FULLY CONNECTED TENSOR NETWORK DECOMPOSITION AND ITS APPLICATION TO HIGHER ORDER TENSOR COMPLETION

19MAT212 PROJECT REPORT

DONE BY

TARUN SANJEEV S [CB.EN.U4AIE19064]

DARSHAN D [CB.EN.U4AIE19021]

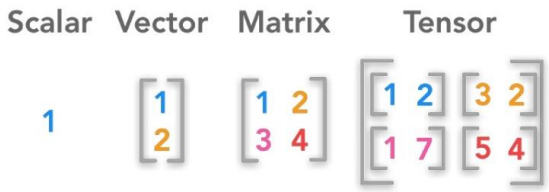
ROHITH G [CB.EN.U4AIE19026]

TABLE OF CONTENTS

INTRODUCTION TO TENSORS

When we represent data for machine learning, we generally do it numerically. Especially when referring specifically of neural network data representation, this is accomplished using something known as the tensor. A tensor can be described as a tool to represent data in N dimensions. Tensors are generalizations of matrices to N-dimensional space.

Mathematically speaking, tensors are more than simply a data containing tool. Aside from holding numeric data, tensors can also include descriptions of the valid linear transformations between tensors. Examples of such transformations, or relations, include the cross product and the dot product. From a computer science perspective, it can be helpful to think of tensors as being objects in an object-oriented sense, as opposed to simply being a data structure.



A tensor is a mathematical entity that lives in a structure and interacts with other mathematical entities. If one transforms the other entities in the structure in a regular way, then the tensor must obey a related transformation rule. This “dynamical” property of a tensor is the key that distinguishes it from a mere matrix. It’s a team player whose numerical values shift around along with those of its teammates when a transformation is introduced that affects all of them.

Any rank 2 tensor can be represented as a matrix, but not every matrix is really a rank 2 tensor. The numerical values of a tensor’s matrix representation depend on what transformation rules have been applied to the entire system.

In this project we will see about tensors and tensor decompositions. Then we see the necessity of the involvement of the fully connected tensor network decompositions and its importance. Finally we see how they can be implemented and how they can be used and manipulated in various different platforms.

SOME KEY TERMS ABOUT TENSORS

TENSOR ORDER

The order of a tensor is the number of its dimensions. Scalars can therefore be interpreted as zeroth-order tensors, vectors as first-order tensors, and matrices as second-order tensors. We will refer to tensors of order three or higher as higher order tensors. Notation-wise, scalars are denoted by lower case letters $x \in R^{I1}$, vectors by lower case bold letters $\mathbf{x} \in R^{I1XI2}$, matrices by upper case bold letters $\mathbf{X} \in R^{I1\times I2}$, and higher order tensors by upper case bold Euler script letters $\boldsymbol{\chi} \in R^{I1\times I2\times\ldots\times IN}$. The Is denote the number of elements in the respective dimension.

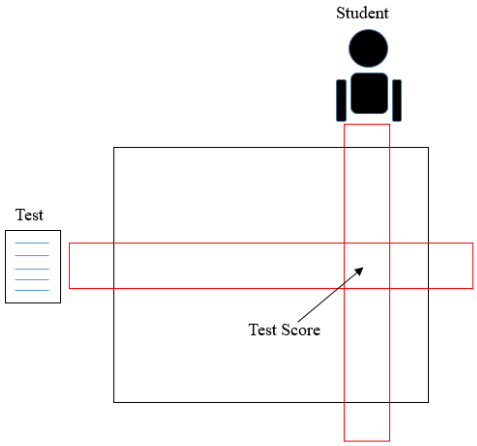
TENSOR INDEXING

We can create subarrays (or subfields) by fixing some of the given tensor's indices. Fibers are created when fixing all but one index, slices (or slabs) are created when fixing all but two indices. For a third order tensor the fibers are given as $\boldsymbol{x}_{:jk} = \boldsymbol{x}_{jk}$ (column), $\boldsymbol{x}_{i:k}$ (row), and $\boldsymbol{x}_{ij:}$ (tube); the slices are given as $\boldsymbol{X}_{::k} = \boldsymbol{X}_{k}$ (frontal), $\boldsymbol{X}_{:j:}$ (lateral), $\boldsymbol{X}_{i::}$ (horizontal).

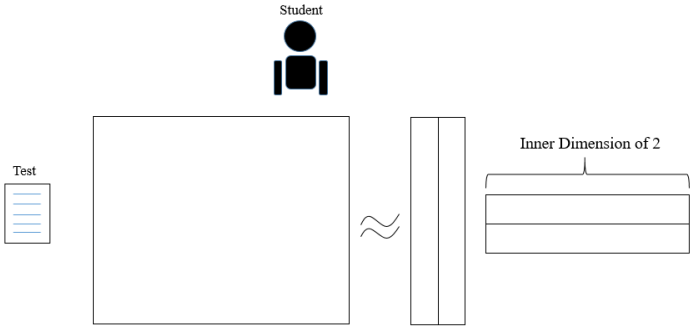
A BRIEF EXPLORATION OF TENSOR DECOMPOSITION

We draw the intuitive idea and necessity behind the theorizing of tensor decomposition through a famous thought experiment known through the history as Spearman's Hypothesis by Charles Spearman, the inventor of Factor Analysis, dated back to 1904. Spearman's hypothesis begins with a controversial argument, he believed that there are only two types of intelligence – Eductive and Reproductive. Eductive is simply defined as "The ability to make sense of out complexity" and Reproductive as "The ability to store and reproduce information".

