectiveness of the proposed method.

The predictor obtained for the 1 factor \((M = 1)\) case using our proposed method is

\[
\hat{x}_t = 0.7378 x_{t-1}.
\]

The contribution or weight of each country is shown in Figure 2. 5 countries have relatively large contributions, which are Japan, Italy, Spain, Germany and Belgium.

The predictor obtained for the 2 factors \((M = 2)\) case is

\[
\hat{x}_t = \begin{bmatrix}
0.7377 & 0.0047 \\
-0.0096 & -0.6207
\end{bmatrix} x_{t-1}.
\]

The contribution or weight of each country is shown in Figure 3. 7 countries have relatively large contributions, which are Japan, France, Italy, Korea, Spain, Germany and Belgium.

### 4.3 GDP dataset prediction

Next, we examine the prediction performance of the predictors found in §4.2 on unseen data. The data we use in this section are composed of data from 1994-Q4 to 2017-Q3 (samples from 135 to 226 in the same dataset as in §4.2). We understand that the mean and variance of variables in this dataset are considerably different from the mean and variance in the dataset analyzed in §4.2, especially due to the financial crisis around 2008. However, we would still like to check whether the predictors found in §4.2 persist.

First, we shift and scale each scalar time series with its corresponding mean and standard deviation found in §4.2. Then, we apply the predictors found in §4.2 to this preprocessed dataset to make predictions. It is worth noting that this is a very challenging task as it covers the financial crisis period from 2007 to 2009.

We use MSE as a measure of how good the predictions are. Table 3 summarizes the MSE when applying single variable AR predictor to Mexico and Belgium, the MSE when applying the predictor to the principal time series obtained using our proposed method and the MSE of naïve zero predictor. The MSE of naïve zero predictor can be used as a reference to evaluate the performance of the other two predictors.

We can see from Table 3 that both single variable AR predictors and our proposed predictors perform better than naïve zero predictor. However, our proposed method provides dramatic improvements over the naïve zero predictor compared to the single variable AR predictors.

Figure 4 shows the prediction results. We can see that the predictions made using the proposed method are more accurate than the predictions using single variable AR predictor. In addition, the predictor found by our proposed method generalizes well even during the financial crisis period. The first factor captures the big drop around 2008.
Figure 1: Prediction results using two approaches from 1961-Q3 to 1994-Q4. The solid blue line represents the true values and the dashed red line represents the predicted values.
Figure 2: Contribution of each country to the most predictable factor when memory is 1 and the number of factors is 1.

Table 3: MSE of Mexico and Belgium using single variable AR fitting, MSE of principal time series extracted by the proposed method and MSE of naïve zero predictor.

<table>
<thead>
<tr>
<th>Country</th>
<th>m=1</th>
<th>MSE</th>
<th>MSE of naïve zero predictor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mexico, m=1</td>
<td>1.4072</td>
<td>1.8671</td>
<td></td>
</tr>
<tr>
<td>Belgium, m=1</td>
<td>0.3844</td>
<td>0.6536</td>
<td></td>
</tr>
<tr>
<td>M = 1, m=1</td>
<td>0.4890</td>
<td>1.3992</td>
<td></td>
</tr>
<tr>
<td>M = 2, m = 1</td>
<td>0.6787</td>
<td>1.5088</td>
<td></td>
</tr>
</tbody>
</table>
Figure 3: Contribution of each country to the most predictable factor when memory is 1 and the number of factors is 2.
Figure 4: Prediction results using two approaches from 1995-Q1 to 2017-Q3. The solid blue line represents the true values and the dashed red line represents the predicted values.
5 Extensions and variations

There are several extensions of and variations on the problems that we describe in this paper.

**Regularization.** We can add regularization on $W$ and $A$ to the objective. We denote the regularized problem as

$$
\text{maximize } f(W) + r(A) + \tilde{r}(W)
$$

subject to $S_0 = I,$

(9)

where $r : \mathbb{R}^{mM \times m} \to \mathbb{R}$ and $\tilde{r} : \mathbb{R}^{n \times m} \to \mathbb{R}$. $r$ and $\tilde{r}$ can be chosen to enforce certain properties in $A$ and $W$. For example, regularization using $r(A) = \|A\|_F$ can be added to avoid overfitting of the AR model; $r(A) = \|A\|_1$ can be added to encourage a parsimonious structure of the AR model; $\tilde{r}(W) = \|W_i\|_1$ enforces $i$th column of $W$ to be sparse, such that the $i$th most predictable time series only depends on a few entries in the high dimensional time series.

