Two-layer decoupling of multivariate polynomials with coupled ParaTuck and CP decompositions

Konstantin Usevich, Yassine Zniyed, Mariya Ishteva, Philippe Dreesen, André L F de Almeida

Two-layer decoupling of multivariate polynomials with coupled ParaTuck and CP decompositions

Konstantin Usevich, Yassine Zniyed, Mariya Ishteva, Philippe Dreesen, André L. F. de Almeida

***Abstract*—In this paper, we propose a new method for multivariate function approximation that generalized the classical decoupling problem. In the context of neural network, this can be seen as a two-layer feedforward network learning problem. In this work, we make use of both first and second-order information of the original function, modeled through ParaTuck and canonical polyadic (CP) decompositions, respectively. ParaTuck decomposition alone is not sufficient due to lack of reliable algorithms for ParaTuck decomposition. Our approach is a methodological work that demonstrates how the ParaTuck and CP decompositions can be combined in a coupled manner to achieve function decoupling according to the new model. Numerical simulations show the effectiveness of the proposed method on a simple synthetic example, demonstrating its ability to approximate multivariate functions accurately.**

***Index Terms*—tensor decomposition, polynomial decoupling, ParaTuck, neural networks, coupled decompositions.**

I. Introduction

The problem of learning to imitate and approximate complex nonlinear functions is crucial for solving many scientific challenges, including nonlinear system identification [1] and neural network learning [2]. The decoupling problem formulated in [3] and motivated by system identification problems, aims at decomposing a multivariate map as linear combinations of univariate functions in linear forms of the input variables. From the neural network point of view, the decoupling model of [3] corresponds to the usage of trainable (flexible) activation functions, see e.g., [4], [5]. Flexible activation functions attracted recent interest in the machine learning community since they can improve the expressive power of neural networks (compared to fixed activation functions).

Several approaches relying on linear and multilinear algebra [3], [6], [7] have been proposed to find the decoupled representations. The most practically relevant approach of [3] relies on the canonical polyadic decomposition (CPD) [8]–[10] of a third-order tensor constructed from stacking evaluations at different points of the Jacobian matrix of the function. It proved to be useful in many tasks in block-structured nonlinear dynamical system identification [1], [11]. While formulated for the decoupling of polynomial maps, the approach of [3] can be also adapted to a wider class of differentiable functions [12]. However, the main drawback of the decoupling approach of [3] is that it applies only to a single hidden nonlinear layer.

In this paper, we introduce a novel decoupled representation that includes two hidden layers. For the proposed new representation, we show that the Jacobian tensor follows a ParaTuck decomposition (PTD) [13]–[15], and that the Hessian of the multivariate map at a single point follows a CPD. Using these results, we provide an algorithm that is based on a coupled factorization of Jacobian and Hessian tensors, which allows for retrieval of the two-layer decoupled representation (*i.e.,* the weights and the flexible activation functions in the context of neural networks) in the polynomial case.

Related work. In the machine learning literature, tensor decompositions of tensors of higher-order derivatives have been already used to obtain guarantees for recovery of weights [16]. (This idea, in fact, goes back to earlier works in blind source separation [17].) However, most of these results apply to the case of a single hidden nonlinear layers and fixed activation functions. The authors are aware of only one work [18] that treats two-layer architecture, however, that work concerns single-output map, fixed activation functions with biases, and also relies on matrix methods (singular value decomposition), rather than tensor decompositions.

II. Notation and Background

The symbols $(\cdot)^\dagger$ and $\mathrm{rank}(\cdot)$ denote, respectively, the pseudo-inverse and the rank of a matrix. The outer, Hadamard and Khatri-Rao products are denoted to by $\otimes$, $\boxdot$, $\odot$, respectively. Tensors are represented by bold calligraphic capital letters, e.g., $\boldsymbol{\mathcal{X}}$. For an $n_1 \times n_2 \times n_3$ tensor $\boldsymbol{\mathcal{X}}$, the i-th horizontal, j-th lateral and k-th frontal slices are denoted $\boldsymbol{\mathcal{X}}_{i,:,:}$, $\boldsymbol{\mathcal{X}}_{:,j,:}$ and $\boldsymbol{\mathcal{X}}_{:,:,k}$, and are of sizes $n_2 \times n_3$, $n_1 \times n_3$ and $n_1 \times n_2$, respectively. The norm of a tensor $\boldsymbol{\mathcal{X}}$ is the square root of the sum of the squares of all its elements, *i.e.,* $||\boldsymbol{\mathcal{X}}|| = \sqrt{\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{k=1}^{n_3}\mathcal{X}_{i,j,k}}$. The contraction on the kth index of a tensor is denoted as $\underset{k}{\bullet}$ [19], while $\mathrm{diag}(\cdot)$ forms a diagonal matrix from its vector argument or captures the diagonal of its argument matrix. $\mathrm{unfold}_k\boldsymbol{\mathcal{X}}$ refers to the unfolding of tensor $\boldsymbol{\mathcal{X}}$ over its k-th mode [20].

A. CPD and matrix diagonalization

The CP decomposition is a decomposition of a tensor $\boldsymbol{\mathcal{X}}$ of size $n_1 \times n_2 \times n_3$ into a sum of r rank-1 tensors,

$$\boldsymbol{\mathcal{X}} = [\![\lambda;\, \mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \overset{\text{def}}{=} \sum_{k=1}^{r} \lambda_k \mathbf{a}_k \otimes \mathbf{b}_k \otimes \mathbf{c}_k, \qquad (1)$$

Konstantin Usevich, Université de Lorraine, CNRS, Nancy, France, e-mail: konstantin.usevich@cnrs.fr; Yassine Zniyed, Univ. de Toulon, Aix Marseille Univ., CNRS, Toulon, France, e-mail: zniyed@univ-tln.fr; Mariya Ishteva, KU Leuven, Leuven, Belgium, e-mail: mariya.ishteva@kuleuven.be; Philippe Dreesen, Maastricht University (DACS), Maastricht, Netherlands, e-mail: philippe.dreesen@gmail.com; André L. F. de Almeida, Federal University of Ceara, Fortaleza, Brazil, e-mail: andre@gtel.ufc.br.