The experiment begins with a data of thousands of student's test scores of around ten tests. The data was arranged in to a matrix where each row represented a test subject and each column represented a different student. Spearman suggested that this matrix can be written as some rank two factorization representing uniquely eductive and reproductive knowledge.

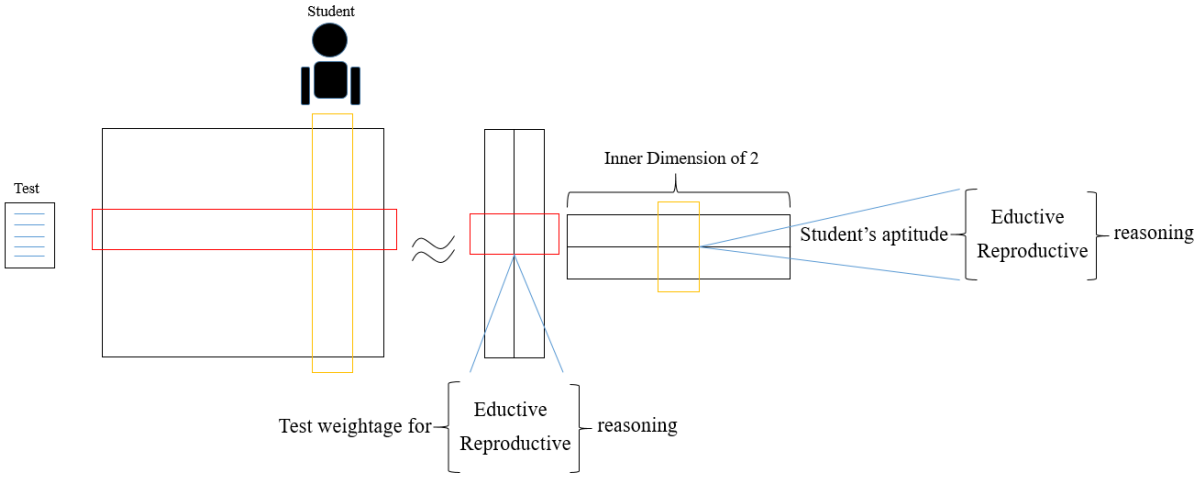


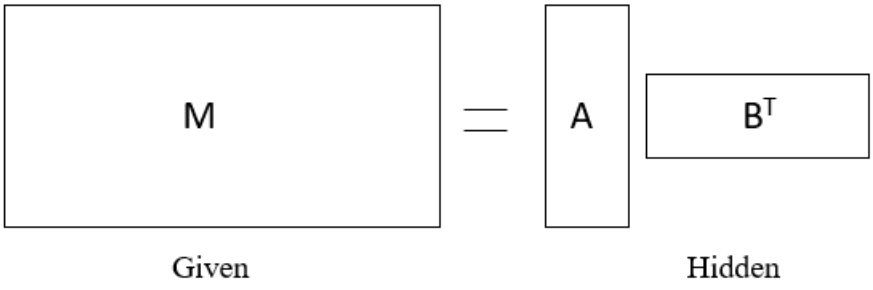
Data of tests and students arranged in a matrix



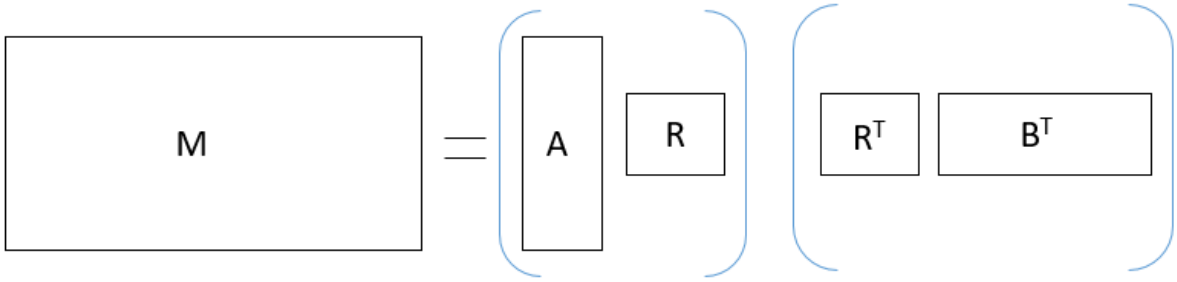
Spearman assumed not only that these matrix had lower rank, but also some meaningful, interpretable low rank factorization. This simply interprets as some rank two factorization, where for a particular row which represents a test, indexing into to the first of the factorized matrices of a length two vector for a given test for the quantitative weightage of eductive

