Kronecker Product Approximation with Multiple Factor Matrices via the Tensor Product Algorithm

King Keung Wu*, Yeung Yam*, Helen Meng† and Mehran Mesbahi‡
*Department of Mechanical and Automation Engineering, The Chinese Univeristy of Hong Kong
Email: {kkwu, yyam}@mae.cuhk.edu.hk
†Department of Systems Engineering and Engineering Management, The Chinese Univeristy of Hong Kong
Email: hmmeng@se.cuhk.edu.hk
‡Department of Aeronautics and Astronautics, University of Washington
Email: mesbahi@aa.washington.edu

***Abstract*—Kronecker product (KP) approximation has recently been applied as a modeling and analysis tool on systems with hierarchical networked structure. In this paper, we propose a tensor product-based approach to the KP approximation problem with arbitrary number of factor matrices. The formulation involves a novel matrix-to-tensor transformation to convert the KP approximation problem to a best rank-$(R_1, \ldots, R_N)$ tensor product approximation problem. In addition, we develop an algorithm based on higher-order orthogonal iteration to solve the tensor approximation problem. We prove that the proposed approach is equivalent to conventional singular value decomposition-based approach for two matrix factor case proposed by Van Loan. Hence, our work is a generalization of Van Loan's approach to more than two factor matrices. We demonstrate our approach by several experiments and case studies. The results indicate that the tensor product formulation is effective for KP approximation.**

I. Introduction

Kronecker product (KP) is an important matrix operation which allows a large block matrix to be represented by two or more smaller factor matrices. KP has been studied for decades and has a wide range of applications including signal processing, image processing, and numerical computation [1]. Recently, due to its "fractal" features, such as the self-similarity property, Kronecker product has been applied to model the hierarchically organized networks of complex systems in biology and social networks [2]. It has also been utilized in dynamics and control of large composite networked system analysis [3]. Hence, the KP approach requires efficient Kronecker product decomposition and approximation techniques to obtain the factor systems.

There are a few studies on the Kronecker approximation of matrices. Most notably, Van Loan and Pitsianis [4] proposed a singular value decomposition (SVD)-based algorithm to efficiently find the optimal factor matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ that minimize the Frobenius norm $\|\mathbf{M} - \mathbf{A}_1 \otimes \mathbf{A}_2\|_F$. However, this work did not consider the approximation using more than two factor matrices.

In this paper, we propose a novel tensor product approach which extends the KP approximation to an arbitrary number of factor matrices, by applying a matrix-to-tensor transformation and the higher-order orthogonal iteration (HOOI) algorithm of tensor [5]. Specifically, we consider the Kronecker product approximation problem as follows: Let $\mathbf{M}$ be a $p \times q$ matrix with $p = p_1 p_2 \cdots p_N$ and $q = q_1 q_2 \cdots q_N$. Then the problem considered is finding the matrices $\mathbf{A}_n \in \mathbb{R}^{p_n \times q_n}$ for $n = 1, \ldots, N$ that solve the following minimization problem:

$$\min_{\mathbf{A}_1, \cdots, \mathbf{A}_N} \|\mathbf{M} - \mathbf{A}_1 \otimes \mathbf{A}_2 \otimes \cdots \otimes \mathbf{A}_N\|_F \, , \tag{1}$$

where $\otimes$ and $\|\cdot\|_F$ denote the Kronecker product and the Frobenius norm, respectively. The Kronecker product of matrices $\mathbf{A} \in \mathbb{R}^{p_1 \times q_1}$ and $\mathbf{B} \in \mathbb{R}^{p_2 \times q_2}$ is defined as the $p_1 p_2$-by-$q_1 q_2$ matrix,

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \cdots & a_{1,q_1}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \cdots & a_{2,q_1}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{p_1,1}\mathbf{B} & a_{p_1,2}\mathbf{B} & \cdots & a_{p_1,,q_1}\mathbf{B} \end{pmatrix}. \tag{2}$$

Moreover, the Frobenius norm is defined as:

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^{p_1} \sum_{j=1}^{q_1} a_{ij}^2} \, . \tag{3}$$

The major difference between this work and Van Loan's presented in [4] is that we allow two or more factor matrices in (1).

We first transform the given matrix $\mathbf{M}$ to a $(N+1)^{th}$-order tensor $\mathcal{M} \in \mathbb{R}^{p_N \times q_N \times \, p_1 q_1 \times \ldots \times p_{N-1} q_{N-1}}$ using the proposed matrix-to-tensor algorithm. Then we convert problem (1) into a best rank-$(R_1, \ldots, R_{N+1})$ tensor product approximation problem. We propose to use the HOOI algorithm to solve for the tensor product. Then, by simple reshaping operations, we obtain the solution for (1). We prove that the proposed approach is equivalent to Van Loan's in the two factor matrices case.

