Journal de Ciencia e Ingeniería, vol.13, no. 1, pp.13-21, 2021
http://jci.uniautonoma.edu.co
DOI: https://doi.org/10.46571/JCI.2021.1.2

Corporación Universitaria Autónoma del Cauca
ISSN 2145-2628, e-ISSN 2539-066X
MATHEMATICS

Canonical tensor scaling

Tensor escalar canónico

Tung Nguyen[+1] **and Jeffrey Uhlmann**[*2]
[+]**Deparment of Physics and Astronomy**
University of Missouri-Columbia
[*]**Deparment of Electrical Engineering and Computer Science**
University of Missouri-Columbia

Abstract. In this paper we generalize the canonical positive scaling of rows and columns of a matrix to the scaling of selected-rank subtensors of an arbitrary tensor. We expect our results and framework will prove useful for sparse-tensor completion required for generalizations of the recommender system problem beyond a matrix of user-product ratings to multidimensional arrays involving coordinates based both on user attributes (e.g., age, gender, geographical location, etc.) and product/item attributes (e.g., price, size, weight, etc.).

Keywords. Kuhn-Tucker conditions; scale invariance; tensor; recommender systems.

Resumen. En este artículo se generaliza el escalamiento canónico de renglones y columnas de una matriz a un escalamiento de subtensores de rango seleccionado de un subtensor arbitrario. Esperamos que nuestros resultados y marco teórico sea útil para completar el tensor disperso necesario para la generalización del problema del sistema de la matriz de evaluación usuario-producto a arreglos multidisciplinarios basados tanto en atributos de usuario (edad, sexo, geolocalización, etc.) como en atributos del producto o artículo (precio, peso, tamaño, etc.)

Palabras Clave. Condición de Kuhn-Tucker; invarianza de escala; tensor; sistemas de recomendación.

How to Cite. T. Nguyen and J. Uhlmann, "Canonical tensor scaling", *Jou. Cie. Ing.*, vol. 13, no. 1, pp. 13-21, 2021. doi:10.46571/JCI.2021.1.2

Received: 20/01/2021 **Revised:** 06/04/2021 **Accepted:** 22/04/2021

1. Introduction

A 1992 algorithm by Rothblum & Zenios [1], which we will refer to as the RZ algorithm, computes a positive diagonal scaling of an arbitrary matrix so that the products of the nonzero normed elements of each row and column (i.e., zero-elements are ignored) are equal to user-specified constants. It has recently been recognized that in the case in which the constants are all unity, a unique scaling always exists and can be used as a canonical scaling for performing scale-consistent or scale-invariant operations on matrices [6]. For example, the algorithm can be applied to define a generalized matrix inverse that is consistent with respect to diagonal scalings

[1] *e-mail: tdn84d@mail.missouri.edu*
[2] *e-mail: uhlmannj@missouri.edu*

(e.g., due to changes of units on variables), as opposed to the Moore-Penrose pseudoinverse which provides consistency with respect to unitary transformations (e.g., rotations) [6]. Scale consistency is critical to applications in which performance is expected to be invariant with respect to the choice of state variables, e.g., whether lengths are represented in centimeters, meters, or kilometers in robotics applications [9, 10, 11, 7] or unknown units specific to each individual's ratings of products in the context of recommender systems [4].

In this paper we generalize the canonical positive scaling of rows and columns of a matrix to the scaling of selected-rank subtensors of an arbitrary tensor. We expect our results and framework will prove useful for sparse-tensor completion required for generalizations of the recommender system problem beyond a matrix of user-product ratings to multidimensional arrays involving coordinates based both on user attributes (e.g., age, gender, geographical location, etc.) and product/item attributes (e.g., price, size, weight, etc.).