This research was partially supported by the ANR (Agence Nationale de Recherche) grant LeaFleT (ANR-19-CE23-0021). André L. F. de Almeida acknowledges CNPq for its financial support under grant 312491/2020-4.

with $\lambda \in \mathbb{R}^r$ and the factor matrices $\mathbf{A} \in \mathbb{R}^{n_1 \times r}, \mathbf{B} \in \mathbb{R}^{n_2 \times r}, \mathbf{C} \in \mathbb{R}^{n_3 \times r}$ are given by $\mathbf{A} = \begin{bmatrix}\mathbf{a}_1 \cdots \mathbf{a}_r\end{bmatrix}$, $\mathbf{B} = \begin{bmatrix}\mathbf{b}_1 \cdots \mathbf{b}_r\end{bmatrix}$, $\mathbf{C} = \begin{bmatrix}\mathbf{c}_1 \cdots \mathbf{c}_r\end{bmatrix}$. Without loss of generality, we can omit λ in (1) and decompose $\boldsymbol{\mathcal{X}}$ as

$$\boldsymbol{\mathcal{X}} \overset{\text{def}}{=} [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] = \sum_{k=1}^{r} \mathbf{a}_k \otimes \mathbf{b}_k \otimes \mathbf{c}_k.$$

Note that the frontal, vertical, and horizontal slices of the third-order tensor $\boldsymbol{\mathcal{X}}$ in (1) can be viewed as three sets of matrices that can be jointly diagonalized by two factor matrices [21]. For example, for the frontal slices, we have

$$\boldsymbol{\mathcal{X}}_{:,:,k} = \mathbf{A}\,\mathrm{diag}(\mathbf{C}_{k,:})\mathbf{B}^{\mathsf{T}}. \tag{2}$$

B. ParaTuck decomposition

The ParaTuck (PT) decomposition can be seen as two-level generalization of the CPD. The PT model has been proposed in psychometrics literature [13] in 1994 but was not widely used due to a lack of reliable algorithms [15]. However, it has been exploited in wireless communication problems [22], [23], [24], [25] mostly assuming prior knowledge on some factor matrices.

The PT decomposition of a $n_1 \times n_2 \times n_3$ tensor $\boldsymbol{\mathcal{X}}$ is defined throughout its slices, by a pair of ranks (r, s) and five factor matrices, as:

$$\boldsymbol{\mathcal{X}}_{:,:,k} = \mathbf{W}\,\mathrm{diag}(\mathbf{g}_k)\mathbf{F}\,\mathrm{diag}(\mathbf{h}_k)\mathbf{U}^{\mathsf{T}}, \tag{3}$$

with $\mathbf{W} \in \mathbb{R}^{n_1 \times r}$, $\mathbf{U} \in \mathbb{R}^{n_2 \times s}$, $\mathbf{F} \in \mathbb{R}^{r \times s}$, $\mathbf{G} = \begin{bmatrix}\mathbf{g}_1 & \cdots & \mathbf{g}_{n_3}\end{bmatrix} \in \mathbb{R}^{r \times n_3}$, and $\mathbf{H} = \begin{bmatrix}\mathbf{h}_1 & \cdots & \mathbf{h}_{n_3}\end{bmatrix} \in \mathbb{R}^{s \times n_3}$. Note that (3) can be viewed as a multilevel version of joint decomposition of frontal slices (2). Alternatively, we can define the PT decomposition in the Tucker [26] form as:

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{C}} \underset{1}{\bullet} \mathbf{W} \underset{2}{\bullet} \mathbf{U},$$

where $\boldsymbol{\mathcal{C}} \in \mathbb{R}^{r \times s \times n_3}$ is the ParaTuck core tensor given by

$$\mathcal{C}_{ijk} = F_{ij} G_{ik} H_{jk}, \tag{4}$$

or using the Hadamard product as $\boldsymbol{\mathcal{C}} = \mathbf{F} \boxdot_{\{r,s\}} \boldsymbol{\mathcal{S}}$, where $\boldsymbol{\mathcal{S}}$ is of size $r \times s \times n_3$, and admits a CPD such that $\boldsymbol{\mathcal{S}} = \boldsymbol{\mathcal{I}}_{3,n_3} \bullet_1 \mathbf{G}^T \bullet_2 \mathbf{H}^T$. It is worth mentioning that if the factor matrices $\mathbf{A}$ and $\mathbf{B}$ have full column rank and are known, then we can recover the PT core $\boldsymbol{\mathcal{C}}$. Moreover, if $\boldsymbol{\mathcal{C}}$ is known, the factor matrices $\mathbf{F}$, $\mathbf{G}$ and $\mathbf{H}$ can be easily retrieved. Indeed, an elementwise division of two slices is a rank-one matrix

$$\frac{\mathcal{C}_{ijk}}{\mathcal{C}_{ijk'}} = \frac{G_{ik}}{G_{ik'}} \frac{H_{jk}}{H_{jk'}}. \tag{5}$$

To sum up, the main difficulty for computing a PT decomposition is to find the factor matrices $\mathbf{W}$ and $\mathbf{U}$ from $\boldsymbol{\mathcal{X}}$. To the best of the authors' knowledge, there are no reliable algorithms that can find the PT decomposition, unless some factors are fixed. For example, an alternating least squares algorithm, introduced in [15], has been shown to have convergence issues. In most cases where the PT decomposition is used [24], some of the factor matrices are assumed to be known, which simplifies considerably the optimization problem.