The structure of this paper is as follows: Section II provides a brief review on existing tools for the KP approximation problems, including the Van Loan's approach for two factor matrix case and useful tensor product operations. Section III introduces the formulation of the Kronecker product approximation problem by describing the details of our tensor

978-1-5090-1897-0/16/$31.00

production transformation. Section IV shows our numerical results. Finally, section V concludes the paper by discussing possible directions for future work.

II. Related Works

In this section, we briefly introduce the existing tools related to the KP approximation problem. First, we present the KP approximation method for two factor matrices proposed by Van Loan *et al.* [4]. Second, we introduce useful tensor product operations including the *n-mode product* and *higher-order singular value decomposition* (HOSVD) [5]–[10]. Third, we review the *alternating least squares* (ALS)-based algorithm known as *higher-order orthogonal iteration* (HOOI) proposed by De Lathauwer *et al.* [5].

A. Van Loan's method for Kronecker product approximation

For the two factor matrix case, the minimization problem (1) becomes

$$\min_{\mathbf{A}_1,\mathbf{A}_2} \|\mathbf{M}-\mathbf{A}_1\otimes\mathbf{A}_2\|_F . \tag{4}$$

Consider the matrix $\mathbf{M}$ as a p_1p_2-by-q_1q_2 uniform blocking matrix:

$$\mathbf{M}=\begin{pmatrix}\mathbf{M}_{11} & \mathbf{M}_{12} & \cdots & \mathbf{M}_{1,q_1}\\ \mathbf{M}_{21} & \mathbf{M}_{22} & \cdots & \mathbf{M}_{2,q_1}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{M}_{p_1,1} & \mathbf{M}_{p_1,2} & \cdots & \mathbf{M}_{p_1,q_1}\end{pmatrix} \tag{5}$$

where $\mathbf{M}_{ij}\in\mathbb{R}^{p_2\times q_2}$. In [4], Van Loan *et al.* prove that

$$\|\mathbf{M}-\mathbf{A}_1\otimes\mathbf{A}_2\|_F=\left\|\mathcal{R}(\mathbf{M})-\mathbf{vec}(\mathbf{A}_1)\mathbf{vec}(\mathbf{A}_2)^T\right\|_F \tag{6}$$

where $\mathbf{vec}(\mathbf{A})$ is the vectorized matrix $\mathbf{A}$ by stacking its columns, and $\mathcal{R}(\mathbf{M})$ is a *rearrangement* of $\mathbf{M}$ defined by

$$\mathcal{R}(\mathbf{M})=\begin{pmatrix}\mathbf{M}_1\\ \vdots\\ \mathbf{M}_{q_1}\end{pmatrix},\quad \mathbf{M}_j=\begin{pmatrix}\mathbf{vec}(\mathbf{M}_{1,j})^T\\ \vdots\\ \mathbf{vec}(\mathbf{M}_{p_1,j})^T\end{pmatrix}, \tag{7}$$

for $j=1,\ldots,q_1$. Based on (6), the KP approximation (1) for two factor matrices becomes a rank-1 matrix approximation problem, and thus can be solved by matrix SVD [4].

B. Tensor product operations

Here we introduce tensor operations that are useful for our problem formulation. First, let $M\in\mathbb{R}^{I_1\times\cdots\times I_N}$ be an N^{th}-order tensor. We state the definition of n-mode unfolding of tensor M as discussed in [9].

Definition 1. (n-mode unfolding of tensor) The n-mode unfolding of tensor $M\in\mathbb{R}^{I_1\times\cdots\times I_N}$ is a matrix denoted by $\mathbf{M}_{(n)}\in\mathbb{R}^{I_n\times(I_{n+1}\times\cdots\times I_N\times I_1\times\cdots\times I_{n-1})}$ that has tensor element $M(i_1,i_2,\ldots,i_N)$ at index (i_n,j), i.e.,

$$\mathbf{M}_{(n)}(i_n,j)=M(i_1,i_2,\ldots,i_N) \tag{8}$$

where $j=1+\sum_{k=1,k\neq n}^{N}(i_k-1)J_k$ and $J_k=\prod_{m=1,m\neq n}^{k-1}I_m$.

By applying the n-mode unfolding of tensor, we define the n-mode product of tensor which is the multiplication of a tensor by a matrix in mode n.