2. Definitions

The objective of this paper is to identify a generalized canonical scaling of subtensors of a given tensor. In the case of a matrix, i.e., a 2-tensor, the rows and columns can be interpreted as the 1-dimensional subtensors that can be scaled by the RZ algorithm. In order to generalize beyond matrices to arbitrary d-dimensional tensors, or d-tensors, we must define notation for operating on the k-dimensional subtensors that are to be scaled, where $1 \leq k < d$ for integer parameter k. This requires distinguishing the overlapping k-dimensional subtensors with respect to each of the d coordinates. In the $d = 2$ case of a matrix with $k = 1$, for example, the dimensional label set $\{1, 2\}$ can be interpreted with (1) denoting the set of row vectors and (2) representing the set of column vectors. In other words, the tuple $(1, i)$ designates the ith row vector while $(2, j)$ designates the jth column vector[1], and $(2, i, j)$ would specify the jth element of the ith column.

In the $d = 3$ case of a 3-tensor, the choice $k \in \{1, 2\}$ can be chosen to specify whether our focus is on 2-dimensional subtensors (matrices) or 1-dimensional subtensors (vectors). For example, $k = 1$ defines $d = 3$ sets of 1-dimensional subtensors, or fibers, each set of which can be thought of intuitively as containing all of the vectors parallel to one of the 3 coordinate axes, e.g., set (1) refers to all 1-dimensional subtensors that live in $\mathbb{R}^{n_1 \times 1 \times 1} \equiv \mathbb{R}^{n_1}$, and $k = 2$ defines all possible 2-dimensional subtensors, or *slices*, comprising $\binom{3}{2} = 3$ sets of slices, the span of which is determined by one of the pairs of indices $(1, 2)$, $(2, 3)$, or $(1, 3)$. For instance, a pair $(1, 3)$ spans all possible 2-dimensional subtensors that live in $\mathbb{R}^{n_1 \times 1 \times n_3} \equiv \mathbb{R}^{n_1 \times n_3}$. In other words, it is possible to relabel these pairs as sets (1), (2), and $\left(\binom{3}{2}\right) = 3$. An important property of the subtensors in each set is that they are disjoint from each other and can be stacked to form the original d-dimensional tensor. In general, the k-dimensional subtensors of the d-dimensional tensor can be partitioned into $\left(\binom{d}{k}\right)$ sets, corresponding to the number of k-tuples chosen from d dimensional indices. An alternative but equivalent interpretation is that there are $\binom{d}{k}$ ways of dividing a d-dimensional tensor into k-subtensors such that all subtensors in each set are disjoint and any two subtensors in two different sets are different. This leads to the following definition.

Definition 2.1. - Sets of subtensors: Let $V_i = \mathbb{R}^{n_i}$ and assume a d-dimensional tensor $A \in V_1 \times V_2 \times \cdots \times V_d$ and a positive integer $1 \leq k < d$. From the set of dimensional indices $[d] = \{1, \ldots, d\}$ we can obtain all $\binom{d}{k}$ possible tuples of k dimensional indices. Specifically,

[1] Matrices are special in that the terms *rows* and *columns* have been conventionally defined to distinguish two sets of overlapping 1-dimensional subtensors/vectors. For our purposes the particular choice of the ordering of the labels, i.e., the distinction between a given matrix and its transpose, is irrelevant. More generally, there is no natural ordering of the d orientations of a d-tensor with names analogous to *rows* and *columns*, so for notational purposes any arbitrary choice will work.

for any tuple $\{i_1, \cdots, i_k\} \subseteq [d]$, we enumerate all possible k-dimensional subtensors of A in dimensions $V_{i_1} \times \cdots \times V_{i_k}$ to obtain the set of subtensors (i). Altogether we obtain $\binom{d}{k}$ sets of k-dimensional subtensors relabeled as $(1), \ldots, \left(\binom{d}{k}\right)$. We denote the number of subtensors in a set (i) as its cardinality $|(i)|$ and label its elements as

$$(i) = \{A_{s_i,i} \mid 1 \leq s_i \leq |(i)|\} \tag{1}$$

for $1 \leq i \leq \binom{d}{k}$.

By this definition, all subtensors in each set (i) are disjoint and can combine to reconstruct the tensor A. Each set (i) corresponds to a partition of the d-dimensional tensor into mutually disjoint k-dimensional subtensors. This property implies the following observation.