**Low rank structure.** In many cases, high dimensional time series have low rank structure. We can use this low rank structure to reduce the computational complexity of our problem. When the high dimensional time series has low rank structure, the covariance matrix $\Sigma_0$ has many eigenvalues close 0. Let $\tilde{U}$ denote the collections of all the eigenvectors of $\Sigma_0$ corresponding to the significantly non-zero eigenvalues, then original inputs $\Sigma_\tau$, $\tau = 0, 1, \ldots, M$ can be approximately transformed into $\Phi_\tau$, $\tau = 0, 1, \ldots, M$, where $\Phi_\tau = \tilde{U}^T \Sigma_\tau \tilde{U}$. Since the dimension of $\Phi_\tau$ is much lower than the dimension of $\Sigma_\tau$, by working with the new input series $\Phi_\tau$, the computational complexity reduces significantly. Let $W_\phi$ and $A_\phi$ denote the solutions with the new inputs $\Phi_\tau$, then the solutions with the original inputs $\Sigma_\tau$ can be obtained as

$$W = \tilde{U} W_\phi, \quad A = A_\phi.$$

**Filtering.** As an extension of the projection method, we can consider extracting a lower-dimensional time series $x_t$ using a finite impulse response (FIR) filter with length $L$:

$$x_t = W_1^T z_t + W_2^T z_{t-1} + \cdots + W_L^T z_{t-L+1}, \quad t \in \mathbb{Z},$$

where $W_1, \ldots, W_L \in \mathbb{R}^{n \times m}$ are the filter coefficients. For $L = 1$, this reduces to the projection problem (4). We can write this as

$$x_t = W^T \begin{bmatrix} z_t \\ z_{t-1} \\ \vdots \\ z_{t-L+1} \end{bmatrix}, \quad t \in \mathbb{Z},$$

16
where

\[ W = \begin{bmatrix} W_1 \\ W_2 \\ \vdots \\ W_L \end{bmatrix}. \]

Define the following auto-covariance matrices of \( \begin{bmatrix} z_t \\ z_{t-1} \\ \vdots \\ z_{t-L+1} \end{bmatrix} \):

\[ \Omega_0 = \mathbb{E} \begin{bmatrix} z_t \\ z_{t-1} \\ \vdots \\ z_{t-L+1} \end{bmatrix} \begin{bmatrix} z_t \\ z_{t-1} \\ \vdots \\ z_{t-L+1} \end{bmatrix}^T = \begin{bmatrix} \Sigma_0 & \Sigma_1^T & \cdots & \Sigma_{L-1}^T \\ \Sigma_1 & \Sigma_0 & \cdots & \Sigma_{L-2}^T \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_{L-1} & \Sigma_{L-2} & \cdots & \Sigma_0 \end{bmatrix}, \]

\[ \Omega_1 = \mathbb{E} \begin{bmatrix} z_t \\ z_{t-1} \\ \vdots \\ z_{t-L+1} \end{bmatrix} \begin{bmatrix} z_{t+1} \\ z_t \\ \vdots \\ z_{t-L+2} \end{bmatrix}^T = \begin{bmatrix} \Sigma_1 & \Sigma_0 & \cdots & \Sigma_{L-2} \\ \Sigma_2 & \Sigma_1 & \cdots & \Sigma_{L-3} \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma_L & \Sigma_{L-1} & \cdots & \Sigma_1 \end{bmatrix}. \]

The goal is to choose \( W \) so that the series \( x_t \) is most predictable by an \( m \)-memory AR predictor. Similar to the projection problem, we add the constraint \( S_0 = W^T \Omega_0 W = I \) to rule out the trivial \( W = 0 \) case. This also ensures that the low-dimensional time series \( x_t \) is standardized.

**Acknowledgements**

We would like to express our appreciation to Professor Peter Stoica for his valuable and constructive suggestions during the preparation of this paper. We also thank Peter Nystrup for pointing us to related work.
Appendix A  Derivation of (6)

We show how to derive the expression (6) in this appendix. For simplicity, we ignore the superscript \( k + 1 \) in \( A^{k+1}, A_i^{k+1}, i = 1, \ldots, M \), and \( S^{k+1}_\tau, \tau \in \mathbb{Z} \), and the superscript \( k \) in \( W^k \).