and reproductive reasoning to the corresponding test. The same indexing on the second matrix represents the student's aptitude of this test on eductive and reproductive reasoning. Charles Spearman hoped for a low rank factorization which is interpretable. Hence a student's performance in a particular test can be explained in terms of the inner product of two different length two vector. This thought process was the origins of Factor Analysis. We can brief the theory conceptually as when having a large data of observation, in this scenario the test-student matrix, explained away using much fewer latent or unobserved variables through a low rank decomposition, that is the hidden features of the test and the student together which determine how well the student performed in the test. This called the Rank Decomposition of Matrices. We can rewrite as $M = AB^T$ with $M \in R^{(m\times n)}$, $A \in R^{(m\times n)}$, $B \in R^{(r\times m)}$, where r represents the rank of the decomposition, aiming to explain the matrix M through r different latent factors.





Assuming there is some true factorization that tests and validates Spearman's Hypothesis. A matrix data represented as the product of low rank A times BT. Now incorporation of any rotations to these factors and would still obtain a valid factorization. Supposedly if a Rotation Matrix of R is used in transformation of the factors, we obtain two new matrices, A times R and R transpose, which is also its inverse, times B transpose.



Now reverse engineering this procedure, we could apply a completely arbitrary rotation and mix up all of our data to result in an absurd data which is difficult to interpret. We would not know the reason for obtaining that data was on course of existence of the hidden factors that wasn't found, i.e. they are non-unique and a number of A and BT, which can give rise to a specific M. This is called Rotation Problem. Mathematically,

$$\mathrm{M} = \sum_{i=1}^{R} a_i b_i^T$$

Assuming that there are R hidden factors of ai times biT for a low rank matrix M. It cannot be uniquely determined from just M, without conditions defining ai and biT, but this entirely dependent on the defining conditions or constraints, like positive-definiteness or orthogonality. If the conditions applied are weak then it would be difficult to formulate and find ai and biT. Given a matrix M, we would like to approximate it as well as possible with another matrix Mˆ of a lower rank, for Spearman's hypothesis: rank of Mˆ as two. We can formulate it as an optimization problem, minimizing the norm of the difference between the two matrices,

$$\min_{\widehat{M}} \|M - \widehat{M}\| \; with \; \widehat{M} = AB^T$$

But as discussed above its not a viable solution as multiple factors can result in same M. Tensor Decompositions comes to play in this scenario, this limitation of matrices overcome through decomposition or factorization of tensors. Tensor Decompositions render strong uniqueness properties, which is why it could provide a solution to Rotation Problem. Low rank tensor decompositions are unique in ways that matrix decompositions are not. A tensor decomposition is called unique if there exists only one combination of rank one tensors that sum to the initial tensor. As we have seen earlier, every slice or the low rank matrices are just different scaling of the rank one matrices.

$$\mathcal{T}_{(:,\ :,\ k)} = \sum_{i=1}^{R} (a_i \otimes b_i) \otimes c_i^k$$

$\Rightarrow \mathcal{T}_{(:,\ :,\ k)} - (a \otimes b)c^k$, i.e. subtracting of the scaling of same rank one matrix decreases the rank of each slice if $a = a_i$, $b = b_i$, $c = c_i$ for some i, under some natural constraint. This ensures that we obtain the true factors in this decomposition and not any arbitrary ones. The symbol $\otimes$ represents the generalized outer product or Kronecker product.

In short the notion or the object is to obtain that one rank one slice or matrix, and scale it by different constants for each of the slices, so that subtracting those rank one matrices decreases the rank of the each of the matrix slices through the tensor.

When we subtract we reduce one of the factors and ranks for each of the matrix. This make tensors more rigid over matrices, the dilemma is that in a matrix there are many rank one terms we can subtract off to reduce its rank that has no close relation with the actual hidden factors.

In notation or representation of tensors, the same of these rank one matrices are in need to be in agreement for all slices which is the convincing factor for identifying the true hidden factors and not an arbitrary one. This is a very viable solution to the rotation problem. But computing the rank of a tensor is classified as an NP-Hard problem.