C. ParaTuck ambiguities

The PT decomposition can be unique, in the same sense as the CPD, under some mild conditions [14]. This means that it is prone to scaling and permutation ambiguities, *i.e.,* multiple solutions can exist for the same decomposition problem. It has be shown in [14], that if the third-order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ with PT decomposition (3), then there exists a family of alternative decompositions [mix-up of $\mathcal{X}$ and $\mathcal{T}$?]

$$\boldsymbol{\mathcal{T}}_{:,:,k} = \tilde{\mathbf{W}}\,\mathrm{diag}(\tilde{\mathbf{g}_k})\tilde{\mathbf{F}}\,\mathrm{diag}(\tilde{\mathbf{h}_k})\tilde{\mathbf{U}}^{\mathsf{T}}, \tag{6}$$

where

$$\mathbf{W} = \tilde{\mathbf{W}} \cdot (\boldsymbol{\Pi}_{\mathbf{W}} \cdot \boldsymbol{\Lambda}_{\mathbf{W}}), \tag{7}$$

$$\mathbf{U} = \tilde{\mathbf{U}} \cdot (\boldsymbol{\Pi}_{\mathbf{U}} \cdot \boldsymbol{\Lambda}_{\mathbf{U}}), \tag{8}$$

$$\mathbf{F} = (\bar{\boldsymbol{\Lambda}}_{\mathbf{W}} \cdot \boldsymbol{\Lambda}_{\mathbf{W}}^{-1} \cdot \boldsymbol{\Pi}_{\mathbf{W}}^{\mathsf{T}}) \cdot \tilde{\mathbf{F}} \cdot (\boldsymbol{\Pi}_{\mathbf{U}} \cdot \boldsymbol{\Lambda}_{\mathbf{U}}^{-1} \cdot \bar{\boldsymbol{\Lambda}}_{\mathbf{U}}), \tag{9}$$

$$\mathbf{g}_k = (\alpha_k \cdot \bar{\boldsymbol{\Lambda}}_{\mathbf{W}}^{-1} \boldsymbol{\Pi}_{\mathbf{W}}^{\mathsf{T}}) \cdot \tilde{\mathbf{g}}_k, \tag{10}$$

$$\mathbf{h}_k = (\alpha_k^{-1} \cdot \bar{\boldsymbol{\Lambda}}_{\mathbf{U}}^{-1} \boldsymbol{\Pi}_{\mathbf{U}}^{\mathsf{T}}) \cdot \tilde{\mathbf{h}}_k, \tag{11}$$

where $\boldsymbol{\Lambda}_{\mathbf{W}}$, $\boldsymbol{\Lambda}_{\mathbf{U}}$, $\bar{\boldsymbol{\Lambda}}_{\mathbf{W}}$ and $\bar{\boldsymbol{\Lambda}}_{\mathbf{B}}$ are diagonal matrices, $\boldsymbol{\Pi}_{\mathbf{W}}$ and $\boldsymbol{\Pi}_{\mathbf{U}}$ are permutation matrices, and α_k are nonzero scalars. The distinction between the CPD and PT ambiguities lies in the slice-wise ambiguities (coefficients α_k), making the ambiguity management task harder than in the CPD case. However, we will show how to handle this later.

III. Decoupling polynomial functions

The problem of decoupling refers to the representation of a multivariate polynomial function as a linear combination of univariate polynomials in terms of the input variables (Fig. 1). In this paper, we take the classical decoupling problem one step further and generalize the representation to a two-layer model. By doing so, we aim to enhance the versatility and expressive power of the decoupling technique and extend its range of applications.

A. Reminder: one-layer structure

Let $\mathbf{f} : \mathbb{R}^m \to \mathbb{R}^n$ be a multivariate polynomial map $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}) \cdots f_n(\mathbf{x})]^{\mathsf{T}}$, with $\mathbf{x} = [x_1 \cdots x_m]^{\mathsf{T}}$. It is said that $\mathbf{f}$ has a decoupled representation (Fig. 1), if

$$\mathbf{f}(\mathbf{x}) = \mathbf{W}\mathbf{g}(\mathbf{V}^{\mathsf{T}}\mathbf{x}), \tag{12}$$

where $\mathbf{W} \in \mathbb{R}^{n \times r}$, $\mathbf{V} \in \mathbb{R}^{m \times r}$ are transformation matrices, $\mathbf{w}_k$ and $\mathbf{v}_k$ and are respectively their columns, and $\mathbf{g} : \mathbb{R}^r \to \mathbb{R}^r$ follows $\mathbf{g}(z_1, \cdots, z_r) = \begin{bmatrix}g_1(z_1) & \cdots & g_r(z_r)\end{bmatrix}^{\mathsf{T}}$, with $g_k : \mathbb{R} \to \mathbb{R}$ univariate functions. This decomposition is thus composed of a single layer, containing a transformation matrix $\mathbf{V}$ and a set of univariate functions g_k, followed by a second transformation matrix $\mathbf{W}$.

B. Proposed two-layer structure

This paper extends the decoupled representation in (12) to a representation with two layers as follows (Fig. 2):

$$\mathbf{f}(\mathbf{x}) = \mathbf{W}\mathbf{g}\big(\mathbf{V}^{\mathsf{T}} \cdot \mathbf{h}(\mathbf{U}^{\mathsf{T}}\mathbf{x})\big), \tag{13}$$

where $\mathbf{W} \in \mathbb{R}^{n \times r}$, $\mathbf{V} \in \mathbb{R}^{s \times r}$, $\mathbf{U} \in \mathbb{R}^{m \times s}$, are transformation matrices, and $\mathbf{h} : \mathbb{R}^s \to \mathbb{R}^s$ and $\mathbf{g} : \mathbb{R}^r \to \mathbb{R}^r$ follow

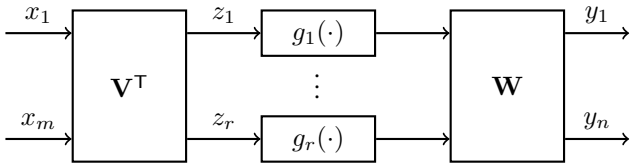

Fig. 1. Decoupled representation of $\mathbf{f}$ into a single-layer model as in (12). This naturally leads to the CPD of the corresponding Jacobian tensor.

$\mathbf{g}(z_1, \cdots, z_r) = [g_1(z_1) \cdots g_r(z_r)]^\mathsf{T}$ and $\mathbf{h}(t_1, \cdots, t_s) = [h_1(t_1) \cdots h_s(t_s)]^\mathsf{T}$, respectively.

This two-layer generalization allows having more flexibility in the decoupling of multivariate nonlinear functions. As we will show next, it is intricately connected to the ParaTuck and CP decompositions when considering first- and second-order information.

