This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

High-Order Temporal Correlation Model Learning for Time-Series Prediction

Peiguang Jing, Yuting Su, Xiao Jin, and Chengqian Zhang

***Abstract*—Time-series prediction has become a prominent challenge, especially when the data are described as sequences of multiway arrays. Because noise and redundancy may exist in the tensor representation of a time series, we focus on solving the problem of high-order time-series prediction under a tensor decomposition framework and develop two novel multilinear models: 1) the multilinear orthogonal autoregressive (MOAR) model and 2) the multilinear constrained autoregressive (MCAR) model. The MOAR model is designed to preserve as much information as possible from the original tensorial data under orthogonal constraints. The MCAR model is an enhanced version that is developed by replacing orthogonal constraints with an inverse decomposition error term. For both models, we project the original tensor into subspaces spanned by basis matrices to facilitate the discovery of the intrinsic temporal structure embedded in the original tensor. To build connections among consecutive slices of the tensor, we generalize a traditional autoregressive model to tensor form to better preserve the temporal smoothness. Experiments conducted on four publicly available datasets demonstrate that our proposed methods converge within a small number of iterations during the training stage and achieve promising results compared with state-of-the-art methods.**

***Index Terms*—Autoregressive (AR), temporal correlation, tensor decomposition, time-series prediction.**

I. Introduction

THE RAPID growth of data in the volume and variability has brought challenges in various branches of science [1]–[4]. In the past decade, mining time-series data has become a very vigorous research area. Major time-series analysis tasks, including prediction [5], [6], monitoring [7], [8], feedback control [9], anomaly detection [10], [11], clustering [12]–[14], classification [15]–[18], and segmentation [19], cover many areas, such as atmosphere environments, ecology, biology, biomedicine, computer vision, and finance. For instance, we attempt to identify what factors affect stock prices over time in the field of finance. In biomedicine, the electrical activity of the heart is the basis of the diagnostic power of electrocardiograms and is observed with millisecond durations. Time-series prediction attempts to forecast the evolution of a system or phenomena represented by previously observed values. The necessary process for time-series prediction is the identification of the nature of the phenomenon represented by the sequence of observations.

With the increasing availability of various technologies, some observed real-world time-series data with multiple features or channels can be naturally described by the multiway structure, known as high-order time series. For example, we can model the spatio-temporal grid of ocean data in meteorology using a fourth-order time series, wherein four factors of this instance, i.e., latitude, longitude, grid points, and time, can be jointly represented. Three prominent challenges [20] shared by high-order time-series prediction are their contextual constraints, temporal smoothness, and high-order. Contextual constraints indicate that observed time-series data often suffer from noise and missing values, especially for sensor-network-based time-series analysis. Temporal smoothness refers to the correlation among the adjacent observations along the temporal dimension. High-order indicates that time-series data are collected based on multiple aspects and organized using multiway arrays.

A typical approach applied to capture the temporal smoothness of data points is autoregressive (AR) model, which shows significant effectiveness and superiority in addressing stationary data. A wide variety of time-series data, such as dynamic textures, spatio-temporal data, and traffic videos, are frequently modeled by AR model and it extensions [21]–[23]. On this basis, Chang *et al.* [23] presented a comprehensive study for the problem of online serial popularity prediction with AR models. Yu *et al.* [24] developed a temporal regularized matrix factorization (TRMF) framework by employing a novel AR temporal regularizer to model temporal dependency among data points. However, when working on multiway data, traditional time-series prediction models generally reshape time-series into vectors or matrices. Approaches of this type not only easily cause a loss of intrinsic relationship information but also often suffer from the curse of dimensionality. Bahadori *et al.* [25] proposed a fast multivariate spatio-temporal analysis (FMSTA) via low-rank tensor learning framework to capture the shared structures of time-series, in which the future values of the time-series are estimated using a vector AR model [26]. Yang *et al.* [27] introduced tensor to tackle high-dimensional

Manuscript received July 9, 2016; revised February 11, 2018; accepted April 24, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 61572356 and Grant 61303208, and in part by the Tianjin Research Program of Application Foundation and Advanced Technology under Grant 15JCQNJC41600. This paper was recommended by Associate Editor P. Tino. *(Corresponding author: Yuting Su.)*

P. Jing, Y. Su, and X. Jin are with the School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China (e-mail: pgjing@tju.edu.cn; ytsu@tju.edu.cn; jinxiao@tju.edu.cn).

C. Zhang is with the School of Electrical Engineering and Information, Southwest Petroleum University, Chengdu 610500, China (e-mail: zhangcqj@tju.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2018.2832085

2168-2267 © 2018 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

representation problem in modeling user-specific spatial temporal activity preference. Takeuchi *et al.* [28] presented a new tensor factorization method to estimate low-rank latent factors by simultaneously learning the spatial and temporal autocorrelations. Yu *et al.* [29] introduced an accelerated online low-rank tensor learning approach for multivariate spatio-temporal streams. Tan *et al.* [30] presented a novel short-term traffic flow prediction approach based on dynamic tensor completion, in which the traffic data are represented as a dynamic tensor pattern to better capture more information of temporal variabilities and spatial characteristics.

In this paper, we focus on the problem of high-order time-series prediction under a tensor decomposition framework and develop two novel multilinear models: 1) the multilinear orthogonal AR (MOAR) model and 2) the multilinear constrained AR (MCAR) model. For both the MOAR and MCAR algorithms, we jointly learn an AR model to build temporal smoothness and a set of projection matrices to obtain the latent core tensor under specified constraints based on the Tucker decomposition framework. The core tensor, which represents the most significant features of the original time series, is regarded as inputs of AR model to discover the latent temporal correlation patterns. The projection matrices are used to project the original tensors into more intrinsic spaces to avoid redundancy and maximally preserve the temporal dependencies among consecutive core tensor slices. We seek orthogonal projection matrices for the MOAR model to alleviate the presence of noisy data as well as to preserve as much information as possible from the original data. For MCAR model, to better alleviate the effects of noise, we enhance the forward process by replacing orthogonal constraints with an inverse decomposition error term to learn a more robust intrinsic temporal correlation model. Compared with other state-of-the-art methods, our algorithms require fewer parameters to tune and have advantages over short-term time-series prediction tasks. We present the core idea of how our proposed methods perform the process in Fig. 1.

The main contributions of this paper are summarized as follows.

1) The AR model and tensor decomposition are collectively integrated into a unified framework to address high-order time-series analysis and prediction.
2) Two multilinear temporal correlation models, i.e., MOAR and MCAR models, are proposed by taking advantages of tensor decomposition. Both the MOAR and MCAR models employ the AR model to capture the temporal correlation structure in the temporal domain and implement the anisotropy analysis of high-order time series.
3) Experimental results on four public datasets illustrate the effectiveness of our proposed models compared to state-of-the-art algorithms.

The remainder of this paper is organized as follows. In Section II, we first briefly review previous work related to time-series prediction. Then, the basic notation and preliminaries are provided in Section III. The proposed methods are presented in Section IV. Experimental evaluation of our methods is reported in Section V, followed by the conclusions in Section VI.

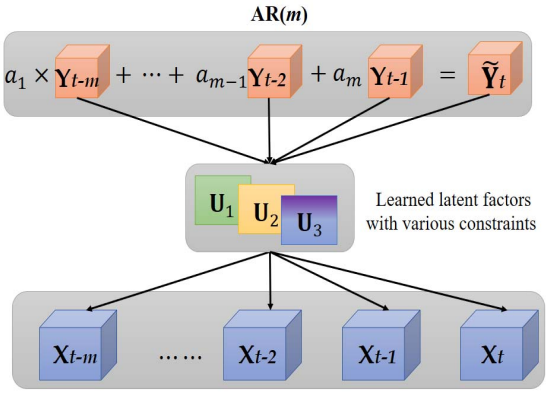


Fig. 1. Schematic illustration of the core idea of our proposed methods.

II. Related Work

There have been numerous works [21], [31]–[34] that have proposed competing models to address prediction tasks, especially for time-series scenario. From the perspective of modeling time-series data, traditional time-series models can roughly be divided into two categories: 1) parametric prediction models and 2) nonparametric prediction models. In a recent study, Cheng *et al.* [35] provided a comparative study of various parametric and nonparametric time-series prediction models used in manufacturing and health informatics areas.

In parametric prediction models, the underlying structure of the time-series can be captured by predetermined functional forms under some unknown parameters. Weron and Misiorek [1] surveyed 12 parametric and semi-parametric time-series to forecast short-term prices in electricity markets. The parametric prediction models cover approaches based on linear methods and approaches based on nonlinear methods. Classical linear prediction approaches include AR, MA, heterogeneous AR [36], AR moving average (ARMA) [37], AR integrated moving average [31], [32], [38], functional-coefficient AR [39], and linear dynamical systems (LDSs) [31]. In nonlinear prediction approaches, Markov chains [40], [41], artificial neural networks [42], and support vector machines (SVMs) are basic structural models that have been widely used to analyze the evolution of the temporal correlations of time series. Typical prediction models include recurrent neural networks [43], fuzzy neural networks [44], hidden Markov models [45], support vector regression (SVR) [46], fuzzy SVR [47], and SVMSA [48]. For example, ElMoaqet *et al.* [8] presented a framework based on weighted SVM for learning a multistep ahead prediction model in physiological monitoring.

In contrast, nonparametric methods generally reduce the particular structure of distributions of time series in a more flexible manner through estimating the conditional distribution, conditional mean, conditional densities, or other characteristics. Advances in nonparametric time-series prediction methods include state-space models [49], [50], Bayesian models [51]–[53], and decomposition models [54]. State-space

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

models predict future values by selectively resampling historical observations, with the basic assumption that future behavior varies smoothly. In nonparametric Bayesian models, the Gaussian process (GP) [51] plays an important role in recovering latent distributions of time-series data. Several classes of Bayesian nonparametric models have been developed to provide flexible and accurate analysis for non-normal time series. For example, the GP convolution model [55] is proposed to model stationary signals as the convolution between a continuous-time white-noise process and a continuous-time linear filter drawn from a GP. Gershman and Blei [53] reviewed Bayesian nonparametric extensions of two common models, mixture models and latent factor models, and illustrated their potential advantages for scientific problems. In [56], a class of nonparametric Bayesian copula models was presented to model the marginal distribution of stationary time-series. Functional decomposition models rely on an adaptive basis, which does not require *a priori* specification of a parametric functional. In [54], Kernel-based nonparametric models were proposed to decompose functions into interpretable components and enable long-range extrapolation on time-series datasets.