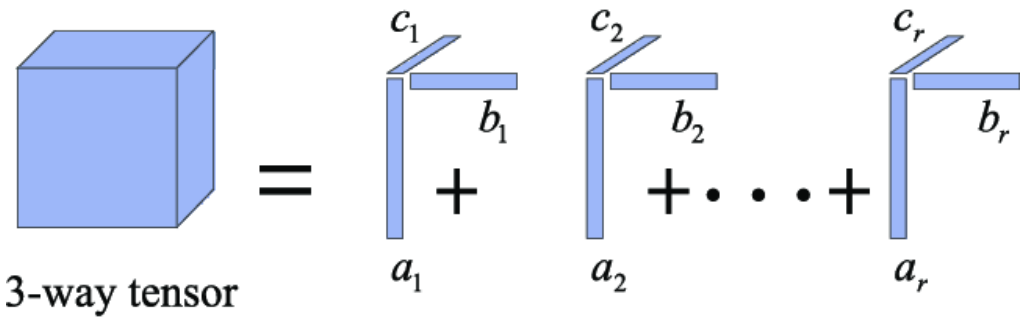
Whereas computing the rank of a matrix is viable and can be solved in linear time and has many significant properties like, row rank is equal to the column rank, which fails for tensors. Tensors holds properties such as,

“There can be an explicit rank three tensor that can be arbitrarily well-approximated by rank two tensor”.

"The best rank k and the best rank k+1 approximations need not share +any rank one factors in common."

"Even tensors with real entries may require complex numbers to find the lowest decomposition, because real rank may not be equivalent to the complex rank".

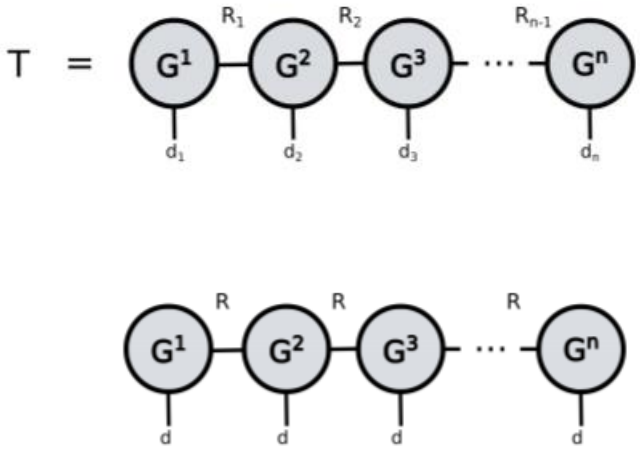
Most tensor problems are defined NP-Hard problem, like computing Singular value over R,C Approximating Eigen Vector over R, Approximating Spectral Norm over R, Rank over R or C etc.



TYPES OF TENSOR NETWORK DECOMPOSITIONS

TENSOR TRAIN DECOMPOSITION

Equipped with the tensor network notation, we can imagine many different decompositions. One of them that we will describe below is the tensor-train. One of the advantages of representing a tensor in this tensor-train representation is compression. One way to see that is by denoting $d = max(d_1, d_2, ..., d_n)$ and $R = max(R_1, R_2, ..., R_n)$ and looking at the following decomposition of τ.



Each core G^s in this decomposition needs $O(dR^2)$ parameters, so the overall train has $O(ndR^2)$ parameters. This is a huge improvement from the exponential number of parameters $O(d^n)$ required to store the original tensor. This seems to good to be true, in fact, in the worst case scenario $R \sim d^{n/2}$ which takes us back to $O(d^n)$. Fortunately for us, for certain natural tensors, we can always choose R to be constant w.r.t n. The TT-rank of a TT decomposition is the expression of the complexity of TT.

TENSOR RING DECOMPOSITION

Tensor-ring decomposition of tensors plays a key role in various applications of tensor network representation in physics as well as in other fields. In most heuristic algorithms for the tensor-ring decomposition, one encounters the problem of local-minima trapping. Particularly, the minima related to the topological structure in the correlation are hard to escape. Therefore, identification of the correlation structure, somewhat analogous to finding matching ends of entangled strings, is the task of central importance.

THE DOWNFALLS OF THESE HIGHER ORDER TENSOR DECOMPOSITION METHODS

There are two limitations to TT and TR decompositions. First, these two decompositions only establish an operation/connection between adjacent two factors, rather than any two factors, which leads to a limited characterization for correlations of tensors. Second, TT decomposition keeps the invariance only when the modes of the target tensor make a reverse permuting, while TR decomposition keeps the invariance only when the modes of the target tensor make a circular shifting or a reverse permuting. These imply that these two decompositions are highly sensitive to the permutation of tensor modes, leading to the inflexibility of decompositions and applications.

FULLY CONNECTED TENSOR NETWORK DECOMPOSITION

To tackle the two limitations, we switch to a fully connected tensor network (FCTN) decomposition, which decomposes an Nth-order tensor into a set of Nth-order factors and establishes a multi-linear operation/connection between any two factors. The proposed FCTN decomposition has the superior capability to characterize directly the intrinsic correlations between any two modes of tensors and is proved to be essentially invariable for any permutations of tensor modes. The main contributions of this paper are summarized as three-folds:

1) We propose an FCTN decomposition, which breaks through the limitations of TT and TR decompositions in terms of correlation characterization and transpositional invariance.

2) We employ the FCTN decomposition to the TC problem and develop an efficient proximal alternating minimization (PAM)-based algorithm to solve it.

3) We theoretically demonstrate the convergence of the developed algorithm by proving the sequence obtained by it globally converges to a critical point (local minima).