IV. Tensor-based function decomposition

A. Jacobian and ParaTuck decomposition

The main idea to find the decomposition (13) of a nonlinear function $\mathbf{f}$ relies on the evaluation of the Jacobian matrix in different points $\mathbf{x}^{(p)}$, for $p = 1, \ldots, P$. This idea mirrors [3], where it has been applied to the classical decoupling model. In the sequel, we will replicate the procedure with the new proposed structure in (13) and will derive the new expression of the Jacobian tensor.

Lemma 1: The first-order derivatives of the parameterization (13) are given by

$$\mathbf{J}_\mathbf{f}(\mathbf{x}) := \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(\mathbf{x}) & \cdots & \frac{\partial f_1}{\partial x_m}(\mathbf{x}) \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1}(\mathbf{x}) & \cdots & \frac{\partial f_n}{\partial x_m}(\mathbf{x}) \end{bmatrix} \tag{14}$$

$$= \mathbf{W} \cdot \operatorname{diag}\Big(\big[g_1'(z_1) \cdots g_r'(z_r)\big]\Big) \cdot \mathbf{V}^\mathsf{T} \cdot \operatorname{diag}\Big(\big[h_1'(t_1) \cdots h_s'(t_s)\big]\Big) \cdot \mathbf{U}^\mathsf{T}. \tag{15}$$

Proof: Proof follows by applying the chain rule to (13). □

Based on Lemma 1, we can see that Jacobian of (13) evaluated at the points $\mathbf{x}^{(p)}$ follows

$$\mathbf{J}_\mathbf{f}(\mathbf{x}^{(p)}) = \mathbf{W} \cdot \operatorname{diag}\Big(\mathbf{g}'(\mathbf{z}^{(p)})\Big) \cdot \mathbf{V}^\mathsf{T} \cdot \operatorname{diag}\Big(\mathbf{h}'(\mathbf{t}^{(p)})\Big) \cdot \mathbf{U}^\mathsf{T}, \tag{16}$$

$$= \mathbf{W} \cdot \mathbf{D}_\mathbf{G}^{(p)} \cdot \mathbf{V}^\mathsf{T} \cdot \mathbf{D}_\mathbf{H}^{(p)} \cdot \mathbf{U}^\mathsf{T}, \tag{17}$$

where $\mathbf{D}_\mathbf{G}^{(p)} \in \mathbb{R}^{r \times r}$ and $\mathbf{D}_\mathbf{H}^{(p)} \in \mathbb{R}^{s \times s}$ are the diagonal matrices given by

$$\mathbf{D}_\mathbf{G}^{(p)} = \operatorname{diag}\big(\mathbf{g}_p\big), \quad \mathbf{D}_\mathbf{H}^{(p)} = \operatorname{diag}\big(\mathbf{h}_p\big),$$

and the vectors $\mathbf{g}_p \in \mathbb{R}^r$ and $\mathbf{h}_p \in \mathbb{R}^s$ are given by

$$\mathbf{g}_p = \mathbf{g}'(\mathbf{z}^{(p)}) = \big[g_1'(z_1^{(p)}) \cdots g_r'(z_r^{(p)})\big]^\mathsf{T}, \tag{18}$$

$$\mathbf{h}_p = \mathbf{h}'(\mathbf{t}^{(p)}) = \big[h_1'(t_1^{(p)}) \cdots h_s'(t_s^{(p)})\big]^\mathsf{T}, \tag{19}$$

with $\mathbf{t}^{(p)} = \big[t_1^{(p)} \cdots t_s^{(p)}\big]^\mathsf{T} = \mathbf{U}^\mathsf{T}\mathbf{x}^{(p)}$ and

$$\mathbf{z}^{(p)} = \big[z_1^{(p)} \cdots z_r^{(p)}\big]^\mathsf{T} = \mathbf{V}^\mathsf{T}\mathbf{h}(\mathbf{U}^\mathsf{T}\mathbf{x}^{(p)}).$$

We can then define the matrices $\mathbf{H} \in \mathbb{R}^{s \times P}$ and $\mathbf{G} \in \mathbb{R}^{r \times P}$, for $p = 1, \ldots, P$, by their columns

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_1 & \cdots & \mathbf{g}_P \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} \mathbf{h}_1 & \cdots & \mathbf{h}_P \end{bmatrix},$$

or equivalently by their rows as

$$\begin{aligned} H_{j,:} &= \big[h_j'(t_j^{(1)}) \cdots h_j'(t_j^{(P)})\big], \\ G_{i,:} &= \big[g_i'(z_i^{(1)}) \cdots g_i'(z_i^{(P)})\big]. \end{aligned} \tag{20}$$

This result shows that the expression of the Jacobian of the new model corresponds to the frontal slices of a PT decomposition (3) of rank (r, s) with factors $\mathbf{U}$, $\mathbf{V}^\mathsf{T}$ and $\mathbf{W}$. It is worth noting that the factors $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{W}$ do not depend on the choice of the point $\mathbf{x}^{(p)}$. Following Lemma 1, a Jacobian tensor $\mathcal{J}$ of size $n \times m \times P$ is constructed by stacking the Jacobian evaluation at P different sampling points $\mathbf{x}^{(p)} \in \mathbb{R}^m$, for $p = 1, \ldots, P$, where

$$\mathcal{J}_{:,:,p} = \mathbf{J}_\mathbf{f}(\mathbf{x}^{(p)}), \tag{21}$$

therefore, tensor $\mathcal{J}$ admits a PT decomposition.