A tensor is a high-order generalization of vectors and matrices. It is often the case that time-series accommodate such a structure. Two remarkable tensor decomposition techniques are CANDECOMP/PARAFAC (CP) [57] and Tucker decomposition [58]. The major distinction between CP and Tucker decomposition lies in the fact that CP decomposition decomposes a tensor as a sum of rank-one tensors, while Tucker decomposition learns a latent low-dimensional core tensor and a set of projection matrices. Using various tensor decomposition techniques, many recent efforts have focused on high-order time-series analysis. For example, Sun *et al.* [59] proposed the dynamic tensor analysis (DTA) algorithm to incrementally summarize dynamic high-order time-series data and reveal hidden patterns. One disadvantage of DTA is that it is not appropriate for performing prediction tasks. Cai *et al.* [20] proposed a fast comprehensive mining method involving the convolution of high-order time-series models from a dynamic graphical perspective to address high-order and temporal smoothness challenges. Ma *et al.* [60] proposed a spatio-temporal tensor-based factorization to solve feature extraction of whole-brain fMRI image analysis. Hu *et al.* [61] proposed a low-rank tensor representation model for moving object detection, in which the local spatio-temporal geometric structure information is incorporated into the tensor total variation to compute the motion saliency of foreground. By expanding Bayesian probabilistic matrix factorization [62] with a third-order tensor, Xiong *et al.* [63] presented Bayesian probabilistic tensor factorization (BPTF) to compute CP decompositions for high-order time series. Based on LDS, Rogers *et al.* [64] developed the multilinear dynamical system (MLDS) approach to simultaneously address both the temporal information and the data structure by combining probabilistic graphical framework and multilinear algebra. Similarly, Goutte and Amini [65] proposed probabilistic tensor factorization (PTF), which is an expanded form of the probabilistic matrix factorization (PMF) model [66]. Compared to LDS and PMF, MLDS, and PTF can preserve the tensor structure and estimate an equal number of parameters. Moreover, tensor probabilistic independent component analysis [67] and generalized coupled tensor factorization [68] can also be applied to high-order time series analysis. Although the aforementioned methods combine tensor decomposition with temporal smoothness analysis in an anisotropic manner, however, they require the estimation of many parameters and are more appropriate for the analysis of long-term time series than for short-term time series.

TABLE I
BASIC TENSOR OPERATIONS AND NOTATIONS

Symbols	Description
$\mathcal{X}$	high-order time series with all timestamps
$\mathcal{Y}$	latent core tensor corresponding to $\mathcal{X}$
$\mathcal{X}_t$	tensor slice at t-th timestamp
$\mathcal{Y}_t$	core tensor slice at t-th timestamp
$\mathcal{E}_t$	prediction error of $\mathcal{X}_t$
$\mathbf{U}_n$	n-th component matrix of tensor $\mathcal{X}$
$\mathbf{X}_{(n)}$	mode-n metricized version of tensor $\mathcal{X}$
$\times_n$	n-mode product of a tensor
$\mathcal{X} \times_{(-n)} \{\mathbf{U}\}$	$= \mathcal{X} \times_1 \mathbf{U}_1 \cdots \times_{n-1} \mathbf{U}_{n-1} \times_{n+1} \mathbf{U}_{n+1} \cdots \times_N \mathbf{U}_N$
a_k	coefficients of autoregressive model
T	time dimension of time series
$\mathbf{\Sigma}_y$	noise covariance matrix
$tr(\mathbf{X})$	the trace of $\mathbf{X}$
$\langle \mathcal{X}, \mathcal{Y} \rangle$	scalar product of $\mathcal{X}$ and $\mathcal{Y}$
$\|\mathcal{X}\|_F$	Frobenius norm of tensor $\mathcal{X}$

III. NOTATION AND PRELIMINARIES

In this section, we first briefly summarize the involved basic tensor notations and operations and then review the AR model. Except for some specific cases, we represent a column vector with lowercase bold letters, e.g., $\mathbf{x}$; a matrix with uppercase bold letters, e.g., $\mathbf{X}$; and a tensor with calligraphic letter, e.g., $\mathcal{X}$. For convenience, the basic tensor notations and their descriptions are listed in Table I.

A. Tensor Algebra

An N-order tensor is defined as $\mathcal{X} \in \mathbf{R}^{I_1 \times I_2 \times \cdots \times I_N}$, where $I_n (1 \leq n \leq N)$ is the dimension of mode n. Its elements are denoted as $x_{i_1 i_2 \cdots i_n \cdots i_N}$, where i_n is the index of mode n, with $1 \leq i_n \leq I_n$.

Definition 1 (Mode-n Unfolding): The mode-n unfolding (matricization) of the tensor $\mathcal{X}$ yields a matrix $\mathbf{X}_{(n)} \in R^{I_n \times (I_1 I_2 \cdots I_{n-1} I_{n+1} \cdots I_N)}$, whose columns are all the mode-n fibers arranged in a specific order. See Fig. 2 for an illustration.

Definition 2 (Mode Product): The n-mode product of a tensor $\mathcal{X} \in \mathbf{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and a matrix $\mathbf{U} \in \mathbb{R}^{J \times I_n}$ is an $I_1 \times \cdots \times I_{n-1} \times J \times I_{n+1} \times \cdots \times I_N$ tensor denoted as $\tilde{\mathcal{X}} = \mathcal{X} \times_n \mathbf{U}$, where $\times_n$ indicates the n-mode product.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

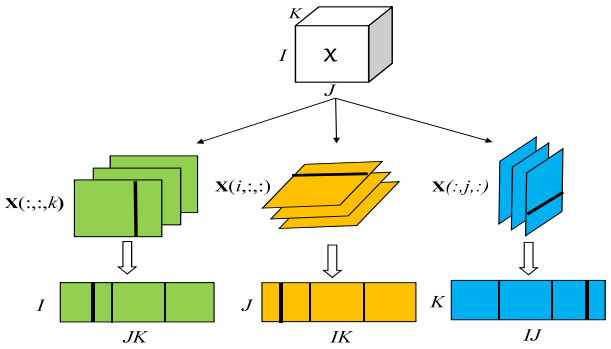


Fig. 2. Graphical illustration of mode-n unfolding for a third-order tensor. The matricization of the tensor $\mathcal{X}$ for the first, second, and third order is shown from left to right.

Its elements are calculated as

$$\tilde{\mathcal{X}}(i_1 \cdots i_{n-1} j i_{n+1} \cdots i_N) = \sum_{i_n=1}^{I_n} \mathcal{X}_{i_1 i_2 \cdots i_N} u_{j i_n}. \tag{1}$$

The n-mode multiplication $\tilde{\mathcal{X}} = \mathcal{X} \times_n \mathbf{U}$ can be manipulated via matrix multiplication by storing the tensors in matricized form, i.e., $\mathbf{B}_{(n)} = \mathbf{U}\mathbf{X}_{(n)}$.

Definition 3 (Scalar Product): The scalar product of two tensors $\mathcal{X}, \mathcal{Y} \in \mathbf{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is denoted as

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} \mathcal{X}_{i_1 i_2 \cdots i_N} \mathcal{Y}_{i_1 i_2 \cdots i_N}. \tag{2}$$

Definition 4 (Tensor Frobenius Norm): The Frobenius norm of a tensor $\mathcal{X}$ is given by

$$\|\mathcal{X}\|_F = \sqrt{\langle \mathcal{X}, \mathcal{X} \rangle} = \sqrt{\sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} \mathcal{X}_{i_1 \cdots i_N}^2}. \tag{3}$$

Definition 5 (Tucker Decomposition): As one of the most commonly used tensor decomposition techniques, Tucker decomposition is often used to extract latent variables or components that can capture the most significant features and remove the redundancy of the original data. For a tensor $\mathcal{X} \in \mathbf{R}^{I_1 \times I_2 \times \cdots \times I_N}$, the Tucker decomposition can be expressed by a set of projection matrices $\mathbf{U}_1, \mathbf{U}_2, \ldots, \mathbf{U}_N$ as follows:

$$\mathcal{Y} = \mathcal{X} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \cdots \times_n \mathbf{U}_n \cdots \times_N \mathbf{U}_N + \mathcal{E} \tag{4}$$

where $\mathcal{Y} \in \mathbf{R}^{J_1 \times J_2 \times \cdots \times J_N}$ is the core tensor, with $J_1 < I_1, J_2 < I_2, \ldots, J_N < I_N$, and $\mathcal{E}$ is the approximation error tensor. Let $\mathbf{U}_i^\dagger$ denote the Moore–Penrose pseudo-inverse of $\mathbf{U}_i$, and the original tensor $\mathcal{X}$ can be reformulated as follows:

$$\mathcal{X} \approx \mathcal{Y} \times_1 \mathbf{U}_1^\dagger \times_2 \mathbf{U}_1^\dagger \cdots \times_n \mathbf{U}_n^\dagger \cdots \times_N \mathbf{U}_N^\dagger. \tag{5}$$

For convenience, the above mathematical operation is called the inverse decomposition in this paper. A visual illustration of a third-order tensor Tucker decomposition is shown in Fig. 3.

The multiplication of a tensor $\mathcal{X}$ by all but one mode n is denoted as

$$\mathcal{X} \times_{(-n)} \{\mathbf{U}\} = \mathcal{X} \times_1 \mathbf{U}_1 \cdots \times_{n-1} \mathbf{U}_{n-1} \times_{n+1} \mathbf{U}_{n+1} \cdots \times_N \mathbf{U}_N. \tag{6}$$

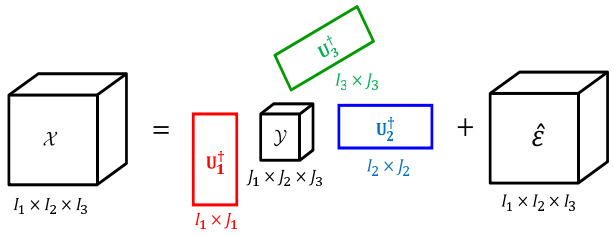


Fig. 3. Illustration of a third-order tensor Tucker decomposition ($\mathcal{Y} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N}$ is the core tensor, ε is the approximation error tensor, $\mathbf{U}_1$, $\mathbf{U}_2$, and $\mathbf{U}_3$ are projection matrices).

