Transform-Based Multilinear Dynamical System for Tensor Time Series Analysis

Weijun Lu^{+}, Xiao-Yang Liu*, Qingwei Wu*, Yue Sun^{+}, and Anwar Walid†
$^{+}$State Key Lab. of Integrated Service Networks, Xidian University,
*Electrical Engineering, Columbia University,
†Nokia-Bell Labs
Emails: 623116319@qq.com, {XL2427, QW2208}@columbia.edu,
sunyue@xidian.edu.cn, anwar.walid@nokia-bell-labs.com

Abstract

We propose a novel multilinear dynamical system (MLDS) in a transform domain, named $\mathcal{L}$-MLDS, to model tensor time series. With transformations applied to a tensor data, the latent multidimensional correlations among the frontal slices are built, and thus resulting in the computational independence in the transform domain. This allows the exact separability of the multidimensional problem into multiple smaller LDS problems. To estimate the system parameters, we utilize the expectation-maximization (EM) algorithm to determine the parameters of each LDS. Further, $\mathcal{L}$-MLDS significantly reduces the model parameters and allows parallel processing. Our general $\mathcal{L}$-MLDS model is implemented based on discrete Fourier transform, discrete cosine transform and discrete wavelet transform, respectively. Due to the nonlinearity of these transformations, $\mathcal{L}$-MLDS is able to capture the nonlinear correlations within the data while the MLDS [1] assumes multi-way linear correlations. On four real datasets, the proposed $\mathcal{L}$-MLDS achieves much higher prediction accuracy than the state-of-the-art MLDS and LDS with an equal number of parameters under different noise models. In particular, the relative errors are reduced by $50\% \sim 99\%$. Simultaneously, $\mathcal{L}$-MLDS achieves an exponential improvement in the model's training time than MLDS.

1 Introduction

Predicting the evolving trends of data sequences is an essential problem arising in various fields such as signal processing, environmental protection and economics. A traditional model to describe a dynamically evolving data sequence is the linear dynamical system (LDS), where the observations and latent states are expressed as vectors. In the era of big data, data in various applications is frequently represented as a time series of multidimensional arrays, called tensors, to preserve the inherent multidimensional correlations. *Of interest is the prediction of future terms of the time tensor series.* The obvious solution is to unfold each tensor to a vector, and then the LDS applies as in [2, 3]. However, LDS can not preserve the data structure, and it does not allow to determine the dimension of each mode of the latent tensor.

A multilinear dynamical system (MLDS) for modeling time tensor series is proposed in [1] to generalize the LDS by vectorizing the input tensors. Expressing the latent states and observations as tensors and replacing the transition and projection matrices with multilinear operators, MLDS preserves the tensorial structure of the data and achieves a higher prediction accuracy than LDS. The multilinear operators are assumed to be factorizable so that the number of model parameters is significantly reduced (compared to LDS). However, MLDS still takes a high computational cost to estimate the large number of covariance parameters. Moreover, the method for estimating the mul-

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

tilinear operators of MLDS in [1] may fall into local optimum, thus compromising the prediction accuracy.

We propose a novel multilinear dynamical system based on transform-based tensor model, named $\mathcal{L}$-MLDS. Working in the transform domain, the multilinear operators and covariances of the $\mathcal{L}$-MLDS model are sparse block diagonal matrices. This allows exact separability of an $\mathcal{L}$-MLDS into multiple smaller LDSs and provides the opportunity for parallel processing. To estimate the model parameters, we utilize the standard EM algorithm to determine the parameters of each LDS in the transform domain. Therefore, the model involves fewer parameters, simple estimation procedures, and efficient computation, leading to improvements in the model training. In addition, $\mathcal{L}$-MLDS allows arbitrary noise relationships among the tensorial elements without the restrictive assumption of isotropic noise used in [4, 5]. For the details of this work, please see [6].