B. Second-order information and structured CPD

To improve the usefulness of the PT formulation, we will examine the second-order information of (13), and show later how this can help in the decomposition since the PT decomposition lacks a reliable algorithm for the moment. In this subsection, we derive an expression for the Hessian tensor at each point. The Hessian tensor $\boldsymbol{\mathcal{H}}(\mathbf{x}) \in \mathbb{R}^{n \times m \times m}$ at a point $\mathbf{x}$ is defined as

$$\mathcal{H}_{ijk}(\mathbf{x}) = \frac{\partial^2 f_i}{\partial x_j \partial x_k}(\mathbf{x}).$$

Next, we show the Hessian tensor has a CP decomposition. But first, we introduce some extra notation for the factors of the Jacobians introduced in the previous subsection. We denote the matrices $\mathbf{A}(\mathbf{x}) \in \mathbb{R}^{n \times s}$ and $\mathbf{B}(\mathbf{x}) \in \mathbb{R}^{r \times m}$ such that

$$\mathbf{A}(\mathbf{x}) = \mathbf{W} \cdot \operatorname{diag}\Big(\mathbf{g}'(\mathbf{z}(\mathbf{x}))\Big) \cdot \mathbf{V}^\mathsf{T}, \tag{22}$$

$$\mathbf{B}(\mathbf{x}) = \mathbf{V}^\mathsf{T} \cdot \operatorname{diag}\Big(\mathbf{h}'(\mathbf{t}(\mathbf{x}))\Big) \cdot \mathbf{U}^\mathsf{T}; \tag{23}$$

so that with this notation, we can reformulate (16) as

$$\begin{aligned} \mathbf{J}_\mathbf{f}(\mathbf{x}) &= \mathbf{A}(\mathbf{x}) \operatorname{diag}\Big(\mathbf{h}'(\mathbf{t}(\mathbf{x}))\Big) \cdot \mathbf{U}^\mathsf{T} \\ &= \mathbf{W} \cdot \operatorname{diag}\Big(\mathbf{g}'(\mathbf{z}(\mathbf{x}))\Big)\mathbf{B}(\mathbf{x}). \end{aligned} \tag{24}$$

Armed with this notation, we are ready to formulate the following result on the structure of the Hessian tensor.

Lemma 2: The Hessian tensor has the following rank $(r+s)$ CP (polyadic) decomposition:

$$\begin{aligned} \boldsymbol{\mathcal{H}}(\mathbf{x}) &= [\![\mathbf{g}''(\mathbf{z}(\mathbf{x})); \mathbf{W}, \mathbf{B}^\mathsf{T}(\mathbf{x}), \mathbf{B}^\mathsf{T}(\mathbf{x})]\!] \\ &+ [\![\mathbf{h}''(\mathbf{t}(\mathbf{x})); \mathbf{A}(\mathbf{x}), \mathbf{U}, \mathbf{U}]\!]. \end{aligned} \tag{25}$$

Proof: The proof follows by applying the Leibniz rule to one of the formulations in (24). □

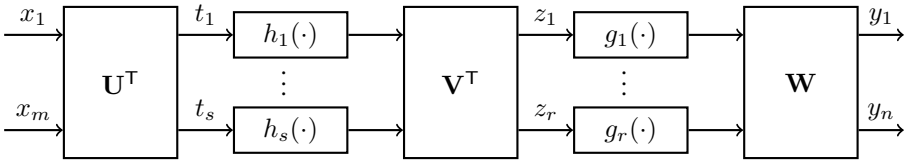

Fig. 2. Decoupled representation of $\mathbf{f}$ into a two-layer model, as in (13). This naturally leads to a PT decomposition of the corresponding Jacobian tensor.

It is worth noting that (i) the Hessian tensor is partially symmetric, *i.e.,* $\mathcal{H}_{ijk} = \mathcal{H}_{ikj}$, and (ii) the rank is too high so that the decomposition cannot be obtained from a CPD due to loss of uniqueness.

V. Constrained coupled decomposition approach

In this section, we propose to tackle the previously mentioned problems by formulating the new decoupling problem as a constrained coupled tensor decomposition, using both the first and second-order information. Before that, we specify the assumptions considered in our approach:

1) $m \geq s$ and $n \geq r \geq s$,
2) $\mathbf{W}$ is known and has full column rank r,
3) $\mathbf{U}$ and $\text{unfold}_2\mathcal{J}$ have full column rank s.

Under the conditions above, we can always reduce the problem to the case $r = n$, $s = m$, and $\mathbf{W} = \mathbf{I}_r$. It is important to mention that (i) despite these assumptions, the PT decomposition remains a challenging problem that cannot be solved by ALS-type algorithms, such as the one proposed in [15], and (ii) these assumptions are not overly restrictive and can easily be met in practical applications, especially in the training of neural networks, such as autoencoders [27].

A. Reformulation as a constrained CPD

We assume that we are given both Jacobians and Hessians $\mathbf{J}_\mathbf{f}(\mathbf{x}^{(p)})$, and $\boldsymbol{\mathcal{H}}(\mathbf{x}^{(p)})$ at P evaluation points, $\mathbf{x}^{(p)} \in \mathbb{R}^m$, for $p = 1, \ldots, P$. Let us define stacked Hessian tensor and Jacobian matrix as follows. We stack all the Hessians into a third-order $Pn \times m \times m$ tensor $\boldsymbol{\mathcal{T}}^{hess}$ as

$$(\boldsymbol{\mathcal{T}}^{hess})_{1+(p-1)n:pn,:,:} = \boldsymbol{\mathcal{H}}(\mathbf{x}^{(p)}). \tag{26}$$

We also stack Jacobians in one matrix $\mathbf{J}^{all} \in \mathbb{R}^{Pn \times m}$ as

$$\mathbf{J}^{all}_{1+(p-1)n:pn,:} = \mathbf{J}_\mathbf{f}(\mathbf{x}^{(p)}). \tag{27}$$

Then the following proposition shows that $\boldsymbol{\mathcal{T}}^{hess}$ admits a particular coupled CP decomposition.

Proposition 1: Under the assumption that the matrix $\mathbf{G}$ does not have nonzero elements (equivalently, all diagonal matrices $\mathbf{D}^{(p)}_\mathbf{G}$ are nonsingular). Then, $\boldsymbol{\mathcal{T}}^{hess}$ has the following CPD with structured factors:

$$\boldsymbol{\mathcal{T}}^{hess} = [\![\text{diag}(\mathbf{c}), (\mathbf{J}^{all})^\mathsf{T}, (\mathbf{J}^{all})^\mathsf{T}]\!] + [\![\mathbf{E}, \mathbf{U}, \mathbf{U}]\!], \tag{28}$$

where

$$\mathbf{c} = \begin{bmatrix} (\mathbf{D}^{(1)}_\mathbf{G})^{-2}\mathbf{g}''(\mathbf{z}^{(1)}) \\ \vdots \\ (\mathbf{D}^{(P)}_\mathbf{G})^{-2}\mathbf{g}''(\mathbf{z}^{(P)}) \end{bmatrix}, \ \mathbf{E} = \begin{bmatrix} (\mathbf{D}^{(1)}_\mathbf{G})\mathbf{V}^\mathsf{T}\,\text{diag}(\mathbf{h}''(\mathbf{t}^{(1)})) \\ \vdots \\ (\mathbf{D}^{(P)}_\mathbf{G})\mathbf{V}^\mathsf{T}\,\text{diag}(\mathbf{h}''(\mathbf{t}^{(P)})) \end{bmatrix}$$