Definition 2. (n-mode product) The n-mode product of tensor $M\in\mathbb{R}^{I_1\times\cdots\times I_N}$ with a matrix $\mathbf{U}\in\mathbb{R}^{J\times I_n}$, denoted by $M\times_n\mathbf{U}$, is a tensor of size $I_1\times\cdots\times I_{n-1}\times J\times I_{n+1}\times\cdots\times I_N$ with element at index $(i_1,\ldots,i_{n-1},j,i_{n+1},\ldots,i_N)$

$$\begin{aligned}&(M\times_n\mathbf{U})(i_1,\ldots,i_{n-1},j,i_{n+1},\ldots,i_N)\\ &=\sum_{i_n=1}^{I_n}M(i_1,\ldots,i_N)\mathbf{U}(j,i_n).\end{aligned} \tag{9}$$

In terms of unfolded tensors, we have

$$Y=M\times_n\mathbf{U}\Leftrightarrow\mathbf{Y}_{(n)}=\mathbf{U}\mathbf{M}_{(n)}. \tag{10}$$

Note that a matrix $\mathbf{M}=\mathbf{U}\mathbf{S}\mathbf{V}^T$ for matrices $\mathbf{U}$, $\mathbf{S}$ and $\mathbf{V}$, it can also be expressed as $\mathbf{M}=\mathbf{S}\times_1\mathbf{U}\times_2\mathbf{V}$. Moreover, we have

$$(M\times_n\mathbf{U})\times_m\mathbf{V}=(M\times_m\mathbf{V})\times_n\mathbf{U} \tag{11}$$

where $m\neq n$.

The concept of higher-order singular value decomposition was introduced in the earlier works [7], [8] and later in more rigorous formulation [5], [9]. Here we state the theorem of HOSVD:

Theorem 1. *(HOSVD) Any real tensor $M\in\mathbb{R}^{I_1\times I_N}$ can be written in the Tucker form as:*

$$M=S\times_1\mathbf{U}_1\times_2\mathbf{U}_2\times_3\cdots\times_N\mathbf{U}_N \tag{12}$$

where

1) $\mathbf{U}_n\in\mathbb{R}^{I_n,I_n}$ *for $n=1,\ldots,N$ is an orthonormal matrix known as the n-mode singular matrix,*
2) $S\in\mathbb{R}^{I_1\times\cdots\times I_N}$ *is known as the core tensor of which the subtensor $S_{i_n=\alpha}$, obtained by fixing the n^{th} index to α, have the following properties:*
   a) *all-orthogonality:*
   $$\langle S_{i_n=\alpha},S_{i_n=\beta}\rangle=0 \tag{13}$$
   whenever $\alpha\neq\beta$ and for all n.
   b) *ordering:*
   $$\|S_{i_n=1}\|_F\geq\|S_{i_n=2}\|_F\geq\cdots\geq\|S_{i_n=I_n}\|_F\geq 0 \tag{14}$$
   for all possible n.

The operation of HOSVD is based on matrix SVD. The n-mode singular matrix $\mathbf{U}_n$ is just the left singular matrix of the n-mode unfolded matrix of tensor M, i.e.

$$\mathbf{M}_{(n)}=\mathbf{U}_n\mathbf{S}_n\mathbf{V}_n^T \tag{15}$$

for $n=1,\ldots,N$. The core tensor S can then be computed by the following formula:

$$S=M\times_1\mathbf{U}_1^T\times_2\mathbf{U}_2^T\times_3\cdots\times_N\mathbf{U}_N^T. \tag{16}$$

Similar to the property of matrix SVD where the zero or small singular values and the associating singular vectors can be discarded for reduced order modeling, it is possible to obtain a lower-rank tensor using HOSVD. However, this approach does not necessarily lead to the best approximation (in terms of the Frobenius norm) of the given tensor M for the same rank condition [5], [11].

Algorithm 1: HOOI Algorithm

input : Original tensor M of size $I_1 \times \cdots \times I_N$ and desired rank of output $R_1, \ldots, R_N$
output: Tensor S and the matrices $\mathbf{U}_n$ for $n = 1, \ldots, N$

Initialize $\mathbf{U}_n \in \mathbb{R}^{I_n \times R_n}$ for $n = 1, \ldots, N$ using HOSVD
while *not converged* **do**
  for $n = 1, \ldots, N$ **do**
    $V = M \times_1 \mathbf{U}_1^T \times_2 \cdots \times_{n-1} \mathbf{U}_{n-1}^T \times_{n+1} \mathbf{U}_{n+1}^T \times_{n+2} \cdots \times_N \mathbf{U}_N^T$;
    $\mathbf{W} = \arg\max_{\mathbf{W}} \left\| V \times_n \mathbf{W}^T \right\|_F$ subject to $\mathbf{W}^T \mathbf{W} = \mathbf{I}$;
    $\mathbf{U}_n \leftarrow \mathbf{W}$;
  $S \leftarrow M \times_1 \mathbf{U}_1^T \times_2 \mathbf{U}_2^T \times_3 \cdots \times_N \mathbf{U}_N^T$;