Observation 2.1. Subtensors by position: Given a d-dimensional tensor A and position $\vec{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_d)$. Then for each value $A(\vec{\alpha})$ and each i for $1 \leq i \leq \binom{d}{k}$, there is a unique s_i for $1 \leq s_i \leq |(i)|$ such that the subtensor $A_{s_i,i}$ contains the position $\vec{\alpha}$ of tensor A, denoted as $\vec{\alpha} \sqsubset_A A_{s_i,i}$.

From this observation, we define the following sets:

Definition 2.2. - Support sets of a subtensor Given a position $\vec{\alpha}$ of A, define the support set type 0 as

$$\sigma_A(\vec{\alpha}) = \left\{(s_1, 1), \ldots, \left(s_{\binom{d}{k}}, \binom{d}{k}\right)\right\} = \left\{(s_i, i) \mid 1 \leq i \leq \binom{d}{k}, \vec{\alpha} \sqsubset_A A_{s_i,i}\right\} \tag{2}$$

From here, given a pair (s_i, i) corresponding to a subtensor of A, define the support set type 1 as

$$\sigma_A(s_i, i) = \{\vec{\alpha} \mid (s_i, i) \in \sigma_A(\vec{\alpha}), A(\vec{\alpha}) \neq 0\} \tag{3}$$

where their union is the set of nonzero elements in the tensor A, denoted by $\sigma(A)$,

$$\sigma(A) = \bigcup_{(s_i,i)} \sigma_A(s_i, i). \tag{4}$$

Now we are equipped to define the scaling process. For notational convenience we define a list $M_{(i)}$ of elements such that $cM_{(i)}$ multiplies/scales each element of every k-dim subtensor of $|(i)|$ as a single scaling process, and the product of a scalar in $M_{(i)}$ and a subtensor $A_{s_i,i}$ scales every elements of that subtensor and thus scales the tensor itself. We now formally define both the set $M_{(i)}$ and notation for this scaling operator following the support set type 1 in definition 2.2.

Definition 2.3. - Scaling a tensor by a set of subtensors: Given a tensor $A \in V_1 \times V_2 \times \cdots \times V_d$. A value of a tensor $A \in V_1 \times V_2 \times \cdots \times V_d$ at is defined as $A(\vec{\alpha}) \equiv A(\alpha_1, \cdots, \alpha_d)$. The list of scaling values with respect to (i) is defined as

$$M_{(i)} = \{M_{s_i,i} \in \mathbb{R}_{\neq 0} \mid 1 \leq s_i \leq |(i)|\},$$

The scaling operation $*_{(i)}$ is defined as

$$A' \equiv A *_{(i)} M_{(i)} \tag{5}$$

such that for each pair (s_i, i) and for every $\vec{\alpha} \in \sigma_A(s_i, i)$

$$A'(\vec{\alpha}) = M_{s_i,i} A(\vec{\alpha}). \tag{6}$$

Consider the case when $d = 3$ and $k = 2$. Let $A \in \mathbb{R}^{3\times4\times2}$ be a tensor with the following 2 frontal slices.

$$A_1 = \begin{bmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \end{bmatrix}, A_2 = \begin{bmatrix} 13 & 16 & 19 & 22 \\ 14 & 17 & 20 & 23 \\ 15 & 18 & 21 & 24 \end{bmatrix} \tag{7}$$

Then the set (3) contains the subtensors A_1 and A_2 and the scaling list $M_{(3)} = \{\alpha_1, \alpha_2\}$, then the scaling process by a list $M_{(3)}$ gives the new tensor A' as

$$A'_1 = \begin{bmatrix} \alpha_1 & 4\alpha_1 & 7\alpha_1 & 10\alpha_1 \\ 2\alpha_1 & 5\alpha_1 & 8\alpha_1 & 11\alpha_1 \\ 3\alpha_1 & 6\alpha_1 & 9\alpha_1 & 12\alpha_1 \end{bmatrix}, A'_2 = \begin{bmatrix} 13\alpha_2 & 16\alpha_2 & 19\alpha_2 & 22\alpha_2 \\ 14\alpha_2 & 17\alpha_2 & 20\alpha_2 & 23\alpha_2 \\ 15\alpha_2 & 18\alpha_2 & 21\alpha_2 & 24\alpha_2 \end{bmatrix} \tag{8}$$

At this point we have defined a particular type of structured scaling that can be applied to elements of specified sets of subtensors of a given tensor. In the next section we iteratively apply scalings of this kind to obtain a unique scaling with prescribed properties.