2 Transform-Based Tensor Model

Let $\mathbb{C}$ denote complex numbers. Vectors are denoted by boldface lowercase letters, e.g., $\boldsymbol{a}$; matrices are denoted by boldface capital letters, e.g., $\boldsymbol{A}$; and tensors are denoted by calligraphic letters, e.g., $\mathcal{A}$. We use $\mathcal{A}^{(k)}$ to denote the k-th frontal slice of $\mathcal{A}$ and $[n]$ to denote the index set $\{1, 2, \cdots, n\}$. In this paper, we just consider the third-order tensor for ease of exposition.

Basic operators [7]: The operator MatView$(\cdot)$ takes a tensor $\mathcal{A} \in \mathbb{C}^{I\times J\times K}$ and returns an $IK \times JK$ block diagonal matrix, with each block being an $I \times J$ matrix, defined as

$$\text{MatView}(\mathcal{A}) = \text{diag}(\mathcal{A}^{(1)}, \cdots, \mathcal{A}^{(k)}, \cdots, \mathcal{A}^{(K)}). \tag{1}$$

The operator Vec$(\cdot)$ takes a tensor $\mathcal{B} \in \mathbb{C}^{I\times 1\times K}$ and returns a vector of length IK, defined as

$$\text{Vec}(\mathcal{B}) = [\mathcal{B}^{(1)}; \cdots; \mathcal{B}^{(k)}; \cdots; \mathcal{B}^{(K)}]. \tag{2}$$

Conversely, the operator TenView$(\cdot)$ folds MatView$(\mathcal{A})$ and Vec$(\mathcal{B})$ back to tensors $\mathcal{A}$ and $\mathcal{B}$, respectively, i.e., $\text{TenView}(\text{MatView}(\mathcal{A})) = \mathcal{A}$ and $\text{TenView}(\text{Vec}(\mathcal{B})) = \mathcal{B}$.

Given an invertible discrete transform $\mathcal{L} : \mathbb{C}^K \to \mathbb{C}^K$, the elementwise multiplication is denoted by $\circ$, and with $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{C}^K$, the *tube multiplication* $\bullet$ is defined [7] as $\boldsymbol{\alpha} \bullet \boldsymbol{\beta} = \mathcal{L}^{-1}(\mathcal{L}(\boldsymbol{\alpha}) \circ \mathcal{L}(\boldsymbol{\beta}))$, and $\mathcal{L}^{-1}$ is the inverse of $\mathcal{L}$.

We use $\widetilde{\mathcal{A}} = \mathcal{L}(\mathcal{A}) \in \mathbb{C}^{I\times J\times K}$ to denote the tensor obtained by taking the transform $\mathcal{L}$ of all the tubes along the third dimension of $\mathcal{A} \in \mathbb{C}^{I\times J\times K}$. The transformation $\mathcal{L}$ builds the correlations among the frontal slices in the transform domain just like threading the wires through them. Therefore, while the frontal slices of a tensor in time domain are dependent, they are in fact *independent* in the transform domain.

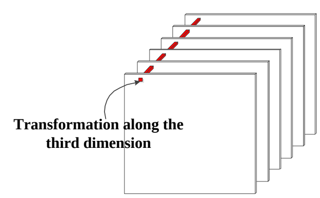


Figure 1: The transform-domain presentation of a tensor.

Definition 1. *The $\mathcal{L}$-product $\mathcal{C} = \mathcal{A} \bullet \mathcal{B}$ of $\mathcal{A} \in \mathbb{C}^{I\times K\times L}$ and $\mathcal{B} \in \mathbb{C}^{K\times J\times L}$ is a tensor in $\mathbb{C}^{I\times J\times L}$, with $\mathcal{C}(i, j, :) = \sum_{k=1}^{K} \mathcal{A}(i, k, :) \bullet \mathcal{B}(k, j, :)$, for $i \in [I]$ and $j \in [J]$.*

Lemma 1. *[7] The $\mathcal{L}$-product $\mathcal{C} = \mathcal{A} \bullet \mathcal{B}$ can be converted to the matrix multiplication in the transform domain, similar to the convolution theorem, MatView$(\widetilde{\mathcal{C}})$ = MatView$(\widetilde{\mathcal{A}}) \cdot$ MatView$(\widetilde{\mathcal{B}})$.*