C. Higher-order orthogonal iteration

De Lathauwer *et al.* [5] proposed HOOI for the best rank-$(R_1, R_2, \ldots, R_N)$ approximation for tensor, as a complement for HOSVD. The best rank-$(R_1, R_2, \ldots, R_N)$ approximation is formulated as the following minimization problem:

$$\min_{S, \mathbf{U}_1, \cdots, \mathbf{U}_N} \left\| \mathbf{M} - S \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \cdots \times_N \mathbf{U}_N \right\|_F \quad (17)$$

where the Frobenius norm of tensor M is defined as

$$\|M\|_F = \left(\sum_{i_1, i_2, \ldots, i_n} M^2_{i_1, i_2, \ldots, i_n} \right)^{1/2} \quad (18)$$

The procedure of HOOI is shown under Algorithm 1.

III. Formulation of Kronecker Product Approximation Problem with arbitrary number of factor matrices

In this session, we formulate the KP approximation problems by minimizing the Frobenius norm of approximation error. The problem is converted to a tensor product approximation using the proposed matrix-to-tensor transformation.

A. Tensor product-based formulation for 2 matrices case

In order to provide some insights on how to apply tensor product to solve the KP approximation problem, we consider the two factor matrices case as in (4), given the matrix $\mathbf{M} \in \mathbb{R}^{p_1 p_2 \times q_1 q_2}$ that can be written as the uniform blocking matrix as in (5). Then we define a third-order tensor $M \in \mathbb{R}^{p_2 \times q_2 \times p_1 q_1}$ by stacking the block matrices $\mathbf{M}_{ij} \in \mathbb{R}^{p_2 \times q_2}$ along the third dimension. Figure 1 illustrates how to convert the given block matrix $\mathbf{M}$ into a third-order tensor M.

Next, we prove the following theorem that allows us to solve problem (4) using tensor product approximation.

Theorem 2. *The Kronecker approximation problem in (4) is equivalent to the following tensor approximation problem:*

$$\min_{S, \mathbf{u}_3} \left\| M - S \times_3 \mathbf{u}_3 \right\|_F . \quad (19)$$

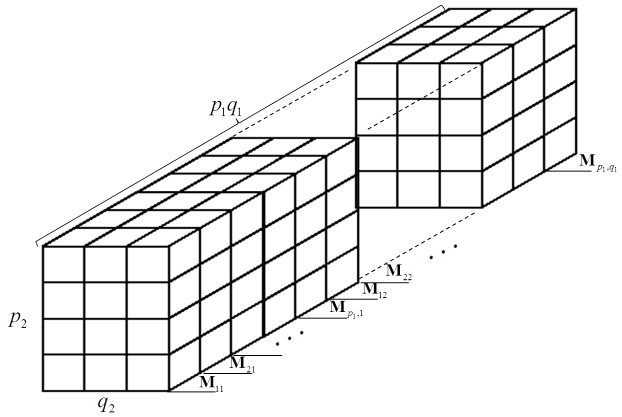

Figure 1. Illustration on how to convert a matrix $\mathbf{M}$ into third-order tensor M for the two factor matrices Kronecker product approximation.

where the tensor $S \in \mathbb{R}^{p_2 \times q_2 \times 1}$ is the matrix $S = \mathbf{A}_2$ and $\mathbf{u}_3 = \mathbf{vec}(\mathbf{A}_1) \in \mathbb{R}^{p_1 q_1 \times 1}$.

Proof. Here we aim to prove the theorem by showing that our method is the same as Van Loan's approach. actually We start from the objective function of (19). By (10), we have

$$\left\| M - S \times_3 \mathbf{u}_3 \right\|_F = \left\| M_{(3)} - \mathbf{u}_3 S_{(3)} \right\|_F . \quad (20)$$

Note that by the definition of n-mode unfolding in Definition 1, $M_{(3)} = \mathcal{R}(\mathbf{M}) \in \mathbb{R}^{p_1 q_1 \times p_2 q_2}$ as defined in (7). Also, $S_{(3)} = \mathbf{vec}(S)^T$. Hence, by letting $S = \mathbf{A}_2$ and $\mathbf{u}_3 = \mathbf{vec}(\mathbf{A}_1)$, we have:

$$\left\| M_{(3)} - \mathbf{u}_3 S_{(3)} \right\|_F = \left\| \mathcal{R}(\mathbf{M}) - \mathbf{vec}(\mathbf{A}_1) \mathbf{vec}(\mathbf{A}_2)^T \right\|_F . \quad (21)$$

By (6) shown by Van Loan in [4], we prove that our formulation is actually the same as Van Loan's, and is thus equivalent to problem (4). ☐

As an alternative to Van Loan's method using matrix SVD to solve (4), we can apply HOOI by considering the problem as the best rank-$(p_2, q_2, 1)$ tensor approximation. We expect the solution of HOOI to be the same as that Van Loan's.

