Multivariate Multi-Order Markov Multi-Modal Prediction With Its Applications in Network Traffic Management

Huazhong Liu, Laurence T. Yang, *Senior Member, IEEE*, Jinjun Chen, *Senior Member, IEEE*, Minghao Ye, Jihong Ding, and Liwei Kuang

***Abstract*—Predicting the future network traffic through big data analysis technologies has been one of the important preoccupations of network design and management. Combining Markov chains with tensors to implement predictions has received considerable attention in the era of big data. However, when dealing with multi-order Markov models, the existing approaches including the combination of states and Z-eigen decomposition still face some shortcomings. Therefore, this paper focuses on proposing a novel multivariate multi-order Markov transition to realize multi-modal accurate predictions. First, we put forward two new tensor operations including tensor join and unified product (UP). Then a general multivariate multi-order (2M) Markov model with its UP-based state transition is proposed. Afterwards, we develop a multi-step transition tensor for 2M Markov models to implement the multi-step state transition. Furthermore, an UP-based power method is proposed to calculate the stationary joint probability distribution tensor (i.e., stationary joint eigentensor, SJE) and realize SJE based multi-modal accurate predictions. Finally, a series of experiments under various Markov models on real-world network traffic datasets are conducted. Experimental results demonstrate that the proposed SJE based approach can improve the prediction accuracy for network traffic by highest up to 38.47 percentage points compared with the Z-eigen based approach.**

***Index Terms*—Multivariate multi-order Markov, multi-step transition tensor, unified product, stationary joint eigentensor, multi-modal accurate prediction, network traffic prediction, network management.**

I. Introduction

Nowadays, with the rapid development of networking and communications, everything interconnects with the networks [1], [2]. Motivated by the continuous improvement of people's requirements for effective communications, some neoteric network architectures are proposed, such as Software Defined Networking (SDN), Network Function Virtualization (NFV), etc. [3]. By breaking vertical integration, SDN is a burgeoning paradigm which separates the network's control planes from the data planes [4]. NFV decouples the software implementation of network functions from the underlying hardware by taking advantages of virtualization technologies and commercial off-the-shelf programmable hardware [5]. Based on these emerging architectures, clusters of network functions can be improved, such as rapid network analysis, comprehensive network design, and efficient network management [6], [7]. Owing to the separation between the control layer and data layer, extensive network data are collected in up-to-date network architectures and served for analyzing and managing the network [8], [9]. By exploiting big data analysis technologies including artificial intelligence and machine learning [10], [11], [12], we can increase flexibility in traffic forwarding, simplify network management, and facilitate network evolution [13].

Predicting the future network traffic has been one of the important preoccupations of network design and management. Accurate traffic prediction can promote people to manage networks and make wise decisions [14]. There are several approaches in traffic prediction, such as multiresolution FIR neural-network-based method [15], naive Bayes [16], deep neural network [17], etc. Besides, another effective prediction approach is to use Markov chains. First-order Markov model and hidden Markov model, due to their well-developed theory, have been extensively utilized in various domains, such as network traffic prediction [18], network traffic modeling [19], as well as trajectory prediction [20],

Manuscript received December 3, 2018; revised April 25, 2019 and July 10, 2019; accepted July 27, 2019. This work was supported in part by the National Natural Science Foundation of China (Nos. 61867002, 71704160, and 71764036). The associate editor coordinating the review of this article and approving it for publication was T. Inoue *(Corresponding author: Laurence T. Yang.)*

H. Liu is with the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China, and also with the School of Information Science and Technology, Jiujiang University, Jiujiang 332005, China (e-mail: sharpshark_ding@163.com).

L. T. Yang is with the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China, and also with the Department of Computer Science, St. Francis Xavier University, Antigonish, NS B2G 2W5, Canada (e-mail: ltyang@gmail.com).

J. Chen is with the Department of Computer Science and Software Engineering, Swinburne University of Technology, Hawthorn, VIC 3122, Australia (e-mail: jchen@swin.edu.au).

M. Ye is with the Department of Electrical and Computer Engineering, New York University, New York, NY 11201 USA (e-mail: my1706@nyu.edu).

J. Ding is with the School of Education Science and Technology, Zhejiang University of Technology, Hangzhou 310014, China (e-mail: jhding@zjut.edu.cn).

L. Kuang is with the Network Business Unit, FiberHome Telecommunication Technologies Company Ltd., Wuhan 430073, China (e-mail: lwkuang@fiberhome.com).

Digital Object Identifier 10.1109/TNSM.2019.2934133

driver intention prediction [21], gene and protein sequences prediction [22], etc.

Recently, studying Markov chains by combining tensors has become an emerging trend in academia. To reflect the diversity of states and improve the prediction accuracy, multivariate Markov chains have been proposed by exploiting tensor-based analysis approaches [23], [24]. In most actual systems, the state may have multiple attributes. For instance, in the location-aware prediction system, each state in traditional first-order Markov chain denotes the point of interest (POI). However, the located POI in real life is influenced by many factors, such as weather, period of time (e.g., morning, afternoon, night), holiday, etc., where the states are considered as multivariate. In [24], Wang *et al.* construct a complex human-spatio-temporal multivariate Markov transition model based on tensor theory and develop an iterative tensor power method to calculate the stationary probability distribution. In multivariate Markov chains, the traditional stationary probability distribution vector (e.g., dominant eigenvector) is extended to stationary probability distribution tensor (e.g., dominant eigentensor). In a Markov chain, the stationary probability distribution represents the ultimate occurring probability of each state at any time epoch in the future, it can be applied to implement future trend prediction when the transition probability tensor keeps roughly stable in the short term of the future. Based on the stationary probability distribution tensor (e.g., dominant eigentensor), the user's mobility trajectory pattern is predicted in [24] and the experimental results demonstrate that the dominant eigentensor based multivariate Markov prediction approach exhibits higher prediction accuracy.

Meanwhile, multi-order (or higher-order) Markov chains have paid more attention in different application areas, a wealth of examples can be found in [25], [26], [27]. In the early stage, the multi-order Markov chains have always been processed by approximating them to the first-order Markov chains through a linear combination of states at multiple time epochs [25], [28], [29]. However, this kind of methods are difficult to deal with some complex multivariate Markov models, i.e., the human-spatio-temporal Markov transition model in [24]. Besides, Gleich *et al.* construct a transition probability tensor for multi-order Markov chains in [29], but in which tensor is just a representation and there are no tensor-based operations and calculations.

Afterwards, immense amounts of research has been carried out by integrating tensor Z-eigenvector and multi-order Markov theories [29], [30], [31]. Tensor Z-eigenvector theory is proposed by Qi [32]. Given a transition probability tensor, the Z-eigen decomposition for the largest Z-eigenvalue (i.e., 1) can be expressed as follows:

$$\underline{P}\mathrm{x}^m = \mathrm{x} \quad \Leftrightarrow \quad \underline{P} \times_2 \mathrm{x} \times_3 \mathrm{x} \cdots \times_n \mathrm{x} \cdots \times_{m+1} \mathrm{x} = \mathrm{x}, \quad (1)$$

where $\times_n$ is the single-mode product, $\underline{P}$ is an $(m+1)$th-order transition probability tensor for an *m*-order Markov chain, x is called dominant Z-eigenvector. In [30], Li and Ng propose an iterative higher-order power method to calculate the stationary probability distribution vector (i.e., dominant Z-eigenvector) for a multi-order Markov chain. Then Gleich *et al.* [29] and Bozorgmanesh and Hajarian [31] further improve the convergence conditions and calculation methods to calculate the dominant eigenvector. In these researches, some exploratory conclusions and complete proofs are provided from the mathematical theory point of view, but there are no applications to implement the future prediction. Furthermore, Kuang *et al.* propose a tensor-based framework for software defined big data center, and then apply the single-mode (or multi-mode) Z-eigen decomposition for the traffic transition probability tensor to implement the future traffic prediction [23].

However, it is notable that there are two key problems in combining dominant Z-eigenvector (or dominant Z-eigentensor for multivariate models) and multi-order Markov chain to realize the future prediction. On the one hand, the constructed Markov models are multi-order, i.e., the state of current time epoch is determined by multiple states at several previous time epochs, but the stationary probability distribution (i.e., dominant Z-eigenvector or Z-eigentensor) is first-order. It is not reasonable to realize future prediction for a multi-order Markov chain by simply using a first-order dominant Z-eigenvector (or Z-eigentensor), resulting in the decrease of prediction accuracy. The experimental results in Section VI will confirm this statement. On the other hand, while computing the dominant Z-eigenvector of an $(m+1)$th-order transition probability tensor for an *m*-order Markov chain in Eq. (1), there exists a strict independence assumption that the multiple states' joint probability at any *m* consecutive time epochs in multi-order Markov model is the product of each state's probability (Please see Section III). However, the independence assumption might not be satisfied in many scenarios.

According to the existing literatures, multivariate Markov models based on tensor theory are studied to describe more complex transition relationship among multiple spaces [24], but they merely deal with the first-order Markov cases. Meanwhile, combining tensor based Z-eigen decomposition and multi-order Markov models has been an alternative approach to handle multi-order Markov models [23], [29], [30], [31], but in which the multivariate cases haven't been considered, and there exist some problems resulted in the decrease of prediction accuracy. Therefore, there is no a general tensor-based multivariate multi-order Markov transition model with the multi-modal prediction approach.

To tackle the aforementioned problems, this paper focuses on proposing a general multivariate multi-order (2M) Markov model and a new transition approach without any assumptions for realizing accurate multi-modal prediction. Concretely, we first propose two new tensor operations including tensor join and unified product (UP). Then we present a general 2M Markov model and a new UP-based transition approach. Afterwards, a multi-step transition approach for 2M Markov models and the multi-step transition tensor are developed. Furthermore, to calculate the stationary joint probability distribution tensor (denoted as stationary joint eigentensor, SJE) for 2M Markov models, we propose an UP-based iterative algorithm with its detailed algorithm analysis. Based on the calculated SJE, we can implement multi-modal predictions. Finally, we conduct a series of experiments on real-world network traffic datasets to verify the performance of the proposed approach under various 2M Markov models. Experimental results demonstrate that the proposed SJE based approach can improve the prediction accuracy by highest up to 38.47 percentage points compared with the Z-eigen based approach.

To summarize, the major contributions of this paper are listed as follows.

- Put forward two new tensor operations including tensor join and unified product.
- Present a general multivariate multi-order Markov model with its UP-based state transition.
- Develop a multi-step transition tensor for 2M Markov models to implement the multi-step state transition.
- Propose an UP-based power method to calculate the stationary joint eigentensor for 2M Markov models and further implement multi-modal accurate predictions.

The rest of the paper is organized as follows. Section II briefly recalls the relative preliminaries of tensor operations and Markov models. Section III describes the problem statement. In Section IV, 2M Markov models are proposed in detail, as well as the multi-step transition tensor. In Section V, the calculation of SJE is discussed in detail. Section VI compares the experimental results, and Section VII concludes the paper.

II. Preliminaries

A. Tensor Operations

In an Nth-order tensor $\underline{X} \in R^{I_1 \times I_2 \times \cdots \times I_N}$, N is the order of the tensor and I_n $(1 \leq n \leq N)$ is the dimensionality of the nth order. In tensor-based data analysis, some tensor operations play significant roles, such as mode-n product, single-mode product, multiple-mode product, Einstein product, etc. For more concrete definition about tensor operations, please refer to [33], [34]. Therein, Einstein product is involved in this paper and defined as follows.