Then, the mode-n unfolding matrix of $\mathcal{Y}$ can be approximated by

$$\mathbf{Y}_{(n)} \approx \mathbf{U}_n \big(\mathcal{X} \times_{(-n)} \{\mathbf{U}\} \big)_{(n)}. \tag{7}$$

B. Autoregressive Model

An AR model forecasts the value of the variable at certain future points using a linear combination of the variable's past values. An m-order AR model, namely, AR(m), is defined as follows:

$$x(i) + \sum_{k=1}^{m} a_k x(i-k) = w(i) \tag{8}$$

where the a_k $(k = 1, 2, \ldots, m)$ are the parameters of AR(m) and $w(i)$ is white noise.

IV. Proposed Methodology

In this section, we propose two novel algorithms, the MOAR and MCAR models, for high-order time-series predictions. We first analyze the idea of the MOAR model, which models temporal correlation analysis into a tensor decomposition framework and explain how to derive the feasible solution in Section IV-A. Then, the extended MCAR algorithm is introduced in Section IV-B.

A. Multilinear Orthogonal Autoregressive

Given a general high-order time series $\mathcal{X} \in \mathbf{R}^{I_1 \times I_2 \times \cdots \times I_N \times T}$, where T is the number of observed time slices, N is the number of aspects affecting the tensor time slices. Supposed that the $(N+1)$th order dimension of the tensor $\mathcal{X}$ is the temporal direction, and $\mathcal{X}_t \in \mathbf{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is the tensor slice at timestamp t. By exploiting the advantages of the Tucker decomposition framework, the goal of the MOAR algorithm is to discover a core tensor $\mathcal{Y}_t \in \mathbf{R}^{J_1 \times J_2 \times \cdots \times J_N} (J_1 < I_1, J_2 < I_2, \ldots, J_N < I_N)$ for $\mathcal{X}_t$ and a series of jointly used projection matrices $\mathbf{U}_1, \mathbf{U}_2, \ldots, \mathbf{U}_N$, which are used to maximally preserve the temporal continuity between consecutive slices of the tensor. Formally, $\mathcal{Y}_t$ can be computed from the original observations by projecting it to subspaces spanned by a set of projection matrices as follows:

$$\mathcal{Y}_t = \mathcal{X}_t \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \cdots \times_N \mathbf{U}_N = \mathcal{X}_t \prod_{i=1}^{N} \times_i \mathbf{U}_i. \tag{9}$$

The core tensors, which reflect the intrinsic interactions between multiple factors, are better able to capture the latent

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

temporal correlation than the original tensor slices. To obtain robust feature representations and preserve the inherent temporal smoothness characteristic of the original tensors, we desire to retain the temporal correlations among core tensor slices during the learning process. More specifically, an m-order AR model is employed to build connections between the current core tensor slice $\mathcal{Y}_t$ and the previous slices $\mathcal{Y}_{t-1}, \mathcal{Y}_{t-2}, \ldots, \mathcal{Y}_{t-m}$ as follows:

$$\begin{aligned}\mathcal{Y}_t &= \sum_{k=1}^{m} a_k \mathcal{Y}_{t-k} + \mathcal{E}_t \\ &= \sum_{k=1}^{m} a_k \mathcal{X}_{t-k} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \cdots \times_N \mathbf{U}_N + \mathcal{E}_t \\ &= \sum_{k=1}^{m} a_k \mathcal{X}_{t-k} \prod_{i=1}^{N} \times_i \mathbf{U}_i + \mathcal{E}_t \end{aligned} \tag{10}$$

where $\{a_i\}_{i=1}^m$ are the AR coefficients and $\mathcal{E}_t$ is Gaussian white noise. $\mathcal{E}_t$ has the normal distribution with expectation $0 \in R^{\mathbf{I}}$ with $\mathbf{I} \in \mathbf{N}^M$ and covariances $\Sigma_t \in \mathbf{R}^{\mathbf{II}}$ and write $\mathcal{E}_t \sim N(0, \Sigma_t)$, which is equivalent to $\text{vec}(\mathcal{E}_t) \sim N(\text{vec}(0), \text{mat}(\Sigma_t))$, where $\text{vec}(\cdot)$ and $\text{mat}(\cdot)$, respectively, represent the vectorization and matricization of tensors [64].

It is worth emphasizing that the AR model is suitable for analyzing stationary and strongly nonrandom time series exhibiting a linear pattern because the AR model is essentially a linear regression model. Nonetheless, for long-term time series data encoding certain underlying linear patterns, AR models remain effective. For example, Ferguson *et al.* [69] employed an AR model to characterize seasonal and interannual variations in elevation change time series superimposed upon a long-term linear trend. Veeraraghavan *et al.* [70] proposed to use an AR model on the tangent space to capture the nature of shape deformations of human gait. As a result, our proposed scheme is intrinsically a multilinear learning case, which shares a similar property in discovering latent temporal patterns.

Based on (10), we can derive the following:

$$\begin{aligned}\mathcal{E}_t &= \mathcal{Y}_t - \sum_{k=1}^{m} a_k \mathcal{Y}_{t-k} \\ &= \left(\mathcal{X}_t - \sum_{k=1}^{m} a_k \mathcal{X}_{t-k}\right) \prod_{i=1}^{N} \times_i \mathbf{U}_i. \end{aligned} \tag{11}$$

Generally, the least squares method can be applied here to find the AR coefficients; however, the computational costs become cumbersome as the order N increases. Thus, after applying the maximum likelihood estimation for $\{a_k\}_{k=1}^m$, we obtain the Yule–Walker (YW) of (11) in tensor form, whose revised form can be formulated as follows:

$$\sum_{k=1}^{m} a_k \left(\sum_{t=1}^{T} \langle \mathcal{Y}_{t-k}, \mathcal{Y}_{t-l} \rangle\right) = \sum_{t=1}^{T} \langle \mathcal{Y}_t, \mathcal{Y}_{t-l} \rangle. \tag{12}$$

By maximizing the correlations between real observations and estimations through the AR model, we obtain the following objective function:

$$\begin{aligned}&\max_{\{\mathbf{U}_1, \mathbf{U}_2, \ldots, \mathbf{U}_N\}} \sum_{t=2}^{T} \text{Cov}\left(\mathcal{Y}_t | \widetilde{\mathcal{Y}}_t\right) \\ &\quad \text{s.t. } \mathbf{U}_i \mathbf{U}_i^T = \mathbf{E}, i = 1, \ldots, N \end{aligned} \tag{13}$$

where $\text{Cov}(\mathcal{Y}_t | \widetilde{\mathcal{Y}}_t)$ measures the consistency between the real latent tensor $\mathcal{Y}_t$ and the predicted latent tensor $\widetilde{\mathcal{Y}}_t$. The orthogonality is imposed for projection matrices in an attempt to avoid an arbitrary scaling of the intermediate representation and to preserve as much information as possible.

In our proposed scheme, $\sum_{t=2}^{T} \text{Cov}(\mathcal{Y}_t | \widetilde{\mathcal{Y}}_t)$ is replaced by the cumulative norm of the noise, i.e.,

$$\sum_{t=2}^{T} \|\mathcal{E}_t\|_F^2 = \sum_{t=2}^{T} \left\| \left(\mathcal{X}_t - \sum_{k=1}^{m} a_k \mathcal{X}_{t-k}\right) \prod_{i=1}^{N} \times_i \mathbf{U}_i \right\|_F^2. \tag{14}$$

By combining (13) and (14), the objective function is thus derived as

$$\begin{aligned}&\min_{\{\mathbf{U}_i\}, \{a_k\}} \sum_{t=2}^{T} \left\| \left(\mathcal{X}_t - \sum_{k=1}^{m} a_k \mathcal{X}_{t-k}\right) \prod_{i=1}^{N} \times_i \mathbf{U}_i \right\|_F^2 \\ &\quad \text{s.t. } \mathbf{U}_i \mathbf{U}_i^T = \mathbf{E}, i = 1, \ldots, N. \end{aligned} \tag{15}$$

Denoting $\tilde{\mathcal{E}}_t = \mathcal{X}_t - \sum_{k=1}^{m} a_k \mathcal{X}_{t-k}$, we can derive $\mathcal{E}_t = \tilde{\mathcal{E}}_t \prod_{i=1}^{N} \times_i \mathbf{U}_i$. The equivalent form of $\sum_{t=2}^{T} \|\mathcal{E}_t\|_F^2$ is formulated as follows:

$$\begin{aligned}\sum_{t=2}^{T} \|\mathcal{E}_t\|_F^2 &= \sum_{t=2}^{T} \left\| \tilde{\mathcal{E}}_t \prod_{i=1}^{N} \times_i \mathbf{U}_i \right\|_F^2 \\ &= \sum_{t=2}^{T} \left\| \mathbf{U}_n \left(\tilde{\mathcal{E}}_t \times_{(-n)} \{\mathbf{U}\}\right)_{(n)} \right\|_F^2 \\ &= \text{tr}\left(\mathbf{U}_n \left(\sum_{t=2}^{T} \mathbf{F}_n^t \mathbf{F}_n^{t\,T}\right) \mathbf{U}_n^T\right) \end{aligned} \tag{16}$$

where $\mathbf{F}_n^t = (\tilde{\mathcal{E}}_t \times_{(-n)} \{\mathbf{U}\})_{(n)}$.