Therefore, by solving (19), we can obtain solutions $\mathbf{A}_1$ and $\mathbf{A}_2$ of (4) by

$$\mathbf{A}_1 = reshape(\mathbf{u}_3, [p_1, q_1]), \quad \mathbf{A}_2 = S; \quad (22)$$

here $reshape(\mathbf{A}, [m, n])$ is a function that reshapes $\mathbf{A}$ into a m-by-n matrix.

B. Extending to N matrix factor case

In this subsection, we extend our matrix-to-tensor transformation in previous subsection to solve the KP approximation for N factor matrices.

Similar to the two matrix case, we first convert the matrix $\mathbf{M}$ into an $(N+1)^{th}$-order tensor $M \in \mathbb{R}^{p_N \times q_N \times p_1 q_1 \times \cdots \times p_{N-1} q_{N-1}}$ as described in Algorithm 2.

Algorithm 2: Convert matrix $\mathbf{M}$ into a tensor M

input : Original matrix $\mathbf{M}$ and dimensions of matrices $\mathbf{A}_1, \ldots, \mathbf{A}_N$ $p_1, q_1, \ldots, p_N, q_N$

output: Tensor M

1. Convert $\mathbf{M}$ to a 3^{rd}-order tensor M_1 of dimension $p_2 \cdots p_N \times q_2 \cdots q_N \times p_1 q_1$ by dividing it into $p_1 \times q_1$ blocks and stacking the blocks along the 3^{rd} dimension;
2. Similar to step 1, convert M_1 to a 4^{th}-order tensor M_2 of dimension $p_3 \cdots p_N \times q_3 \cdots q_N \times p_1 q_1 \times p_2 q_2$ by dividing it into $p_2 \times q_2$ blocks and stacking the blocks along the 4^{th} dimension;
3. Repeat the step $(N-1)$ times until we get a tensor M of dimension $p_N \times q_N \times p_1 q_1 \times \cdots \times p_{N-1} q_{N-1}$;

Then, the minimization problem (1) is equivalent to the following problem:

$$\min_{S, \mathbf{u}_3, \cdots, \mathbf{u}_{N+1}} \| M - S \times_3 \mathbf{u}_3 \times_4 \mathbf{u}_4 \times \cdots \times_{N+1} \mathbf{u}_{N+1} \|_F \tag{23}$$

where $\mathbf{A}_N = S$ and $\mathbf{u}_3, \ldots, \mathbf{u}_{N+1}$ are vectors that can be reshaped to $\mathbf{A}_1, \ldots, \mathbf{A}_{N-1}$, respectively, by the following formula:

$$\mathbf{A}_n = reshape\left(\mathbf{u}_{n+2}, [p_n, q_n]\right) \tag{24}$$

for $n = 1, \ldots, N-1$.

Note that the problem (23) is equivalent to solving the best rank-$(p_N, q_N, 1, \ldots, 1)$ approximation of the tensor M. We apply the HOOI as shown in Algorithm 1 to solve the problem (23). Bader *et al.* from Sandia National Laboratories has kindly provided a Matlab toolbox for HOOI which is available online [12].

IV. Numerical Experiments

We carried out four numerical experiments to study the properties of the proposed methods. We implemented our methods in Matlab and ran the experiments on a personal computer with quad-core CPU of 2.5GHz. For the HOOI algorithm, we applied the tensor toolbox developed by the Sandia National Laboratories [13].

Experiment 1: Two factor matrices in 1,000 trials

In Experiment 1, we compared our solution of problem (1) with the conventional method proposed by Van Loan *et al.* [4]. We first generated 1,000 different random matrices $\mathbf{M}$ of variable size with maximum 100 columns/rows. The entries of the matrices were chosen uniformly at random between 0 and 1. Then we approximated the matrices by Kronecker product of two matrices $\hat{\mathbf{A}}_1$ and $\hat{\mathbf{A}}_2$ using the two algorithms. We compared our method using HOOI with that of Van Loan by computing the mean square difference between the entries of the resulting Kronecker product approximations $\hat{\mathbf{M}} = \hat{\mathbf{A}}_1 \otimes \hat{\mathbf{A}}_2$ by the two algorithms. We found that the average mean square difference is 1.3878×10^{-17} for each trial, which is extremely small and potentially due to numerical errors. Note that although our algorithm is different from Van Loan's, the objective functions are identical. We verified empirically that the two algorithms achieve the same results.