Definition 1 (Einstein Product [35]): Given two tensors $\underline{A} \in R^{I_1 \times I_2 \times \cdots \times I_M \times K_1 \times K_2 \times \cdots \times K_P}$ and $\underline{B} \in R^{K_1 \times K_2 \times \cdots \times K_P \times J_1 \times J_2 \times \cdots \times J_N}$ with the same dimensionality on P common orders $K_1, K_2, \ldots, K_P$, the Einstein product of two tensors $\underline{A}$ and $\underline{B}$ yields a new tensor $\underline{C} \in R^{I_1 \times I_2 \times \cdots \times I_M \times J_1 \times J_2 \times \cdots \times J_N}$ with entry $c_{i_1,i_2,\ldots,i_M,j_1,j_2,\ldots,j_N} = \sum_{k_1,k_2,\ldots,k_P} a_{i_1,i_2,\ldots,i_M,k_1,k_2,\ldots,k_P} b_{k_1,k_2,\ldots,k_P,j_1,j_2,\ldots,j_N}$, which can be represented as $\underline{C} = \underline{A} *_P \underline{B}$.

Especially, if the common orders are not consecutive, we can represent Einstein product as $\underline{C} = \underline{A} *_{m\cdots p}^{n\cdots q} \underline{B}$ $(I_m = J_n, \ldots, I_p = J_q)$.

B. Multivariate Markov Chain

Suppose $\{X_t, t = 0, 1, 2, \ldots\}$ is a stochastic process and S denotes the finite unary state set

$$S \equiv \{1, 2, \ldots, I\}.$$

In a first-order Markov chain, the state at the current time epoch is only determined by the state at the previous time epoch.

$$\begin{aligned} &Pr(X_t = i | X_{t-1} = j, X_{t-2} = i_{t-2}, \ldots, X_0 = i_0) \\ &\quad = Pr(X_t = i | X_{t-1} = j) = p_{ij}, \end{aligned} \tag{2}$$

where $i, j, i_{t-2}, \ldots, i_0 \in S$. Based on Eq. (2), we construct a transition probability matrix P for the first-order Markov chain.

$$\begin{aligned} &P = (p_{ij}),\ P \in R^{I \times I}, i, j \in S, \\ &p_{ij} \geq 0\ \ and\ \sum_{i=1}^{I} p_{ij} = 1,\ j = 1, 2, \ldots, I. \end{aligned} \tag{3}$$

The probability transition principle in a first-order Markov chain can be represented as follows:

$$\begin{aligned} Pr(X_t = x_t) &= \sum_{x_{t-1}} Pr(X_t = x_t, X_{t-1} = x_{t-1}) \\ &= \sum_{x_{t-1}} Pr(X_t = x_t | X_{t-1} = x_{t-1}) \\ &\quad \times Pr(X_{t-1} = x_{t-1}). \end{aligned} \tag{4}$$

It can be easily found that the function in Eq. (4) can exactly be realized by matrix-vector multiplication, i.e., $\mathrm{x}_t = P\mathrm{x}_{t-1}$, where x_t and x_{t-1} denote the probability distribution vector of states. Therefore, calculating the stationary probability distribution vector for a first-order Markov chain is equivalent to calculating the dominant eigenvector of the transition probability matrix P associated with the largest eigenvalue [36], i.e., $\lambda \mathrm{v} = P\mathrm{v}$ $(\lambda = 1)$, where $\mathrm{v} \in R^I$. Then it is further converted to a fix-point problem and solved through the power method [30].

However, the state in real life may be influenced by many attributes. For instance, the state in the network traffic system can be jointly determined by {*Holiday, TimePeriod, …, Traffic*}. Therefore the traditional first-order Markov model is extended to multivariate first-order Markov model in which the state is multi-attribute. Suppose each state in a multivariate Markov model is determined by k attributes and each dimensionality is I_i $(i = 1, 2, \ldots, k)$. The finite multivariate state set can be represented as:

$$S' \equiv \{(1, 1, \ldots, 1), (1, 1, \ldots, 2), \ldots, (I_1, I_2, \ldots, I_k)\}. \tag{5}$$

Let $Pr(X_{t,1}, X_{t,2}, \ldots, X_{t,k} = i_1, i_2, \ldots, i_k \mid X_{t-1,1}, X_{t-1,2}, \ldots, X_{t-1,k} = j_1, j_2, \ldots, j_k) = p'_{i_1,i_2,\ldots,i_k,j_1,j_2,\ldots,j_k}$, where $i_1, i_2, \ldots, i_k$ and $j_1, j_2, \ldots, j_k \in S'$. Then the transition probability matrix is transformed to a transition probability tensor.

$$\begin{aligned} &\underline{P'} \in R^{(I_1 \times I_2 \times \cdots \times I_k) \times (I_1 \times I_2 \times \cdots \times I_k)}, p'_{i_1,i_2,\ldots,i_k,j_1,j_2,\ldots,j_k} > 0, \\ &\sum_{i_1,i_2,\ldots,i_k=1}^{I_1,I_2,\ldots,I_k} p'_{i_1,i_2,\ldots,i_k,j_1,j_2,\ldots,j_k} = 1,\ \forall\ j_1, j_2, \ldots, j_k \in S'. \end{aligned} \tag{6}$$

Accordingly, the dominant eigenvector problem is extended to dominant eigentensor problem for the transition probability tensor $\underline{P'}$, i.e., $\lambda \underline{T'} = \underline{P'} *_k \underline{T'}$ $(\lambda = 1)$, where $*$ denotes Einstein product and $\underline{T'} \in R^{I_1 \times I_2 \times \cdots \times I_k}$. The dominant eigentensor can be calculated by exploiting tensor power method [24]. Finally, based on the dominant eigentensor, we can realize multi-modal accurate prediction according to different attributes, e.g., the network traffic prediction under various time periods (e.g., morning or afternoon or night) and different days (e.g., working day or holiday).

C. Irreducible Tensor

In a first-order Markov model, concerning $P\overline{\mathrm{v}} = \overline{\mathrm{v}}$, if the transition probability matrix P is irreducible, $\overline{\mathrm{v}}$ will be positive and unique [30]. However, in multi-order Markov chains, the definition of irreducibility needs to be extended to irreducible tensor accordingly.

Definition 2 (Irreducible Tensor [25], [30]): Given an $(m+1)$th-order I-dimensional transition probability tensor $\underline{Q}$ for an m-order Markov chain, in which $q_{i_1,i_2,\ldots,i_{m+1}} = Pr(X_t = i_1 | X_{t-1} = i_2, X_{t-2} = i_3, \ldots, X_{t-m} = i_{m+1})$. Tensor $\underline{Q}$ is called reducible if there exists a nonempty proper index subset $J \subset \{1, 2, \ldots, I\}$ and

$$q_{i_1,i_2,\ldots,i_{m+1}} = 0, \ \ \forall i_1 \in J, \ \ \forall i_2, \ldots, i_{m+1} \notin J.$$

If $\underline{Q}$ is not reducible, then we call $\underline{Q}$ irreducible.

III. Problem Statement

For the convenience of expression, we simplify some expressions in the following sections of the paper.

Notation 1: Simplified probability notation:

$$Pr(X_t = x_t) \Leftrightarrow Pr(X_t), \ \sum_{x_t} Pr(X_t = x_t) \Leftrightarrow \sum_{t} Pr(X_t).$$

Notation 2: Simplified k-variate state notation:

$$X_{t,1}, X_{t,2}, \ldots, X_{t,k} \Leftrightarrow X_t, \ i_{t,1}, i_{t,2}, \ldots, i_{t,k} \Leftrightarrow i_t.$$

To illustrate multi-order Markov chains, we take a second-order Markov chain as an example and have

$$\begin{aligned} &Pr(X_t = i | X_{t-1} = j, X_{t-2} = k, X_{t-3} = i_{t-3}, \ldots, X_0 = i_0) \\ &= Pr(X_t = i | X_{t-1} = j, X_{t-2} = k) = p''_{ijk}. \end{aligned} \tag{7}$$

Based on Eq. (7), we construct a transition probability tensor $\underline{P}''$ for the second-order Markov chain as follows:

$$\begin{aligned} &\underline{P}'' = \left(p''_{ijk}\right), \ \underline{P}'' \in R^{I \times I \times I}, i, j, k \in S, \\ &p''_{ijk} \geq 0 \ \ and \ \ \sum_{i=1}^{I} p''_{ijk} = 1, \ \ j, k = 1, 2, \ldots, I. \end{aligned} \tag{8}$$

To calculate the stationary probability distribution vector of the second-order Markov chain, combining Z-eigenvector theory and Markov theory is extensively adopted. The dominant Z-eigenvector $\mathrm{v}' \in R^I$ of $\underline{P}''$ associated with the largest Z-eigenvalue ($\lambda = 1$) can be described as follows:

$$\mathrm{v}' = \underline{P}'' \times_2 \mathrm{v}' \times_3 \mathrm{v}'. \tag{9}$$

In fact, Eq. (9) is equivalent to the following representation:

$$\mathrm{v}' = \underline{P}'' *_2 \left(\mathrm{v}' \circ \mathrm{v}'\right), \tag{10}$$

where $\circ$ denotes outer product. The Z-eigen based state transition is depicted in Fig. 1.

From the perspective of probability theory, the nature of Eq. (10) is to perform the following operations:

$$\begin{aligned} Pr(X_t) &= \sum_{t-1,t-2} Pr(X_t X_{t-1} X_{t-2}) \\ &= \sum_{t-1,t-2} Pr(X_t | X_{t-1} X_{t-2}) Pr(X_{t-1} X_{t-2}). \end{aligned} \tag{11}$$

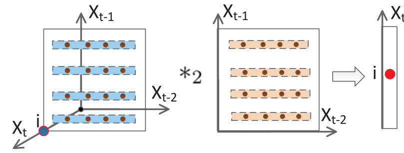


Fig. 1. Illustration of Z-eigen based state transition for a second-order Markov model.

Therefore, if we calculate the stationary probability distribution vector by achieving the dominant Z-eigenvector in Eq. (9) through some iterative approaches, there implies an independent assumption:

$$Pr(X_{t-1} X_{t-2}) = Pr(X_{t-1}) P(X_{t-2}). \tag{12}$$

The assumption means that any two consecutive states in the second-order Markov model must be independent.

Therefore, it can be easily found that there exist two problems directly by using Z-eigen based approach to deal with the multi-order Markov model: (1) The assumption may not be true in most scenarios. (2) The prediction accuracy will decrease if the dominant Z-eigenvector/Z-eigentensor are directly exploited to implement future predictions in multi-order Markov models. Because the next state in a multi-order Markov model is jointly determined by multiple previous states. The future state should be predicted according to the multi-order stationary joint probability distribution, not the first-order stationary probability distribution. Therefore, we shall resolve these concrete problems in the following sections:

(1) How to propose a general 2M Markov model and further implement the state transition without any assumption?

(2) How to obtain the stationary joint probability distribution (i.e., stationary joint eigentensor) for a 2M Markov model?

(3) How to implement the multi-modal accurate prediction based on the stationary joint eigentensor?

IV. Multivariate Multi-Order Markov Model

This section first presents two new tensor operations, and then proposes a general 2M Markov model with its state transition, as well as a multi-step transition tensor.

A. Proposed Tensor Operations

To establish a general 2M Markov model, we need to seek for an operation to satisfy the following two requirements. (1) Each transition operation must follow the probability transition principle. (2) The transition operation can be consecutively implemented without any other assumptions. Therefore, we define two new operations as follow.

Definition 3 (Tensor Join): Given two tensors $\underline{A} \in R^{I_1 \times I_2 \times \cdots \times I_M \times K_1 \times K_2 \times \cdots \times K_Q}$ and $\underline{B} \in R^{K_1 \times K_2 \times \cdots \times K_Q \times J_1 \times J_2 \times \cdots \times J_N}$ with Q common modes $K_1, K_2, \ldots, K_Q$, tensor join of tensors $\underline{A}$ and $\underline{B}$ generates a new tensor $\underline{C} \in R^{I_1 \times I_2 \times \cdots \times I_M \times J_1 \times J_2 \times \cdots \times J_N \times K_1 \times K_2 \times \cdots \times K_Q}$ with entries $c_{i_1,i_2,\ldots,i_M,j_1,j_2,\ldots,j_N,k_1,k_2,\ldots,k_Q} = a_{i_1,i_2,\ldots,i_M,k_1,k_2,\ldots,k_Q} b_{k_1,k_2,\ldots,k_Q,j_1 j_2,\ldots,j_N}$, which can be represented as $\underline{C} = \underline{A} \bowtie_Q \underline{B}$. If the common orders are

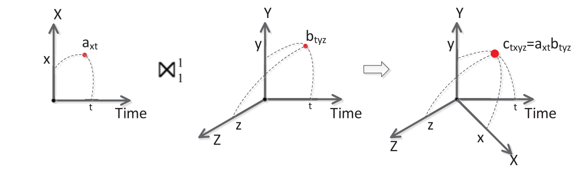


Fig. 2. An example of tensor join $\underline{C} = \underline{A} \bowtie^{Time}_{Time} \underline{B}$.