Proof: Based on (24), we remark that the matrices $\mathbf{B}(\mathbf{x}^{(p)})$ can be expressed through the known Jacobians as

$$\mathbf{B}(\mathbf{x}^{(p)}) = (\mathbf{D}^{(p)}_\mathbf{G})^{-1}\mathbf{J}^{(p)}, \tag{29}$$

where $\mathbf{J}^{(p)} = \mathbf{J}_\mathbf{f}(\mathbf{x}^{(p)})$. In the same way, and based on (22), matrix $\mathbf{A}(\mathbf{x}^{(p)})$ can be expressed as

$$\mathbf{A}(\mathbf{x}^{(p)}) = \mathbf{D}^{(p)}_\mathbf{G}\mathbf{V}^\mathsf{T}. \tag{30}$$

Substituting (29) and (30) in (25), the Hessian at the p-th sampling point has the decomposition

$$\begin{aligned} \boldsymbol{\mathcal{H}}(\mathbf{x}^{(p)}) = \ & [\![((\mathbf{D}^{(p)}_\mathbf{G})^{-2}\mathbf{g}''(\mathbf{z}(\mathbf{x}^{(p)})))^\mathsf{T}; \mathbf{I}_r, (\mathbf{J}^{(p)})^\mathsf{T}, (\mathbf{J}^{(p)})^\mathsf{T}]\!] \\ & + [\![\mathbf{h}''(\mathbf{t}(\mathbf{x}^{(p)})); \mathbf{D}^{(p)}_\mathbf{G}\mathbf{V}^\mathsf{T}, \mathbf{U}, \mathbf{U}]\!] \\ = \ & [\![\text{diag}((\mathbf{D}^{(p)}_\mathbf{G})^{-2}\mathbf{g}''(\mathbf{z}(\mathbf{x}^{(p)}))), (\mathbf{J}^{(p)})^\mathsf{T}, (\mathbf{J}^{(p)})^\mathsf{T}]\!] \\ & + [\![\mathbf{D}^{(p)}_\mathbf{G}\mathbf{V}^\mathsf{T}\,\text{diag}(\mathbf{h}''(\mathbf{t}(\mathbf{x}^{(p)}))), \mathbf{U}, \mathbf{U}]\!]. \end{aligned}$$

Stacking the expressions for $\boldsymbol{\mathcal{H}}(\mathbf{x}^{(p)})$ in (26) completes the proof. □

We see that the decomposition in (28) is a CPD where the first term has two known factors $\mathbf{J}^{all}$ and one diagonal factor, and a second term with unknown factors. Note that the second CPD has very low rank s and can be retrieved with matrix methods as we explain in the next subsection.

B. Reformulation as structured low-rank matrix completion

Instead of the tensor $\boldsymbol{\mathcal{T}}^{hess}$, we consider its transposed first unfolding $\mathbf{T}^{all} \in \mathbb{R}^{m^2 \times nP}$, which has the factorization

$$\mathbf{T}^{hess} = \left((\mathbf{J}^{all})^\mathsf{T} \odot (\mathbf{J}^{all})^\mathsf{T}\right)\text{diag}(\mathbf{c}) + (\mathbf{U} \odot \mathbf{U})\mathbf{E}^\mathsf{T}.$$

Assume we are in the exact case, then we just need to find vector $\mathbf{c}$ so that the matrix

$$\mathscr{S}(\mathbf{c}) = \mathbf{T}^{hess} - \left((\mathbf{J}^{all})^\mathsf{T} \odot (\mathbf{J}^{all})^\mathsf{T}\right)\text{diag}(\mathbf{c})$$

has rank s. We pose this problem as rank minimization of $\mathscr{S}(\mathbf{c})$, which can be solved as the following minimization problem over the low-rank manifold:

$$\min_{\mathbf{P},\mathbf{L}} \|\Pi_\mathscr{S}(\mathbf{P}\mathbf{L} - \mathbf{T}^{hess})\|_F, \tag{31}$$

where $\Pi_\mathscr{S}$ is the projection on the set of structured matrices, see e.g., [28] for more details on the reformulation (31).

After the minimizer of (31) is found, there is still some work need to be done even if the low-rank matrix is exactly recovered ($\mathbf{P}\mathbf{L} = (\mathbf{U} \odot \mathbf{U})\mathbf{E}^\mathsf{T}$). In fact, this is due to the ambiguity of low-rank matrix factorization. Therefore, in general, we have

$$\mathbf{P} = (\mathbf{U} \odot \mathbf{U})\mathbf{S}^\mathsf{T},$$

and hence $\mathbf{U}$ can be retrieved from the CPD $[\![\mathbf{U}, \mathbf{U}, \mathbf{S}]\!]$ of the reshaping of $\mathbf{P} \in \mathbb{R}^{m^2 \times d}$ as an $m \times m \times s$ tensor.

C. Recovering of other factors and functions

After estimation of $\mathbf{U}$, we can find the core tensor (4) by multiplying by its pseudoiverse the Jacobian tensor ($\boldsymbol{\mathcal{C}} = \boldsymbol{\mathcal{J}} \bullet_2 \mathbf{U}^\dagger$). From the core tensor (4), it is easy to estimate the other factors. Indeed, without loss of generality, let us assume assuming that all elements of a frontal slice $\boldsymbol{\mathcal{C}}_{:,:,k'}$ are nonzero, we set $\mathbf{V} = \boldsymbol{\mathcal{C}}_{:,:,k'}$, so we can set $\widehat{\mathbf{G}}_{:,k'} \equiv 1$, $\widehat{\mathbf{H}}_{:,k'} \equiv 1$ thanks to the ambiguities. Then by (5) we must have

$$\frac{\boldsymbol{\mathcal{C}}_{i,j,k}}{\boldsymbol{\mathcal{C}}_{i,j,k'}} = \widehat{G}_{ik}\widehat{H}_{jk},$$

thus the k-th columns of $\widehat{\mathbf{G}}$ and $\widehat{\mathbf{H}}$ can be found from rank-one factorization of the elementwise division of the slices $\boldsymbol{\mathcal{C}}_{:,:,k}$ and $\boldsymbol{\mathcal{C}}_{:,:,k'}$.