Experiment 2: Three factor matrices case study

In Experiment 2, we illustrated our method using the following example. Given the matrix:

$$\mathbf{M} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 2 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 3 & 10 & 16 & 17 & 18 & 19 & 20 & 21 \\ 4 & 11 & 17 & 22 & 23 & 24 & 25 & 26 \\ 5 & 12 & 18 & 23 & 27 & 28 & 29 & 30 \\ 6 & 13 & 19 & 24 & 28 & 31 & 32 & 33 \\ 7 & 14 & 20 & 25 & 29 & 32 & 34 & 35 \\ 8 & 15 & 21 & 26 & 30 & 33 & 35 & 36 \end{pmatrix}, \tag{25}$$

we aim to approximate it by KP of three 2-by-2 matrices and compare the results of using our method with Van Loan's. Different approximation methods were compared and results were presented as follows. For better understanding, the resulting KP approximation is expressed as $\hat{\mathbf{M}} = \sigma \hat{\mathbf{A}}_1 \otimes \hat{\mathbf{A}}_2 \otimes \hat{\mathbf{A}}_3$ where $\hat{\mathbf{A}}_1$, $\hat{\mathbf{A}}_2$ and $\hat{\mathbf{A}}_3$ are normalized by explicitly extracting the norm σ.

1) Our tensor product method: Directly apply HOOI to solve (1). We transformed $\mathbf{M}$ into a fourth-order tensor $M \in \mathbb{R}^{2 \times 2 \times 4 \times 4}$ by Algorithm 2. Then we solved the best rank-$(2,2,1,1)$ problem using HOOI.
The resulting KP approximation is

$$\begin{aligned}\hat{\mathbf{M}} =& 161.8575 \begin{pmatrix} 0.2435 & 0.4142 \\ 0.4142 & 0.7730 \end{pmatrix} \\ & \otimes \begin{pmatrix} 0.3779 & 0.4832 \\ 0.4832 & 0.6246 \end{pmatrix} \otimes \begin{pmatrix} 0.4394 & 0.4954 \\ 0.4954 & 0.5622 \end{pmatrix}\end{aligned} \tag{26}$$

and $\left\| \mathbf{M} - \hat{\mathbf{M}} \right\|_F = 35.3576$.

2) Van Loan's method #1: Apply Van Loan's method to solve (4) twice. First approximate $\mathbf{M}$ by KP of a 2-by-2 matrix and a 4-by-4 matrix, then approximate the resulting 4-by-4 matrix by two 2-by-2 matrices. The resulting KP approximation is

$$\begin{aligned}\hat{\mathbf{M}} =& 161.8573 \begin{pmatrix} 0.2450 & 0.4144 \\ 0.4144 & 0.7724 \end{pmatrix} \\ & \otimes \begin{pmatrix} 0.3778 & 0.4832 \\ 0.4832 & 0.6247 \end{pmatrix} \otimes \begin{pmatrix} 0.4394 & 0.4954 \\ 0.4954 & 0.5622 \end{pmatrix}\end{aligned} \tag{27}$$

and $\left\| \mathbf{M} - \hat{\mathbf{M}} \right\|_F = 35.3584$.

3) Van Loan's method #2: Apply Van Loan's method to solve (4) twice. First approximate $\mathbf{M}$ by KP of a 4-by-4 matrix and a 2-by-2 matrix, then approximate the

Method	Average Frobenius norm error
Proposed method	**11.2207**
Van Loan's	11.2216

Table I
THE AVERAGE FROBENIUS NORM ERRORS OVER ALL 1,000 TRIALS USING THE TWO METHODS IN EXPERIMENT 3.

resulting 4-by-4 matrix by two 2-by-2 matrices. The resulting KP approximation is

$$\hat{\mathbf{M}} = 161.8542 \begin{pmatrix} 0.2433 & 0.4141 \\ 0.4141 & 0.7733 \end{pmatrix} \otimes \begin{pmatrix} 0.3779 & 0.4832 \\ 0.4832 & 0.6247 \end{pmatrix} \otimes \begin{pmatrix} 0.4443 & 0.4956 \\ 0.4956 & 0.5580 \end{pmatrix} \tag{28}$$

and $\left\|\mathbf{M} - \hat{\mathbf{M}}\right\|_F = 35.3727$.