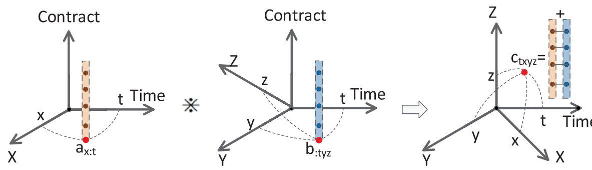


Fig. 3. An example of unified product $\underline{C} = \underline{A} \circledast^{Contract,Time}_{Contract,Time} \underline{B}$.

not consecutive, it can also be represented as $\underline{C} = \underline{A} \bowtie^{s \cdots v}_{r \cdots u} \underline{B} (I_r = J_s, \ldots, I_u = J_v)$.

Generally, tensor join can integrate two tensors according to their common orders, which can be used to implement data fusion. Fig. 2 depicts a simple example of tensor join for two tensors $\underline{A} \in R^{I_X \times I_{Time}}$ and $\underline{B} \in R^{I_{Time} \times I_Y \times I_Z}$ with the same *Time* order.

Definition 4 (Unified Product): Given two tensors $\underline{A} \in R^{I_1 \times I_2 \times \cdots \times I_M \times L_1 \times L_2 \times \cdots \times L_P \times K_1 \times K_2 \times \cdots \times K_Q}$ and $\underline{B} \in R^{L_1 \times L_2 \times \cdots \times L_P \times K_1 \times K_2 \times \cdots \times K_Q \times J_1 \times J_2 \times \cdots \times J_N}$ with two groups of common modes including P common modes for contraction $L_1, L_2, \ldots, L_P$ and Q common modes for join $K_1, K_2, \ldots, K_Q$, the unified product of tensors $\underline{A}$ and $\underline{B}$ will yield a new tensor $\underline{C} \in R^{I_1 \times \cdots \times I_M \times J_1 \times \cdots \times J_N \times K_1 \times \cdots \times K_Q}$ with entry

$$c_{i_1,\ldots,i_M,j_1,\ldots,j_N,k_1,\ldots,k_Q} = \sum_{l_1,\ldots,l_P} a_{i_1,\ldots,i_M,l_1,\ldots,l_P,k_1,\ldots,k_Q} b_{l_1,\ldots,l_P,k_1,\ldots,k_Q,j_1,\ldots,j_N}.$$

Unified product of two tensors can be represented as $\underline{C} = \underline{A} \circledast_{P,Q} \underline{B}$. And if the common orders are not consecutive, it can also be represented as $\underline{C} = \underline{A} \circledast^{n \cdots q}_{m \cdots p};^{s \cdots v}_{r \cdots u} \underline{B}$ $(I_m = J_n, \ldots, I_p = J_q; I_r = J_s, \ldots, I_u = J_v)$.

Fig. 3 gives an example of the unified product for two tensors $\underline{A} \in R^{I_X \times I_{Contract} \times I_{Time}}$ and $\underline{B} \in R^{I_{Contract} \times I_{Time} \times I_Y \times I_Z}$ with the same *Contract* order to contract and the same *Time* order to join. According to Def. 4 and Fig. 3, we can divide all orders in unified product into three parts. The first part is the contracted orders, e.g., $L_1, L_2, \ldots, L_P$, these common orders will be contracted and disappear. The second part is the join orders, e.g., $K_1, K_2, \ldots, K_Q$, these common orders will be merged to one part. The third part is the expanded orders, e.g., $I_1, I_2, \ldots, I_M$ and $J_1, J_2, \ldots, J_N$, these orders will be expanded, which is similar to outer product.

Unified product is a general and useful operation, it can cover many tensor operations and meet various scenarios when P, Q, M, N are set to different values. We summarize various cases of unified product and illustrate them in Table I. Some important cases are illustrated as follows:

TABLE I
DIFFERENT CASES OF UNIFIED PRODUCT

P	Q	M	N	Order	Tensor Operation (Notation)
$\neq 0$	$\neq 0$	$\neq 0$	$\neq 0$	M+N+Q	Unified Product ($\circledast_{p,q}$)
$= 0$	$\neq 0$	$\neq 0$	$\neq 0$	M+N+Q	Tensor Join ($\bowtie_q$)
$= 0$	$= 0$	$\neq 0$	$\neq 0$	M+N	Outer Product ($\circ$)
> 1	$= 0$	$\neq 0$	$\neq 0$	M+N	Tensor Time Tensor ($\times^{n \cdots s}_{m \cdots r}$) Einstein Product ($*_p$) Multi-mode Product ($\times^{n \cdots s}_{m \cdots r}$)
$= 1$	$= 0$	$\neq 0$	$\neq 0$	M+N	Single-mode Product ($\times^n_m$)
$= 1$	$= 0$	> 1	$= 1$	M+N =M+1	Tensor Time Matrix ($\times^{n \cdots s}_{m \cdots r}$) Mode-n Product ($\times_n$)
$= 1$	$= 0$	$= 1$	$= 1$	M+N=2	Matrix Product ($\times$)
$= 1$	$= 0$	> 1	$= 0$	M+N=M	Tensor Time Vector ($\times_n$)
$= 1$	$= 0$	$= 1$	$= 0$	M+N=1	Matrix Time Vector ($\times$)
$= 0$	$\neq 0$	$= 0$	$= 0$	Q	Hadamard Product ($\circledast$)
$\neq 0$	$= 0$	$= 0$	$= 0$	1	Inner Product ($\cdot$)
$\neq 0$	$\neq 0$	$= 0$	$= 0$	Q	Multi-order Inner Product ($\cdot_p$)

1) Unified product will convert to tensor join if $P = 0$. Further, it will be outer product if $P, Q = 0$. Thus outer product is a special case of tensor join, and tensor join can also be considered as multi-mode outer product.

2) It will convert to Einstein product (or multi-mode product) if $Q = 0$. Further, it will be single-mode product if $P = 1$. Besides, other operations can be obtainted when M and N are set to different values, such as tensor time matrix (or mode-n product) if $M > 1$ and $N = 1$, matrix product if $M = N = 1$, tensor time vector if $M > 1$ and $N = 0$, and matrix time vector if $M = 1$ and $N = 0$.

3) It will convert to Hadamard Product if P, M, $N = 0$.

4) It will convert to inner product if Q, M, $N = 0$. Further, if $Q \neq 0$, we call it multi-mode inner product.

B. Multivariate Multi-Order Markov Model

In a stochastic process, if the state has k attributes, we call the state k-variate; if the state at the current time epoch is determined by the states at previous m time epochs, we call the Markov chain m-order. Therefore, the multivariate multi-order Markov model is also called k-variate m-order Markov model.

1) First-Variate Second-Order Markov Transition: We take a second-order Markov model as an example to illustrate the unified product based (UP-based) multi-order Markov transition. Suppose the settings of the second-order Markov stochastic process are the same as that in Section III and the transition probability tensor is $\underline{P}''$ satisfying Eq. (8). Suppose the joint probability distribution matrix is represented as $M \in R^{I \times I}$, in which each entry $m_{ij} = Pr(X_t = i, X_{t-1} = j)$. According to the probability transition principle of second-order Markov models, we can obtain the following equations:

$$Pr(X_t X_{t-1} X_{t-2}) = Pr(X_t | X_{t-1} X_{t-2}) Pr(X_{t-1} X_{t-2}),$$
$$Pr(X_t X_{t-1}) = \sum_{t-2} Pr(X_t X_{t-1} X_{t-2}). \quad (13)$$

By combining Def. 4 and Eq. (13), we can find that the proposed unified product can be directly exploited to realize the function in Eq. (13). Therefore, the one-step transition for a second-order Markov chain can be represented as follows:

$$M^{(t,t-1)} = \underline{P}'' \circledast^{X_{t-2}, X_{t-1}}_{X_{t-2}; X_{t-1}} M^{(t-1,t-2)}. \quad (14)$$

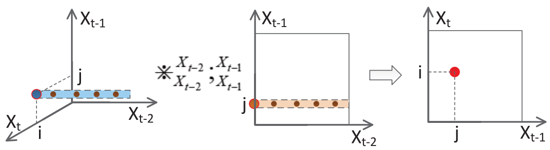


Fig. 4. UP-based state transition for a first-variate second-order Markov model.

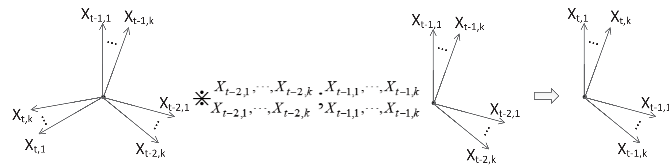


Fig. 5. UP-based state transition for a k-variate second-order Markov model.

The implementation process is illustrated in Fig. 4.

By integrating the definition of unified product in Def. 4 and Eq. (13), we give the detailed analysis about Eq. (14) from the probability transition point of view as follows:

$$\begin{aligned}\left(M^{(t,t-1)}\right)_{(i,j)} &= m_{ij} = Pr(X_t = i, X_{t-1} = j)\\ &= \sum_{t-2} \begin{matrix} Pr(X_t = i|X_{t-1} = j, X_{t-2} = k)\\ Pr(X_{t-1} = j, X_{t-2} = k)\end{matrix}\\ &= \sum_k p''_{ijk} m_{jk} = \left(\underline{P}'' \ast^{t-2,t-1}_{t-2;t-1} M^{(t-1,t-2)}\right)_{(i,j)}.\end{aligned} \tag{15}$$

From Eq. (15), we notice that the UP-based transition can be consecutively implemented without any other assumptions.

2) k-Variate Second-Order Markov Transition: Further, if the state is k-variate, then the k-variate second-order Markov transition can be accordingly realized based on the proposed transition principle in Section IV-B1. Suppose the transition probability tensor $\underline{P}''' \in R^{(I_1\times\cdots\times I_k)\times(I_1\times\cdots\times I_k)\times(I_1\times\cdots\times I_k)}$ in which $p'''_{i_{1,1}\cdots i_{1,k} i_{2,1}\cdots i_{2,k} i_{3,1}\cdots i_{3,k}} \geq 0$ and $\sum_{i_{1,1},\ldots,i_{1,k}=1}^{I_1,\ldots,I_k} p'''_{i_{1,1}\cdots i_{1,k} i_{2,1}\cdots i_{2,k} i_{3,1}\cdots i_{3,k}} = 1, \forall i_{j,1}\cdots i_{j,k} \in S' (j = 2, 3)$ and the joint probability distribution tensor is expressed as $\underline{M}' \in R^{(I_1\times\cdots\times I_k)\times(I_1\times\cdots\times I_k)}$ in which each entry is greater than or equal to 0 and the summation of all entries is 1.

Then the k-variate second-order Markov transition can be expressed as follows:

$$\begin{aligned}\underline{M}'^{(t,t-1)} &= \underline{P}''' \ast \underline{M}'^{(t-1,t-2)},\\ \ast &:= \ast_{k,k}\left(or \ast^{X_{t-2,1},\ldots,X_{t-2,k};X_{t-1,1},\ldots,X_{t-1,k}}_{X_{t-2,1},\ldots,X_{t-2,k};X_{t-1,1},\ldots,X_{t-1,k}}\right).\end{aligned} \tag{16}$$

The illustration is depicted in Fig. 5. The derivation can be easily achieved through the similar method in Section IV-B1. The difference is that each state in multivariate models is determined by k tensor orders in UP-based transition.