The FCTN decomposition aims to decompose an Nth-order tensor into a set of Nth-order factor tensors.

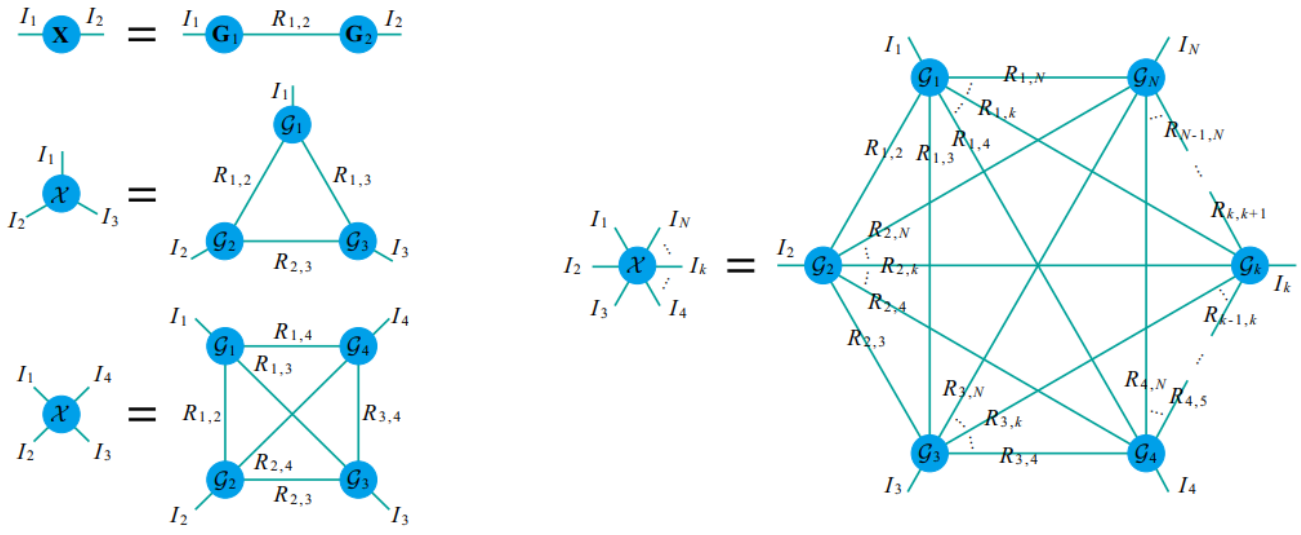
For higher-order tensors, any two FCTN factors G_{k1} and G_{k2} have an equal-sized mode $R_{k1;k2}$ used to conduct the tensor contraction operation, which enables the FCTN decomposition to characterize adequately the intrinsic correlations between any two modes of the target tensor. This indicates an essential advantage of the FCTN decomposition over the TT and TR decompositions, which establish only the connection between adjacent two factors, leading to a limited characterization for correlations of tensors. Besides, the FCTN decomposition can degenerate to the TT and TR decompositions by simply setting the corresponding modes of factors to 1.

In second-order case, it is well known that the matrix factorization is essentially invariable under the transpositional condition, i.e., $\mathbf{X} = \mathbf{G}_1\mathbf{G}_2 \Rightarrow \mathbf{X}^T = \mathbf{G}^T_2\mathbf{G}^T_1$. Naturally, it is expected to extend this property to higher-order cases.

There is another essential advantage of the FCTN decomposition as compared with the TT and TR decompositions. More specifically, the FCTN decomposition is essentially invariable, no matter how to permute the modes of the target tensor. But TR decomposition keeps the invariance only when the modes of the target tensor make a circular shifting or a reverse permuting. And TT decomposition keeps the invariance only when the modes of the target tensor make a reverse permuting.

Since the FCTN decomposition aims to characterize the intrinsic correlations between any two modes by establishing a connection between any two factors, the factors have to be designed as Nth-order tensors, which inevitably leads to the increment of the storage cost as compared to TT and TR decompositions.

To illustrate the FCTN decomposition vividly, Figure gives a graphical representation of it.



For second-order tensors, the FCTN decomposition is actually the matrix factorization and the FCTN-rank is actually the matrix rank. Furthermore, for higher-order tensors, any two FCTN factors Gk1 and Gk2 have an equal-sized mode Rk1,k2 used to conduct the tensor contraction operation, which enables the FCTN decomposition to characterize adequately the intrinsic correlations between any two modes of the target tensor. This indicates an essential advantage of the FCTN decomposition over the TT and TR decompositions, which establish only the connection between adjacent two factors, leading to a limited characterization for correlations of tensors. Besides, the FCTN decomposition can degenerate to the TT and TR decompositions by simply setting the corresponding modes of factors to 1. In second-order case, it is well known that the matrix factorization is essentially invariable under the transpositional condition, i.e., X = G1G2 ⇔ XT = GT2GT1. Naturally, it is expected to extend this property to higher-order cases.