3. Unique Tensor Scaling

Using the representation and notation defined in the previous section, we demonstrate the uniqueness and correctness of our generalization of the RZ algorithm for matrices (i.e., $n = 2$ and $k = 1$) to arbitrary tensors with $1 \leq k < n$.

3.1. Defining the k-dim scaling program:

3.1.1. Program I: We now follow and extend the approach of Rothblum and Zenios (RZ) in [1] for the iterative scaling of 1-dimensional row and column vector sets of a matrix to the general case for arbitrary scaling of $\binom{d}{k}$ sets of subtensors of a given nonnegative tensor A, which may be obtained by replacing the elements of a given tensor A' with their magnitudes. The algorithm begins with tensor $A \in V_1 \times V_2 \times \cdots \times V_d$ and a list of strictly positive numbers $T = \{T_{s_i,i} \mid 1 \leq s_i \leq |(i)| \text{ and } 1 \leq i \leq \binom{d}{k}\}$. Let the scaling elements of $\binom{d}{k}$ list $M_{(i)}$ of scaling elements be applied in the scaling process of tensor A in label order from set (1) to set $\binom{d}{k}$. We observe that the choice of ordering is free due to the the commutative property of the scaling process.

Observation 3.1. The scaling of indices $i_1, i_2 \in \left[\binom{d}{k}\right]$ is commutative:

$$A *_{(i_1)} M_{(i_1)} *_{(i_2)} M_{(i_2)} = A *_{(i_2)} M_{(i_2)} *_{(i_1)} M_{(i_1)}. \tag{9}$$

In other words, a permutation of the dimensional indices will not affect the correctness of the following problem formulation. Using definitions 2.2 and 2.3, we define the scaling problem:

Problem Statement: Find a non-negative tensor $A' \in V_1 \times V_2 \times \cdots \times V_d$ and $\binom{d}{k}$ scaling lists $M_{(i)}$ of positive elements with cardinality $|(i)|$ such that:

$$A' = A *_{(1)} M_{(1)} \cdots *_{(\binom{d}{k})} M_{(\binom{d}{k})} \tag{10}$$

or

$$A'(\vec{\alpha}) \equiv \begin{cases} A(\vec{\alpha}) \cdot \prod\limits_{(s_i,i)\in\sigma_A(\vec{\alpha})} m_{s_i,i} & \text{if} \quad \vec{\alpha} \in \sigma(A) \\ 0 \quad \text{otherwise} \end{cases}$$

and for each subtensor by a pair (s_i, i)

$$\prod_{\vec{\alpha} \in \sigma_A(s_i,i)} A'(\vec{\alpha}) = T_{s_i,i} \tag{11}$$

A logarithm conversion for tensor A, A', and T is applied via the below-defined operator $\hat{L}$ to obtain tensors a, a', and s:

$$\begin{aligned} a(\vec{\alpha}) &\equiv \hat{L}(A(\vec{\alpha})) \equiv \begin{cases} ln(A(\vec{\alpha})) & \text{if} \quad \vec{\alpha} \in \sigma(A) \\ 0 \quad \text{otherwise} \end{cases} \\ a'(\vec{\alpha}) &\equiv \hat{L}(A'(\vec{\alpha})) \end{aligned} \tag{12}$$

We also let $t_{s_i,i} \equiv ln(T_{s_i,i})$ and $m_{s_i,i} \equiv ln(M_{s_i,i})$. Having performed the logarithm conversion of the operator $\hat{L}$, we present the following program,