3) k-Variate m-Order Markov Transition: First, we define a k-variate m-order Markov chain as follows.

Definition 5 (k-Variate m-Order Markov Chain): Suppose the finite k-variate state set is S' defined in Eq. (5). Then a k-variate m-order Markov chain is formed when there is a fixed probability independent of the time epoch such that

$$\begin{aligned}&Pr\big(X_t = i_t|X_{t-1} = i_{t-1}, X_{t-2} = i_{t-2}, \ldots, X_0 = i_0\big)\\ &\quad = Pr\big(X_t = i_t|X_{t-1} = i_{t-1}, X_{t-2} = i_{t-2}, \ldots, X_{t-m} = i_{t-m}\big),\end{aligned} \tag{17}$$

where $X_t, X_{t-1}, \ldots, X_0$ and $i_t, i_{t-1}, \ldots, i_0$ are same as that in Notation 2, and $i_l \in S'$ $(l = t, t-1, \ldots, 0)$.

Suppose the probability in Eq. (17) is represented as $p_{i_{1,1},\ldots,i_{1,k},i_{2,1},\ldots,i_{2,k},\ldots,i_{m+1,1},\ldots,i_{m+1,k}}$. Then we can construct a $k{*}(m+1)$th-order transition probability tensor $\underline{P}$ for the k-variate m-order Markov model as follows.

$$\begin{aligned}&\underline{P} \in R^{(I_{1,1}\times\cdots\times I_{1,k})\times(I_{2,1}\times\cdots\times I_{2,k})\times\cdots\times(I_{m+1,1}\times\cdots\times I_{m+1,k})},\\ &\quad 0 \leq p_{i_{1,1},\ldots,i_{1,k},i_{2,1},\ldots,i_{2,k},\ldots,i_{m+1,1},\ldots,i_{m+1,k}} \leq 1,\\ &\quad \sum_{i_{1,1},\ldots,i_{1,k}=1}^{I_1,\ldots,I_k} p_{i_{1,1},\ldots,i_{1,k},i_{2,1},\ldots,i_{2,k},\ldots,i_{m+1,1},\ldots,i_{m+1,k}} = 1.\end{aligned} \tag{18}$$

Suppose the joint probability distribution is represented as a $(k{*}m)$th-order tensor $\underline{M}$ and defined as follows, in which each entry denotes the probability of joint states.

$$\begin{aligned}&\underline{M} \in R^{(I_{1,1}\times\cdots\times I_{1,k})\times(I_{2,1}\times\cdots\times I_{2,k})\times\cdots\times(I_{m,1}\times\cdots\times I_{m,k})},\\ &\quad m_{i_{1,1},i_{1,2},\ldots,i_{1,k},\ldots,i_{m,1},i_{m,2},\ldots,i_{m,k}}\\ &\quad = Pr\big(X_{1,1}, X_{1,2}, \ldots, X_{1,k} = i_{1,1}, i_{1,2}, \ldots, i_{1,k}, \cdots,\\ &\qquad X_{m,1}, X_{m,2}, \ldots, X_{m,k} = i_{m,1}, i_{m,2}, \ldots, i_{m,k}\big) \geq 0,\\ &\quad \sum \underline{M} = 1.\end{aligned} \tag{19}$$

According to the probability transition principle of the m-order Markov model, we can obtain:

$$\begin{aligned}Pr(X_t X_{t-1} \cdots X_{t-m+1}) &= \sum_{t-m} Pr(X_t X_{t-1} \cdots X_{t-m})\\ &= \sum_{t-m} Pr(X_t|X_{t-1} X_{t-2} \cdots X_{t-m})\\ &\quad \times Pr(X_{t-1} X_{t-2} \cdots X_{t-m}).\end{aligned} \tag{20}$$

Next, we implement the UP-based transition for a k-variate m-order Markov model and give two theorems. Note that the expressions of multivariate state and probability representation follow Notations 1 and 2 in the following section.

Theorem 1: Given a k-variate m-order Markov chain, the one-step transition by exploiting the proposed unified product can be implemented as follows:

$$\begin{aligned}&\underline{M}^{(t,t-1,\ldots,t-m+1)} = \underline{P} \ast \underline{M}^{(t-1,t-2,\ldots,t-m)},\\ &\quad \ast := \ast_{k,(m-1)k}\ or\\ &\quad \ast^{X_{t-m,1},\ldots,X_{t-m,k};X_{t-1,1},\ldots,X_{t-1,k},\ldots,X_{t-m+1,1},\ldots,X_{t-m+1,k}}_{X_{t-m,1},\ldots,X_{t-m,k};X_{t-1,1},\ldots,X_{t-1,k},\ldots,X_{t-m+1,1},\ldots,X_{t-m+1,k}}.\end{aligned} \tag{21}$$

Proof: According to Eqs. (20) and (21) and the definition of unified product defined in Def. 4, we have

$$\begin{aligned}&\left(\underline{M}^{(t,t-1,\ldots,t-m+1)}\right)_{i_1,i_2,\ldots,i_m}\\ &\quad = Pr(X_t = i_1, X_{t-1} = i_2, \ldots, X_{t-m+1} = i_m)\\ &\quad = \sum_{t-m} Pr(X_t = i_1, X_{t-1} = i_2, \ldots, X_{t-m} = i_{m+1})\end{aligned}$$

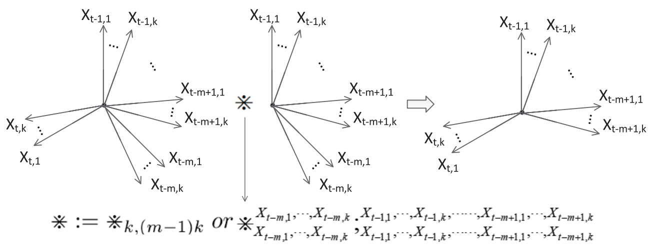


Fig. 6. UP-based transition for a *k*-variate *m*-order Markov model.

$$= \sum_{t-m} \begin{array}{l} Pr(X_t = i_1 | X_{t-1} = i_2, \ldots, X_{t-m} = i_{m+1}) \\ \times\ Pr(X_{t-1} = i_2, \ldots, X_{t-m} = i_{m+1}) \end{array}$$

$$= \sum_{i_{m+1}=1}^{I_1,\ldots,I_k} p_{i_1, i_2, \ldots, i_{m+1}} m_{i_2,\ldots,i_{m+1}}$$

$$= \left(\underline{P} \circledast \underline{M}^{(t-1,t-2,\ldots,t-m)} \right)_{i_1, i_2, \ldots, i_m}.$$

It is clear that the UP-based transition in Eq. (21) exactly follows the probability transition principle of a *k*-variate *m*-order Markov model. ■

The graphical representation of UP-based one-step transition is illustrated in Fig. 6. Therefore, we notice that the consecutive transitions can be implemented through the joint probability distribution tensor in UP-based transition for a multivariate multi-order Markov model.

Theorem 2: The sum of the joint probability distribution tensor remains 1 after implementing the UP-based transition in a *k*-variate *m*-order Markov model.

Proof: By combining Eq. (20), we can obtain:

$$\sum_{t,t-1,\ldots,t-m+1} Pr(X_t X_{t-1} \cdots X_{t-m+1})$$

$$= \sum_{t,t-1,\ldots,t-m+1} \sum\nolimits_{t-m} \begin{array}{l} Pr(X_t | X_{t-1} X_{t-2} \cdots X_{t-m}) \\ \times\ Pr(X_{t-1} X_{t-2} \cdots X_{t-m}) \end{array}$$

$$= \sum_{t-1,t-2,\ldots,t-m} \begin{array}{l} Pr(X_{t-1} X_{t-2} \cdots X_{t-m}) \\ \times \sum_t Pr(X_t | X_{t-1} X_{t-2} \cdots X_{t-m}). \end{array} \tag{22}$$

Based on Eq. (18), we have $\sum_t\ Pr(X_t | X_{t-1} X_{t-2} \cdots X_{t-m}) = 1$, then substitute it to Eq. (22) and exploit $\sum \underline{M}^{t-1,t-2,\ldots,t-m} = 1$ in Eq. (19), we can obtain:

$$\sum_{t,t-1,\ldots,t-m+1} Pr(X_t X_{t-1} \cdots X_{t-m+1})$$

$$= \sum_{t-1,t-2,\ldots,t-m} Pr(X_{t-1} X_{t-2} \cdots X_{t-m}) = 1.$$

■

Besides, based on the joint probability distribution tensor $\underline{M}$ at *m* consecutive time epochs, i.e., $t, t-1, \ldots, t-m+1$, we can calculate the probability distribution tensor $\underline{X}$ at the *t*th time epoch.

$$\underline{X} \in R^{I_1 \times \cdots \times I_k}, \underline{X}^{(t)} = \sum_{\widetilde{X_{t,1},\ldots,X_{t,k}}} \underline{M}^{(t,t-1,\ldots,t-m+1)}, \tag{23}$$

where $\widetilde{X_{t,1},\ldots,X_{t,k}}$ represents all orders of tensor $\underline{M}$ except for these $X_{t,1}, \ldots, X_{t,k}$ orders. In fact, Eq. (23) can be inferred from the following probability equation:

$$Pr(X_t) = \sum_{t-1,\ldots,t-m+1} Pr(X_t X_{t-1} \cdots X_{t-m+1}).$$

C. Multivariate Multi-Order Markov Multi-Step Transition

In the traditional first-order Markov, if the transition probability matrix is *P* defined in Eq. (3), and the probability distribution of states at the *t*th time epoch is $\mathrm{v}^{(t)}$, we can obtain the probability distribution $\mathrm{v}^{(t+q)}$ of states after *q*-step transitions as follows:

$$\begin{aligned} \mathrm{v}^{(t+q)} &= P \times \left(P \times \cdots \times \left(P \times \mathrm{v}^{(t)} \right) \right) \\ &= P \times_{n-1}^{n-1} \left(P \times_{n-1}^{n-1} \cdots \times_{n-1}^{n-1} \left(P \times_{n-1}^{n-1} \mathrm{v}^{(t)} \right) \right). \end{aligned} \tag{24}$$

On the other hand, Eq. (24) is equivalent to the following form.

$$\begin{aligned} \mathrm{v}^{(t+q)} &= \left(P \times_{n-1}^{n} P \times_{n-1}^{n} \cdots \times_{n-1}^{n} P \right) \times_{n-1}^{n-1} \mathrm{v}^{(t)} \\ &= P^{q}_{\times_{n-1}^{n}} \times_{n-1}^{n-1} \quad \mathrm{v}^{(t)}. \end{aligned} \tag{25}$$

In general, P^q is called *q*-step transition probability matrix. Note that the two single-mode product operations in P^q and Eq. (24) are different in nature.

Next, we generalize the idea of the *q*-step transition probability matrix to 2M Markov model and compute the *q*-step transition probability tensor.