Theorem 2 (Transpositional Invariance) Supposing that an Nth-order tensor X has the following FCTN decomposition: $\mathcal{X} = FCTN(\mathcal{G}_1, \mathcal{G}_2, \cdots, \mathcal{G}_N)$Then, its vector n-based generalized tensor transposition X~ n can be expressed as $\vec{\mathcal{X}}^{\mathbf{n}} = FCTN(\vec{\mathcal{G}}^{\mathbf{n}}_{n_1}, \vec{\mathcal{G}}^{\mathbf{n}}_{n_2}, \cdots, \vec{\mathcal{G}}^{\mathbf{n}}_{n_N})$, where n = (n1, n2,· · ·, nN) is a reordering of the vector (1, 2, · · · , N).

Theorem 2 illustrates another essential advantage of the FCTN decomposition as compared with the TT and TR decompositions. More specifically, the FCTN decomposition is essentially invariable, no matter how to permute the modes of the target tensor. But TR decomposition keeps the invariance only when the modes of the target tensor make a circular shifting or a reverse permuting. And TT decomposition keeps the invariance only when the modes of the target tensor make a reverse permuting.

The following theorem presents that the FCTN-ranks can bound the rank of all generalized tensor unfolding.

Theorem 3 Supposing that an Nth-order tensor X can be represented by Equation (1), the following inequality holds:

$$\mathrm{Rank}\left(\mathbf{X}_{[n_{1:d};n_{d+1:N}]}\right) \leq \prod_{i=1}^{d}\prod_{j=d+1}^{N} R_{n_i,n_j},$$

Where Rni,nj = Rnj ,ni if ni > nj and (n1, n2,· · ·, nN) is a reordering of the vector (1, 2, · · · , N). Since the FCTN decomposition aims to characterize the intrinsic correlations between any two modes by establishing a connection between any two factors, the factors have to be designed as Nth-order tensors, which inevitably leads to the increment of the storage cost as compared to TT and TR decompositions. For an Nth-order X ∈ R I×I×···×I, whose FCTN-ranks are the same value R1, the FCTN decomposition requires O(NIRN−1 1) parameters to express it. It seems to stay on the same order of magnitude with that of the Tucker decomposition (O (NIR2 + RN2) parameters). But when we express real-world data, the required FCTN rank R1 is usually far less than Tucker rank R2, because the FCTN decomposition uses R N−1 1 to bound Tucker rank R2 (as shown in Theorem 3). This indicates that the FCTN decomposition is superior to the Tucker decomposition regarding the storage cost.

Definition 5 (FCTN Composition) We call the process of generating X by its FCTN factors Gk (k = 1, 2, · · · N) as the FCTN composition, which is also denotes as $FCTN(\{\mathcal{G}_k\}_{k=1}^{N})$.

Furthermore, if one of the factors Gt (t ∈ {1, 2, · · · , N}) does not participate in the composition, we denote it as $FCTN(\{\mathcal{G}_k\}_{k=1}^{N}, /\mathcal{G}_t)$.

Theorem 4: Supposing that $\mathcal{X} = FCTN(\{\mathcal{G}_k\}_{k=1}^{N})$ and $\mathcal{M}_t = FCTN(\{\mathcal{G}_k\}_{k=1}^{N}, /\mathcal{G}_t)$, we obtain that:

$$\mathbf{X}_{(t)} = (\mathbf{G}_t)_{(t)}(\mathbf{M}_t)_{[m_{1:N-1};n_{1:N-1}]},$$

where

$$m_i = \begin{cases} 2i, & if\ i < t, \\ 2i-1, & if\ i \geq t, \end{cases} \quad and \quad n_i = \begin{cases} 2i-1, & if\ i < t, \\ 2i, & if\ i \geq t. \end{cases}$$

Theorem 4 reveals the relationship between one FCTN factor and the composition of the other factors. It is of great importance to the computation of the FCTN decomposition since computing one factor usually needs to fix the others.

FCTN Decomposition-Based TC Method

Model of the Algorithm

Here the FCTN decomposition is applied to one representative task, i.e., TC, which aims to recover missing elements of a higher-order tensor from its an incomplete observation. Giving an incomplete observation of $\mathcal{F} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ the target tensor $X \in R\ I_1 \times I_2 \times \cdots \times I_N$, the proposed FCTN decomposition-based TC (FCTN-TC) model can be formulated as

$$\min_{\mathcal{X},\mathcal{G}} \ \frac{1}{2}\|\mathcal{X} - \text{FCTN}(\mathcal{G}_1, \mathcal{G}_2, \cdots, \mathcal{G}_N)\|_F^2 + \iota_{\mathbb{S}}(\mathcal{X}), \qquad (2)$$

where $\mathcal{G} = (\mathcal{G}_1, \mathcal{G}_2, \cdots, \mathcal{G}_N)$ and

$$\iota_{\mathbb{S}}(\mathcal{X}) := \begin{cases} 0, & \text{if } \mathcal{X} \in \mathbb{S}, \\ \infty, & \text{otherwise}, \end{cases} \quad \text{with } \mathbb{S} := \{\mathcal{X} : \mathcal{P}_\Omega(\mathcal{X} - \mathcal{F}) = 0\}.$$

