

HOMC: A MATLAB Package for Higher Order Markov Chains

Jianhong Xu*

October 6, 2025

Abstract

We present a MATLAB package, which is the first of its kind, for Higher Order Markov Chains (HOMC). It can be used to easily compute all important quantities in our recent works relevant to higher order Markov chains, such as the k -step transition tensor, limiting probability distribution, ever-reaching probability tensor, and mean first passage time tensor. It can also be used to check whether a higher order chain is ergodic or regular, to construct the transition matrix of the associated reduced first order chain, and to determine whether a state is recurrent or transient. A key function in the package is an implementation of the tensor “box” product which has a probabilistic interpretation and is different from other tensor products in the literature. This HOMC package is useful to researchers and practitioners alike for tasks such as numerical experimentation and algorithm prototyping involving higher order Markov chains.

keywords: higher order Markov chains, MATLAB, tensors, transition probabilities, ever-reaching probabilities, limiting probability distributions, mean first passage times, classification of states

1 Introduction

Let $X = \{X_t : t = 1, 2, \dots\}$ be a stochastic process, where t and X_t can be regarded as time and the state of a system at time t , respectively. For

*School of Mathematical and Statistical Sciences, Southern Illinois University Carbondale, Carbondale, IL 62901, USA. Email: jhxu@siu.edu

convenience, the states of the system are often labeled as 1 through n with $n \geq 2$. The set $S = \{1, 2, \dots, n\}$ is called the state space. Then, this process X is called a first order Markov chain if the future state X_{t+1} of the system can be determined by its present state X_t , meaning

$$\Pr(X_{t+1} = i | X_t = j, \dots, X_1 = k) = \Pr(X_{t+1} = i | X_t = j)$$

for all $t \geq 1$ and $i, j, \dots, k \in S$. In addition, the chain is called homogeneous if the above probability depends only on i and j but is independent of t . Denote $p_{ij} = \Pr(X_{t+1} = i | X_t = j)$, which is called the transition probability from states j to i . The n by n matrix $P = [p_{ij}]$ is called the transition matrix of the chain. The matrix P is stochastic, that is, $0 \leq p_{ij} \leq 1$ for all $i, j \in S$ and $\sum_{i \in S} p_{ij} = 1$ for all $j \in S$.

On the other hand, a higher order Markov chain arises if the future state of the system depends on not only its present state but also one or more of its past states. Specifically, on letting $m \geq 3$, the process X is called a higher order Markov chain of order $m - 1$, or simply an $(m - 1)$ th order Markov chain, if

$$\begin{aligned} \Pr(X_{t+1} = i_1 | X_t = i_2, \dots, X_{t-m+2} = i_m, \dots, X_1 = i_{t+1}) \\ = \Pr(X_{t+1} = i_1 | X_t = i_2, \dots, X_{t-m+2} = i_m) \end{aligned} \quad (1)$$

for all $t \geq m - 1$ and $i_1, i_2, \dots, i_m, \dots, i_{t+1} \in S$. In particular, such a higher order chain reduces to the first order case if $m = 2$. Similar to the first order one, the higher order chain is called homogeneous if the probability in (1) depends only on $i_1, i_2, \dots, i_m$ yet is independent of t . Denote $p_{i_1 i_2 \dots i_m} = \Pr(X_{t+1} = i_1 | X_t = i_2, \dots, X_{t-m+2} = i_m)$, which is called the transition probability from states $(i_2, \dots, i_m)$ to i_1 . The m th order, n dimensional tensor $\mathcal{P} = [p_{i_1 i_2 \dots i_m}]$ is called the transition tensor of the higher order chain. This tensor $\mathcal{P}$ is stochastic too in the sense that $0 \leq p_{i_1 i_2 \dots i_m} \leq 1$ for all $i_1, i_2, \dots, i_m \in S$ and $\sum_{i_1 \in S} p_{i_1 i_2 \dots i_m} = 1$ for all $i_2, \dots, i_m \in S$. Throughout this article, a Markov chain of any order is always assumed to be homogeneous and is often referred to simply as a chain.

Higher order chains have been the focus of many recent works, see [2, 6, 8, 9, 10, 11, 12, 18, 19, 26] and the references therein. They have also appeared in a variety of applications [1, 3, 5, 7, 16, 13, 17, 20, 22, 24, 27, 28]. To the best of our knowledge, however, there is no dedicated MATLAB package for higher order chains. The goal of this article is to present for the first time such a package, named by the acronym HOMC for Higher Order Markov Chains.

MATLAB is a high-level computing environment that has been widely adopted by researchers and practitioners in various fields as a powerful tool for numerical experiment, simulation, and visualization. When it comes to higher order chains, MATLAB is particularly attractive because of its capability of dealing with multidimensional data such as the transition tensor $\mathcal{P}$. To store $\mathcal{P}$ in MATLAB, for example, we can easily achieve that by assigning the respective transition probabilities to each frontal slice [21] $\mathcal{P}(:, :, i_3, \dots, i_m)$, where $i_3, \dots, i_m \in S$. These frontal slices are n by n matrices and hence can be conveniently handled and manipulated in MATLAB.