Theorem 3: Given a *k*-variate *m*-order Markov model, suppose the transition probability tensor is $\underline{P}$ satisfying Eq. (18), and the current joint probability distribution tensor is $\underline{M}^{(t,t-1,\ldots,t-m+1)}$ satisfying Eq. (19). Then the UP-based *q*-step transition for *k*-variate *m*-order Markov model is presented.

$$\begin{aligned} &\underline{M}^{(t+q,t+q-1,\ldots,t+q-m+1)} \\ &\quad = \underline{P} \circledast \left(\underline{P} \circledast \cdots \circledast \left(\underline{P} \circledast \underline{M}^{(t,t-1,\ldots,t-m+1)} \right) \right). \end{aligned} \tag{26}$$

Eq. (26) can also be implemented by the following approach.

$$\underline{M}^{(t+q,t+q-1,\ldots,t+q-m+1)} = \underline{P}^{q} \circledast \underline{M}^{(t,t-1,\ldots,t-m+1)}, \tag{27}$$

$$\underline{P}^{q} = \underline{P} \circledast \underline{P} \circledast \cdots \circledast \underline{P}. \tag{28}$$

The unified product in Eqs. (26) and (27) is the defined operation in Eq. (21), and the unified product in Eq. (28) should be:

$$\circledast^{X_{t,1},\ldots,X_{t,k}}_{X_{t-1,1},\ldots,X_{t-1,k}} {}^{;X_{t-1,1},\ldots,X_{t-1,k},\ldots,\ldots,X_{t-m+1,1},\ldots,X_{t-m+1,k}}_{;X_{t-2,1},\ldots,X_{t-2,k},\ldots,\ldots,X_{t-m,1},\ldots,X_{t-m,k}}. \tag{29}$$

$\underline{P}^q$ in Eq. (28) is called the UP-based *q*-step transition probability tensor.

Proof: According to the principle of conditional probability and the definition of *k*-variate *m*-order Markov chain in Def. 5, we have

$$\begin{aligned} &Pr\left(X_{t+q-1} X_{t+q-2} \cdots X_t | X_{t-1} X_{t-2} \cdots X_{t-m}\right) \\ &\quad = \frac{Pr\left(X_{t+q-1} X_{t+q-2} \cdots X_t X_{t-1} X_{t-2} \cdots X_{t-m}\right)}{Pr(X_{t-1} X_{t-2} \cdots X_{t-m})} \end{aligned}$$

$$
\begin{aligned}
&= \frac{Pr\big(X_{t+q-1}X_{t+q-2}\cdots X_t X_{t-1}X_{t-2}\cdots X_{t-m}\big)}{Pr\big(X_{t+q-2}X_{t+q-3}\cdots X_{t-m}\big)} \\
&\quad\times \frac{Pr\big(X_{t+q-2}X_{t+q-3}\cdots X_{t-m}\big)}{Pr\big(X_{t+q-3}X_{t+q-4}\cdots X_{t-m}\big)}\cdots \\
&\quad\times \frac{Pr(X_t X_{t-1}\cdots X_{t-m})}{Pr(X_{t-1}X_{t-2}\cdots X_{t-m})} \\
&= Pr\big(X_{t+q-1}|X_{t+q-2}\cdots X_{t-m}\big) \\
&\quad\times Pr\big(X_{t+q-2}|X_{t+q-3}\cdots X_{t-m}\big)\cdots \\
&\quad\times Pr(X_t|X_{t-1}\cdots X_{t-m}) \\
&= Pr\big(X_{t+q-1}|X_{t+q-2}\cdots X_{t+q+m-2}\big) \\
&\quad\times Pr\big(X_{t+q-2}|X_{t+q-3}\cdots X_{t+q+m-3}\big)\cdots \\
&\quad\times Pr(X_t|X_{t-1}\cdots X_{t-m}). \qquad (30)
\end{aligned}
$$

We know that each entry in $\underline{P}^q$ denotes a q-step transition probability. By combining Eq. (30), we have

$$
\begin{aligned}
&Pr\big(X_{t+q-1}|X_{t-1}X_{t-2}\cdots X_{t-m}\big) \\
&= \sum_{t+q-2,\dots,t} Pr\big(X_{t+q-1}X_{t+q-2}\cdots X_t|X_{t-1}\cdots X_{t-m}\big) \\
&= \sum_{t+q-2,\dots,t} \begin{matrix} Pr\big(X_{t+q-1}|X_{t+q-2}\cdots X_{t+q+m-2}\big) \\ Pr\big(X_{t+q-2}|X_{t+q-3}\cdots X_{t+q+m-3}\big) \\ \cdots Pr(X_t|X_{t-1}\cdots X_{t-m}) \end{matrix} \\
&= \sum_{t+q-2} Pr\big(X_{t+q-1}|X_{t+q-2}\cdots X_{t+q+m-2}\big) \\
&\quad\times \Bigg(\sum_{t+q-3} Pr\big(X_{t+q-2}|X_{t+q-3}\cdots X_{t+q+m-2}\big) \\
&\qquad\times \Bigg(\cdots\Bigg(\sum_{t} Pr(X_{t+1}|X_t\cdots X_{t-m+1}) \\
&\qquad\qquad\times Pr(X_t|X_{t-1}\cdots X_{t-m})\Bigg)\Bigg)\Bigg) \qquad (31)
\end{aligned}
$$

According to the definition of unified product in Def. 4, we can infer that the operations in Eqs. (27) and (31) are equivalent. ■

Furthermore, if we expect to calculate the probability distribution tensor $X^{(t+q)}$ at the ($t+q$)th time epoch, the implementation approach can be represented as follows by integrating Eqs. (23) and (27).

$$
\begin{aligned}
\underline{X}^{(t+q)} &= \underline{P}^q * \underline{M}^{(t,t-1,\dots,t-m+1)} \\
* &:= *^{X_{t-1,1},\dots,X_{t-1,k},\dots,\dots,X_{t-m,1},\dots,X_{t-m,k}}_{X_{t-1,1},\dots,X_{t-1,k},\dots,\dots,X_{t-m,1},\dots,X_{t-m,k}}. \qquad (32)
\end{aligned}
$$

V. Multivariate Multi-Order Markov Prediction

In this section, we propose an iterative algorithm to calculate the stationary joint probability distribution for a 2M Markov model and then present a multi-modal prediction approach.

A. Stationary Joint Probability Distribution Tensor

In general, the stationary distribution in Markov models can be used to implement future predictions. Motivated by the idea of power method in PageRank [29] and dominant Z-eigenvector [30], we propose an iterative UP-based power method (UP-PM) to calculate the stationary joint probability distribution tensor for a 2M Markov model, i.e., stationary joint eigentensor (SJE). Specifically, to guarantee that the UP-PM is convergent, one attempt is to ensure the transition probability tensor should be aperiodic and irreducible, i.e., $\underline{P}' = \alpha\underline{P} + (1-\alpha)\underline{A}$, where $\underline{A}$ is an adjustment transition tensor satisfying Eq. (18), whose entry is equal to $\frac{1}{(I_1 I_2 \cdots I_k)^m}$. By combining Eq. (21), another equivalent approach is to perform the following stochastic and primitivity adjustment.

Algorithm 1: Algorithm of UP-PM for Calculating the Stationary Joint Eigentensor for a 2M Markov Model

Input:
A $k*(m+1)$th-order transition probability tensor $\underline{P}$ in Eq. (18) and the convergence threshold ε.
Output:
A stationary joint eigentensor $\underline{M}$ satisfying Eq. (19) and a stationary eigentensor $\underline{X}$ in Eq. (23).

1 **begin**
2 Select an initial random tensor $\underline{M}_0$ satisfying Eq. (19);
3 $j \leftarrow 0$;
4 **repeat**
5 $j \leftarrow j+1$;
6 Execute $\underline{M}_j = \alpha\underline{P} \circledast \underline{M}_{j-1} + (1-\alpha)\underline{E}$;
7 **until** $\|\underline{M}_j - \underline{M}_{j-1}\| < \varepsilon$;
8 $\underline{M} \leftarrow \underline{M}_j$;
9 Compute stationary eigentensor $\underline{X}$ based on stationary joint eigentensor $\underline{M}$ according to Eq. (23);
10 return $\underline{M}$ and $\underline{X}$.

$$\underline{M} = \alpha\underline{P} \circledast \underline{M} + (1-\alpha)\underline{E}. \qquad (33)$$

Therein, $\circledast$ is the unified product in Eq. (21), $\underline{E}$ is an adjustment joint distribution tensor satisfying Eq. (19), whose entry is equal to $\frac{1}{(I_1 I_2 \cdots I_k)^m}$. $0 < \alpha < 1$ is an adjustment parameter and will affect the convergence speed. The pseudocode of UP-PM is illustrated in Algorithm 1. On line 7 of Algorithm 1, $\|\bullet\|$ represents the norm and we can select a suitable norm type according to practical situations.

B. Algorithm Analysis

In this section, we shall analyze the existence, uniqueness, and convergence of UP-PM, as well as its time complexity.

1) Existence: We first prove the existence of UP-PM.

Theorem 4: Let $\underline{P}$ be a transition probability tensor for a 2M Markov model satisfying Eq. (18), then there exists a nonzero non-negative tensor $\widehat{\underline{M}}$ satisfied Eq. (19) such that $\widehat{\underline{M}} = \alpha\underline{P} \circledast \widehat{\underline{M}} + (1-\alpha)\underline{E}$ and $\sum\widehat{\underline{M}} = 1$. $\widehat{\underline{M}}$ is called stationary joint eigentensor.

Proof: The problem can be considered as a fixed point problem. Based on the properties in Eq. (19), let $\Omega = \{m_{i_{1,1},i_{1,2},\dots,i_{1,k},\dots,i_{m,1},i_{m,2},\dots,i_{m,k}}\}$. It is clear that Ω is a closed and convex set. We define the following nonlinear map

$$\Psi(\underline{M}) = \alpha\underline{P} \circledast \underline{M} + (1-\alpha)\underline{E}. \qquad (34)$$

We can see that Ψ is well-defined and continuous. According to the Brouwer Fixed Point Theorem [30], there exists $\underline{\widehat{M}} \in \Omega$ such that $\Psi(\underline{\widehat{M}}) = \underline{\widehat{M}}$.

According to Eqs. (18) and (19), every entry in tensors $\underline{P}$ and $\underline{M}$ is greater than or equal to 0, and every entry in tensor $\underline{E}$ is $\frac{1}{(I_1 I_2 \cdots I_k)^m}$ which is greater than 0, hence, every entry in $\alpha \underline{P} \circledast \underline{M} + (1-\alpha)\underline{E}$ is greater than 0, i.e., tensor $\underline{\widehat{M}}$ is nonzero and non-negative. Besides, according to Theorem 2, it is clear that $\sum \underline{\widehat{M}} = 1$. ■

2) Uniqueness: The uniqueness of the solution in Eq. (33) is proved in the following theorem.

Theorem 5: Let $\underline{P}$ be a transition probability tensor for a 2M Markov model satisfying Eq. (18), then there exists a unique solution in Eq. (33) when $0 < \alpha < 1$.

Proof: We shall prove the theorem by using reduction to absurdity. Assume there are two distinct stationary solution $\underline{\widehat{M}}_1$ and $\underline{\widehat{M}}_2$ in Eq. (33), then we can obtain

$$\underline{\widehat{M}}_1 = \alpha \underline{P} \circledast \underline{\widehat{M}}_1 + (1-\alpha)\underline{E},$$
$$\underline{\widehat{M}}_2 = \alpha \underline{P} \circledast \underline{\widehat{M}}_2 + (1-\alpha)\underline{E}.$$

Then, by subtracting these two equations, we get

$$\left\| \underline{\widehat{M}}_1 - \underline{\widehat{M}}_2 \right\| = \alpha \left\| \underline{P} \circledast \left(\underline{\widehat{M}}_1 - \underline{\widehat{M}}_2 \right) \right\|. \tag{35}$$

According to the definition of unified product in Def. 4 and the property (the sum is 1) of transition probability tensor $\underline{P}$ in Eq. (18), by combining $0 < \alpha < 1$, the right side of Eq. (35) can be converted to

$$\alpha \left\| \underline{P} \circledast \left(\underline{\widehat{M}}_1 - \underline{\widehat{M}}_2 \right) \right\| = \alpha \left\| \underline{\widehat{M}}_1 - \underline{\widehat{M}}_2 \right\| < \left\| \underline{\widehat{M}}_1 - \underline{\widehat{M}}_2 \right\|. \tag{36}$$

By integrating Eqs. (35) and (36), we can infer

$$\left\| \underline{\widehat{M}}_1 - \underline{\widehat{M}}_2 \right\| < \left\| \underline{\widehat{M}}_1 - \underline{\widehat{M}}_2 \right\|. \tag{37}$$

It is clear that Eq. (37) is a contradiction. Therefore, the theorem is proved. ■

3) Convergence: The following theorem is to prove the convergence of UP-PM.