We introduce an alternating descent algorithm to solve our objective function. First, given a random initialization $\mathbf{U}_1, \mathbf{U}_2, \ldots, \mathbf{U}_N$, we can estimate the coefficients $\{a_k\}_{k=1}^m$ of AR(m) through (12). Second, we obtain the transformed objective function with fixed $\mathbf{U}_2, \mathbf{U}_3, \ldots, \mathbf{U}_N$ and $\{a_k\}_{k=1}^m$

$$\begin{aligned}&\min_{\mathbf{U}_1} tr\left(\mathbf{U}_1 \left(\sum_{t=2}^{T} \mathbf{F}_1^t \mathbf{F}_1^{t\,T}\right) \mathbf{U}_1^T\right) \\ &\quad \text{s.t. } \mathbf{U}_1 \mathbf{U}_1^T = \mathbf{E}. \end{aligned} \tag{17}$$

A typical way to minimize the above objective function is to use Lagrange multipliers and eigenvalue decomposition. The partial derivative of the above objective function with respect to $\mathbf{U}_1$ is calculated as

$$\sum_{t=2}^{T} \mathbf{F}_1^t \mathbf{F}_1^{t\,T} \mathbf{u}_{1j}^T = \lambda_{1j} \mathbf{u}_{1j}^T \tag{18}$$

where $\mathbf{u}_{1j}$ is a generalized eigenvector of the matrix $\sum_{t=2}^{T} \mathbf{F}_1^t \mathbf{F}_1^{t\,T}$ and λ_{1j} is the corresponding eigenvalue.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Algorithm 1 MOAR for Time-Series Analysis

Input: A tensor time slice $\mathcal{X}_t \in R^{I_1 \times I_2 \times \cdots \times I_N \times T}$;
Output: A set of orthogonal matrices $\{\mathbf{U}_i \in R^{J_i \times I_i}\}_{i=1}^{N}$;
Processing:
1) Give initialized projection matrices $\mathbf{U}_1, \mathbf{U}_2, \cdots, \mathbf{U}_N$;
2) Compute the latent intrinsic core tensor slices $\mathcal{Y}_t = \mathcal{X}_t \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \cdots \times_N \mathbf{U}_N$ at all timestamps;
3) Estimate AR coefficients $\{a_k\}_{k=1}^{m}$ using YW equation in Eq. (12);
4) Compute $\tilde{\mathcal{E}}_t = \mathcal{X}_t - \sum_{k=1}^{m} a_k \mathcal{X}_{t-k}$ for all t;
5) For $i = 1, \cdots, N$
- Compute $\mathbf{F}_i^t = \left(\tilde{\mathcal{E}}_t \times_{(-i)} \{\mathbf{U}\}\right)_{(i)}$ with fixed matrices $\mathbf{U}_1, \cdots, \mathbf{U}_{i-1}, \mathbf{U}_{i+1}, \cdots, \mathbf{U}_N$;
- Update $\mathbf{U}_i$ by conducting eigen-decomposition on matrix $\sum_{t=2}^{T} \mathbf{F}_i^t \mathbf{F}_i^{tT}$;

6) Iterate steps 2)∼5) until convergence;
7) Output orthogonal projection matrices $\mathbf{U}_1, \cdots, \mathbf{U}_N$.

Similarly, we fix $\mathbf{U}_1, \ldots, \mathbf{U}_{i-1} \cdots \mathbf{U}_{i+1} \cdots \mathbf{U}_N$ and obtain the partial derivative of the objective function with respect to $\mathbf{U}_i$

$$\sum_{t=2}^{T} \mathbf{F}_i^t \mathbf{F}_i^{tT} \mathbf{u}_{ij}^T = \lambda_{ij} \mathbf{u}_{ij}^T \tag{19}$$

where $\mathbf{u}_{ij}$ is a generalized eigenvector of the matrix $\sum_{t=2}^{T} \mathbf{F}_i^t \mathbf{F}_i^{tT}$ and λ_{ij} is the corresponding eigenvalue. The whole iterative procedure of the MOAR algorithm is shown in Algorithm 1.

In the prediction stage, the new observations $\mathcal{X}_{T+1}$ can be performed in two steps: 1) computing the new core tensor slices based on the previous core tensor slices $\mathcal{X}_{T-m}, \mathcal{X}_{T-m+1}, \ldots, \mathcal{X}_T$ and 2) building the connections between the new core tensor slices $\mathcal{Y}_{T+1}$ and the new observations $\mathcal{X}_{T+1}$ based on the projection matrices $\mathbf{U}_1, \mathbf{U}_2, \ldots, \mathbf{U}_N$. The prediction procedure can be expressed as follows:

$$\mathcal{X}_{T+1} = \left(\sum_{k=1}^{m} a_k \mathcal{Y}_{T-k}\right) \prod_{i=1}^{N} \times_i \mathbf{U}_i^{\dagger}. \tag{20}$$

B. Multilinear Constrained Autoregressive Algorithm

Similar to the MOAR algorithm, the basic idea of the MCAR algorithm is also to employ the AR model to capture the temporal correlations of consecutive core tensor slices under the Tucker tensor decomposition framework. Although the original time series associated with multiple features and high dimensionality are represented by tensors, distortion, and redundancy remain common in the tensor representations of time series. To alleviate the effects of noise, we improve our algorithm and make it robust to complex time-series prediction situations. To achieve this, we add an inverse decomposition error item as a constraint, which measures the error between the core tensor slices and the corresponding transformed original tensor slices using the learned projection matrices. We write the inverse decomposition error as follows:

$$\min_{\{\mathbf{U}_i\}} \sum_{t=1}^{T} \|\mathcal{Y}_t - \mathcal{X}_t \prod_{i=1}^{N} \times_i \mathbf{U}_i\|_F^2 \tag{21}$$

where $\mathcal{X}_t$ is the real observed tensor slice at timestamp t and $\mathcal{Y}_t$ is the core tensor corresponding to $\mathcal{X}_t$.

When calculating the projection matrices, we adopt the above error term as a constraint to replace the original orthogonality constraint, which not only increases stability of solution but also incorporates more prior information in the training stage. The term can alternately minimize each projection matrix after fixing all other projection matrices. Then, the added inverse decomposition error term allows the objective function to be integrated and rewritten as follows:

$$\begin{aligned} \min_{\{\mathbf{U}_i, a_k, \mathcal{Y}_t\}} & \sum_{t=2}^{T} \left\| \left(\mathcal{X}_t - \sum_{k=1}^{m} a_k \mathcal{X}_{t-k}\right) \prod_{i=1}^{N} \times_i \mathbf{U}_i \right\|_F^2 \\ & + \varphi \sum_{t=1}^{T} \left\| \mathcal{Y}_t - \mathcal{X}_t \prod_{i=1}^{N} \times_i \mathbf{U}_i \right\|_F^2 \end{aligned} \tag{22}$$

where φ is a weighting parameter used to balance the influence between the prediction error and the inverse decomposition error.

Equation (19) can be equivalently rewritten as follows:

$$\mathcal{Y}_t - \mathcal{X}_t \prod_{i=1}^{N} \times_i \mathbf{U}_i \Rightarrow \mathbf{Y}_{t(n)} - \mathbf{U}_n \left(\mathcal{X}_t \times_{(-n)} \{\mathbf{U}\}\right)_{(n)}. \tag{23}$$

By denoting $\mathbf{G}_n^t = (\mathcal{X}_t \times_{(-n)} \{\mathbf{U}\})_{(n)}$, the above function becomes

$$\|\mathcal{Y}_t - \mathcal{X}_t \prod_{i=1}^{N} \times_i \mathbf{U}_i\|_F^2 = \left\|\mathbf{Y}_{t(n)} - \mathbf{U}_n \mathbf{G}_n^t\right\|_F^2. \tag{24}$$

The objective function of MCAR can be reduced to solving the following problem:

$$\begin{aligned} \min_{\{\mathbf{U}_1, \mathbf{U}_2, \ldots, \mathbf{U}_N\}} & tr\left(\mathbf{U}_n \left(\sum_{t=2}^{T} \mathbf{F}_n^t \mathbf{F}_n^{t\,T}\right) \mathbf{U}_n^T\right) \\ & + \varphi \sum_{t=2}^{T} \left\|\mathbf{Y}_{t(n)} - \mathbf{U}_n \mathbf{G}_n^t\right\|_F^2. \end{aligned} \tag{25}$$

A typical way of minimizing the above objective function is to use Lagrange multipliers. Similar to the MOAR model, we use the same alternating strategy to compute the solutions efficiently. Taking the matrix $\mathbf{U}_n$ as a example to illustrate the details of solving process here. We first fix $\mathbf{U}_1, \mathbf{U}_2, \ldots, \mathbf{U}_{n-1}, \mathbf{U}_{n+1}, \ldots \mathbf{U}_N$, and the partial derivative of the objective function with respect to $\mathbf{U}_n$ is calculated as follows:

$$\sum_{t=2}^{T} \mathbf{F}_n^t \mathbf{F}_n^{t\,T} \mathbf{U}_n^T - \varphi\left(\mathbf{Y}_{t(n)} \mathbf{G}_n^{t\,T} - \mathbf{U}_n \mathbf{G}_n^t \mathbf{G}_n^{t\,T}\right) = 0. \tag{26}$$

Then, we can obtain the formulation of $\mathbf{U}_n (1 \leq n \leq N)$ as follows:

$$\mathbf{U}_n^T = \varphi \left(\sum_{t=2}^{T} \mathbf{F}_n^t \mathbf{F}_n^{t\,T}\right)^{-1} \left(\mathbf{Y}_{t(n)} \mathbf{G}_n^{t\,T} - \mathbf{U}_n \mathbf{G}_n^t \mathbf{G}_n^{t\,T}\right). \tag{27}$$

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Algorithm 2 MCAR for Time-Series Analysis

Input: A set of time slices $\{\mathcal{X}_t \in R^{I_1 \times I_2 \times \cdots \times I_N}\}_{t=1}^{T}$;
AR order m;
Output: A set of projection matrices $\{\mathbf{U}_n \in R^{J_n \times I_n}\}_{n=1}^{N}$;
Processing:
1) Give initialized projection matrices $\mathbf{U}_1, \mathbf{U}_2, \cdots, \mathbf{U}_N$;
2) Compute latent low-dimensional tensor series $\mathcal{Y}_t = \mathcal{X}_t \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \cdots \times_N \mathbf{U}_N$ for all t;
3) Estimate AR coefficients $\{a_k\}_{k=1}^{m}$ based on YW equation of Eq.(12);
4) Compute $\tilde{\mathcal{E}}_t = \mathcal{X}_t - \sum_{k=1}^{m} a_k \mathcal{X}_{t-k}$ for all t;
5) For $i = 1, \cdots, N$
- With fixed matrices $\mathbf{U}_1, \cdots, \mathbf{U}_{i-1}, \mathbf{U}_{i+1}, \cdots, \mathbf{U}_N$, compute $\mathbf{F}_i^t = \left(\tilde{\mathcal{E}}_t \times_{(-i)} \{\mathbf{U}\}\right)_{(i)}$;
- With fixed matrices $\mathbf{U}_1, \cdots, \mathbf{U}_{i-1}, \mathbf{U}_{i+1}, \cdots, \mathbf{U}_N$, compute $\mathbf{G}_i^t = (\mathcal{X}_t \times_{(-n)} \{\mathbf{U}\})_{(i)}$;
- Update $\mathbf{U}_i$ with the formulation $2\varphi\left(\sum_{t=2}^{T} \mathbf{F}_i^t \mathbf{F}_i^{t\,T}\right)^{-1}\left(\mathbf{Y}_{t(i)} \mathbf{G}_i^{t\,T} - \mathbf{U}_i \mathbf{G}_i^t \mathbf{G}_i^{t\,T}\right)$;

6) Iterate steps 2)∼5) until convergence;
7) Output matrices $\mathbf{U}_1, \mathbf{U}_2, \cdots, \mathbf{U}_N$.