In our HOMC package, a key function is an implementation of the $\boxtimes$, or “box”, product of m th order, n dimensional tensors [9, 11]. This operation has a useful probabilistic interpretation and is different from other types of existing tensor multiplications [15, 23]. The $\boxtimes$ product may be directly implemented by first constructing all $(i_1, i_2, \dots, i_m)$ for $i_1, i_2, \dots, i_m \in S$. This approach clearly requires a huge amount of storage memory and, therefore, becomes impractical when the size of the chain is large. Accordingly, we have taken a different approach by exploiting MATLAB’s built-in function `ind2sub` to generate $(i_1, i_2, \dots, i_m)$ on the fly. The same idea has been used across the other functions in the package when necessary so as to cut on their memory usage. Overall, the main storage utilization of the package is dominated by that of the transition tensor $\mathcal{P}$ and, if applicable, output tensors of the same size. Thus, its space complexity is $O(n^m)$. This is the best that can be accomplished in coping with a dense tensor problem such as general higher order chains.

As seen in their definitions, a higher order chain reduces to a first order one when $m = 2$. Consistent with that, our HOMC package can also deal with first order chains. Our focus in this article, however, is on the higher order case. It should be pointed out that not all questions related to higher order chains can be addressed through first order chains [9, 11]. Consequently, the HOMC package is not superseded by any software tools that can only handle first order chains.

The MATLAB source files of the HOMC package are available publicly at <https://neumann.math.siu.edu/homc>.

2 Description of Main Functions in HOMC Package

Now, we describe the main functions in the HOMC package along with necessary and brief background information. These functions are for general purposes and are useful in many situations. We also provide examples to illustrate how these functions can be used.

2.1 Linear Indexing

Linear indexing [21] is one way of ordering $(i_1, i_2, \dots, i_m)$ with $i_1, i_2, \dots, i_m \in S$. Simply put, it means that we run these indices from 1 through n in the precedence from left to right. The reversed linear indexing, on the other hand, means in the precedence from right to left. The function for these tasks is

$$\text{ind} = \text{lind}(\mathbf{s}, \mathbf{r}),$$

where $\mathbf{s}$ is the row vector

$$\mathbf{s} = [\underbrace{n \ n \ \dots \ n}_m] \quad (2)$$

and $\mathbf{r}$ is optional unless the reversed linear indexing is needed. Observe that for an $(m-1)$ th order chain on state space $S = \{1, 2, \dots, n\}$, $\mathbf{s}$ is exactly the size of its transition tensor $\mathcal{P}$ and can be easily obtained in MATLAB by $\mathbf{s} = \text{size}(\mathcal{P})$. The default value of $\mathbf{r}$ is 1, which leads to the linear indexing of $(i_1, i_2, \dots, i_m)$ for $i_1, i_2, \dots, i_m \in S$. The reversed linear indexing is achieved by setting $\mathbf{r}$ to -1 .

EXAMPLE 2.1 Let $s = [2 \ 2 \ 2 \ 2]$. Then, $\text{ind} = \text{lind}(\mathbf{s})$ gives:

1	1	1	1
2	1	1	1
1	2	1	1
2	2	1	1
1	1	2	1
2	1	2	1
1	2	2	1
2	2	2	1

1	1	1	2
2	1	1	2
1	2	1	2
2	2	1	2
1	1	2	2
2	1	2	2
1	2	2	2
2	2	2	2

while `ind=lind(s,-1)` gives:

1	1	1	1
1	1	1	2
1	1	2	1
1	1	2	2
1	2	1	1
1	2	1	2
1	2	2	1
1	2	2	2
2	1	1	1
2	1	1	2
2	1	2	1
2	1	2	2
2	2	1	1
2	2	1	2
2	2	2	1
2	2	2	2

Clearly, the reversed linear indexing can be obtained from linear indexing by MATLAB's `fliplr`. This is what `lind` does when `r` is set to `-1`.

2.2 Tensor $\boxtimes$ Product and Powers

Given two m th order, n dimensional tensors $\mathcal{A} = [a_{i_1 i_2 \dots i_m}]$ and $\mathcal{B} = [b_{i_1 i_2 \dots i_m}]$, $\mathcal{A} \boxtimes \mathcal{B}$ is defined [9, 11] to be an m th order, n dimensional tensor $\mathcal{C} = [c_{i_1 i_2 \dots i_m}]$ such that

$$c_{i_1 i_2 \dots i_m} = \sum_{j=1}^n a_{i_1 j i_2 \dots i_{m-1}} b_{j i_2 \dots i_m}$$

for all $1 \leq i_1, i_2, \dots, i_m \leq n$. As a special case, when $\mathcal{A}$ and $\mathcal{B}$ are both n by n matrices, $\mathcal{A} \boxtimes \mathcal{B}$ reduces to the usual matrix multiplication $\mathcal{A}\mathcal{B}$. Unlike its matrix counterpart, however, the $\boxtimes$ product is not associative when $m \geq 3$. In other words, for m th order, n dimensional tensors $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$,

$$\mathcal{A} \boxtimes (\mathcal{B} \boxtimes \mathcal{C}) \neq (\mathcal{A} \boxtimes \mathcal{B}) \boxtimes \mathcal{C}$$

in general.

The function for computing $\mathcal{A} \boxtimes \mathcal{B}$ can be called by

$$\boxed{\mathbf{C}=\mathbf{bprod}(\mathbf{A},\mathbf{B}),}$$

where $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are self-explanatory.