Theorem 6: Let $\underline{P}$ be a transition probability tensor for a 2M Markov model satisfying Eq. (18) and $\underline{M}^{(0,-1,\ldots,-m+1)}$ be an any initial tensor satisfying Eq. (19). If $0 < \alpha < 1$, then the fixed-point iteration

$$\underline{M}^{(t,t-1,\ldots,t-m+1)} = \alpha \underline{P} \circledast \underline{M}^{(t-1,t-2,\ldots,t-m)} + (1-\alpha)\underline{E}. \tag{38}$$

will converge to a unique solution in Theorem 5.

Proof: Suppose the unique solution in Theorem 5 is $\underline{\widehat{M}}$ which satisfies Eq. (19), we have

$$\underline{\widehat{M}} = \alpha \underline{P} \circledast \underline{\widehat{M}} + (1-\alpha)\underline{E}. \tag{39}$$

According to the definition of unified product in Def. 4 and the property (the sum is 1) of transition probability tensor $\underline{P}$ in Eq. (18), by subtracting Eqs. (38) and (39), we have

$$\begin{aligned} &\left\| \underline{M}^{(t,t-1,\ldots,t-m+1)} - \underline{\widehat{M}} \right\| \\ &= \alpha \left\| \underline{P} \circledast \left(\underline{M}^{(t-1,t-2,\ldots,t-m)} - \underline{\widehat{M}} \right) \right\| \\ &= \alpha \left\| \underline{M}^{(t-1,t-2,\ldots,t-m)} - \underline{\widehat{M}} \right\|. \end{aligned}$$

Further, we can obtain

$$\begin{aligned} \left\| \underline{M}^{(t,t-1,\ldots,t-m+1)} - \underline{\widehat{M}} \right\| &= \alpha \left\| \underline{M}^{(t-1,t-2,\ldots,t-m)} - \underline{\widehat{M}} \right\| \\ &= \alpha^2 \left\| \underline{M}^{(t-2,t-3,\ldots,t-m-1)} - \underline{\widehat{M}} \right\| \\ &= \cdots = \alpha^t \left\| \underline{M}^{(0,-1,\ldots,-m+1)} - \underline{\widehat{M}} \right\|. \end{aligned}$$

Since $0 < \alpha < 1$, then $\lim_{t \to \infty} \alpha^t = 0$. Thus, for an arbitrary tensor $\underline{M}^{(0,-1,\ldots,-m+1)}$, we can obtain

$$\lim_{t \to \infty} \left\| \underline{M}^{(t,t-1,\ldots,t-m+1)} - \underline{\widehat{M}} \right\| = 0.$$

Therefore, the fixed-point iteration in Eq. (38) can converge to $\underline{\widehat{M}}$ and the convergence speed is determined by the adjustment parameter α. ■

4) Time Complexity: In Algorithm 1, the time complexity is mainly determined by the execution of unified product on line 6. Without loss of generality, for a *k*-variate *m*-order Markov model, suppose $I = max\{I_1, I_2, \ldots, I_k\}$. According to Def. 4 and Fig. 6, the time complexity of one-step transition in Eq. (21) is $O(I^{k(m+1)})$, thus the time complexity of UP-PM is

$$Time = O\left(N * I^{k(m+1)}\right), \tag{40}$$

where *N* is the iterative number.

C. Stationary Joint Eigentensor Based Multi-Modal Prediction

1) Multi-Modal Prediction Approaches: In the SJE based approach, the stationary joint distribution is used to implement future predictions. For a first-order Markov model, there is no joint probabilities. And if the model is first-variate, the SJE degrades to a vector and the Top-*K* predicted values can be directly used to perform predictions. If the model is multivariate, take the traffic prediction as an example, we need to first extract the *Traffic* fiber from the SJE by specifying all orders except *Traffic* according to the given state attributes, then we can use the Top-*K* predicted values in the *Traffic* fiber to perform predictions. Therefore, the prediction results will be distinct under different state attributes, we call it multi-modal prediction. However, when it comes to a multi-order Markov model, previous states should be jointly taken into consideration when implementing future predictions. For a *k*-variate *m*-order Markov model, we need to first specify the values of all states at *m*-1 past time epochs according to practical scenarios and then extract the *k*th-order tensor from the SJE, which represents the stationary probability distribution of states at next time epoch when recent *m*-1 states are given. Afterwards, we can exploit the aforementioned multi-modal prediction approach to implement future predictions based on the extracted *k*th-order tensor.

To verify whether the assumption in Z-eigen based approach is reasonable, we expect to analyze the impact of the stationary eigentensors generated from Z-eigen based and SJE based approaches on the prediction accuracy. Therefore, we can calculate the stationary eigentensor (SE) according to Eq. (23) after obtaining the stationary joint eigentensor. It is obtained by performing summations over all states at *m*-1 past time

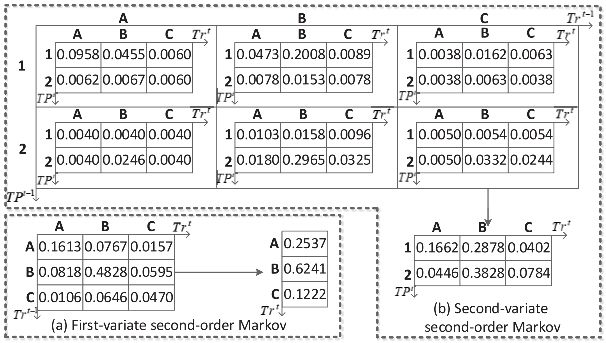


Fig. 7. Examples of network traffic predictions in SJE based and SE based approaches.

epochs. The result will be a stationary probability distribution vector (or tensor for multivariate). Therefore, the result form is similar to that in SJE based approach under first-order Markov models, and we can use the same approach to implement future predictions. We represent the prediction approach as SE based approach.

2) Multi-Modal Prediction Examples: We take a second-order Markov model for network traffic as an example to illustrate the prediction details of the two approaches. Suppose the state is *Traffic* in the first-variate model and it is *(TimePeriod, Traffic)* in the second-variate model. First, we introduce the first-variate situation, which is depicted in Fig. 7(a). For the SJE based approach, each entry in SJE denotes the stationary joint probability of traffic states at two consecutive time epochs, illustrated in the left part of Fig. 7(a). To predict next traffic state, we should first determine the current traffic state, such as $Tr^{t-1} = C$. Then, we can extract the *Traffic* fiber [(*A*: 0.0106), (*B*: 0.0646), (*C*: 0.0470)], which represents the probability distribution of traffic states at next time epoch. After sorting the *Traffic* fiber in descending order like [*B*, *C*, *A*], we can apply its Top-*K* predicted values to predict the next traffic state. If Top-2 predicted values are used, we expect that the next traffic state should lay in the prediction set {*B*, *C*}. For the SE based approach, the resulting vector by summing the values on Tr^{t-1} is illustrated in the right part of Fig. 7(a). To predict the network traffic, we can only directly use the Top-*K* predicted values from the sorted vector [*B*, *A*, *C*]. We can see from Fig. 7(a) that the prediction results are different by using SJE based and SE based approaches.

Then, the second-variate situation is discussed as follows. For the SJE based prediction approach, the generated SJE is a 4th-order tensor, depicted in the upper part of Fig. 7(b). Differing from the first-variate second-order Markov model, each state in this SJE is determined by two orders (*TimePeriod*, *Traffic*), i.e., (*TP*, *Tr*) in Fig. 7(b). To predict the next traffic state, first we need to specify the current state (*TimePeriod*, *Traffic*), such as $(TP^{t-1}, Tr^{t-1}) = (1, A)$. In this way, we can obtain a matrix containing the probability distribution of next state. Suppose the *TimePeriod* of next state is 1 (i.e., $TP^t = 1$), then we can extract the *Traffic* fiber [(*A*: 0.0958), (*B*: 0.0455), (*C*: 0.0060)] for the next state. After that, we can choose the Top-*K* predicted values from sorted fiber [*A*, *B*, *C*] to build a prediction set and perform traffic predictions. Besides, the lower part of Fig. 7(b) depicts a matrix obtained by SE based approach, which represents the stationary probability distribution of next states (*TimePeriod*, *Traffic*). Given the value of *TimePeriod* of next state (e.g., $TP^t = 1$), the *Traffic* fiber [(*A*: 0.1662), (*B*: 0.2878), (*C*: 0.0402)] can be extracted from the matrix, and we can further predict the traffic according to the Top-*K* predicted values in the sorted fiber [*B*, *A*, *C*].

VI. Experiments

In this section, a series of experiments are conducted using real-world network traffic data to verify the prediction performance of the proposed SJE based approach. We compare the prediction accuracy of SJE based approach and other state-of-the-art approaches under various 2M Markov models. Furthermore, the influence of different variates and orders on prediction accuracy is discussed.

A. Metric

To evaluate the prediction performance, the prediction accuracy measure is applied and defined as follows.

Definition 6 (Prediction Accuracy): Suppose the predicted Top-K traffic values constitute a prediction set $PS_{Top-K} = \{PV_1, PV_2, \ldots, PV_K\}$. Given a testing traffic sequence $TS = \{Tf_1, Tf_2, \ldots, Tf_i, \ldots, Tf_N\}$. For every entry in *TS*, if $Tf_i \in PS_{Top-K}$, we call it one time of hit, namely,

$$Hit\big(Tf_i, PS_{Top-K}\big) = \begin{cases} 1, & Tf_i \in PS_{Top-K} \\ 0, & Tf_i \notin PS_{Top-K} \end{cases}$$

Then, the prediction accuracy is calculated as follows:

$$Accuracy = \frac{\sum_{i=1}^{N} Hit\big(Tf_i, PS_{Top-K}\big)}{N}.$$

B. Experimental Design

The experiments are implemented through NumPy package in Python. All experiments are executed on a cloud platform which configures an Intel's 16-core Xeon E5-2630 processor with 2.4 GHZ and a 125 GB memory.

1) Datasets: The real-world network traffic data is collected from *FiberHome* packet transport network device deployed in telecommunication operator. *FiberHome* is a leading network solution provider in the telecommunications equipment manufacturing industry of China. The traffic data totally contains 11196 network flow records generated from four different ports, which is collected for a consecutive time of 30 days. After analyzing the raw data, we construct two datasets from 640 MSK XGE Port 1 (dataset 1) and 640 XSK XGE Port 1 (dataset 2), and each dataset contains 2801 network traffic records. The average network traffic is stored in a record for every 15 minutes, e.g., "2018/6/5 00:00-00:15 17.588Mbps $\cdots$". Then we remove irrelevant data fields and preprocess these data according to the experimental requirements.

2) Parameters Settings: Based on these two preprocessed datasets, we set three variates for each state in 2M Markov models, i.e., *Holiday*, *TimePeriod*, and *Traffic*. The value of

Holiday is determined by whether the current day is a holiday (including weekend), if yes, the value is 1, otherwise 0. To reflect the regular patterns of network traffic, *TimePeriod* is set to 4 periods for one day, i.e., 0:00-6:00, 6:00-12:00, 12:00-18:00, and 18:00-24:00. As regards *Traffic*, the average network inflow traffic is adopted. According to the traffic distribution of datasets, the traffic of dataset 1 and dataset 2 are equally divided into 20 slices and 19 slices, where the interval of each slice is 0.3 Mbps and 2 Mbps, respectively. During all experiments, the ratio of training to testing data is 8:2, the adjustment parameter α is 0.85, the convergence threshold ε is 1e-6, and the norm measure is 2-norm.