Here Ω denotes the index of the known elements and PΩ(X) is a projection operator which projects the elements in Ω to themselves and all others to zeros. Since all optimization variables are coupled with each other, we employ the framework of PAM (Attouch, Bolte, and Svaiter 2013) to solve (2), whose solution can be obtained by alternately updating.

$$\begin{cases} \mathcal{G}_k^{(s+1)} = \underset{\mathcal{G}_k}{\text{argmin}} \Big\{ f(\mathcal{G}_{1:k-1}^{(s+1)}, \mathcal{G}_k, \mathcal{G}_{k+1:N}^{(s)}, \mathcal{X}^{(s)}) \\ \qquad\qquad + \frac{\rho}{2}\|\mathcal{G}_k - \mathcal{G}_k^{(s)}\|_F^2 \Big\}, \ k = 1, 2, \cdots, N, \quad (3) \\ \mathcal{X}^{(s+1)} = \underset{\mathcal{X}}{\text{argmin}} \Big\{ f(\mathcal{G}^{(s+1)}, \mathcal{X}) + \frac{\rho}{2}\|\mathcal{X} - \mathcal{X}^{(s)}\|_F^2 \Big\}, \end{cases}$$

where f(G, X) is the objective function of (2) and ρ > 0 is a proximal parameter.

1) Update Gk: According to Theorem 4, the Gk (k = 1, 2,· · ·, N)-sub problems can be rewritten as

$$\mathcal{G}_k^{(s+1)} = \underset{\mathcal{G}_k}{\text{argmin}} \Big\{ \frac{\rho}{2}\|(\mathbf{G}_k)_{(k)} - (\mathbf{G}_k^{(s)})_{(k)}\|_F^2 + \frac{1}{2}\|\mathbf{X}_{(k)}^{(s)} - (\mathcal{G}_k)_{(k)}(\mathbf{M}_k^{(s)})_{[m_{1:N-1}; n_{1:N-1}]}\|_F^2 \Big\}, \qquad (4)$$

where and vectors m and n have the same setting as that in Theorem 4. The problem (4) can be directly solved as

$$(\mathbf{G}_k^{(s+1)})_{(k)} = \left[\mathbf{X}_{(k)}^{(s)}(\mathbf{M}_k^{(s)})_{[n_{1:N-1};m_{1:N-1}]} + \rho(\mathbf{G}_k^{(s)})_{(k)}\right] \left[(\mathbf{M}_k^{(s)})_{[m_{1:N-1};n_{1:N-1}]}(\mathbf{M}_k^{(s)})_{[n_{1:N-1};m_{1:N-1}]} + \rho\mathbf{I}\right]^{-1}, \quad (5)$$

and $\mathcal{G}_k^{(s+1)} = \texttt{GenFold}\big((\mathbf{G}_k^{(s+1)})_{(k)}, k; 1, \cdots, k-1, k+1, \cdots, N\big).$

2) Update X : The X -subproblem has the following closed-form solution since it is a least square problem

$$\mathcal{X}^{(s+1)} = \mathcal{P}_{\Omega^c}\left(\frac{\text{FCTN}\big(\{\mathcal{G}_k^{(s+1)}\}_{k=1}^N\big) + \rho\mathcal{X}^{(s)}}{1+\rho}\right) + \mathcal{P}_{\Omega}(\mathcal{F}). \quad (6)$$

Computational Complexity Analysis

For an Nth-order incomplete tensor, $\mathcal{F} \in \mathbb{R}^{I\times I\times\cdots\times I}$ we analyze the computational complexity of the proposed FCTN-TC method by simply setting the FCTN ranks $R_{k1,k2}$ ($1 \le k_1 < k_2 \le N$ and $k_1, k_2 \in N+$) as the same value R. The computational cost lies on two part: 1) updating G_k ($k = 1, 2, \cdots, N$) and 2) updating X . In (5), updating G_k involves the FCTN composition, the matrix multiplication, and the matrix inversion, which costs

$$\mathcal{O}\big(N\sum_{k=2}^{N} I^k R^{k(N-k)+k-1} + NI^{N-1}R^{2(N-1)} + NR^{3(N-1)}\big)$$

In (6), updating X requires the FCTN composition costing $\mathcal{O}\big(\sum_{k=2}^{N} I^k R^{k(N-k)+k-1}\big)$.

Therefore the whole computational complexity at each stage is

$$\mathcal{O}\big(N\sum_{k=2}^{N} I^k R^{k(N-k)+k-1} + NI^{N-1}R^{2(N-1)} + NR^{3(N-1)}\big).$$

Convergence Analysis

In this section, a theoretical guarantee for the convergence of the developed PAM-based algorithm is provided.