The power of an m th order, n dimensional tensor $\mathcal{A}$ is defined [9, 11] recursively according to

$$\mathcal{A}^{k+1} = \mathcal{A}^k \boxtimes \mathcal{A}, \quad k = 0, 1, 2, \dots,$$

where, in particular, $\mathcal{A}^0$ is the m th order, n dimensional identity tensor $\mathcal{I} = [\delta_{i_1 i_2 \dots i_m}]$ with

$$\delta_{i_1 i_2 \dots i_m} = \begin{cases} 1, & i_1 = i_2; \\ 0, & \text{otherwise.} \end{cases}$$

Clearly, $\mathcal{I} \boxtimes \mathcal{A} = \mathcal{A}$ but, in general, $\mathcal{A} \boxtimes \mathcal{I} \neq \mathcal{A}$. Observe that this identity tensor is different from those in [21, 23].

To compute the k th power of $\mathcal{A}$, we can use the function

$$\boxed{\mathbf{C}=\mathbf{bpow}(\mathbf{A},\mathbf{k}).}$$

We comment that because the $\boxtimes$ product is not associative, $\mathcal{A}^6$, for example, cannot be computed as $\mathcal{A}^3 \boxtimes \mathcal{A}^3$. To form the m th order, n dimensional identity tensor, the command is

$$\boxed{\mathbf{I}=\mathbf{eyet}(\mathbf{s}),}$$

where $\mathbf{s}$ is as in (2).

The $\boxtimes$ product and power play a key role in the study of higher order Markov chains [9, 10, 11]. To begin with, denote $\mathcal{P}^k = [p_{i_1 i_2 \dots i_m}^{(k)}]$, where $\mathcal{P}$ is the transition tensor and $k \geq 1$. Then, $p_{i_1 i_2 \dots i_m}^{(k)}$ is the k -step transition probability from states $(i_2, \dots, i_m)$ to i_1 , that is,

$$p_{i_1 i_2 \dots i_m}^{(k)} = \Pr(X_{t+k} = i_1 | X_t = i_2, \dots, X_{t-m+2} = i_m).$$

Accordingly, $\mathcal{P}^k$ is called the k -step transition tensor. Clearly, $\mathcal{P}^1 = \mathcal{P}$.

EXAMPLE 2.2 Consider a second order chain on $S = \{1, 2, 3, 4\}$ with transition tensor

$$\begin{aligned}\mathcal{P}(:, :, 1) &= \begin{bmatrix} 0.5 & 0 & 0 & 0 \\ 0.5 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathcal{P}(:, :, 2) = \begin{bmatrix} 0 & 0 & 0.5 & 1 \\ 0 & 0.5 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 \\ 0.5 & 0 & 0.5 & 0 \end{bmatrix}, \\ \mathcal{P}(:, :, 3) &= \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0.5 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathcal{P}(:, :, 4) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0.5 \\ 0 & 0 & 0 & 0.5 \end{bmatrix}.\end{aligned}$$

Then, using $\text{bpow}(\mathbf{P}, 10)$, we see

$$\begin{aligned}\mathcal{P}^{10}(:, :, 1) &= \begin{bmatrix} 0.2959 & 0.3750 & 0.2500 & 0.5000 \\ 0.2158 & 0.3750 & 0.1250 & 0.2500 \\ 0.3066 & 0.1250 & 0.3750 & 0.1250 \\ 0.1816 & 0.1250 & 0.2500 & 0.1250 \end{bmatrix}, \\ \mathcal{P}^{10}(:, :, 2) &= \begin{bmatrix} 0.3750 & 0.3350 & 0.1875 & 0.1250 \\ 0.3750 & 0.2959 & 0.3750 & 0.2500 \\ 0.1875 & 0.2158 & 0.3750 & 0.5000 \\ 0.0625 & 0.1533 & 0.0625 & 0.1250 \end{bmatrix}, \\ \mathcal{P}^{10}(:, :, 3) &= \begin{bmatrix} 0.3750 & 0.1250 & 0.3066 & 0.1250 \\ 0.1250 & 0.3750 & 0.2158 & 0.2500 \\ 0.2500 & 0.3750 & 0.2959 & 0.5000 \\ 0.2500 & 0.1250 & 0.1816 & 0.1250 \end{bmatrix}, \\ \mathcal{P}^{10}(:, :, 4) &= \begin{bmatrix} 0.3750 & 0.2949 & 0.2500 & 0.3633 \\ 0.1250 & 0.2168 & 0.1250 & 0.3066 \\ 0.2500 & 0.3066 & 0.3750 & 0.2158 \\ 0.2500 & 0.1816 & 0.2500 & 0.1143 \end{bmatrix}.\end{aligned}$$

Since there exists $k \geq 1$, i.e., $k = 10$ here, such that $\mathcal{P}^k > 0$, this higher order chain is called regular [11]. An important consequence of such regularity is that the higher order chain must have a unique limiting probability distribution [11].

EXAMPLE 2.3 Take a second order chain on $S = \{1, 2, 3\}$ whose transition tensor is

$$\mathcal{P}(:, :, i_3) = \begin{bmatrix} 0 & 0.5 & 0 \\ 1 & 0 & 1 \\ 0 & 0.5 & 0 \end{bmatrix}$$

for $i_3 = 1, 2, 3$. Then, with `bpow(P,k)`, it is easy to check that $\mathcal{P}^k = \mathcal{P}$ for $k = 1, 3, 5, \dots$, and

$$\mathcal{P}(:, :, i_3) = \begin{bmatrix} 0.5 & 0 & 0.5 \\ 0 & 1 & 0 \\ 0.5 & 0 & 0.5 \end{bmatrix}, \quad i_3 = 1, 2, 3,$$

for $k = 2, 4, 6, \dots$. Clearly, this chain is not regular. Given any $i_1, i_2, i_3 \in S$, however, there exists some $k \geq 1$, which usually depends on i_1, i_2, i_3 , such that $p_{i_1 i_2 i_3}^{(k)} > 0$. Such a higher order chain is called ergodic [9]. It is shown in [9] that when a higher order chain is ergodic, its mean first passages times are well-defined.