The whole iterative procedure of the MCAR algorithm is shown in Algorithm 2.

Notably, because real-world tensor data are generally incomplete or include noise noise, tensor completion [71], [72] is developed as a promising technique to extract the intrinsic representation of the tensor data. To some degree, our proposed MCAR algorithm can be deemed as a simplified tensor completion method and especially suits data having a spatio-temporal property.

V. Experiments

In this section, we evaluate how well the proposed methods model high-order time series when applied to time-series prediction. We investigate our methods on four real-world available datasets, including SST, NASDAQ-100, the U.S. historical climatology network (USHCN), and electroencephalogram (EEG). Their sizes range from approximately 100 to approximately 2000. In the following sections, we first give a brief introduction to datasets and experimental settings and then present the prediction results.

A. Dataset and Experimental Settings

In this section, we consider the following four real-word datasets for time-series prediction.

SST: Real-time data of sea surface temperature (SST) from moored ocean buoys. Maintain an array of moored buoys across the Pacific basin. The array took ten years to build and was completed in 1994. Oceanic data are transmitted several times per day to NOAA's polar-orbiting satellites. In this paper, the data are 5-by-6 grids of SST from 5°N, 180°W to 5°S, 110°W recorded hourly from 7:00 P.M. on April 26, 1994 to 3:00 A.M. on July 19, 1994, yielding 2000 epochs [73].

USHCN[1]*:* The USHCN monthly temperature dataset records monthly climatological data from the USHCN to quantify temperature changes in the conterminous United States. This dataset includes four climate statistics: 1) monthly mean maximum temperature; 2) monthly mean minimum temperature; 3) monthly temperature; and 4) total monthly precipitation. In this experiment, we use the climatological data of 120 stations spanning from the year 1920 to the year 2014 to test our algorithms. The averaged statistics are calculated for each year for each of the 120 stations, and finally, a high-order time series of size $120 \times 4 \times 95$ is obtained.

NASDAQ-100: The NASDAQ-100 index includes 100 of the largest domestic and international nonfinancial securities based on market capitalization listed on the stock market. In this paper, the data are opening, closing, adjusted-closing, high, and low prices and volume for 20 randomly chosen NASDAQ-100 companies, from January 1, 2005 to December 31, 2009 (1259 epochs) [64].

EEG: Human brain activity can often be described by EEGs. The EEG dataset [74] was downsampled to a 128 Hz sampling rate from 32 electrodes (Fp1, Fp2, AF3, AF4, F7, F3, Fz, F4, F8, FC1, FC5, FC6, FC2, T7, C3, Cz, C4, T8, CP1, CP5, CP6, CP2, P7, P3, Pz, P4, P8, PO3, PO4, O1, Oz, and O2) placed at the standard positions of the active two system. In total, 942 epochs, starting from the onset of a flash, are extracted from the raw data.

In our experiments, all datasets are divided into training sets and test sets. To make fair comparison, we conduct 15 experiments and report the average performance of the proposed method and other baselines. For the SST dataset, the first 1940 epochs are chosen as the training set, and the remaining epochs are treated as the test set. For the USHCN dataset, we train all models with the first 60 time steps and predict the next 25 time steps. For the NASDAQ-100 dataset, we use the first 1200 epochs to train models and predict the following 59 epochs. For the EEG dataset, the first 900 epochs are used to train the models, and the remaining frames constitute the test set. Typically, we use grid search to set the best parameters on the development portion, and set a maximum number of 30 iterations as the stopping criteria for the MOAR and MCAR algorithms. To assess the predicted accuracy with the ground truth results, we adopt the relative Euclidean distance and compute the cumulative prediction error to evaluate the prediction performance. Specifically, the prediction performance at timestamp t' is evaluated using the following formulation:

$$e(t') = \sum_{t=t_{\text{train}}+1}^{t_{\text{train}}+t'} \frac{\|\mathcal{X}_t - \mathcal{P}_t\|_2}{\|\mathcal{X}_t\|_2} \tag{28}$$

where t_{train} is the length of training set, $\mathcal{P}_t$ is the predicted tensor slice at timestamp t of a time series of length N, and $\mathcal{X}_t$ is the ground-truth tensor slice value. A lower $e_{t'}$ indicates a better algorithm.

[1] http://www.ncdc.noaa.gov/oa/climate/research/ushcn/

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

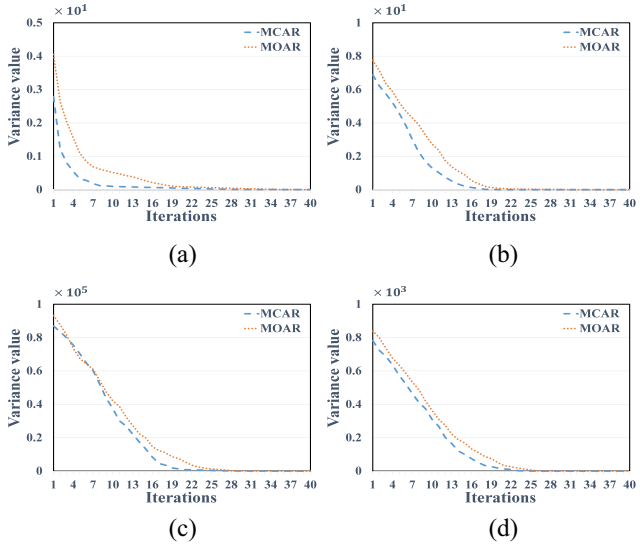


Fig. 4. Convergence curves of our algorithms for the (a) SST dataset, (b) USHCN dataset, (c) NASDAQ-100 dataset, and (d) EEG dataset. The x-axis represents the number of iterations, and the y-axis is the divergence between two consecutive measured $\mathbf{U}$s.

B. Performance Evaluation

To obtain a comprehensive evaluation of our proposed algorithms, in the following experiments, we evaluate the proposed algorithms from the following five perspectives.

1) *Convergency:* We test the convergence of our algorithms based on the proposed alternating algorithm.
2) *Parameter Sensitivity:* How do the various weighting parameters φ and AR order m influence the prediction accuracy.
3) *Training Size:* We evaluate the effect of the size of the training set on two proposed algorithms.
4) *Effectiveness:* The performance comparison with several state-of-the-art algorithms is investigated.
5) *Running Time:* We evaluate the time complexity of the proposed algorithms.

1) Evaluation on Convergence: In this section, we test the convergence and effectiveness of objective functions based on the proposed alternating algorithms and randomly select a trial to report the results. Because $\mathbf{U}$ is used for projecting original tensors, we would like to measure the variance between two sequential $\mathbf{U}$s by the following metric:

$$D(t) = \sum_{i=1}^{N} \left\| \mathbf{U}_i^{t+1} \right\|_F - \left\| \mathbf{U}_i^{t} \right\|_F. \tag{29}$$

This will guarantee that the final feature results will not be changed drastically.

Fig. 4 presents the absolute values of the variance during the iterations. As can be seen, the divergence values obtained for both the MOAR and MCAR algorithms decrease rapidly with increasing iteration number and converge no more than 30 iterations. Compared with the MOAR algorithm, the MCAR algorithm achieves faster convergence, indicating higher efficiency and lower complexity. Based on the above analysis, the iterative criteria are essential to guaranteeing the convergency of our objective function. Therefore, in this paper, we use the relative change between two consecutive iterations falling below a threshold of 1e-1 and a maximum of 30 iterations as the stopping criteria for our proposed methods.

2) Evaluation of Parameter Sensitivity: Because there are two different main parameters, i.e., m and φ, we would like to provide the results with different values of parameters. The influence of different values of m and φ in our algorithms on four datasets is shown in Figs. 5 and 6, respectively. In our experiments, the parameters m and φ are selected via grid search in a heuristic manner.

Fig. 5 depicts the performance of our proposed algorithms with various m values. From this figure, we can observe that the MCAR algorithm significantly outperforms the MOAR algorithm both in terms of prediction performance and AR order.

1) Due to the desired property of the inverse decomposition term, the MCAR algorithm achieves a better prediction performance than the MOAR algorithm for almost all AR model orders. This indicates that the inverse decomposition term is conducive to reducing the prediction error effectively.
2) The optimal AR model orders for the MOAR and MCAR algorithms on four datasets are $\{40, 2, 86, 13\}$ and $\{17, 2, 37, 5\}$, respectively. This phenomenon apparently indicates that the MCAR algorithm can capture more intrinsic latent subspaces spanned by spatial basis matrices with a lower AR model order.

Fig. 6 depicts the performance of the proposed MCAR algorithm with different values of φ ranging from 0.1 to 1. From this figure, we can see that the best φ on the four dataset are the set 0.8, 0.6, 0.2, and 0.7, respectively. For the SST dataset, the proposed MCAR algorithm is not sensitive to the parameter φ, whereas other datasets are particularly sensitive. For the USCHN and EEG datasets, the best φ all appear in the second half of the curve, meaning that the inverse decomposition term plays a key role in finding the optimal projection matrices. Moreover, when φ increases, the prediction error does not monotonically decrease. For the NASDAQ-100 dataset, the best performance is achieved when $\varphi = 0.2$, and the prediction errors do not monotonically decrease as φ increases.

3) Evaluation of Training Size: To further demonstrate the effect of the size of the training sets on the MOAR and MCAR algorithms, we conduct experiments on four datasets while varying the training set size from 10% to 90% of the tensor slices. For each dataset, the remaining 10% of the time slices followed the training set are considered as the test set.