We found that our method obtains the minimal error norm $\left\|\mathbf{M} - \hat{\mathbf{M}}\right\|_F$ among the three methods. This is reasonable because using Van Loan's methods, it cannot guarantee more than two matrices Kronecker product to minimize the error norm. The errors of Van Loan's methods deviate more from that of our method with more factor matrices. Besides, the choice of different factorization order for Van Loan's method also affect the approximation. Hence, in this case, we verified that our method can simultaneously solve the three matrices KP approximation problem (1) which achieves better results than that of Van Loan's.

Experiment 3: Three factor matrices in 1,000 trials

In this experiment, we empirically verified the efficacy of our tensor product approach by approximating different random matrices using KP of 3 factor matrices for 1,000 trials. We generated a different random matrix $\mathbf{M}$ of variable size for each trial. The sizes of the factor matrices p_n and q_n for $n = 1, \ldots, 3$ were also chosen randomly from integers between 1 to 5. Thus, the maximum possible number of columns and rows was 125. The entries of the matrices were chosen uniformly at random between 0 and 1. In each trial, we compared the error norm $\left\|\mathbf{M} - \hat{\mathbf{M}}\right\|_F$ of the results using the following two methods:

1) Our tensor product method: Directly apply HOOI to solve (1).
2) Van Loan's method: Apply Van Loan's method to solve (4) repeatedly, obtaining $\hat{\mathbf{A}}_n$ in the ascending order of n.

We found that our method achieved the smallest error among the two methods for all 1,000 trials. The average Frobenius norm errors over all 1,000 trials using the two methods were summarized in Table 1.

Experiment 4: Four factor matrices case study

In experiment 4, we studied the KP approximation of a 4 factor matrices with different sizes. Consider the following 24-by-24 matrix:

$$\mathbf{M} = \textbf{reshape}(\mathbf{v}, [24, 24]) = \begin{pmatrix} 1 & 25 & \cdots & 553 \\ 2 & 26 & \cdots & 554 \\ \vdots & \vdots & \ddots & \vdots \\ 24 & 48 & \cdots & 576 \end{pmatrix}, \tag{29}$$

where column vector $\mathbf{v} = (1 \quad 2 \quad \ldots \quad 576)^T$. We approximate $\mathbf{M}$ by $\hat{\mathbf{M}} = \sigma \hat{\mathbf{A}}_1 \otimes \hat{\mathbf{A}}_2 \otimes \hat{\mathbf{A}}_3 \otimes \hat{\mathbf{A}}_4$ where $\hat{\mathbf{A}}_1 \in \mathbb{R}^{2\times 3}$, $\hat{\mathbf{A}}_2 \in \mathbb{R}^{2\times 2}$, $\hat{\mathbf{A}}_3 \in \mathbb{R}^{3\times 2}$ and $\hat{\mathbf{A}}_4 \in \mathbb{R}^{3\times 3}$ are normalized by explicitly extracting the norm σ. We applied three different methods to find the approximation. The results were as shown:

1) Our tensor product method: Directly apply HOOI to solve (1). We transformed $\mathbf{M}$ into a fifth-order tensor $M \in \mathbb{R}^{2\times 2\times 6\times 4\times 6}$ by Algorithm 2. Then solved the best rank-$(2, 2, 1, 1, 1)$ problem using HOOI. The resulting KP approximation is

$$\hat{\mathbf{M}} = 7966.86 \begin{pmatrix} 0.1200 & 0.3536 & 0.5872 \\ 0.1346 & 0.3682 & 0.6018 \end{pmatrix} \otimes \begin{pmatrix} 0.4282 & 0.5557 \\ 0.4361 & 0.5636 \end{pmatrix} \otimes \begin{pmatrix} 0.3794 & 0.4311 \\ 0.3816 & 0.4333 \\ 0.3837 & 0.4354 \end{pmatrix} \otimes \begin{pmatrix} 0.4833 & 0.5149 \\ 0.4846 & 0.5162 \end{pmatrix} \tag{30}$$

and $\left\|\mathbf{M} - \hat{\mathbf{M}}\right\|_F = 629.3953$.

2) Van Loan's method #1: Apply Van Loan's method to solve (4) repeatedly, obtaining $\hat{\mathbf{A}}_n$ in the ascending order of n. The resulting KP approximation is similar to (30), but the $\left\|\mathbf{M} - \hat{\mathbf{M}}\right\|_F = 629.3957$, which is larger than that of our method.
3) Van Loan's method #2: Apply Van Loan's method to solve (4) repeatedly, by first factorizing the matrix $\mathbf{M}$ by two smaller matrices, and then applying the method again for these two smaller matrices to obtain $\hat{\mathbf{A}}_n$ for $n = 1, \ldots, 4$. The resulting KP approximation is also similar to (30), but the $\left\|\mathbf{M} - \hat{\mathbf{M}}\right\|_F = 629.4170$, which is larger than that of our method and Van Loan's method #1.