As mentioned earlier, the HOMC package works for first order chains too. Below is one such example.

EXAMPLE 2.4 Let

$$P = \begin{bmatrix} 0.5 & 0.5 & 0 \\ 0.5 & 0 & 1 \\ 0 & 0.5 & 0 \end{bmatrix}$$

be the transition matrix of a first order chain with three states. Then, from `bpow(P,5)`, we get the 5-step transition matrix of the chain as

$$\begin{bmatrix} 0.3750 & 0.4688 & 0.3125 \\ 0.4688 & 0.2188 & 0.6250 \\ 0.1562 & 0.3125 & 0.0625 \end{bmatrix}.$$

which is the same as the result from `P^5` in MATLAB, the usual way how this matrix is formed in the first order case [14].

2.3 Diagonal Tensors

The identity tensor introduced in the previous subsection is a special diagonal tensor. More generally, if $\mathcal{A} = [a_{i_1 i_2 \dots i_m}]$ is an m th order, n dimensional

tensor, then the diagonal tensor obtained from $\mathcal{A}$ is defined as $\mathcal{A}_d = [a_{i_1 i_2 \dots i_m}^{(d)}]$ such that

$$a_{i_1 i_2 \dots i_m}^{(d)} = \begin{cases} a_{i_1 i_2 \dots i_m}, & i_1 = i_2; \\ 0, & \text{otherwise.} \end{cases}$$

In other words, $\mathcal{A}_d$ is derived from $\mathcal{A}$ by setting all its off-diagonal entries to zero. We comment that the diagonal tensor here is different from those in [15, 23].

To extract the diagonal tensor $\mathcal{A}_d$ from $\mathcal{A}$, we simply call

$$\boxed{\text{D=diagt}(\mathcal{A}).}$$

This function is useful for dealing with problems such as ever-reaching probabilities and mean first passage times as we shall see in Section 3.

2.4 Matricization and Tensorization

In certain situations, it is more convenient to represent a tensor as a matrix or vice versa. These are called matricization and tensorization, respectively. In such conversions, linear indexing is usually adopted for consistency of results.

Suppose that $\mathcal{A} = [a_{i_1 i_2 \dots i_m}]$ is an m th order, n dimensional tensor. Let $N = n^{m-1}$. Then, for any $1 \leq k \leq m$, the mode- k matricization of $\mathcal{A}$ is an n by N matrix $B = [b_1 \ b_2 \ \dots \ b_N]$ in columnwise form such that its columns are the mode- k fibers of $\mathcal{A}$, namely, $\mathcal{A}(i_1, \dots, i_{k-1}, 1 : n, i_{k+1}, \dots, i_m)$, and are arranged via the linear indexing order of $(i_1, \dots, i_{k-1}, i_{k+1}, \dots, i_m)$ [15]. The converse of this process, on the other hand, produces the mode- k tensorization, written as $\mathcal{A}$ for example, of an n by N matrix $B = [b_1 \ b_2 \ \dots \ b_N]$.

The mode- k matricization and its reverse tensorization are done by the functions

$$\boxed{B=\text{t2mat}(\mathcal{A},k)}$$

and

$$\boxed{\mathcal{A}=\text{mat2t}(B,k),}$$

respectively. Observe that the size of B must be n by n^{m-1} for some $n, m \geq 2$. In addition, when $\mathcal{A}$ is an n by n matrix, $\text{t2mat}(\mathcal{A},1)$ gives $\mathcal{A}$ and $\text{t2mat}(\mathcal{A},2)$ gives $\mathcal{A}^T$.

EXAMPLE 2.5 Let $\mathcal{A} = \text{reshape}(1 : 16, 2, 2, 2, 2)$, i.e.,

$$\begin{aligned}\mathcal{A}(:, :, 1, 1) &= \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}, \quad \mathcal{A}(:, :, 2, 1) = \begin{bmatrix} 5 & 7 \\ 6 & 8 \end{bmatrix}, \\ \mathcal{A}(:, :, 1, 2) &= \begin{bmatrix} 9 & 11 \\ 10 & 12 \end{bmatrix}, \quad \mathcal{A}(:, :, 2, 2) = \begin{bmatrix} 13 & 15 \\ 14 & 16 \end{bmatrix}.\end{aligned}$$

Then, by $\mathbf{B} = \text{t2mat}(\mathbf{A}, 3)$, we arrive at the following mode-3 matricization of $\mathcal{A}$:

$$B = \begin{bmatrix} 1 & 2 & 3 & 4 & 9 & 10 & 11 & 12 \\ 5 & 6 & 7 & 8 & 13 & 14 & 15 & 16 \end{bmatrix}.$$

In addition, the command $\mathbf{A} = \text{mat2t}(\mathbf{B}, 3)$ yields back $\mathcal{A}$.

Applications of matricization to the study of higher order chains will be demonstrated soon.