In particular, given $\mathcal{A} \in \mathbb{C}^{I\times J\times K}$ and $\mathcal{B} \in \mathbb{C}^{J\times 1\times K}$, the $\mathcal{L}$-product $\mathcal{C} = \mathcal{A} \bullet \mathcal{B}$ can be calculated as $\text{Vec}(\widetilde{\mathcal{C}}) = \text{MatView}(\widetilde{\mathcal{A}}) \cdot \text{Vec}(\widetilde{\mathcal{B}})$. In this case, we call $\mathcal{A}$ a multilinear operator of $\mathcal{B}$ [7–9].

Different from the tensor normal distribution in [10] which is restricted to symmetric second-order tensors, we define an $\mathcal{L}$-normal distribution in a transform domain for arbitrary second-order tensors. The corresponding random tensors, called $\mathcal{L}$-random tensors, are used to construct $\mathcal{L}$-MLDS.

Definition 2. ***($\mathcal{L}$-Normal Distribution)*** *Given a tensor $\mathcal{X} \in \mathbb{C}^{J\times 1\times K}$, let $\widetilde{\mathcal{X}} = \mathcal{L}(\mathcal{X})$, and then we say $\mathcal{X}$ has the $\mathcal{L}$-normal distribution with expectation $\mathcal{U} \in \mathbb{C}^{J\times 1\times K}$ and covariance $\mathcal{Q} \in \mathbb{C}^{J\times J\times K}$, denoted by*

$$\mathcal{X} \sim \mathcal{CN}_{\mathcal{L}}(\mathcal{U}, \mathcal{Q}), \tag{3}$$

if and only if

$$Vec(\widetilde{\mathcal{X}}) \sim \mathcal{CN}(Vec(\widetilde{\mathcal{U}}), MatView(\widetilde{\mathcal{Q}})), \tag{4}$$

where $\mathcal{CN}$ is the traditional complex normal distribution.

3 Transform-Based Multilinear Dynamical System

The $\mathcal{L}$-MLDS model consists of a sequence $\mathcal{X}_1, \cdots, \mathcal{X}_N$ of latent tensors, where $\mathcal{X}_n \in \mathbb{C}^{J\times 1\times K}$ for all n. Each latent tensor $\mathcal{X}_n$ associates with an observation $\mathcal{Y}_n \in \mathbb{C}^{I\times 1\times K}$. The $\mathcal{L}$-MLDS is initialized by a latent tensor $\mathcal{X}_1$ distributed as

$$\mathcal{X}_1 \sim \mathcal{CN}_{\mathcal{L}}(\mathcal{U}_0, \mathcal{Q}_0). \tag{5}$$

Given $\mathcal{X}_n$, $1 \leq n \leq N-1$, we generate $\mathcal{X}_{n+1}$ according to the conditional distribution

$$\mathcal{X}_{n+1} \mid \mathcal{X}_n \sim \mathcal{CN}_{\mathcal{L}}(\mathcal{A} \bullet \mathcal{X}_n, \mathcal{Q}), \tag{6}$$

where $\mathcal{Q}$ is the conditional covariance tensor shared by all $\mathcal{X}_n, 2 \leq n \leq N$, and $\mathcal{A} \in \mathbb{C}^{J\times J\times K}$ is the transition tensor which describes the dynamics of the evolving sequence $\mathcal{X}_1, \cdots, \mathcal{X}_N$. For each $\mathcal{X}_n$, the corresponding observation $\mathcal{Y}_n$ is generated by the conditional distribution

$$\mathcal{Y}_n \mid \mathcal{X}_n \sim \mathcal{CN}_{\mathcal{L}}(\mathcal{C} \bullet \mathcal{X}_n, \mathcal{R}), \tag{7}$$

where $\mathcal{R}$ is the conditional covariance tensor shared by all $\mathcal{Y}_n$, and $\mathcal{C} \in \mathbb{C}^{I\times J\times K}$ is the projection tensor which transforms latent $\mathcal{X}_n$ to the corresponding observation $\mathcal{Y}_n$.