Program I: Finding the tensor $a' \in V_1 \times V_2 \times \cdots \times V_d$ and $\binom{d}{k}$ lists $M_{(i)}$ such that:

$$a'(\vec{\alpha}) \equiv \begin{cases} a(\vec{\alpha}) + \sum\limits_{(s_i,i) \in \sigma_A(\vec{\alpha})} m_{s_i,i} & \text{if} \quad \vec{\alpha} \in \sigma(A) \\ \qquad 0 \quad \text{otherwise} \end{cases}$$

and for each subtensor by a pair (s_i, i)

$$\sum_{\vec{\alpha} \in \sigma_A(s_i,i)} a'(\vec{\alpha}) = t_{s_i,i} \tag{13}$$

3.1.2. Program II: The RZ model provides a basis for establishing the existence and uniqueness properties of our tensor generalization. Briefly, let $u \in R^p, b \in R^q$, and $C \in R^{p \times q}$ be the original convex optimization problem of finding a vector $u' \in R^p$ and $\omega \in R^q$ such that

$$u'^T = u^T + \omega^T C \tag{14}$$

and

$$Cu' = b \tag{15}$$

It is proven in [1] that this problem is equivalent to the following optimization problem, for which the properties of uniqueness and existence can be established and for $x \in R^p$:

$$\begin{aligned} \min \, 2^{-1} \sum_{j=1}^{p} (x_j - u_j)^2 \\ \text{subject to} \qquad Cx = b. \end{aligned} \tag{16}$$

The details of the proof can be found in [1], but we will only apply the Kuhn-Tucker (KT) conditions from **Program I** for our results. To establish the corresponding properties of the RZ algorithm for our **Program I**, we apply the following transformations. We begin with a transformation to *unfold* tensors a and a' into the corresponding vectors u and u' according to the following definition.

Definition 3.1. - Unfolding tensor: Given a tensor $a = V_1 \times V_2 \times \cdots \times V_d$ and $P = \prod_{j=1}^{d} n_j$. The unfolding vector $u \in \mathbb{R}^P$ defined via the mapping $a(\vec{\alpha}) \mapsto u(J(\vec{\alpha}))$ for

$$J(\vec{\alpha}) = 1 + \sum_{s=1}^{d} \left((\alpha_s - 1) \prod_{m=1}^{s-1} n_m \right)$$

Regarding $Cx = b$ and $\omega^T C$, we show (following [1]) KT conditions that **Program I** is equivalent to

Program II: For any pair (s_i, i):

$$\begin{aligned} \text{find min } 2^{-1} \sum_{\vec{\alpha} \in \sigma_A(s_i,i)} (x(\vec{\alpha}) - a(\vec{\alpha}))^2, \\ \text{subject to} \sum_{\vec{\alpha} \in \sigma_A(s_i,i)} x(\vec{\alpha}) = t_{s_i,i}. \end{aligned} \tag{17}$$

3.1.3. Proof: We define the node-arc incidence matrix of tensor A as a matrix C of real numbers with the objective of transforming the tensor problem to a matrix problem for which the solution becomes equivalent to the RZ solution for C. Specifically, we establish formulas such that each position of $\omega^T C$ defined by a function $J(\vec{\alpha})$ equals the sum of $m_{s_i,i}$ of **Program I** and each position on Cx defined by a pair (s_i, i) equals to the nonzero sum of x of **Program II**. For equation $Cx = b$, a function P with an ordering for (s_i, i) is defined as

$$P(s_i, i) = \sum_{i'=0}^{i-1} |(i')| + s_i, \tag{18}$$

with the assumption $|(0)| = 0$. Then all elements $t_{s_i,i}$ maps onto b and all elements $m_{s_i,i}$ maps onto ω as

$$b(P(s_i, i)) = t_{s_i,i}, \quad \omega(P(s_i, i)) = m_{s_i,i}. \tag{19}$$

Note that the tensor x agrees with tensor a in terms of nonzero elements. With a slight abuse of notation, we transform tensor x to a vector x by the unfolding mapping from definition 3.1. The matrix C is defined with its column size equals $N = \prod_{i=1}^{d} n_i$ and its row size equals $\sum_{i=1}^{\binom{d}{k}} |(i)|$.