2.5 Reduced First Order Chain

It is well known that a higher order chain can be associated with a first order chain [4]. Even though not all problems regarding a higher order chain can be solved by its associated first order chain [9, 11], there are circumstances where such an associated first order chain is quite useful, especially when it comes to the problem of finding the limiting probability distribution [11].

Given the higher order chain $X = \{X_t : t = 1, 2, \dots\}$ with an m th order, n dimensional transition tensor $\mathcal{P}$ as laid out in the introductory section, define $T = \{i_1 i_2 \dots i_{m-1} : i_1, i_2, \dots, i_{m-1} \in S\}$ and $Y_t = [X_t \ X_{t-1} \ \dots \ X_{t-m+2}]^T$ for $t \geq m-1$. The set T consists of all $N = n^{m-1}$ multi-indices $i_1 i_2 \dots i_{m-1}$ of length $m-1$, ordered according to linear indexing of $(i_1, i_2, \dots, i_{m-1})$. By identifying $Y_t = [i_1 \ i_2 \ \dots \ i_{m-1}]$ with $Y_t = i_1 i_2 \dots i_{m-1}$, the vector-valued stochastic process $Y = \{Y_t : t = m-1, m, \dots\}$ turns out to be a first order chain on state space T [4]. It is called [11] the reduced first order chain obtained from the higher order chain X .

The probability distribution of the chain X at t is defined to be

$$x_t = [\Pr(X_t = 1) \ \Pr(X_t = 2) \ \dots \ \Pr(X_t = n)]^T. \quad (3)$$

Similarly, the probability distribution of the reduced first order chain Y at t is defined to be an N by 1 column vector y_t whose, in multi-index form, $i_1 i_2 \dots i_{m-1}$ th entry is $\Pr(Y_t = i_1 i_2 \dots i_{m-1})$. Then, it is known [19]

$$x_{t+1} = P y_t,$$

where P is the mode-1 matricization of $\mathcal{P}$. This is where matricization in the previous subsection can be useful since it establishes a connection here between the probability distributions x_t and y_t .

Another usage of matricization is for the construction of the N by N transition matrix Q of the reduced first order chain Y , see [11]. We give a brief description as follows.

Let

$$G = \underbrace{\begin{bmatrix} I_{n^{m-2}} & I_{n^{m-2}} & \dots & I_{n^{m-2}} \end{bmatrix}}_n,$$

where $I_{n^{m-2}}$ is the n^{m-2} by n^{m-2} identity matrix. Then,

$$Q = G * P,$$

where $*$ stands for the columnwise Khatri-Rao product [25].

Accordingly, there are two functions related to the formation of Q , one is

$$\boxed{\text{krprod}(A,B),}$$

which computes the columnwise Khatri-Rao product of matrices A and B , and the other is

$$\boxed{Q=\text{rcmat}(P),}$$

which takes the transition tensor $\mathcal{P}$ as input, calls `t2mat` and `krprod`, and returns the matrix Q . In the special case of a first order chain with transition matrix P , the output from above is the same as P .

EXAMPLE 2.6 *Let us revisit the second order chain in Example 2.2. Us-*

ing $Q = \text{rcmat}(P)$, we obtain

$$Q = \begin{bmatrix} 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5 \end{bmatrix}.$$

It is shown in [11] that when a higher order chain is regular, its limiting probability distribution $\pi = \lim_{t \rightarrow \infty} x_t$, where x_t is given in (3), exists and is unique. In addition, π can be determined by

$$\pi = P^{(0)}y, \quad (4)$$

where $P^{(0)}$ is the mode-1 matricization of the m th order, n dimensional identity tensor and y is any right eigenvector corresponding to the dominant eigenvalue $\lambda = 1$ of Q , normalized such that $y \geq 0$ and $\|y\|_1 = 1$. With MATLAB's `eig` function, we find that $\lambda = 1$ has multiplicity 2 and its eigenspace is spanned by

$$y^{(1)} = [0 \ 0 \ 0.1429 \ 0.1429 \ 0.2857 \ 0 \ 0 \ 0 \ 0 \ 0.2857 \ 0 \ 0 \ 0 \ 0 \ 0.1429 \ 0]^T$$

and

$$y^{(2)} = [0 \ 0.2857 \ 0 \ 0 \ 0 \ 0 \ 0.2857 \ 0 \ 0.1429 \ 0 \ 0 \ 0.1429 \ 0.1429 \ 0 \ 0 \ 0]^T.$$

Because of (4), both $y^{(1)}$ and $y^{(2)}$ lead to

$$\pi = [0.2857 \ 0.2857 \ 0.2857 \ 0.1429]^T.$$

The examples being discussed thus far have demonstrated that the HOMC package, together with a few MATLAB's built-in functions, can be a handy tool in numerical experimentation and quick prototyping on higher order chains.

3 Additional Functions in HOMC Package

Next, let us explore some more advanced and specialized functions in the HOMC package. We continue to use the settings in the introductory section for the higher order chain X on state space $S = \{1, 2, \dots, n\}$ with an m th order, n dimensional transition tensor $\mathcal{P} = [p_{i_1 i_2 \dots i_m}]$.