Suppose $I = J = K = n$. Then, the parameter complexities of LDS and MLDS are $O(n^4)$ and that of $\mathcal{L}$-MLDS is $O(n^3)$. Thus $\mathcal{L}$-MLDS significantly reduces the number of parameters as the dimensions of the tensors increase. Conversely, with equal number of parameters, $\mathcal{L}$-MLDS tends to have a greater dimensionality ($J \times K$) of the latent state. Generally, the longer the vectorized latent tensor is, the more information of the corresponding observation it has.

The problem of $\mathcal{L}$-MLDS identification is to estimate the parameters $\Theta = \{\mathcal{U}_0, \mathcal{Q}_0, \mathcal{A}, \mathcal{Q}, \mathcal{C}, \mathcal{R}\}$ from the given time series of observations $\mathcal{Y}_1, \cdots, \mathcal{Y}_N$. For the existence of unknown latent states $\mathcal{X}_n$ in the $\mathcal{L}$-MLDS, we cannot directly maximize the likelihood of the data with respect to Θ. According to Definition 2, the $\mathcal{L}$-MLDS specified by (5), (6), and (7) can be exactly divided into K indenpendent LDSs in the transform domain with each LDS being defined as

$$\begin{cases} \widetilde{\mathcal{X}}_1^{(k)} \sim \mathcal{CN}(\widetilde{\mathcal{U}}_0^{(k)}, \widetilde{\mathcal{Q}}_0^{(k)}), \\ \widetilde{\mathcal{X}}_{n+1}^{(k)} | \widetilde{\mathcal{X}}_n^{(k)} \sim \mathcal{CN}(\widetilde{\mathcal{A}}^{(k)} \cdot \widetilde{\mathcal{X}}_n^{(k)}, \widetilde{\mathcal{Q}}^{(k)}), \\ \widetilde{\mathcal{Y}}_n^{(k)} | \widetilde{\mathcal{X}}_n^{(k)} \sim \mathcal{CN}(\widetilde{\mathcal{C}}^{(k)} \cdot \widetilde{\mathcal{X}}_n^{(k)}, \widetilde{\mathcal{R}}^{(k)}). \end{cases} \tag{8}$$

Hence, the problem of estimating Θ is exactly separated into K independent subproblems of estimating $\theta^{(k)} = \{\widetilde{\mathcal{U}}_0^{(k)}, \widetilde{\mathcal{Q}}_0^{(k)}, \widetilde{\mathcal{A}}^{(k)}, \widetilde{\mathcal{Q}}^{(k)}, \widetilde{\mathcal{C}}^{(k)}, \widetilde{\mathcal{R}}^{(k)}\}$ with incomplete data [11]. Then, we use the EM algorithm to estimate each $\theta^{(k)}$, $k \in [K]$, and finally convert all those subsystem components to time domain. For the specific process, see Figure 2.