3) Markov Model Construction: Based on the preprocessed datasets and parameters, we can construct various Markov models according to various variate (k = 1, 2, 3) and order (m = 1, 2, 3). In the constructed Markov models, the state in the first-variate models is *Traffic*, it becomes *(TimePeriod, Traffic)* in the second-variate models, and it will be *(Holiday, TimePeriod, Traffic)* in the third-variate models. For every k-variate m-order Markov model, we first count the total transition number for every pair of k-variate states according to the definition of k-variate m-order Markov in Def. 5. Then we normalize these occurring number and construct the transition probability tensor. The concrete construction process can be referred to [23], [24].

4) Baselines: To verify the performance of different prediction approaches for 2M Markov models, three approaches are compared, i.e., SJE based, SE based, and Z-eigen based approaches. The prediction process of the SJE based and SE based approaches have been illustrated in detail in Section V-C. In the Z-eigen based approach, a dominant Z-eigenvector (or Z-eigentensor for multivariate models) can be obtained after executing dominant Z-eigen decomposition [23], [30], which denotes the stationary probability distribution of states. Even though the values of dominant Z-eigenvector/Z-eigentensor in the Z-eigen based approach and stationary eigenvector/eigentensor in the SE based approach are somewhat different, they have similar structures and both represent the stationary probability distribution of states. Thus, the prediction process of Z-eigen based approach is similar to that in SE based approach. In these experiments, other machine learning based prediction approaches, such as naive Bayes, deep neural network, etc., are not selected as the baselines. This is because this paper focuses on studying the prediction of Markov models, especially the tensor-based multivariate multi-order Markov transition model.

C. Evaluations of Prediction Accuracy

1) Comparisons of Prediction Accuracy Among Different Prediction Approaches: To verify the advantages of the proposed SJE based approach in multi-order Markov models, we construct a series of k-variate second-order Markov models (k = 1, 2, 3) on dataset 1 and dataset 2, and then compare their prediction accuracies among SJE based, SE based and Z-eigen based approaches. Fig. 8 illustrates the prediction accuracy comparisons of the three approaches under different second-order models, where the x-axis and y-axis represent the Top-K value and prediction accuracy, respectively. It can be seen from Fig. 8 that the SJE based approach gains the highest prediction accuracy among the three approaches for all models. Especially, it exhibits more superiorities when the value of Top-K is smaller. Table II gives the prediction accuracy of different approaches under various Markov models on dataset 2. Compared with the Z-eigen based approach, the SJE based approach can improve the prediction accuracy by 22.85, 24.92, 15.14 percentage points in average when the value of Top-K is 4, 8, 12, respectively, and the highest improvement reaches to 38.47 percentage points. These experimental results show that the SJE based approach is more efficient. They further confirm our aforementioned analysis in Section III. In multi-order Markov models, the prediction approach based on the stationary joint probability distribution is more reasonable and efficient than on the first-order stationary probability distribution.

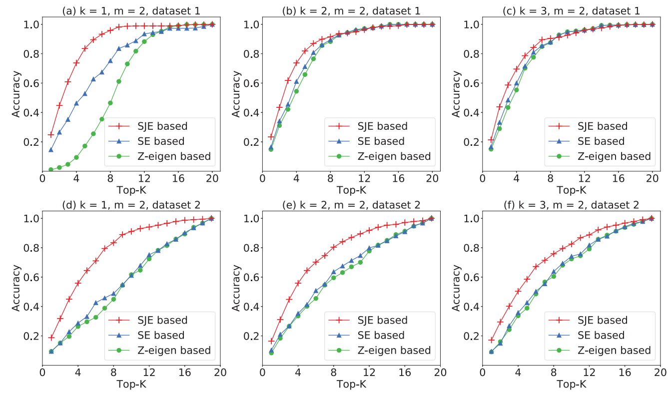


Fig. 8. Comparisons of prediction accuracy among different approaches under various Markov models.

Meanwhile, we can see from Fig. 8 that SE based approach slightly outperforms Z-eigen based approach under most Top-K values. Table II shows that the SE based approach can improve the prediction accuracy by 2.08, 3.80, and 2.56 percentage points in average when the value of Top-K is 4, 8, and 12, respectively, compared with the Z-eigen based approach. According to the analysis in Section VI-B4, the prediction process of SE based and Z-eigen based approaches are similar. Therefore, we infer that the difference of prediction accuracy is likely caused by the independence assumption in calculating the stationary probability distribution. Note that if we perform small-scale experiments in [30] by exploiting the SE based and Z-eigen based approaches, we can obtain the same results. Besides, we perform these approaches based on another people's trajectory dataset (i.e., GeoLife), the SE based approach also shows its superiority in prediction accuracy. Thus, as we discussed in Section III, we can see that the independence assumption in Z-eigen based approach is not necessarily satisfied for all scenarios.

2) Comparisons of Prediction Accuracy Under Different Variates: To explore the influence of different variates on prediction accuracy in Markov models, we select three k-variate first-order Markov models (i.e., k = 1, 2, 3), i.e., *Traffic*, *TimePeriod-Traffic*, and *Holiday-TimePeriod-Traffic*,

TABLE II
PREDICTION ACCURACY OF DIFFERENT APPROACHES UNDER VARIOUS SECOND-ORDER MARKOV MODELS ON DATASET 2

	k = 1			k = 2			k = 3		
Top-K	4	8	12	4	8	12	4	8	12
SJE based	55.89%	83.39%	94.11%	55.89%	80.36%	91.79%	50.36%	75.89%	88.75%
SE based	28.70%	48.84%	75.22%	35.29%	63.64%	79.86%	35.83%	63.81%	81.82%
Z-eigen based	26.38%	44.92%	72.37%	33.51%	59.54%	77.72%	33.69%	60.43%	79.14%

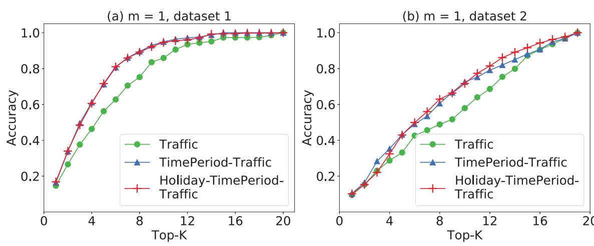


Fig. 9. Comparisons of prediction accuracy under different variates in first-order Markov models.

and then compare their prediction accuracies. The reason of choosing first-order Markov models is that there is no difference among the three prediction approaches in first-order models and the general influence of different variates can be demonstrated.

Fig. 9 gives the comparisons and shows that second-variate and third-variate models can achieve higher prediction accuracy than first-variate model. These experimental results verify the efficiency of multi-modal prediction by comprehensively considering the diversity of states. For instance, network traffic is not only related to past traffic but also influenced by current time and date. However, compared with second-variate model, third-variate model does not have distinct superiority. We analyze the possible reason is that the influence of holiday on network traffic might not be very prominent for the selected two datasets.

3) Comparisons of Prediction Accuracy Under Different Orders: To explore the influence of different Markov orders on prediction accuracy with three approaches, we conduct six groups of experiments under various situations.

Fig. 10 shows the comparisons of prediction accuracy among various Markov models with different orders ($m = 1, 2, 3$) for every approach. For the SJE based approach, the experimental results from Figs. 10(a)(d) depict that second-order and third-order models perform better than first-order model, while second-order model gains the highest accuracy. It demonstrates that the second-order Markov model is more suitable for the network traffic dataset. However, for the Z-eigen based and SE based approaches, we can see from Figs. 10(b)(e) and Figs. 10(c)(f) that increasing orders has negligible influence on prediction accuracy. The results also confirm our proposed wondering in Section III, namely, it is not reasonable to implement future predictions by directly adopting the first-order stationary distribution in multi-order Markov models. Instead, the SJE based approach has higher prediction accuracy by using stationary joint eigentensor to predict network traffic under multi-order Markov models. This is because the SJE based approach takes recent states into consideration during traffic prediction, which is consistent with

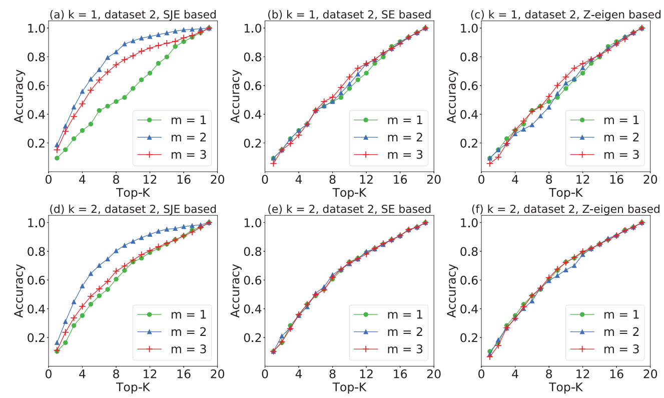


Fig. 10. Comparisons of prediction accuracy under different orders in all prediction approaches.

TABLE III
NUMBER OF ITERATIONS UNDER VARIOUS α VALUES

	$\alpha = 0.70$	$\alpha = 0.75$	$\alpha = 0.8$	$\alpha = 0.85$	$\alpha = 0.9$
k=1,m=2	20	23	28	35	45
k=2,m=2	21	25	31	40	55

the concept of multi-order Markov process. Besides, we can see that second-order models have better performance than third-order models when using the SJE based approach. The possible reason for this phenomenon is that the current state is closely related to previous two states, but not three states in the real-world network traffic datasets.

D. Convergence Analysis

To analyze the convergence of UP-PM in the SJE based approach, we conduct several experiments for two Markov models on dataset 1. One is a first-variate second-order Markov model, the size of whose transition probability tensor is 20∗20∗20. Another is a second-variate second-order Markov model, the size of whose transition probability tensor is 4∗20∗4∗20∗4∗20. Fig. 11 illustrates the convergence trend of UP-PM when adopting various adjustment factors α. It shows that the number of iterations will increase as the α value increases. We can see that the convergence is consistent with the analysis in Theorem 6. Meanwhile, Table III exhibits the number of iterations of UP-PM under different α value for the two Markov models. It shows that the number of iterations will increase slightly as the size of transition probability tensor increases. For instance, just three more times of iterations are required as the size of transition tensor increases from 20∗20∗20 to 4∗20∗4∗20∗4∗20 when $\alpha = 0.8$.

Therefore, from the extensive experimental results, it is clear that the proposed 2M Markov model and SJE based multi-modal prediction approach can obtain excellent prediction

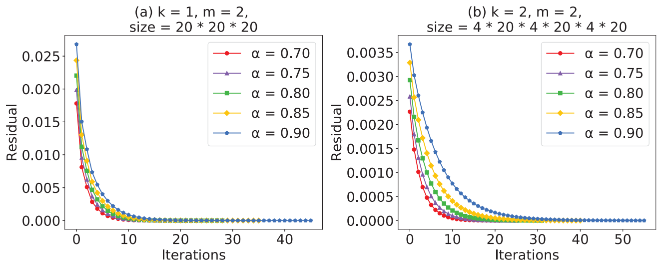


Fig. 11. Convergence trend of UP-PM under various α values.

performance for network traffics, which is conducive to improving the Quality of Service in network management system. The proposed approaches will play a significant role to data-driven network management in the network big data era.

VII. Conclusion

To realize accurate future predictions, this paper proposes a general multivariate multi-order Markov model and a SJE based multi-modal prediction approach. First, we propose two new useful tensor operations including tensor join and unified product, which will play an important role in tensor-based data analysis. Based on the unified product, we develop a general 2M Markov model with its UP-based transition. Meanwhile, the multi-step transition tensor for a 2M Markov model is presented. Afterwards, an UP-based power method is proposed to calculate the stationary joint probability distribution tensor and further implement the SJE based multi-modal prediction. Extensive experimental results based on real-world network traffic datasets demonstrate that the proposed SJE based approach has distinct superiority in prediction accuracy compared with other state-of-the-art approaches. By exploiting the accurate multi-modal prediction approach, we are capable of providing right service in right location at right time. These accurate prediction services can significantly improve the efficiency of network traffic management. In fact, the proposed prediction approaches can also be applied to other domains as long as we construct a suitable Markov model according to practical requirements, e.g., location-aware trajectory prediction, social network application, targeted advertisement delivery, accurate trend prediction, etc.