3.1 Ever-Reaching Probabilities

Let $\eta_{i_1 i_2 \dots i_m} = \min\{j \geq 1 : X_{m+j-1} = i_1 | X_{m-1} = i_2, \dots, X_1 = i_m\}$ be the random variable of the first passage time from states $(i_2, \dots, i_m)$ to i_1 . The probability that such a first passage occurs at $j = k$ is denoted by $f_{i_1 i_2 \dots i_m}^{[k]} = \Pr(\eta_{i_1 i_2 \dots i_m} = k)$. It is known [9] that $f_{i_1 i_2 \dots i_m}^{[1]} = p_{i_1 i_2 \dots i_m}$ and

$$f_{i_1 i_2 \dots i_m}^{[k+1]} = \sum_{j \in S, j \neq i_1} f_{i_1 j i_2 \dots i_{m-1}}^{[k]} p_{j i_2 \dots i_m}, \quad k = 1, 2, \dots$$

Denote $\mathcal{F}^{[k]} = [f_{i_1 i_2 \dots i_m}^{[k]}]$ as a tensor. It follows that $\mathcal{F}^{[1]} = \mathcal{P}$ and, with the $\boxtimes$ product notation,

$$\mathcal{F}^{[k+1]} = (\mathcal{F}^{[k]} - \mathcal{F}_d^{[k]}) \boxtimes \mathcal{P}, \quad k = 1, 2, \dots$$

The ever-reaching probability tensor $\mathcal{F} = [f_{i_1 i_2 \dots i_m}]$ is defined to be [9]

$$\mathcal{F} = \sum_{k=1}^{\infty} \mathcal{F}^{[k]}, \quad (5)$$

where the summation is in the entrywise sense, exactly in the way how MATLAB handles it.

As an application, the diagonal entries of $\mathcal{F}$ can be used to classify the states of the chain [10]. Specifically, a state i is called recurrent if $f_{iii_3 \dots i_m} = 1$ for all $i_3, \dots, i_m \in S$; otherwise, it is called transient. In addition, a state i is called fully transient if $f_{iii_3 \dots i_m} < 1$ for all $i_3, \dots, i_m \in S$. Incidentally, if

the chain is ergodic, then $\mathcal{F} = \mathcal{E}$, the m th order, n dimensional tensor of all ones [9]. In this case, all the states are known to be recurrent [10].

Based on the function `bprod`, we have implemented (5) in the HOMC package as a function

$$\boxed{\mathbf{F}=\text{erp}(\mathbf{P},\text{tol}),}$$

where `tol` is the tolerance and is optional. The default value of `tol` is 10^{-6} . If for some $k \geq 1$, the largest entry in absolute value of $\mathcal{F}^{[k]}$ is less than `tol`, then `erp` returns the sum of the first k terms in (5) as an approximation to $\mathcal{F}$.

EXAMPLE 3.1 Consider a second order chain on $S = \{1, 2, 3\}$ whose transition tensor is given by

$$\mathcal{P}(:, :, 1) = \begin{bmatrix} 1/2 & 1/3 & 1/2 \\ 1/2 & 1/3 & 0 \\ 0 & 1/3 & 1/2 \end{bmatrix}, \quad \mathcal{P}(:, :, 2) = \begin{bmatrix} 1 & 0 & 1/2 \\ 0 & 1 & 1/2 \\ 0 & 0 & 0 \end{bmatrix},$$

$$\mathcal{P}(:, :, 3) = \begin{bmatrix} 0 & 0 & 1/2 \\ 1 & 0 & 1/2 \\ 0 & 1 & 0 \end{bmatrix}.$$

Using `erp(P,tol)` with `tol`= 10^{-8} and MATLAB's `format long`, we obtain from the first 67 terms in (5) that

$$\mathcal{F}(:, :, 1) = \begin{bmatrix} 0.8333333333313931 & 0.6666666666627862 & 0.999999999941792 \\ 1.0000000000000000 & 1.0000000000000000 & 1.0000000000000000 \\ 0.499999975649230 & 0.499999986536366 & 0.749999988833857 \end{bmatrix},$$

$$\mathcal{F}(:, :, 2) = \begin{bmatrix} 1.0000000000000000 & 0 & 0.999999999941792 \\ 1.0000000000000000 & 1.0000000000000000 & 1.0000000000000000 \\ 0.499999968638406 & 0 & 0.749999988833857 \end{bmatrix},$$

$$\mathcal{F}(:, :, 3) = \begin{bmatrix} 0.6666666666627862 & 0.9999999999883585 & 0.999999999941792 \\ 1.0000000000000000 & 1.0000000000000000 & 1.0000000000000000 \\ 0.499999982660054 & 1.0000000000000000 & 0.749999988833857 \end{bmatrix}.$$

Besides, we see from the diagonal entries of $\mathcal{F}$ that state 1 is transient but not fully transient, state 2 is recurrent, and state 3 is fully transient.