$$C_{P(s_i,i)J(\vec{\alpha})} \equiv \begin{cases} 1 & \text{if} \quad \vec{\alpha} \in \sigma_A(s_i, i) \\ 0 & \text{otherwise.} \end{cases} \tag{20}$$

Then the vector ω and C matrix satisfy

$$[\omega^T C]_{J(\vec{\alpha})} = \sum_{(s_i,i) \in \sigma_A(\vec{\alpha})} m_{s_i,i} \tag{21}$$

where the matrix b is defined using $t_{s_i,i}$ such that $Cx = b$ is equivalent to

$$\sum_{\vec{\alpha} \in \sigma_A(s_i,i)} x(\vec{\alpha}) = t_{s_i,i}. \tag{22}$$

Satisfaction of the KT conditions guarantees

$$\begin{aligned} 0 =& \frac{\partial}{\partial x(\vec{\alpha})} \left[2^{-1} \sum_{\vec{\alpha} \in \sigma_A(s_i,i)} (x(\vec{\alpha}) - a(\vec{\alpha}))^2 - \omega^T (Cx - b) \right] \\ =& x(\vec{\alpha}) - a(\vec{\alpha}) - [\omega^T C]_{J(\vec{\alpha})} = x(\vec{\alpha}) - a(\vec{\alpha}) - \sum_{(s_i,i) \in \sigma_A(\vec{\alpha})} m_{s_i,i}. \end{aligned} \tag{23}$$

We now exploit the equivalence of our transformed formulation to that of the optimization problem of section 3 of [1] to establish the characterization of solution. Before that, we have the following lemma from Theorem 3.2 in [1].

Lemma 3.2. If $Cx = b$ has a real solution x and there exists a vector λ with $\lambda^T C = 0$ then $\lambda^T b = 0$.

Now, we have the characterization of solutions.

Theorem 3.3. (Characterization) The following statements are equivalent:

(i) There exists a solution to **Program I**

(ii) There exists a solution to **Program II**

(iii) There exists a non-negative tensor $A' \in V_1 \times V_2 \times \cdots \times V_d$ that satisfies **Program I** and has $\sigma(A) = \sigma(A')$

(iv) **Program II** is feasible

(v) **Program II** has a optimal solution

(vi) Given $\binom{d}{k}$ list of real numbers $\mu_{(i)} = \{\mu_{s_i,i} \in \mathbb{R} \mid 1 \leq s_i \leq |(i)|\}$ with cardinality $|(i)|$ satisfying $\sum\limits_{(s_i,i)\in\sigma_A(\vec{\alpha})} \mu_{s_i,i} = 0$ for all $\vec{\alpha} \in \sigma(A)$, then

$$\prod_{i=1}^{\binom{d}{k}} \prod_{s_i=1}^{|(i)|} (T_{s_i,i})^{\mu_{s_i,i}} = 1. \tag{24}$$

(vii) Similarly from condition 6,

$$\sum_{i=1}^{\binom{d}{k}} \sum_{s_i=1}^{|(i)|} t_{s_i,i} \cdot \mu_{s_i,i} = 0 \tag{25}$$

Proof. Using the proof from Section 3 of [1] is sufficient to establish from statement (1) to statement (6). Since statement (6) is equivalent to statement (7), we only show the equivalency of statement (7) with former statements. Statements from (1) to (5) give that for the formulations of vector $\vec{b}$ from (19) and matrix C from (20), $Cx = b$ has an optimal solution x. Now, we use lemma 3.2 by the following. Let's consider λ such that $\lambda\left(P(s_i, i)\right) = \mu_{s_i,i}$, then

$$\lambda^T C\left[J(\vec{\alpha})\right] = \sum_{(s_i,i)\in\sigma_A(\vec{\alpha})} \mu_{s_i,i} = 0 \tag{26}$$

By lemma 3.2, we obtain statement (6) and the theorem is fully proved, followed specifically by expanding the result $\vec{\lambda}^T\vec{b} = 0$ by the expressions of numbers in list $\mu_{(i)}$ and numbers $t_{s_i,i}$. □

Specifically, we know that if there exists a solution x such that $Ax = b$, and vector ω such that $A\omega = 0$, then $x + \omega$ is another solution. Thus, additional solutions can be obtained from the known solution of **Program I** based on statements (6) and (7) in Theorem 3.3.