However, there is a trade-off between the prediction accuracy and storage in the SJE based approach, since the stationary joint eigentensor will consume more storage space. In the future, we shall study how to improve the computation efficiency by adopting sparse representation or exploiting tensor decomposition. Beside, since network data are generated in a streaming way, we shall further study an incremental approach to calculate the stationary joint eigentensor.

References

[1] H. Liu, L. T. Yang, M. Lin, D. Yin, and Y. Guo, "A tensor-based holistic edge computing optimization framework for Internet-of-Things," *IEEE Netw.*, vol. 1, no. 32, pp. 88–95, Jan./Feb. 2018.

[2] H. Liu *et al.*, "A holistic optimization framework for mobile cloud task scheduling," *IEEE Trans. Sustain. Comput.*, vol. 4, no. 2, pp. 217–230, Apr./Jun. 2019. doi: 10.1109/TSUSC.2017.2765520.

[3] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, "Network function virtualization: Challenges and opportunities for innovations," *IEEE Commun. Mag.*, vol. 53, no. 2, pp. 90–97, Feb. 2015.

[4] L. Kuang, L. T. Yang, X. Wang, P. Wang, and Y. Zhao, "A tensor-based big data model for QoS improvement in software defined networks," *IEEE Netw.*, vol. 30, no. 1, pp. 30–35, Jan./Feb. 2016.

[5] W. Borjigin, K. Ota, and M. Dong, "In broker we trust: A double-auction approach for resource allocation in NFV markets," *IEEE Trans. Netw. Service Manag.*, vol. 15, no. 4, pp. 1322–1333, Dec. 2018.

[6] J. Zhang, Y. Xiang, Y. Wang, W. Zhou, Y. Xiang, and Y. Guan, "Network traffic classification using correlation information," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 1, pp. 104–117, Jan. 2013.

[7] A. Laghrissi, T. Taleb, and M. Bagaa, "Conformal mapping for optimal network slice planning based on canonical domains," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 519–528, Mar. 2018.

[8] W. Xiong *et al.*, "Anomaly secure detection methods by analyzing dynamic characteristics of the network traffic in cloud communications," *Inf. Sci.*, vol. 258, pp. 403–415, Feb. 2014.

[9] Y. Wang, Y. Xiang, J. Zhang, W. Zhou, G. Wei, and L. T. Yang, "Internet traffic classification using constrained clustering," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 11, pp. 2932–2943, Nov. 2014.

[10] P. Li, Z. Chen, L. T. Yang, Q. Zhang, and M. J. Deen, "Deep convolutional computation model for feature learning on big data in Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 14, no. 2, pp. 790–798, Feb. 2017.

[11] Q. Zhang, L. T. Yang, Z. Chen, and P. Li, "High-order possibilistic c-means algorithms based on tensor decompositions for big data in IoT," *Inf. Fusion*, vol. 39, pp. 72–80, Jan. 2018.

[12] H. Liu, L. T. Yang, Y. Guo, X. Xie, and J. Ma, "An incremental tensor train decomposition for cyber-physical-social big data," *IEEE Trans. Big Data*, to be published. doi: 10.1109/TBDATA.2018.2867485.

[13] F. Z. Yousaf, M. Bredel, S. Schaller, and F. Schneider, "NFV and SDN—Key technology enablers for 5G networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2468–2478, Nov. 2017.

[14] H. Liu, L. T. Yang, J. Ding, Y. Guo, and S. S. Yau, "Tensor-train-based high order dominant Eigen decomposition for multi-modal prediction services," *IEEE Trans. Eng. Manag.*, to be published. doi: 10.1109/TEM.2019.291292.

[15] V. Alarcon-Aquino and J. A. Barria, "Multiresolution FIR neural-network-based learning algorithm applied to network traffic prediction," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 36, no. 2, pp. 208–220, Mar. 2006.

[16] J. Zhang, C. Chen, Y. Xiang, W. Zhou, and Y. Xiang, "Internet traffic classification by aggregating correlated naive Bayes predictions," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 1, pp. 5–15, Jan. 2013.

[17] R. Boutaba *et al.*, "A comprehensive survey on machine learning for networking: Evolution, applications and research opportunities," *J. Internet Services Appl.*, vol. 9, no. 1, p. 16, 2018.

[18] Y. Xie, J. Hu, S. Tang, and X. Huang, "A forward-backward algorithm for nested hidden semi-Markov model and application to network traffic," *Comput. J.*, vol. 56, no. 2, pp. 229–238, Sep. 2012.

[19] Y. Xie, J. Hu, Y. Xiang, S. Yu, S. Tang, and Y. Wang, "Modeling oscillation behavior of network traffic by nested hidden Markov model with variable state-duration," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 9, pp. 1807–1817, Sep. 2013.

[20] S. Qiao, D. Shen, X. Wang, N. Han, and W. Zhu, "A self-adaptive parameter selection trajectory prediction approach via hidden Markov models," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 284–296, Feb. 2015.

[21] D. Tran, W. Sheng, L. Liu, and M. Liu, "A hidden Markov model based driver intention prediction system," in *Proc. IEEE Int. Conf. Cyber Technol. Autom. Control Intell. Syst.*, 2015, pp. 115–120.

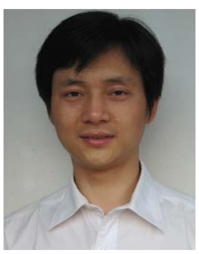
[22] L. A. Kelley, S. Mezulis, C. M. Yates, M. N. Wass, and M. J. Sternberg, "The Phyre2 Web portal for protein modeling, prediction and analysis," *Nat. Protocols*, vol. 10, no. 6, pp. 845–858, 2015.

[23] L. Kuang, L. T. Yang, S. C. Rho, Z. Yan, and K. Qiu, "A tensor-based framework for software-defined cloud data center," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 12, no. 5s, p. 74, 2016.

[24] P. Wang, L. T. Yang, Y. Peng, J. Li, and X. Xie, "M^2T^2: The multivariate multi-step transition tensor for user mobility pattern prediction," *IEEE Trans. Netw. Sci. Eng.*, to be published. doi: 10.1109/TNSE.2019.2913669.

[25] W.-K. Ching and M. K. Ng, *Markov Chains: Models, Algorithms and Applications* (International Series in Operations Research and Management Science), vol. 83. Heidelberg, Germany: Springer, 2006.

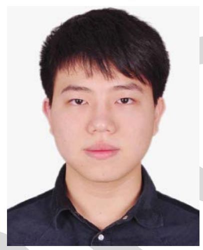
[26] J. Kwak, C.-H. Lee, and D. Y. Eun, "A high-order Markov-chain-based scheduling algorithm for low delay in CSMA networks," *IEEE/ACM Trans. Netw.*, vol. 24, no. 4, pp. 2278–2290, Aug. 2016.
[27] W. Sha, Y. Zhu, M. Chen, and T. Huang, "Statistical learning for anomaly detection in cloud server systems: A multi-order Markov chain framework," *IEEE Trans. Cloud Comput.*, vol. 6, no. 2, pp. 401–413, Apr./Jun. 2018.
[28] S. Hu and L. Qi, "Convergence of a second order Markov chain," *Appl. Math. Comput.*, vol. 241, no. 3, pp. 183–192, 2014.
[29] D. F. Gleich, L.-H. Lim, and Y. Yu, "Multilinear PageRank," *SIAM J. Matrix Anal. Appl.*, vol. 36, no. 4, pp. 1507–1541, 2015.
[30] W. Li and M. K. Ng, "On the limiting probability distribution of a transition probability tensor," *Linear Multilinear Algebra*, vol. 62, no. 3, pp. 362–385, 2014.
[31] H. Bozorgmanesh and M. Hajarian, "Convergence of a transition probability tensor of a higher-order Markov chain to the stationary probability vector," *Numer. Linear Algebra Appl.*, vol. 23, no. 6, pp. 972–988, 2016.
[32] L. Qi, "Eigenvalues of a real supersymmetric tensor," *J. Symbolic Comput.*, vol. 40, no. 6, pp. 1302–1324, 2005.
[33] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.
[34] A. Cichocki, N. Lee, I. V. Oseledets, A. H. Phan, Q. Zhao, and D. Mandic, "Low-rank tensor networks for dimensionality reduction and large-scale optimization problems: Perspectives and challenges part 1," *Found. Trends® Mach. Learn.*, vol. 9, nos. 4–5, pp. 249–429, 2016.
[35] M. Brazell, N. Li, C. Navasca, and C. Tamon, "Solving multilinear systems via tensor inversion," *SIAM J. Matrix Anal. Appl.*, vol. 34, no. 2, pp. 542–570, 2013.
[36] T. H. Haveliwala, "Topic-sensitive PageRank: A context-sensitive ranking algorithm for Web search," *IEEE Trans. Knowl. Data Eng.*, vol. 15, no. 4, pp. 784–796, Jul./Aug. 2003.



Huazhong Liu received the B.S. degree in computer science from the School of Computer Science and Technology, Jiangxi Normal University, Nanchang, China, in 2004, and the M.S. degree in computer science from the College of Mathematics and Computer Science, Hunan Normal University, Changsha, China, in 2009. He is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China. He is also a Lecturer with the School of Information Science and Technology, Jiujiang University, Jiujiang, China. His research interests include big data, cloud computing, Internet of Things, and scheduling optimization.

Laurence T. Yang (SM'15) received the B.E. degree in computer science and technology from Tsinghua University, China, and the Ph.D. degree in computer science from the University of Victoria, Canada. He is a Professor with the School of Computer Science and Technology, Huazhong University of Science and Technology, China, and the Department of Computer Science, St. Francis Xavier University, Canada. His research has been supported by the National Sciences and Engineering Research Council, Canada, and the Canada Foundation for Innovation. His research interests include parallel and distributed computing, embedded and ubiquitous/pervasive computing, and big data.

Jinjun Chen (SM'13) received the Ph.D. degree in information technology from the Swinburne University of Technology, Australia, where he is a Professor. He is the Deputy Director of Swinburne Data Science Research Institute, and the Director of Swinburne Big Data Lab. His research results have been published over 130 papers in international journals and conferences, including various IEEE/ACM TRANSACTIONS. His research interests include scalability, big data, data science, software systems, cloud computing, data privacy and security, and related various research topics. He is an Associate Editor of *ACM Computing Surveys*, the IEEE TRANSACTIONS ON BIG DATA, the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, and the IEEE TRANSACTIONS ON CLOUD COMPUTING, as well as other journals, such as the *Journal of Computer and System Sciences*, and the *Journal of Network and Computer Applications*. He is the Chair of the IEEE Computer Society Technical Committee on Scalable Computing.



Minghao Ye received the first B.E. degree in microelectronic science and engineering from Sun Yat-sen University, Guangzhou, China, and the second B.E. degree (Hons.) in electronic engineering from Hong Kong Polytechnic University, Hong Kong, in 2017, the M.S. degree in electrical engineering from New York University, New York, NY, USA, in 2019, where he is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering. His research interests include traffic engineering, software-defined networks, mobile edge computing, and reinforcement learning.

Jihong Ding received the B.S. and M.S. degrees in computer science from Hunan Normal University, Changsha, China, in 2004 and 2009, respectively, and the Ph.D. degree in educational technology from Central China Normal University, Wuhan, China, in 2016. She is currently a Lecture with the Zhejiang University of Technology. Her research interests include educational resource recommendation and big data analysis.

Liwei Kuang received the M.S. degree from the School of Computer Science, Hubei University of Technology, Wuhan, China, and the Ph.D. degree from the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan. He is currently a Software Engineer with FiberHome Telecommunication Technologies Company Ltd., Wuhan. His main research interests include software defined networks, network function virtualization, cloud computing, and big data.