3.2 Mean First Passage Times

Mean first passage times are important characteristics of a chain since they provide useful information about the short-term dynamics of the stochastic process. Recall that the first passage time random variable from states $(i_2, \dots, i_m)$ to i_1 is denoted by $\eta_{i_1 i_2 \dots i_m}$. The mean first passage time from states $(i_2, \dots, i_m)$ to i_1 is defined as [8]

$$\mu_{i_1 i_2 \dots i_m} = E(\eta_{i_1 i_2 \dots i_m}) = \sum_{k=1}^{\infty} k f_{i_1 i_2 \dots i_m}^{[k]}.$$

The m th order, n dimensional tensor $\mu = [\mu_{i_1 i_2 \dots i_m}]$ is called the mean first passage time tensor. It is shown in [9] that when the chain is ergodic, μ is uniquely determined by

$$\mu_{i_1 i_2 \dots i_m} = 1 + \sum_{j \in S, j \neq i_1} \mu_{i_1 j i_2 \dots i_{m-1}} p_{j i_2 \dots i_m}, \quad i_1, i_2, \dots, i_m \in S. \quad (6)$$

With the $\boxtimes$ notation, the above can be written as a tensor equation

$$\mu = \mathcal{E} + (\mu - \mu_d) \boxtimes \mathcal{P}, \quad (7)$$

where $\mathcal{E}$ is the m th order, n dimensional tensor of all ones. We comment that $\mathcal{E}$ can be easily formed in MATLAB by `ones(s)` with `s=size(P)`.

The mean first passage time tensor μ can be computed by either a direct method or an iterative method [9]. We now give a brief description of each of these approaches as follows.

To compute μ directly, we treat (6) as a linear system. It is shown in [9] that when the chain is ergodic, this system is nonsingular. Moreover, by arranging the unknown $\mu_{i_1 i_2 \dots i_m}$ in the reversed linear indexing order, the coefficient matrix of the system in (6) has an n by n block diagonal structure, in which the k th diagonal block corresponds to $i_1 = k$ in (6), $k = 1, 2, \dots, n$. This splits (6) naturally into n smaller subsystems. We have implemented this approach by the function

`mu=mfptd(P).`

It takes the transition tensor $\mathcal{P}$ as input, solves each smaller subsystem from (6), and piece together these solutions to form μ . We comment that it is not advisable to solve (6) directly as an entire n^m by n^m system without

exploiting its underlying block diagonal structure, especially when the chain is large.

To compute μ iteratively, we rewrite (7) as

$$\mu^{(k+1)} = \mathcal{E} + (\mu^{(k)} - \mu_d^{(k)}) \boxtimes \mathcal{P}, \quad k = 0, 1, 2, \dots$$

It is shown in [9] that this iterative algorithm is guaranteed to converge when the chain is ergodic. A good choice of $\mu^{(0)}$ is $\mathcal{E}$ since $\mu_{i_1 i_2 \dots i_m} \geq 1$ for any $i_1, i_2, \dots, i_m \in S$ [9]. The algorithm terminates when the largest entrywise difference in absolute value between two successive iterations of μ is within a prescribed tolerance. We have also implemented this approach by the function

$$\boxed{\text{mu=mfpti(P,mu0,tol),}}$$

where $\mathbf{P}$ is the transition tensor, mu0 is the initial approximation to μ , and tol is the tolerance. The default mu0 is $\mathcal{E}$ while the default tol is 10^{-6} . Both mu0 and tol are optional, but both must be specified if any of them needs to be set differently from its default value.

EXAMPLE 3.2 *Now we revisit the ergodic second order chain in Example 2.3. First, we use mu=mfptd(P) to compute μ . The result is*

$$\mu(:, :, i_3) = \begin{bmatrix} 4 & 3 & 4 \\ 1 & 2 & 1 \\ 4 & 3 & 4 \end{bmatrix}, \quad i_3 = 1, 2, 3.$$

To check this answer, according to (7), we compute

$$\mu - \mathcal{E} - (\mu - \mu_d) \boxtimes \mathcal{P}$$

by $\text{mu=ones(s)-bprod(mu-diagt(mu),P)}$ with $\mathbf{s}=\text{size(P)}$, which yields a zero tensor as we expect. Next, we use mu=mfpti(P) to compute μ . The algorithm converges in 40 iterations and the output in format long is

$$\mu(:, :, i_3) = \begin{bmatrix} 3.999997138977051 & 2.999998092651367 & 3.999997138977051 \\ 1.000000000000000 & 2.000000000000000 & 1.000000000000000 \\ 3.999997138977051 & 2.999998092651367 & 3.999997138977051 \end{bmatrix},$$

for $i_3 = 1, 2, 3$.

Finally, let us give one more example to illustrate the applicability of the HOMC package to first order chains. It is known [14] that for an ergodic first order chain with n by n transition matrix P , its mean first passage matrix M satisfies

$$M = E + (M - M_d)P,$$

where E is the n by n matrix of all ones and M_d is the diagonal matrix obtained from M by setting all its off-diagonal entries to zero. As pointed out in [9], however, the above does not yield (7) although the reverse is obviously true. As a matter of fact, not all problems involving higher order chains can be solved by resorting to first order chains. Thus a package such as HOMC cannot be replaced by one that deals only with first order chains.

EXAMPLE 3.3 *Let us revisit the first order chain in Example 2.4. By `M=mfptd(P)`, we obtain*

$$M = \begin{bmatrix} 2.5000 & 3.0000 & 4.0000 \\ 2.0000 & 2.5000 & 1.0000 \\ 6.0000 & 4.0000 & 5.0000 \end{bmatrix}.$$

On the other hand, using `M=mfpti(P)` with the default `tol` and `format long`, we find

$$M = \begin{bmatrix} 2.499999999825377 & 2.999999999767169 & 3.999999999650754 \\ 2.000000000000000 & 2.500000000000000 & 1.000000000000000 \\ 5.999995824987868 & 3.999997419700599 & 4.999996810574538 \end{bmatrix}$$

in 66 iterations.