Table II shows prediction performance of MOAR and MCAR on four datasets. From the table, we can see that our methods achieve better performance with increased size of training set. Among four datasets, the best performance always is achieved on SST dataset, while the worst performance is obtained on EEG dataset. The main reason is that despite EEG dataset has strong dependency in temporal domain, but the data sampled from various electrodes remain relatively separate in spatial domain. Moreover, both SST and USHCN are temperature related dataset, the performance on USHCN dataset is poor than SST dataset. One possible reason is that SST dataset is produced with more distinct spatio-temporal

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

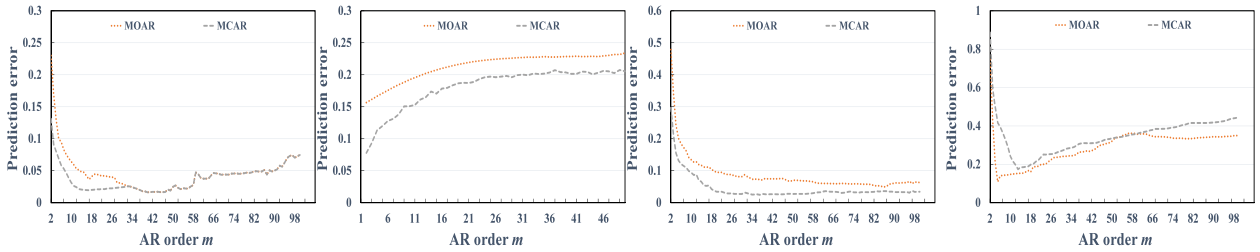


Fig. 5. Effect of the parameter m for the MOAR and MCAR algorithms on different datasets. (a) SST dataset, (b) USHCN dataset, (b) NASDAQ-100 dataset, and (d) EEG dataset.

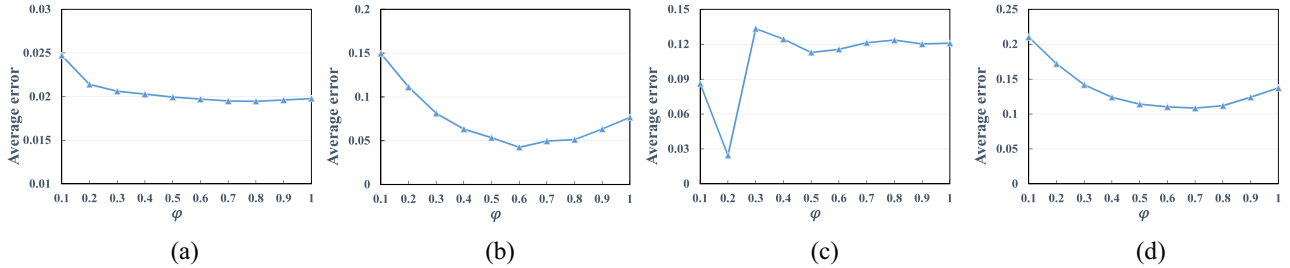


Fig. 6. Effect of the parameter φ for the MCAR algorithm on different datasets. (a) SST dataset, (b) USHCN dataset, (b) NASDAQ-100 dataset, and (d) EEG dataset.

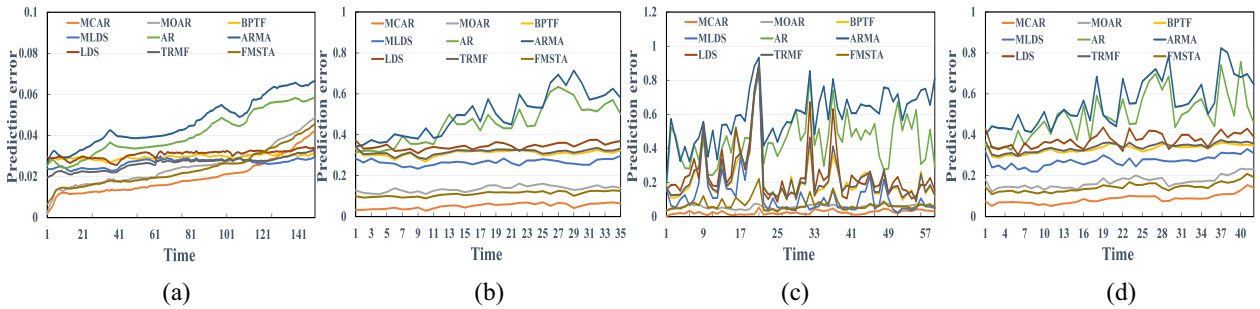


Fig. 7. Prediction accuracy in comparison with state-of-the-art algorithms. (a) SST dataset, (b) USHCN dataset, (b) NASDAQ-100 dataset, and (d) EEG dataset.

TABLE II
PREDICTION PERFORMANCE FOR MOAR AND MCAR WITH DIFFERENT PERCENTAGES OF THE TRAINING SET

Training Size		10%	20%	30%	40%	50%	60%	70%	80%	90%
MOAR	SST	0.422	0.238	0.163	0.124	0.095	0.082	0.075	0.061	0.057
	USHCN	0.572	0.427	0.285	0.212	0.158	0.122	0.092	0.066	0.039
	NASDAQ	0.744	0.522	0.470	0.412	0.323	0.264	0.197	0.132	0.102
	EEG	0.954	0.924	0.712	0.626	0.544	0.487	0.397	0.287	0.223
MCAR	SST	0.388	0.227	0.159	0.116	0.093	0.070	0.055	0.040	0.026
	USHCN	0.481	0.365	0.282	0.196	0.132	0.078	0.046	0.020	0.004
	NASDAQ	0.443	0.375	0.162	0.122	0.089	0.075	0.066	0.046	0.033
	EEG	0.945	0.657	0.512	0.487	0.410	0.365	0.291	0.242	0.169

characteristics, indicating that our proposed algorithms may be more appropriate to dataset with strong spatio-temporal consistency. NASDAQ dataset is a finance related dataset, the observations from stock market often fluctuate dedicated to various reasons, the performance on NASDAQ is not ideal. Furthermore, MCAR outperforms better than MOAR for the same size of training set showing the superiority for tacking various prediction tasks.

4) Evaluation of Effectiveness: To judge the performance of our proposed algorithms, we also perform a comparison with the following state-of-the-art algorithms: AR, ARMA, LDS, MLDS, BPTF, FMSTA, and TRMF. It is worth noting that we conduct our experiments using the following prediction strategy: predict the values at a time point using prior estimated values during the test stage. Although this approach may lead to error accumulation to some

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

degree, the approach becomes more practical for real-world applications.

Fig. 7 depicts the prediction performance of our proposed algorithms on four datasets. From the figure, we can observe the following conclusions.

1) In most cases, the best performance is always achieved with the MCAR algorithm, thus demonstrating the superiority of our proposed scheme.
2) For each dataset, MLDS achieves a better prediction accuracy than LDS. The MCAR and MOAR algorithms are found to be significantly better than AR and ARMA. This is an encouraging result because it highlights the importance of multilinear analysis for high-order time series analysis. Moreover, MOAR, MCAR, MLDS, BPTF, and FMSTA outperform tractional methods, such as TAR, ARMA, LDS, and TMRF, owing to the superiority of tensor in modeling tensor time series.
3) FMSTA offers competitive performance compared with other tensor-based methods over four datasets, which illustrates the effectiveness of low-rank structure assumption in solving spatio-temporal data analysis.

Although MCAR and MLDS use the similar strategies to compute a latent core tensor, but MCAR achieves better prediction performance than MLDS. This is because that MLDS incorporates the concepts of probabilistic graphical models and multilinear algebra to capture the temporal structure of tensor data and the latent parameters in MLDS are estimated by an expectation-maximization algorithm; by contrast, MCAR combines AR and multilinear algebra to discover the latent temporal correlation models and estimates parameters through a simple algebraic equation, which is well suited for dealing with scenarios that encodes a certain underlying linear pattern. Moreover, MCAR also introduces inverse decomposition term to further enhance connections with original tensor representation in spatial domain. For the SST dataset, the curves of our algorithms are relatively stable but change drastically with increasing number of time points. The prediction errors become larger than those of MLDS and LDS when the time point comes to 1911. Indeed, the main reason for this phenomenon is that MLDS and LDS predict values using real observations, not estimated values. Moreover, this also reflects that our proposed algorithms are robust for short-term prediction tasks.

We also performed a pair-wise t-test analysis between our proposed methods and other methods, i.e., MLDS and BPTF, based upon predicted values. P-value is adopted to assess whether the superiority of our methods are statistically significant [4], [75]. According to the results, we find that the P-values of both two proposed algorithms are smaller than the significance level of 0.05, which indicates that the null hypothesis is clearly rejected and the proposed methods are statistically significant.

5) Evaluation of Running Time: In this section, we conduct a survey of the execution time and report the average results in Table III. From the table, we can see that the smallest running time is achieved on the USHCN dataset, whereas the most time-consuming dataset is the NASDAQ-100 dataset. When the size of time-series fixed, the MCAR algorithm requires shorter running time than the MOAR algorithm on all datasets because the optimal order of the AR in the MCAR is smaller than that in the MOAR. This indicates that the running time depends on both the order of the AR model and the size of the time series, as predicted. Moreover, we can observe that MLDS and BPTF are relatively are time-consuming due to the complex computation process. By contrast, FMSTA has lower time complexity due to the usage of the fast greedy solver to guarantee its convergence.

TABLE III
AVERAGE RUNNING TIME (IN SECONDS) COMPARISON WITH DIFFERENT TENSOR-BASED ALGORITHMS ON FOUR DATASETS

	SST	USHCN	NASDAQ-100	EEG
MLDS	436.8	113.1	655.2	242.9
BPTF	382.3	94.8	492.2	223.0
FMSTA	161.5	44.3	323.1	64.7
MOAR	337.9	57.2	601.3	167.5
MCAR	88.4	26.4	76.3	51.5

VI. CONCLUSION

To address high-order time-series prediction tasks, in this paper, we proposed two novel algorithms for high-order time-series: the MOAR and the MCAR. By representing high-order time-series by tensors. Both two methods integrate the AR model into the Tucker decomposition framework to learn the latent core tensor through a series of projection matrices simultaneously. The core tensor, which represents the most significant features, is used to maximally preserve the temporal correlations among consecutive core tensor slices. The experimental results on four datasets demonstrate that our proposed algorithms achieves significant performance improvements.

APPENDIX
YULE-WALKER ESTIMATION

We give an explicit explanation for the tensor-based Yule-Walker (YW) estimation used in (12). Given a set of observation data $\{\mathcal{Y}_n\}_{n=1}^T$, if those data are assumed to follow an m-order AR model AR(m)

$$\mathcal{Y}_n + a_1\mathcal{Y}_{n-1} + a_2\mathcal{Y}_{n-2} + \cdots + a_N\mathcal{Y}_{n-m} = \mathcal{W}_n \tag{30}$$

where $a_1, a_2, \ldots, a_N$ are the parameters of the AR model and $\mathcal{W}_n$ is a white noise.

The YW equations build connections between AR model parameters and the auto-covariance of given tensor time-series. Two steps are employed to find the AR model parameters.