Theorem 3.4. (Uniqueness) There exists at most one tensor $A' \in V_1 \times V_2 \times \cdots \times V_d$ such that there exist $\binom{d}{k}$ scaling lists $M_{(i)} = \{M_{s_i,i} \mid 1 \leq s_i \leq |(i)|\}$ of positive elements with cardinality $|(i)|$ such that the solution $\left(A', \left\{M_{(i)} \mid 1 \leq i \leq \binom{d}{k}\right\}\right)$ satisfies **Program I**. Furthermore, if $\left(A', \left\{M^1_{(i)} \mid 1 \leq i \leq \binom{d}{k}\right\}\right)$ satisfies **Program I**, then for $\binom{d}{k}$ lists $D_{(i)} = \{D_{s_i,i} \in \mathbb{R}_{>0}| \mid 1 \leq s_i \leq |(i)|\}$ such that

$$\prod_{(s_i,i)\in\sigma_A(\vec{\alpha})} D_{s_i,i} = 1 \qquad \forall \vec{\alpha} \in \sigma(A), \tag{27}$$

and $MD^1_{(i)} = \left\{M^1_{s_i,i} D_{s_i,i} \mid 1 \leq s_i \leq |(i)|, D_{s_i,i} \in D_{(i)}\right\}$, its general solution is

$$\left(A', \left\{MD^1_{(i)} \mid 1 \leq i \leq \binom{d}{k}\right\}\right) \tag{28}$$

Proof. The proof of uniqueness follows from the RZ model, and the general solution follows directly from Theorem 3.3 for the transformed expression of **Program I**. □

4. Algorithm

To simplify the algorithmic representation, we define a function that maps the number of elements in $A_{s,i}$ from an input as a pair (s, i). This establishes a relationship between the number of nonzero elements in $A_{s_i,i}$ and the cardinality of $\sigma_A(s_i, i)$. One important property is that the cardinality of $\sigma_A(s_i, i)$ equals to the number of nonzero elements in $A_{s,i}$. We establish the following algorithm.

4.1. Algorithm for general $k < d$:

SCALING ALGORITHM FOR PROGRAM II

(i) Step 0 [initialization]: Set *count* ← 0. Select $\binom{d}{k}$ list of scaling $m_{(i)} = \{m_{s,i} \in \mathbb{R} \mid 1 \leq s \leq |(i)|\}$ and $\binom{d}{k}$ sets of subtensors (i) by definition 2.1 for d as the dimension of tensor A. Define tensor $a^0 \in V_1 \times V_2 \times \cdots \times V_d$ such that

$$a^0(\vec{\alpha}) \equiv \begin{cases} a(\vec{\alpha}) + \sum\limits_{(s_i,i)\in\sigma_A(\vec{\alpha})} m_{s_i,i} & \text{for} \quad \vec{\alpha} \in \sigma(A) \\ 0 & \text{for} \quad \vec{\alpha} \notin \sigma(A) \end{cases} \tag{29}$$

(ii) Step 1 [iterative step over constraint]: Let $i = 1$ and our goal is to iterate until $i = \binom{d}{k}$. For $s_i \in [|(i)|]$, let

$$\begin{aligned} \rho_{s_i} &= [|\sigma_A(s_i, i)|]^{-1} \left[t_{s_i,i} - \sum_{\vec{\alpha}\in\sigma_A(s_i,i)} a^{step}(\vec{\alpha}) \right] \\ a^{step}(\vec{\alpha}) &\leftarrow a^{step}(\vec{\alpha}) + \rho_{s_i} \quad \text{for} \quad \vec{\alpha} \in \sigma_A(s_i, i) \\ m_{s_i,i} &\leftarrow m_{s_i,i} + \rho_{s_i} \end{aligned} \tag{30}$$

We then set $i \leftarrow i + 1$ and continue until $i = \binom{d}{k}$.