4 Conclusions

In this article, we have presented a MATLAB package for higher order Markov chains. Coupled with a few built-in functions in MATLAB such as `size`, `ones`, and `eig`, this package can be used to conveniently compute all the important quantities relevant to higher order chains in our recent works [8, 9, 10, 11], including the k -step transition tensor, limiting probability distribution, ever-reaching probability tensor, and mean first passage time tensor. It can also be used to check the ergodicity or regularity of a chain, to construct the transition matrix of the reduced first order chain, and

to classify the states of the chain [10, 11]. These functionalities are useful to the research community in both theoretical investigation on and practical application of higher order Markov chains.

References

- [1] S. Baena-Mirabete, P. Puig, Parsimonious higher order Markov models for rating transitions, *Journal of the Royal Statistical Society Series A: Statistics in Society* 181 (2018): 107–131.
- [2] K. Chang, T. Zhang, On the uniqueness and non-uniqueness of the positive Z -eigenvector for transition probability tensors, *Journal of Mathematical Analysis and Applications* 408 (2013): 525–540.
- [3] V. Deneshkumar, M. Manoprabha, K. Annan, Markov chain modeling for immunological status of HIV patients, *Communications in Statistics: Case Studies, Data Analysis and Applications* 5 (2019): 85–91.
- [4] J. Doob, *Stochastic Processes*, Wiley Classic Library Edition, Wiley, New York, 1990.
- [5] G. Flett, N. Kelly, An occupant-differentiated, higher-order Markov chain method for prediction of domestic occupancy, *Energy and Buildings* 125 (2016): 219–230.
- [6] B. Geiger, A sufficient condition for a unique invariant distribution of a higher-order Markov chain, *Statistics and Probability Letters* 130 (2017): 49–56.
- [7] D. Gleich, L. Lim, Y. Yu, Multilinear pagerank, *SIAM Journal on Matrix Analysis and Applications* 36 (2015): 1507–1541.
- [8] L. Han, K. Wang, J. Xu, Higher order ergodic Markov chains and first passage times, *Linear and Multilinear Algebra* 70 (2022): 6772–6779.
- [9] L. Han, J. Xu, Ever-reaching probabilities and mean first passage times of higher order ergodic Markov chains, *Linear and Multilinear Algebra* 72 (2024): 59–75.

- [10] L. Han, J. Xu, On classification of states in higher order Markov chains, *Linear Algebra and Its Applications* 685 (2024): 24–45.
- [11] L. Han, J. Xu, On limiting probability distributions of higher order Markov chains, under review, [arXiv.2506.08874v2](#).
- [12] S. Hu, L. Qi, Convergence of a second order Markov chain, *Applied Mathematics and Computation* 241 (2014): 183–192.
- [13] M. Islam, R. Chowdhury, A higher order Markov model for analyzing covariate dependence, *Applied Mathematical Modelling* 30 (2006): 477–488.
- [14] J. Kemeny, J. Snell, *Finite Markov Chains*, Springer-Verlag, New York, 1960.
- [15] T. Kolda, B. Bader, Tensor decompositions and applications, *SIAM Review* 51 (2009): 455–500.
- [16] D. Ky, L. Tuyen, A higher order Markov model for time series forecasting, *International Journal of Applied Mathematics and Statistics* 57 (2018): 1–18.
- [17] J. Lan, X. Li, V. Jilkov, C. Mu, Second-order Markov chain based multiple-model algorithm for maneuvering target tracking, *IEEE Transactions on Aerospace and Electronic Systems* 49 (2013): 3–19.
- [18] C. Li, S. Zhang, Stationary probability vectors of higher-order Markov chains, *Linear Algebra and Its Applications* 473 (2016): 114–125.
- [19] W. Li, M. Ng, On the limiting probability distribution of a transition probability tensor, *Linear and Multilinear Algebra* 62 (2014): 362–385.
- [20] Z. Liu, Y. Luo, Y. Zhu, State estimation for linear dynamic system with multiple-step random delays using high-order Markov chain, *IEEE Access* 8 (2020): 76218–76227.
- [21] C. Martin, R. Shafer, B. Larue, An order- p tensor factorization with applications in imaging, *SIAM Journal on Scientific Computing* 35 (2013): 474–490.

- [22] N. Masseran, Markov chain model for the stochastic behaviors of wind-direction data, *Energy Conversion and Management* 92 (2015): 266–274.
- [23] L. Qi, Z. Luo, *Tensor Analysis: Spectral Theory and Special Tensors*, SIAM, Philadelphia, 2017.
- [24] M. Sanjari, H. Gooi, Probabilistic forecast of PV power generation based on higher order Markov chain, *IEEE Transactions on Power Systems* 32 (2017): 2942–2952.
- [25] A. Smilde, R. Bro, P. Geladi, *Multi-Way Analysis: Applications in the Chemical Sciences*, Wiley, West Sussex, UK, 2004.
- [26] S. Wu, M. Chu, Markov chains with memory, tensor formulation, and the dynamics of power iteration, *Applied Mathematics and Computation* 303 (2017): 226–239.
- [27] H. Xiong, R. Mamon, A higher-order Markov chain-modulated model for electricity spot-price dynamics, *Applied Energy* 233 (2019): 495–515.
- [28] Y. Yang, H. Jang, B. Kim, A hybrid recommender system for sequential recommendation: combining similarity models with Markov chains, *IEEE Access* 8 (2020): 190136–190146.