The results revealed that the proposed method achieves the smallest approximation error comparing with different combinations of Van Loan's method, verifying our claim that our formulation gives the best Kronecker product approximation in term of Frobenius norm.

Experiment 5: Four factor matrices in 1,000 trials

This experiment setup is basically the same as Experiment 3, except we considered 4 factor matrices approximation instead. We generated a different random matrices $\mathbf{M}$ of variable size for each trial. The sizes of the factor matrices p_n and q_n for $n = 1, \ldots, 4$ were also chosen randomly from integers between 1 to 5. Thus, the maximum possible number of columns and rows are 625. The entries of the matrices were chosen uniformly at random between 0 and 1. In each trial, we

Method	Average Frobenius norm error
Proposed method	**39.0003**
Van Loan's # 1	39.0007
Van Loan's # 2	39.0005

Table II
THE AVERAGE FROBENIUS NORM ERRORS OVER ALL 1,000 TRIALS USING THE THREE METHODS IN EXPERIMENT 5.

compared the error norm $\left\|\mathbf{M} - \hat{\mathbf{M}}\right\|_F$ of the results using the three methods as in Experiment 4: our tensor product method, Van Loan's methods #1 and #2.

Similar to Experiment 3, we found that our method achieved the smallest error among the three methods for all 1,000 trials. The average Frobenius norm errors and the total errors per entry over all 1,000 trials using the three methods are summarized in Table 2.

V. CONCLUSIONS

In this paper, we solved the Kronecker product approximation of an arbitrary number of factor matrices. We proposed to use a matrix-to-tensor transformation to convert the original problem to a best rank-$(R_1, \ldots, R_N)$ tensor product approximation problem, where efficient tensor approximation algorithm such as the higher-order orthogonal iteration can subsequently be applied. Numerical experiments showed that the proposed method generally outperform the well-known Van Loan's method for approximation of two factor matrices. Our method can be applied as an analysis tool for a wide range of applications such as networked systems. In future work, we will give the detailed proof for the case of more than two factor matrices. We will also study its applications in different fields, by incorporating additional application-specific constraints. We notice that the proposed matrix-to-tensor transformation algorithm can also be applied to decompose a matrix into sum of Kronecker products.

ACKNOWLEDGMENT

This research is affiliated with the Stanley Ho Big Data Decision Analytics Research Centre of The Chinese University of Hong Kong.

REFERENCES

[1] C. F. Van Loan, "The ubiquitous kronecker product," *Journal of computational and applied mathematics*, vol. 123, no. 1, pp. 85–100, 2000.
[2] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani, "Kronecker graphs: An approach to modeling networks," *The Journal of Machine Learning Research*, vol. 11, pp. 985–1042, 2010.
[3] A. Chapman and M. Mesbahi, "Kronecker product of networked systems and their approximates," in *In 21st International Symposium on Mathematical Theory of Networks and Systems*, 2014, pp. 1426–1431.
[4] C. F. Van Loan and N. Pitsianis, *Approximation with Kronecker products*. Springer, 1993.
[5] L. De Lathauwer, B. De Moor, and J. Vandewalle, "On the best rank-1 and rank-(r_1, r_2,..., r_n) approximation of higher-order tensors," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1324–1342, 2000.
[6] Y. Yam, "Fuzzy approximation via grid point sampling and singular value decomposition," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 27, no. 6, pp. 933–951, 1997.
[7] Y. Yam, P. Baranyi, and C.-T. Yang, "Reduction of fuzzy rule base via singular value decomposition," *Fuzzy Systems, IEEE Transactions on*, vol. 7, no. 2, pp. 120–132, 1999.
[8] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.
[9] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.
[10] P. Baranyi, Y. Yam, and P. Várlaki, *Tensor product model transformation in polytopic model-based control*. CRC Press, 2013.
[11] J. Chen and Y. Saad, "On the tensor svd and the optimal low rank orthogonal approximation of tensors," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 4, pp. 1709–1734, 2009.
[12] B. W. Bader, T. G. Kolda *et al.*, "Matlab tensor toolbox version 2.6," Available online, February 2015. [Online]. Available: http://www.sandia.gov/ tgkolda/TensorToolbox/
[13] B. W. Bader and T. G. Kolda, "Algorithm 862: Matlab tensor classes for fast algorithm prototyping," *ACM Transactions on Mathematical Software (TOMS)*, vol. 32, no. 4, pp. 635–653, 2006.