When \( A \) is fixed, we have

\[
f(w) = \text{Tr} \left( -2A \begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_M \end{bmatrix} \right) + A \begin{bmatrix} S_0 & S_1^T & \cdots & S_{M-1}^T \\ S_1 & S_0 & \cdots & S_{M-2}^T \\ \vdots & \vdots & \ddots & \vdots \\ S_{M-1} & S_{M-2} & \cdots & S_0 \end{bmatrix} A^T \right)
\]

\[
= -2 \sum_{i=1}^{M} \text{Tr}(A_i S_i) + \text{Tr} \left( \begin{bmatrix} S_0 & S_1^T & \cdots & S_{M-1}^T \\ S_1 & S_0 & \cdots & S_{M-2}^T \\ \vdots & \vdots & \ddots & \vdots \\ S_{M-1} & S_{M-2} & \cdots & S_0 \end{bmatrix} \begin{bmatrix} A_i^T A_1 & A_i^T A_2 & \cdots & A_i^T A_M \\ A_2^T A_1 & A_2^T A_2 & \cdots & A_2^T A_M \\ \vdots & \vdots & \ddots & \vdots \\ A_M^T A_1 & A_M^T A_2 & \cdots & A_M^T A_M \end{bmatrix} \right)
\]

We divide \( A_i, i = 1, 2, \ldots, M \) into the following submatrices,

\[
A_i = \begin{bmatrix} A_{i,11} & A_{i,12} \\ A_{i,21} & A_{i,22} \end{bmatrix} \text{ for } i = 1, 2, \ldots, M,
\]

where \( A_{i,11} \in \mathbb{R}^{k \times k}, A_{i,12} \in \mathbb{R}^{k \times 1}, A_{i,21} \in \mathbb{R}^{1 \times k}, A_{i,22} \in \mathbb{R} \). With this notation, we can expand \( \text{Tr}(A_i S_i) \) as

\[
\text{Tr}(A_i S_i) = \text{Tr} \left[ \begin{bmatrix} A_{i,11} & A_{i,12} \\ A_{i,21} & A_{i,22} \end{bmatrix} \begin{bmatrix} W^T \Sigma_i W & W^T \Sigma_i w \\ w^T \Sigma_i W & w^T \Sigma_i w \end{bmatrix} \right] = w^T(A_{i,22} \Sigma_i)w + (\Sigma_i W A_{i,12} + \Sigma_i^T W A_{i,21}^T)^T w + d,
\]

where \( d \) is a constant. For the second term in \( f(w) \), we have

\[
\text{Tr} \left( S_{M-i}^T A_i A_i^T A_j \right) = \sum_{j-i} S_{j-i} A_i^T A_j,
\]

where \( \text{Tr}(S_{j-i} A_i^T A_j) \) can be expanded as

\[
\text{Tr}(S_{j-i} A_i^T A_j) = \begin{bmatrix} W^T \Sigma_{j-i} W & W^T \Sigma_{j-i} w \\ w^T \Sigma_{j-i} W & w^T \Sigma_{j-i} w \end{bmatrix} \begin{bmatrix} A_{i,11}^T & A_{i,21}^T & A_{i,12}^T & A_{i,22}^T \end{bmatrix}
\]

\[
= \left( A_{i,12}^T A_{j,11} + A_{i,22}^T A_{j,21} \right) W^T \Sigma_{j-i} w + \left( A_{i,11}^T A_{j,12} + A_{i,21}^T A_{j,22} \right) W^T \Sigma_{j-i} w.
\]

18
Summing all terms, we can obtain the following expression for $f(w)$,

$$f(w) = w^T B w - 2c^T w + d,$$

where $d$ is a constant and

$$B = \sum_{1 \leq i, j \leq M} (A_{i,12} A_{j,12} + A_{i,22} A_{j,22}) \Sigma_{j-i} - \sum_{i=1}^{M} A_{i,22}(\Sigma_i + \Sigma_i^T),$$

$$c = \sum_{i=1}^{M} (\Sigma_i W A_{i,12} + \Sigma_i^T W A_{i,21}) - \sum_{1 \leq i < j \leq M} \Sigma_{j-i}^T W (A_{j,11} A_{i,12} + A_{i,22} A_{j,21})$$

$$- \sum_{1 \leq i < j \leq M} \Sigma_{j-i} W (A_{i,11} A_{j,12} + A_{j,22} A_{i,21}).$$

The constant term can be ignored when we want to minimize $f(w)$. It is easy to show that $B \succ 0$. 

19
References