1) Estimating the auto-covariance sequence for a given $\{\mathcal{Y}_n\}_{n=1}^T$.
2) Finding the AR parameters $a_1, a_2, \ldots, a_N$ of the AR(m) model by solving the YW equations.

Equation (30) can be written in a compact form as

$$\sum_{i=0}^{m} a_k\mathcal{Y}_{n-k} = \mathcal{W}_n, \quad a_0 = 1. \tag{31}$$

It is noted that a_0 is the scaling factor corresponding to $\mathcal{Y}_n$, which is set as $a_0 = 1$.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

By multiplying (31) with $\mathcal{Y}_{n-l}$, we can derive that

$$\sum_{k=0}^{m} a_k E\{\langle \mathcal{Y}_{n-k}, \mathcal{Y}_{n-l}\rangle\} = E\{\langle W_n, \mathcal{Y}_{n-l}\rangle\}. \tag{32}$$

One can readily identify the auto-correlation and cross-correlation terms as $E\{\langle \mathcal{Y}_{n-k}, \mathcal{Y}_{n-l}\rangle\} = r_{xx}(l-k)$ and $E\{\langle W_n, \mathcal{Y}_{n-l}\rangle\} = r_{wx}(l)$; then, we have

$$\sum_{k=0}^{m} a_k r_{xx}(l-k) = r_{wx}(l). \tag{33}$$

The next step is to compute the identified cross-correlation $r_{wx}(l)$ term and relate it to the auto-correlation term $r_{xx}(l-k)$. The term $\mathcal{Y}_{n-l}$ can also be obtained from (30) as

$$\mathcal{Y}_{n-l} = -\sum_{k=1}^{m} a_k \mathcal{Y}_{n-k-l} + \mathcal{W}_{n-l}. \tag{34}$$

Generally, we note that $\langle \mathcal{Y}_{n-k-l}, \mathcal{W}_{n-l}\rangle = 0$ because the random perturbation $\mathcal{W}_{n-l}$ of time $n-l$ is uncorrelated with the previous time-series data of the process. Similarly, the auto-correlation $\langle \mathcal{W}_{n-l}, \mathcal{W}_n\rangle = 0 (l \neq 0)$ is zero at all time lags. When $l = 0$, its value is defined as σ^2. According to the above discussion, we draw the following conclusion:

$$\begin{aligned} r_{wx}(l) &= E\{\langle \mathcal{W}_n \cdot \mathcal{Y}_{n-l}\rangle\} \\ &= E\left\{\left\langle \mathcal{W}_n, -\sum_{k=1}^{m} a_k \mathcal{Y}_{n-k-l} + \mathcal{W}_{n-l}\right\rangle\right\} \\ &= E\left\{-\sum_{k=1}^{m} a_k \langle \mathcal{Y}_{n-k-l}, \mathcal{W}_n\rangle + \langle \mathcal{W}_{n-l}, \mathcal{W}_n\rangle\right\} \\ &= E\{0 + \langle \mathcal{W}_{n-l}, \mathcal{W}_n\rangle\} \\ &= E\{\langle \mathcal{W}_{n-l}, \mathcal{W}_n\rangle\} \\ &= \begin{cases} 0, & l > 0 \\ \sigma^2, & l = 0. \end{cases} \end{aligned} \tag{35}$$

By substituting (35) into (33)

$$\sum_{k=0}^{m} a_k r_{xx}(l-k) = \begin{cases} 0, & l > 0 \\ \sigma^2, & l = 0. \end{cases} \tag{36}$$

For the $l > 0$ case, (36) takes on the following form:

$$\sum_{k=1}^{m} a_k r_{xx}(l-k) = -r_{xx}(l). \tag{37}$$

Now, (37) can be written in matrix form

$$\begin{bmatrix} r_{xx}(0) & r_{xx}(-1) & \cdots & r_{xx}(1-m) \\ r_{xx}(1) & r_{xx}(0) & \cdots & r_{xx}(2-m) \\ \vdots & \vdots & \ddots & \vdots \\ r_{xx}(m-1] & r_{xx}(m-2) & \cdots & r_{xx}(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} = -\begin{bmatrix} r_{xx}(1) \\ r_{xx}(2) \\ \vdots \\ r_{xx}(m) \end{bmatrix}. \tag{38}$$

These are the YW equations, which consist of a set of m linear equations and m unknown parameters. By representing (38) in a compact form, we have

$$\mathbf{Ha} = -\mathbf{r}. \tag{39}$$

Since $\mathbf{H}$ is full rank and symmetric, and thus the invertability of $\mathbf{H}$ is guaranteed. The solutions $\mathbf{a}$ can be obtained by solving

$$\mathbf{a} = -\mathbf{H}^{-1}\mathbf{r}. \tag{40}$$

REFERENCES

[1] R. Weron and A. Misiorek, "Forecasting spot electricity prices: A comparison of parametric and semiparametric time series models," *Int. J. Forecast.*, vol. 24, no. 4, pp. 744–763, 2008.

[2] L. Nie, M. Wang, Y. Gao, Z.-J. Zha, and T.-S. Chua, "Beyond text QA: Multimedia answer generation by harvesting Web information," *IEEE Trans. Multimedia*, vol. 15, no. 2, pp. 426–441, Feb. 2013.

[3] D. Cao *et al.*, "Embedding factorization models for jointly recommending items and user generated lists," in *Proc. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2017, pp. 585–594.

[4] L. Nie, Y.-L. Zhao, M. Akbari, J. Shen, and T.-S. Chua, "Bridging the vocabulary gap between health seekers and healthcare knowledge," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 2, pp. 396–409, Feb. 2015.

[5] O. Anava, E. Hazan, and A. Zeevi, "Online time series prediction with missing data," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2191–2199.

[6] J. Zhao, Z. Han, W. Pedrycz, and W. Wang, "Granular model of long-term prediction for energy system in steel industry," *IEEE Trans. Cybern.*, vol. 46, no. 2, pp. 388–400, Feb. 2016.

[7] S. Yin, S. X. Ding, X. Xie, and H. Luo, "A review on basic data-driven approaches for industrial process monitoring," *IEEE Trans. Ind. Electron.*, vol. 61, no. 11, pp. 6418–6428, Nov. 2014.

[8] H. ElMoaqet, D. M. Tilbury, and S. K. Ramachandran, "Multi-step ahead predictions for critical levels in physiological time series," *IEEE Trans. Cybern.*, vol. 46, no. 7, pp. 1704–1714, Jul. 2016.

[9] M. W. Mosconi *et al.*, "Feedforward and feedback motor control abnormalities implicate cerebellar dysfunctions in autism spectrum disorder," *J. Neurosci.*, vol. 35, no. 5, pp. 2015–2025, 2015.

[10] H. Wang, M. Tang, Y.-S. Park, and C. E. Priebe, "Locality statistics for anomaly detection in time series of graphs," *IEEE Trans. Signal Process.*, vol. 62, no. 3, pp. 703–717, Feb. 2014.

[11] F. Rasheed and R. Alhajj, "A framework for periodic outlier pattern detection in time-series sequences," *IEEE Trans. Cybern.*, vol. 44, no. 5, pp. 569–582, May 2014.

[12] J. Hensman, M. Rattray, and N. D. Lawrence, "Fast nonparametric clustering of structured time-series," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 2, pp. 383–393, Feb. 2015.

[13] M. Chaouch, "Clustering-based improvement of nonparametric functional time series forecasting: Application to intra-day household-level load curves," *IEEE Trans. Smart Grid*, vol. 5, no. 1, pp. 411–419, Jan. 2014.

[14] J. Zhang, X. Li, P. Jing, J. Liu, and Y. Su, "Low-rank regularized heterogeneous tensor decomposition for subspace clustering," *IEEE Signal Process. Lett.*, vol. 25, no. 3, pp. 333–337, Mar. 2018.

[15] E. S. García-Treviño and J. A. Barria, "Structural generative descriptions for time series classification," *IEEE Trans. Cybern.*, vol. 44, no. 10, pp. 1978–1991, Oct. 2014.

[16] J. Zhang, C. Xu, P. Jing, C. Zhang, and Y. Su, "A tensor-driven temporal correlation model for video sequence classification," *IEEE Signal Process. Lett.*, vol. 23, no. 9, pp. 1246–1249, Sep. 2016.

[17] J. Mei, M. Liu, Y.-F. Wang, and H. Gao, "Learning a Mahalanobis distance-based dynamic time warping measure for multivariate time series classification," *IEEE Trans. Cybern.*, vol. 46, no. 6, pp. 1363–1374, Jun. 2016.

[18] Y. Su, H. Wang, P. Jing, and C. Xu, "A spatial-temporal iterative tensor decomposition technique for action and gesture recognition," *Multimedia Tools Appl.*, vol. 76, no. 8, pp. 10635–10652, 2017.

[19] J. Qi *et al.*, "Indexable online time series segmentation with error bound guarantee," *World Wide Web*, vol. 18, no. 2, pp. 359–401, 2015.

[20] Y. Cai, H. Tong, W. Fan, P. Ji, and Q. He, "Facets: Fast comprehensive mining of coevolving high-order time series," in *Proc. ACM Int. Conf. Knowl. Disc. Data Min.*, 2015, pp. 79–88.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

[21] J. Angeby, "Properties of the structured auto-regressive time-frequency distribution," in *Proc. IEEE Conf. Acoustics Speech Signal Process.*, 1997, pp. 2017–2020.

[22] A. B. Chan and N. Vasconcelos, "Probabilistic kernels for the classification of auto-regressive visual processes," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, vol. 1. San Diego, CA, USA, 2005, pp. 846–851.

[23] B. Chang *et al.*, "Predicting the popularity of online serials with autoregressive models," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2014, pp. 1339–1348.

[24] H.-F. Yu, N. Rao, and I. S. Dhillon, "Temporal regularized matrix factorization for high-dimensional time series prediction," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 847–855.

[25] M. T. Bahadori, Q. R. Yu, and Y. Liu, "Fast multivariate spatio-temporal analysis via low rank tensor learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3491–3499.

[26] S. Johansen *et al.*, *Likelihood-Based Inference in Cointegrated Vector Autoregressive Models*. Oxford, U.K.: Oxford Univ. Press, 1995.

[27] D. Yang, D. Zhang, V. W. Zheng, and Z. Yu, "Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 1, pp. 129–142, Jan. 2015.

[28] K. Takeuchi, H. Kashima, and N. Ueda, "Autoregressive tensor factorization for spatio-temporal predictions," in *Proc. IEEE Int. Conf. Data Mining*, New Orleans, LA, USA, 2017, pp. 1105–1110.