Theorem 5: The sequence $\{\mathcal{G}^{(s)}, \mathcal{X}^{(s)}\}_{s\in\mathbb{N}}$ obtained by the Algorithm 1 globally converges to a critical point of (2). To prove the Theorem 5, we only need to justify that the following four conditions hold (Attouch, Bolte, and Svaiter 2013):

1) $\mathcal{G}^{(s)}$ and $\mathcal{X}^{(s)}$ $(s \in \mathbb{N})$ are bounded.
2) $f(\mathcal{G}, \mathcal{X})$ is a proper lower semi-continuous function;
3) $f(\mathcal{G}, \mathcal{X})$ satisfies the K-Ł property at $\{\mathcal{G}^{(s)}, \mathcal{X}^{(s)}\}_{s\in\mathbb{N}}$;
4) $\{\mathcal{G}^{(s)}, \mathcal{X}^{(s)}\}_{s\in\mathbb{N}}$ satisfies Lemmas 1 and 2.

Lemma 1 (Sufficient Decrease)

Letting $\{\mathcal{G}^{(s)}, \mathcal{X}^{(s)}\}_{s\in\mathbb{N}}$ be the sequence obtained by the Algorithm 1, then it satisfies

$$f(\mathcal{G}_{1:k}^{(s+1)}, \mathcal{G}_{k+1:N}^{(s)}, \mathcal{X}^{(s)}) + \frac{\rho}{2}\|\mathcal{G}_k^{(s+1)} - \mathcal{G}_k^{(s)}\|_F^2 \leq f(\mathcal{G}_{1:k-1}^{(s+1)}, \mathcal{G}_{k:N}^{(s)}, \mathcal{X}^{(s)}),\ k = 1, 2, \cdots, N;$$

$$f(\mathcal{G}^{(s+1)}, \mathcal{X}^{(s+1)}) + \frac{\rho}{2}\|\mathcal{X}^{(s+1)} - \mathcal{X}^{(s)}\|_F^2 \leq f(\mathcal{G}^{(s+1)}, \mathcal{X}^{(s)}).$$

Lemma 2 (Relative Error)

Letting $\{\mathcal{G}^{(s)}, \mathcal{X}^{(s)}\}_{s\in\mathbb{N}}$ be the sequence obtained by the Algorithm 1, then there exists $\mathcal{A}_k^{(s+1)} \in \mathbf{0}$ *and* $\mathcal{A}^{(s+1)} \in \partial_{\mathcal{X}} \iota_{\mathbb{S}}(\mathcal{X}^{(s+1)})$ Satisfied

$$\|\mathcal{A}_k^{(s+1)} + \nabla_{\mathcal{G}_k} h(\mathcal{G}_{1:k}^{(s+1)}, \mathcal{G}_{k+1:N}^{(s)}, \mathcal{X}^{(s)})\|_F \leq \rho\|\mathcal{G}_k^{(s+1)} - \mathcal{G}_k^{(s)}\|_F,\ k = 1, 2, \cdots, N;$$

$$\|\mathcal{A}^{(s+1)} + \nabla_{\mathcal{X}} h(\mathcal{G}^{(s+1)}, \mathcal{X}^{(s+1)})\|_F \leq \rho\|\mathcal{X}^{(s+1)} - \mathcal{X}^{(s)}\|_F,$$

where $h(\mathcal{G}, \mathcal{X}) = \frac{1}{2}\|\mathcal{X} - FCTN(\{\mathcal{G}_k\}_{k=1}^N)\|_F^2$.

APPLICATION

In recent years, the application of tensors has become more widespread in fields that involve data analytics and numerical computation. Due to the explosive growth of data, low-rank tensor decompositions have become a powerful tool to harness the notorious curse of dimensionality. The main forms of tensor decomposition include CP decomposition, Tucker decomposition, tensor train (TT) decomposition, etc. Each of the existing TT decomposition algorithms, including the TT-SVD and randomized TT-SVD, is successful in the field, but neither can both accurately and efficiently decompose large-scale sparse tensors. Based on previous research, this paper proposes a new quasi-best fast TT decomposition algorithm for large-scale sparse tensors with proven correctness and the upper bound of its complexity is derived. In numerical experiments, we verify that the proposed algorithm can decompose sparse tensors faster than the TT-SVD, and have more speed, precision and versatility than randomized TT-SVD, and it can be used to decomposes arbitrary high-dimensional tensor without losing efficiency when the number of non-zero elements is limited. The new algorithm implements a large-scale sparse matrix TT decomposition that was previously unachievable, enabling tensor decomposition based algorithms to be applied in larger-scale scenarios.

CONCLUSION

The FCTN decomposition, which factorizes an Nthorder tensor into a set of Nth-order factors with full connections. The FCTN decomposition showed its outstanding capability to adequately characterize the correlations between any two modes of tensors and was proved to be essentially transpositional invariable. Experimental results demonstrated that the FCTN-TC method delivered an overall better performance than the compared ones.