(iii) Step 2: Set *count* ← *count* + 1 and return to step 1. We also set the convergence condition as the variance of the tensor elements projected by each of its dimensions is smaller than a very small positive ϵ.

The time complexity of this algorithm is $O\left(\sum_{i=1}^{\binom{d}{k}} \sum_{s_i=1}^{|(i)|} |\sigma_A(s_i, i)|\right) \equiv O\left(\binom{d}{k}|\sigma(A)|\right)$. This complexity is unaffected by the final step of converting our solution over logarithms to our desired solution as $A(\vec{\alpha}) = \exp(a(\vec{\alpha}))$, which is proportional to the number of nonzero elements of A.

5. Discussion

In this paper we have generalized the RZ algorithm [1] as specialized for the unique canonical scaling of matrices [6, 8, 5] for the canonical scaling of arbitrary k-dimensional subtensors of a given n-dimensional tensor. This algorithm provides a means for generalizing scale-invariant methods presently applied to matrices to higher-dimensional structures, e.g., extending the scaling-based image interpolation method of [8] to the interpolation of a video sequence treated as a 3d array/tensor.

6. Acknowledgements

This work was sponsored in part by University of Missouri's A&S Undergraduate Research and Creativity Activity Mentorship Program. The authors thank Mark Ashbaugh for helpful comments.

References

[1] U.G. Rothblum and S.A. Zenios, "Scalings of Matrices Satisfying Line-Product Constraints and Generalizations," *Linear Algebra and Its Applications*, 175:159-175, 1992.

[2] Stephan Rabanser, Oleksandr Shchur, and Stephan Gunnemann, "Introduction to tensor decompositions and their applications in machine learning," *arXiv preprint*, arXiv:1711.10781, 2017.

[3] G.H. Golub, C.F. van Loan, *Matrix Computations*, JHU Press, ISBN: 1421407949 9781421407944, 2013.

[4] Jeffrey Uhlmann, " A Scale-Consistent Approach for Recommender Systems," (arxiv.org/pdf/1905.00055.pdf), 2019.

[5] Jeffrey Uhlmann, "Unit Consistency, Generalized Inverses, and Effective System Design Methods," arXiv:1604.08476v2 [cs.NA] 11 Jul 2017 (2015).

[6] Jeffrey Uhlmann, "A Generalized Matrix Inverse that is Consistent with Respect to Diagonal Transformations," *SIAM Journal on Matrix Analysis* (SIMAX), 2018.

[7] Jeffrey Uhlmann, "A Rank-Preserving Generalized Matrix Inverse for Consistency with Respect to Similarity," *IEEE Control Systems Letters*, ISSN: 2475-1456, 2018.

[8] Rumana Aktar, K. Palaniappan, and Jeffrey Uhlmann, "A Method for Real-Time Interpolation of Packet-Loss Blocks in Streaming Video," *Proceedings of 48th Annual IEEE AIPR 2019*, (Washington, DC), 6 October, 2019.

[9] J.K. Uhlmann, "On the Relative Gain Array (RGA) with Singular and Rectangular Matrices," *Applied Mathematics Letters*, Vol. 93, 2019.

[10] Bo Zhang and Jeffrey Uhlmann, "Applying a Unit-Consistent Generalized Matrix Inverse for Stable Control of Robotic Systems," *ASME J. of Mechanisms and Robotics*, 11(3), 2019.

[11] Bo Zhang and Jeffrey Uhlmann, "Examining a Mixed Inverse Approach for Stable Control of a Rover," *International Journal of Control Systems and Robotics*, 5, 1-7, 2020.