However, there still remains the problem of slicewise ambiguities α_k, which is particularly important for the reconstruction of functions. In fact, the approach suggested in [3] (regression of $\mathbf{H}_{j,:}$ versus $(t_j^{(1)}, \ldots, t_j^{(P)})$, see (20)) does not directly work. This happens because, even in the case of exact decomposition, the columns of the estimated matrices $\widehat{\mathbf{G}}$ and $\widehat{\mathbf{H}}$ contain information about functions only up to the slice-wise ambiguities:

$$H_{j,:} = \left[\alpha_1^{-1} h_j'(t_j^{(1)}) \cdots \alpha_P^{-1} h_j'(t_j^{(P)})\right],$$
$$G_{i,:} = \left[\alpha_1 g_i'(z_i^{(1)}) \cdots \alpha_P g_i'(z_i^{(P)})\right].$$

To recover the functions, we need more assumptions (for example, impose that h_k can be polynomials of low order). To estimate the slice-wise ambiguities, we propose to solve the following system of equations

$$\mathbf{a}_k^\mathsf{T} \mathbf{X}(\mathbf{t}, d) = \mathbf{H}_{k,:} \operatorname{diag}(\boldsymbol{\alpha}), \quad k = 1, \ldots, s,$$

where d is the degree of the polynomial, $\mathbf{X}(\mathbf{t}, d)$ is the Vandermonde matrix (for points t and up to degree d), and we solve for $\mathbf{a}_k^\mathsf{T}$ (coefficients of polynomials) and α (scalings). This is a problem of intersection of linear subspaces and can be solved with alternating projections, see Fig. 3.

D. Overall algorithm

The overall algorithm can be summarized as follows: **Algorithm: Two-layer decoupling using PTD**

assume $r = n$, $s = m$, $\mathbf{W} = \mathbf{I}_r$
input: Jacobian evaluations $\mathbf{J}_f(\mathbf{x}_1), \ldots, \mathbf{J}_f(\mathbf{x}_P)$
Hessian evaluations $\boldsymbol{\mathcal{T}}(\mathbf{x}_1), \ldots, \boldsymbol{\mathcal{T}}(\mathbf{x}_P)$
output: factors $\mathbf{U}$, $\mathbf{V}$; coefficients of $\mathbf{g}$, $\mathbf{h}$

1) stack the Jacobians into matrix $\mathbf{J}^{all}$, see (27)
2) stack the Hessians into tensor $\boldsymbol{\mathcal{T}}^{hess}$, see (26)
3) find $\mathbf{PL}$ from the rank minimization problem (31)
4) reshape $\mathbf{P}$ into a $m \times m \times m$ tensor
5) compute its rank-m CPD $[\![\mathbf{U}, \mathbf{U}, \mathbf{Q}]\!]$ and extract $\mathbf{U}$
6) find the ParaTuck core tensor $\boldsymbol{\mathcal{C}} = \boldsymbol{\mathcal{J}} \bullet_2 \mathbf{U}^\dagger$
7) from $\boldsymbol{\mathcal{C}}$ and (4), recover $\mathbf{V}$, $\mathbf{G}$, and $\mathbf{H}$
8) fix α_k ambiguities by imposing polynomial structure.

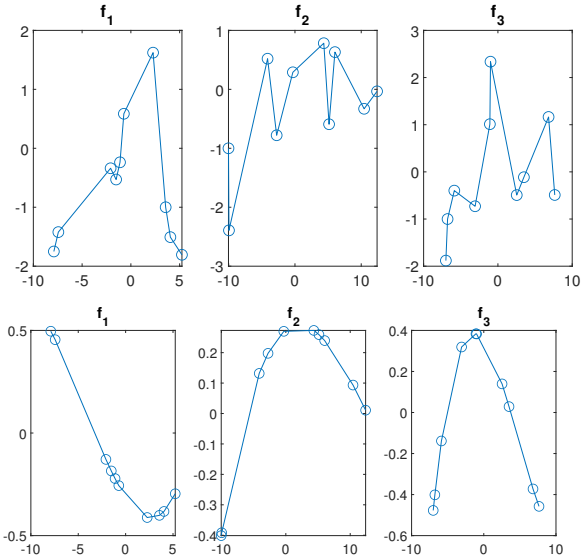

Fig. 3. Due to the presence of slice ambiguities, recovering the functions $\mathbf{g}$ and $\mathbf{h}$ from the PTD requires additional attention. Plot of the estimation of $\mathbf{h}$ without (top) and with (bottom) slice ambiguity correction.

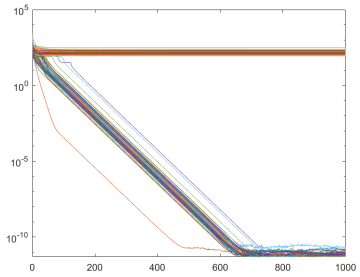

Fig. 4. The convergence plot shows a linear convergence to an error of order 10^{-10} in 46 out of 100 random initializations.

VI. Numerical examples

A. Example of decoupling with $r = s = 3$

We consider a concrete example of decoupling for the polynomials functions, which are shifts of the same function

$$h_1(t) = \phi(t - 0.5), \quad h_2(t) = \phi(t + 0.2), \quad h_3(t) = \phi(t)$$
$$g_1(t) = \phi(t + 0.1), \quad g_2(t) = \phi(t + 0.4), \quad g_3(t) = \phi(t)$$

where $\phi(t) = t^2 - 0.25t^4 + t^3 - 3t$. In our experiment, the matrices $\mathbf{V}$ and $\mathbf{U}$ were generated randomly, with i.i.d. elements from the uniform distribution on $[-1; 1]$. The $P = 100$ sampling points $\mathbf{x}^{(p)}$ are drawn uniformly from $[-0.5; 0.5]$.