[29] R. Yu, D. Cheng, and Y. Liu, "Accelerated online low rank tensor learning for multivariate spatiotemporal streams," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 238–247.

[30] H. Tan, Y. Wu, B. Shen, P. J. Jin, and B. Ran, "Short-term traffic prediction based on dynamic tensor completion," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 8, pp. 2123–2133, Aug. 2016.

[31] Z. Ghahramani and G. E. Hinton, "Parameter estimation for linear dynamical systems," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Rep. CRG-TR-96-2, 1996.

[32] C. M. Bishop, *Pattern Recognition and Machine Learning* (Information Science and Statistics). New York, NY, USA: Springer, 2006.

[33] L. Nie, M. Wang, Z.-J. Zha, and T.-S. Chua, "Oracle in image search: A content-based approach to performance prediction," *ACM Trans. Inf. Syst.*, vol. 30, no. 2, p. 13, 2012.

[34] L. Nie *et al.*, "Beyond doctors: Future health prediction from multimedia and multimodal observations," in *Proc. ACM Int. Conf. Multimedia*, 2015, pp. 591–600.

[35] C. Cheng *et al.*, "Time series forecasting for nonlinear and non-stationary processes: A review and comparative study," *IIE Trans.*, vol. 47, no. 10, pp. 1053–1071, 2015.

[36] M. McAleer and M. C. Medeiros, "A multiple regime smooth transition heterogeneous autoregressive model for long memory and asymmetries," *J. Econometrics*, vol. 147, no. 1, pp. 104–119, 2008.

[37] R. Ramachandran and V. R. Bhethanabotla, "Generalized autoregressive moving average modeling of the bellcore data," in *Proc. 25th Annu. IEEE Conf. Local Comput. Netw.*, Tampa, FL, USA, 2000, pp. 654–661.

[38] W. H. Lin, A. Kulkarni, and P. Mirchandani, "Short-term arterial travel time prediction for advanced traveler information systems," *J. Intell. Transp. Syst.*, vol. 8, no. 3, 2004, pp. 143–154.

[39] Z. Cai, J. Fan, and Q. Yao, "Functional-coefficient regression models for nonlinear time series," *J. Amer. Stat. Assoc.*, vol. 95, no. 451, pp. 941–956, 2000.

[40] R. R. Sarukkai, "Link prediction and path analysis using Markov chains," *Comput. Netw.*, vol. 33, nos. 1–6, pp. 377–386, 2000.

[41] M. R. Hassan and B. Nath, "Stock market forecasting using hidden Markov model: A new approach," in *Proc. IEEE Int. Conf. Intell. Syst. Design Appl.*, 2005, pp. 192–196.

[42] G. A. Barreto, "Time series prediction with the self-organizing map: A review," in *Perspectives of Neural-Symbolic Integration*. Heidelberg, Germany: Springer, 2007, pp. 135–158.

[43] G. Grudnitski and L. Osburn, "Forecasting S&P and gold futures prices: An application of neural networks," *J. Futures Markets*, vol. 13, no. 6, pp. 631–643, 1993.

[44] N. Amjady, "Day-ahead price forecasting of electricity markets by a new fuzzy neural network," *IEEE Trans. Power Syst.*, vol. 21, no. 2, pp. 887–896, May 2006.

[45] M. R. Hassan, "A combination of hidden Markov model and fuzzy model for stock market forecasting," *Neurocomputing*, vol. 72, nos. 16–18, pp. 3439–3446, 2009.

[46] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Stat. Comput.*, vol. 14, no. 3, pp. 199–222, 2004.

[47] R. Khemchandani, Jayadeva, and S. Chandra, "Regularized least squares fuzzy support vector regression for financial time series forecasting," *Expert Syst. Appl.*, vol. 36, no. 1, pp. 132–138, 2009.

[48] P.-F. Pai and W.-C. Hong, "Support vector machines with simulated annealing algorithms in electricity load forecasting," *Energy Convers. Manage.*, vol. 46, no. 17, pp. 2669–2688, 2005.

[49] E. Vernet *et al.*, "Posterior consistency for nonparametric hidden Markov models with finite state space," *Electron. J. Stat.*, vol. 9, no. 1, pp. 717–752, 2015.

[50] É. Gassiat, A. Cleynen, and S. Robin, "Inference in finite state space non parametric hidden Markov models and applications," *Stat. Comput.*, vol. 26, nos. 1–2, pp. 61–71, 2016.

[51] C. E. Rasmussen, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.

[52] S. Volant, C. Bérard, M.-L. Martin-Magniette, and S. Robin, "Hidden Markov models with mixtures as emission distributions," *Stat. Comput.*, vol. 24, no. 4, pp. 493–504, 2014.

[53] S. J. Gershman and D. M. Blei, "A tutorial on Bayesian nonparametric models," *J. Math. Psychol.*, vol. 56, no. 1, pp. 1–12, 2012.

[54] D. Duvenaud, J. R. Lloyd, R. Grosse, J. B. Tenenbaum, and Z. Ghahramani, "Structure discovery in nonparametric regression through compositional kernel search," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1166–1174.

[55] F. Tobar, T. D. Bui, and R. E. Turner, "Learning stationary time series using Gaussian processes with nonparametric kernels," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 3483–3491.

[56] Z. Xu, S. MacEachern, and X. Xu, "Modeling non-Gaussian time series with nonparametric Bayesian model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 2, pp. 372–382, Feb. 2015.

[57] A.-H. Phan, P. Tichavský, and A. Cichocki, "CANDECOMP/PARAFAC decomposition of high-order tensors through tensor reshaping," *IEEE Trans. Signal Process.*, vol. 61, no. 19, pp. 4847–4860, Oct. 2013.

[58] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.

[59] J. Sun, D. Tao, and C. Faloutsos, "Beyond streams and graphs: Dynamic tensor analysis," in *Proc. ACM SIGKDD Int. Conf. Knowl. Disc. Data Mining*, 2006, pp. 374–383.

[60] G. Ma *et al.*, "Spatio-temporal tensor analysis for whole-brain fMRI classification," in *Proc. SIAM Int. Conf. Data Mining*, 2016, pp. 819–827.

[61] W. Hu, Y. Yang, W. Zhang, and Y. Xie, "Moving object detection using tensor-based low-rank and saliently fused-sparse decomposition," *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 724–737, Feb. 2017.

[62] R. Salakhutdinov and A. Mnih, "Bayesian probabilistic matrix factorization using Markov chain Monte Carlo," in *Proc. Int. Conf. Mach. Learn.*, 2008, pp. 880–887.

[63] L. Xiong, X. Chen, T. K. Huang, J. G. Schneider, and J. G. Carbonell, "Temporal collaborative filtering with Bayesian probabilistic tensor factorization," in *Proc. Syst. Design Manage.*, vol. 10, 2010, pp. 211–222.

[64] M. Rogers, L. Li, and S. J. Russell, "Multilinear dynamical systems for tensor time series," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 2634–2642.

[65] C. Goutte and M. R. Amini, "Probabilistic tensor factorization and model selection," in *Proc. Tensors Kernels Mach. Learn.*, 2010, pp. 1–4

[66] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 1257–1264.

[67] C. F. Beckmann and S. M. Smith, "Tensorial extensions of independent component analysis for multisubject FMRI analysis," *Neuroimage*, vol. 25, no. 1, pp. 294–311, 2005.

[68] K. Y. Yılmaz, A. T. Cemgil, and U. Şimşekli, "Generalised coupled tensor factorisation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 2151–2159.

[69] A. C. Ferguson, C. H. Davis, and J. E. Cavanaugh, "An autoregressive model for analysis of ice sheet elevation change time series," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 11, pp. 2426–2436, Nov. 2004.

[70] A. Veeraraghavan, A. K. Roy-Chowdhury, and R. Chellappa, "Matching shape sequences in video with applications in human movement analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 12, pp. 1896–1909, Dec. 2005.

[71] C. Jia, G. Zhong, and Y. Fu, "Low-rank tensor learning with discriminant analysis for action classification and image recovery," in *Proc. AAAI Conf. Artif. Intell.*, 2014, pp. 1228–1234.

[72] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 208–220, Jan. 2013.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

[73] Nutritional and Microbial Ecology Laboratory. (2013). *Tropical Atmosphere Ocean Project*. [Online]. Available: http://www.pmel.noaa.gov/tao/data_deliv/deliv.html
[74] U. Hoffmann, G. Garcia, J.-M. Vesin, K. Diserens, and T. Ebrahimi, "A boosting approach to P300 detection with application to brain-computer interfaces," in *Proc. IEEE Int. EMBS Conf. Neural Eng.*, 2005, pp. 97–100.
[75] B. Bhattacharya and D. Habtzghi, "Median of the p value under the alternative hypothesis," *Amer. Stat.*, vol. 56, no. 3, pp. 202–206, 2002.

Peiguang Jing received the M.S. and Ph.D. degrees in signal and information processing from Tianjin University, Tianjin, China, in 2012 and 2018, respectively.

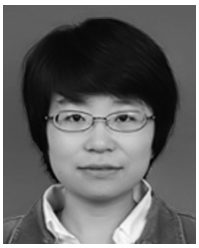
He was a Visiting Student with the National University of Singapore, Singapore, from 2014 to 2015. He is currently an Assistant Professor with the Multimedia Institute, Tianjin University. His current research interests include computer vision, multimedia content analysis, and tensor decomposition.

Yuting Su received the B.S. degree in electronic information engineering, and the M.S. and Ph.D. degrees in signal processing from Tianjin University, Tianjin, China, in 1995, 1998, and 2001, respectively.

He is currently a Professor with the School of Electrical and Information Engineering, Tianjin University. His current research interests include multimedia content analysis, information security, and tensor decomposition.

Xiao Jin received the B.S. degree in electronic information engineering from Tianjin University, Tianjin, China, in 2013, where he is currently pursuing the Ph.D. degree in signal and information processing with the School of Electrical and Information Engineering.

His current research interests include multimedia content analysis, information security, and tensor decomposition.



Chengqian Zhang received the B.S. degree in electronic information engineering, and the M.S. and Ph.D. degrees in signal processing from Tianjin University, Tianjin, China, in 2003, 2006, and 2009, respectively.

She is currently an Associate Professor with the School of Electrical Engineering and Information, Southwest Petroleum University, Chengdu, China. Her current research interests include multimedia security and image processing.