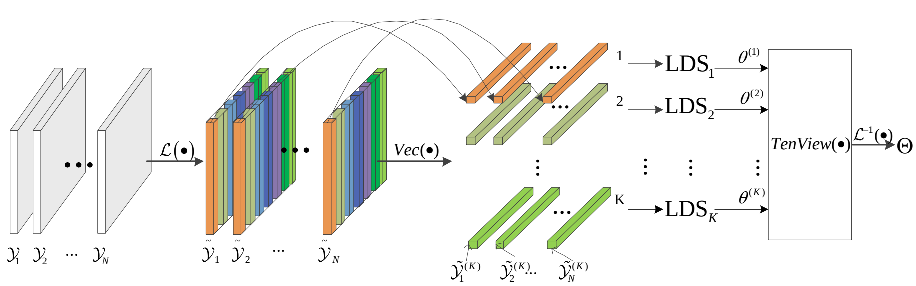


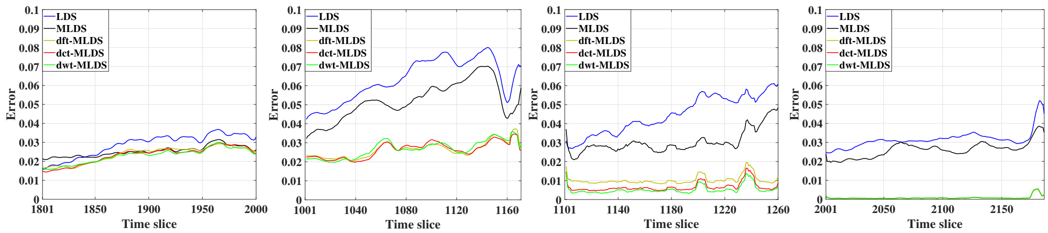
Figure 2: The process of $\mathcal{L}$-MLDS training.

4 Performance Evaluation on Real Data

We conduct experiments with the noise covariances in the models being diagonal and non-diagonal, respectively. In all the experiments, the LDS latent dimensionality is always set to the smallest value such that the number of parameters of the LDS is greater than or equal to that of the MLDS. Also, the latent dimensionality J of each LDS in the transform domain of $\mathcal{L}$-MLDS is set to the largest value such that the number of parameters of the $\mathcal{L}$-MLDS is less than or equal to that of MLDS.

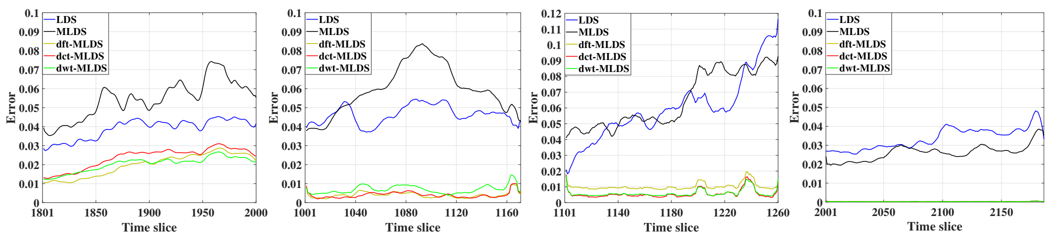
We use the following datasets in evaluations, and the codes are available online [12].

SST [1]: A 5-by-6 grid of sea-surface temperatures. Each model was trained on the first 1800 epochs and tested on the last 200 epochs.
Video [1]: A 10×10 patch for each frame. Each model was trained on the first 1000 frames and tested on the last 171 frames.
Tesla [13]: A 14×5 patch for each epoch. Each model was trained on the first 1100 epochs and tested on the last 160 epochs.
NASDAQ-100 [14]: Opening, closing, high, and low for 50 randomly-chosen NASDAQ-100 companies. Each model was trained on the first 2000 epochs and tested on the last 186 epochs.



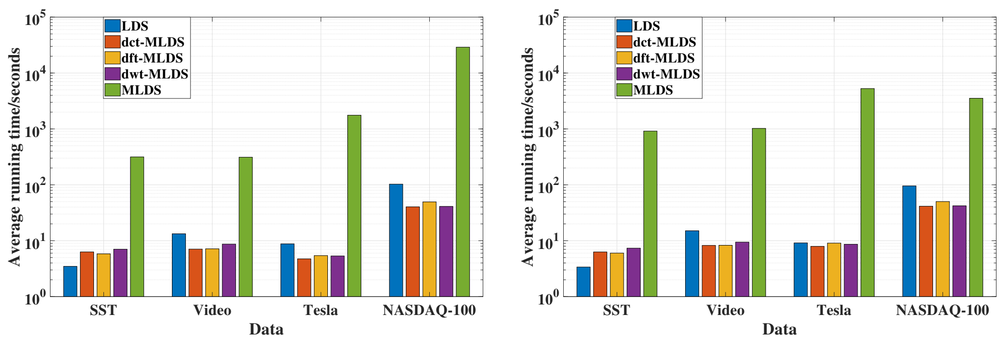
(a) SST (b) Video (c) Tesla (d) NASDAQ-100

Figure 3: Performance results for LDS, MLDS, dct-MLDS, dft-MLDS and dwt-MLDS using real data with the covariances of the noises being diagonal.