In Fig. 4, we show the cost function (31) as a function of iteration (out of maximum 1000 iterations). We run 100 random initializations of $\mathbf{P}$ as $\mathbf{P}_0 = \mathbf{U}_0 \odot \mathbf{U}_0$ with $\mathbf{U}_0$ drawn from standard Gaussian i.i.d. distribution. We observe that in 46 cases out of 100, the algorithm shows linear convergence to an error of order 10^{-10}. Taking one of the runs with the best cost function value, we are able to recover the original $\mathbf{U}$ and the nonlinearities.

VII. Conclusion

We presented a new method for multivariate function approximation that couples the PT and CP decompositions. Our approach utilizes both first and second-order information of the original function and has been shown to be effective through numerical simulations on a simple synthetic example. Although the PT decomposition remains a challenging problem, our results demonstrate the potential of the proposed method for addressing this issue and provide a promising direction for future work in the field of multivariate function approximation.

References

[1] J. Schoukens and L. Ljung, "Nonlinear system identification: A user-oriented road map," *IEEE Control Systems Magazine*, vol. 39, no. 6, pp. 28–99, 2019.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[3] P. Dreesen, M. Ishteva, and J. Schoukens, "Decoupling multivariate polynomials using first-order information and tensor decompositions," *SIAM Journal on Matrix Analysis and Applications*, vol. 36, no. 2, pp. 864–879, 2015.

[4] A. Apicella, F. Donnarumma, F. Isgrò, and R. Prevete, "A survey on modern trainable activation functions," *Neural Networks*, vol. 138, pp. 14–32, 2021.

[5] Y. Zniyed, K. Usevich, S. Miron, and D. Brie, "Tensor-based approach for training flexible neural networks," in *55th Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, United States, Oct. 2021.

[6] M. Schoukens and Y. Rolain, "Cross-term elimination in parallel Wiener systems using a linear input transformation," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, pp. 845–847, 2012.

[7] A. Van Mulders, L. Vanbeylen, and K. Usevich, "Identification of a block-structured model with several sources of nonlinearity," in *European Control Conference (ECC)*, Strasbourg, France, 2014.

[8] F. L. Hitchcock, "Multiple invariants and generalized rank of a p-way matrix or tensor," *Journal of Mathematics and Physics*, vol. 7, pp. 39–79, 1927.

[9] R. A. Harshman, "Foundations of the PARAFAC procedure: Models and conditions for an 'explanatory' multi-modal factor analysis," *UCLA Working Papers in Phonetics*, vol. 16, pp. 1–84, 1970.

[10] R. Bro, "PARAFAC. tutorial and applications," *Chemometrics and Intelligent Laboratory Systems*, vol. 38, no. 2, pp. 149–171, 1997.

[11] G. Hollander, "Multivariate polynomial decoupling in nonlinear system identification," Ph.D. dissertation, Vrije Universiteit Brussel, 2018.

[12] J. Decuyper, K. Tiels, S. Weiland, M. C. Runacres, and J. Schoukens, "Decoupling multivariate functions using a nonparametric filtered tensor decomposition," *Mechanical Systems and Signal Processing*, vol. 179, p. 109328, 2022.

[13] R. A. Harshman and M. E. Lundy, "PARAFAC: Parallel factor analysis," *Computational Statistics & Data Analysis*, vol. 18, no. 1, pp. 39–72, 1994.

[14] ——, "Uniqueness proof for a family of models sharing features of Tucker's three-mode factor analysis and PARAFAC/candecomp," *Psychometrika*, vol. 61, no. 1, pp. 133–154, March 1996.

[15] R. Bro, "Multi-way analysis in the food industry — models, algorithms & applications," Ph.D. dissertation, Universiteit van Amsterdam, 1998.

[16] M. Janzamin, H. Sedghi, and A. Anandkumar, "Beating the perils of non-convexity: Guaranteed training of neural networks using tensor methods," *arXiv preprint arXiv:1506.08473*, 2015.

[17] P. Comon and M. Rajih, "Blind identification of under-determined mixtures based on the characteristic function," *Signal Processing*, vol. 86, no. 9, pp. 2271–2281, 2006.

[18] M. Fornasier, T. Klock, and M. Rauchensteiner, "Robust and resource-efficient identification of two hidden layer neural networks," *Constructive Approximation*, pp. 1–62, 2019.

[19] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3551–3582, 2017.

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

[21] R. André, X. Luciani, and E. Moreau, "Joint eigenvalue decomposition algorithms based on first-order Taylor expansion," *IEEE Transactions on Signal Processing*, vol. 68, pp. 1716–1727, 2020.

[22] A. L. de Almeida, G. Favier, and J. C. Mota, "Space–time spreading–multiplexing for MIMO wireless communication systems using the PARATUCK2 tensor model," *Signal Processing*, vol. 89, no. 11, pp. 2103–2116, 2009.

[23] A. L. F. de Almeida, G. Favier, and L. R. Ximenes, "Space-time-frequency (STF) MIMO communication systems with blind receiver based on a generalized PARATUCK2 model," *IEEE Transactions on Signal Processing*, vol. 61, no. 8, pp. 1895–1909, 2013.

[24] L. R. Ximenes, G. Favier, A. L. F. de Almeida, and Y. C. B. Silva, "Parafac-paratuck semi-blind receivers for two-hop cooperative mimo relay systems," *IEEE Transactions on Signal Processing*, vol. 62, no. 14, pp. 3604–3615, 2014.

[25] P. Marinho R. de Oliveira, C. A. R. Fernandes, G. Favier, and R. Boyer, "PARATUCK semi-blind receivers for relaying multi-hop MIMO systems," *Digital Signal Processing*, vol. 92, pp. 127–138, 2019.

[26] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.

[27] C.-Y. Liou, W.-C. Cheng, J.-W. Liou, and D.-R. Liou, "Autoencoder for words," *Neurocomputing*, vol. 139, pp. 84–96, 2014.

[28] M. Ishteva, K. Usevich, and I. Markovsky, "Factorization approach to structured low-rank approximation with applications," *SIAM Journal on Matrix Analysis and Applications*, vol. 35, no. 3, pp. 1180–1204, 2014.