(a) SST (b) Video (c) Tesla (d) NASDAQ-100

Figure 4: Performance results for LDS, MLDS, dct-MLDS, dft-MLDS and dwt-MLDS using real data with the covariances of noises being non-diagonal.



(a) (b)

Figure 5: The running time for LDS, MLDS, dct-MLDS, dft-MLDS and dwt-MLDS (the runtime of transformation is considered in each $\mathcal{L}$-MLDS). (a) corresponds to Figure 3 and (b) to Figure 4.

The comparisons (shown in Figure 3 and 4) demonstrate that our $\mathcal{L}$-MLDS is able to achieve a high prediction accuracy for arbitrary noise relationships among the tensorial elements, and reduces the relative errors by $50\% \sim 99\%$. In addition to the higher prediction accuracy, $\mathcal{L}$-MLDS reduces the training time by orders of magnitude compared to MLDS, see Figure 5. Simultaneously, the longer the vectorized inputs are, the more obvious the improvement will be.

5 Acknowledgement

This work was supported by the National Natural Science Foundation of China under Grant 61671345.

References

[1] Mark Rogers, Lei Li, and Stuart J Russell, "Multilinear dynamical systems for tensor time series," in *Advances in Neural Information Processing Systems (NIPS)*, 2013, pp. 2634–2642.

[2] Christopher M Bishop, "Pattern recognition and machine learning," Springer, 1st edition, 2006.

[3] Zoubin Ghahramani and Geoffrey E Hinton, "Parameter estimation for linear dynamical systems," Tech. Rep., CRG-TR-96-2, University of Totronto, Dept. of Computer Science, 1996.

[4] Amit Surana, Geoff Patterson, and Indika Rajapakse, "Dynamic tensor time series modeling and analysis," in *IEEE 55th Conference on Decision and Control (CDC)*, 2016, pp. 1637–1642.

[5] Jimeng Sun, Dacheng Tao, and Christos Faloutsos, "Beyond streams and graphs: dynamic tensor analysis," in *Proceedings of International Conference on Knowledge Discovery and Data Mining*, 2006, pp. 374–383.

[6] Weijun Lu, Xiao-Yang Liu, Qingwei Wu, Yue Sun, and Anwar Walid, "Transform-based multilinear dynamical system for tensor time series analysis," *arXiv preprint arXiv:1811.07342*, 2018.

[7] Xiao-Yang Liu and Xiaodong Wang, "Fourth-order tensors with multidimensional discrete transforms," *arXiv preprint arXiv:1705.01576*, pp. 1–37, 2017.

[8] Misha E Kilmer, Karen Braman, Ning Hao, and Randy C Hoover, "Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging," *Journal on Matrix Analysis and Applications*, pp. 148–172, 2013.

[9] Xiao-Yang Liu, Shuchin Aeron, Vaneet Aggarwal, and Xiaodong Wang, "Low-tubal-rank tensor completion using alternating minimization," in *(Major revision) IEEE Transactions on Information Theory*, 2018.

[10] Peter J Basser and Sinisa Pajevic, "A normal distribution for tensor-valued random variables to analyze diffusion tensor mri data," in *Proceedings of International Symposium on Biomedical Imaging*. IEEE, 2002, pp. 927–930.

[11] Arthur P Dempster, Nan M Laird, and Donald B Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society. Series B (methodological)*, pp. 1–38, 1977.

[12] "Our codes," *http://www.tensorlet.com/*.

[13] "Tesla stock prices data," *https://xueqiu.com/*.

[14] Yao Qin, Dongjin Song, Haifeng Chen, Wei Cheng, Guofei Jiang, and Garrison Cottrell, "A dual-stage attention-based recurrent neural network for time series prediction," *arXiv preprint arXiv:1704.02971*, 2017.