ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 5, Issue 7, July 2016

# SCALABLE HIGH DIMENSIONAL MINING FOR BIG DATA

**1.** Diksha Negi , **2**.Sneh Chauhan, **3**.Assoc. Prof. Mr. Manish Sharma
Department of Computer Science and Engineering Graphic Era University Dehradun

## ABSTRACT

Tucker decomposition (TD) is a powerful tool for the removal of nonnegative or non-useful parts and physically significant latent components from high-dimensional tensor data while preserving the natural multi-linear structure of data. However, as the data tensor often has multiple modes and is large-scale, existing TD algorithms suffer from a very high computational difficulty in terms of both storage and computation time, which has been one major problem for practical applications of TD. To overcome existing disadvantages, we show how linear rank approximation (LRA) of tensors is able to significantly simplify the computation of the gradients of the cost function, upon which a family of well-organized first-order TD algorithms are developed. Besides intensely reducing the storage complexity and running time, the new algorithms are quite flexible and robust to noise because any well-established LRA-ALS Bisection approaches can be applied. We also show how density incorporating abnormal Tensor substantially improves the uniqueness property and partially alleviates the curse of dimensionality of the Tucker decompositions. The Proposed technique strongly provide accurate dimension and efficiency calculation of N*N factor, it work with 2-D and 3-D. Simulation results on synthetic and real-world data justify the validity and high efficiency of the proposed algorithms.

**Keyword:** ***Tensor, Tucker decompositions, alternating least squares, bisection method etc.***

## I.INTRODUCTION

A tensor is a multidimensional array. More formally, an N-way or Nth-order tensor is an element of the tensor product of N vector spaces, each of which has its own coordinate system. This notion of tensors is not to be confused with tensors in physics and engineering (such as stress tensors) [175], which are generally referred to as tensor fields in mathematics [69]. A third-order tensor has three indices as shown in Figure 1.1. A first-order tensor is a vector, a second-order tensor is a matrix, and tensors of order three or higher are called higher-order tensors. The goal of this survey is to provide an overview of higher-order tensors and their decompositions. Though there has been active research on tensor decompositions and models (i.e., decompositions applied to data arrays for extracting and explaining their properties) for four decades, very little of this work has been published in applied mathematics journals. Therefore, we wish to bring this research to the attention of SIAM readers.

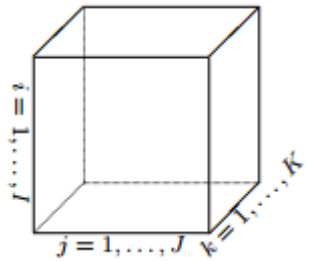


**Fig. 1.1: A third-order tensor: $X \in^{R\, I \times J \times K}$**

The N-way Toolbox for MATLAB, by Andersson and Bro [9], provides a large collection of algorithms for computing different tensor decompositions. It provides methods for computing CP and Tucker, as well as many of the other models such as multilinear partial least squares (PLS). Additionally, many of the methods can handle constraints (e.g., nonnegativity) and missing data. CuBatch is a graphical user interface in MATLAB for the analysis of data that is built on top of the N-way Toolbox. Its focus is data centric, offering an environment for preprocessing data through diagnostic assessment, such as jack-knifing and bootstrapping. The interface allows custom extensions through an open architecture.

## II. PROBLEM FORMULATION

To evaluate such tensor data, numerous tensor decompositions are projected in the previous work. The Tucker decomposition [1] is has been applied in many different domains such as web search mining , network forensics and social network. Despite its popularity, how to apply Tucker on a large sparse tensor is still an open problem. One surprising phenomenon is perceived by specialists:

Despite that both the input (huge, sparse) tensor and the output (small, dense, factorized) tensor can be stored in memory, memory overflows may occur during the Tucker decomposition process. For these challenge we proposed a new technique namely ALS- Tucker- Bisection method which work as a efficiency calculation and minimize the size dimension.

## III . SYSTEM MODEL

### A . ALS (Alternating Least Square)

Tensor decomposition solves iteratively the below Equation(1) by an Alternating Least Squares algorithm which calculates concentration **C** and pure spectra $\mathbf{S^T}$ matrices that optimally fit the experimental data matrix **D** with non-modelled residuals **E**.

$$\mathbf{D} = \mathbf{CS^T} + \mathbf{E} \quad (1)$$

This optimization is carried out for a proposed number of components and using original estimates of either **C** or $\mathbf{S^T}$. In this work, initial estimates were determined using a pure variable method based on the Bisection approach [1].

ALS multistep analysis of multiple independent experiments run under different experimental conditions is a useful and powerful strategy to improve resolution. This strategy implies the analysis of a column-wise and row-wise data matrix, in which the resolved pure spectra of the same species are common for all experiments and experiment-to-experiment variation is allowed for the resolved concentration profiles. Runs combined in the multi-set structure can be of different nature and size, e.g., runs from an experimental design, calibration and/or test runs, spectral information about pure compounds, etc. depending on the purpose of the analysis. Equation 2 shows the bilinear ALS model for a multi-set system ($\mathbf{D_1}$, $\mathbf{D_2}$, ..., $\mathbf{D_n}$), formed by $n\mathbf{D_i}$ experiments, which is expressed by a common pure spectra matrix $\mathbf{S^T}$ and sub-matrices of process profiles $\mathbf{C_1}$, $\mathbf{C_2}$, …, $\mathbf{C_n}$ related to $\mathbf{D_1}$, $\mathbf{D_2}$ … $\mathbf{D_n}$ respectively.

$$\begin{bmatrix}\mathbf{D_1}\\ \mathbf{D_2}\\ \vdots\\ \mathbf{D_n}\end{bmatrix} = \begin{bmatrix}\mathbf{C_1}\\ \mathbf{C_2}\\ \vdots\\ \mathbf{C_n}\end{bmatrix}\mathbf{S^T} + \begin{bmatrix}\mathbf{E_1}\\ \mathbf{E_2}\\ \vdots\\ \mathbf{E_n}\end{bmatrix} \quad (2)$$

Convergence during the optimization is achieved when in two consecutive iterative cycles, relative differences in standard deviations of the residuals between experimental and ALS calculated data values are less than a previously selected value, in this case 0.1%. Figures of merit of the optimization procedure are the percent of lack of fit (%LOF) and the percent of variance explained (%R2). Lack of fit is defined as the difference among the input data **D** and the data reproduced from the $\mathbf{CS^T}$ product obtained by ALS. This value is calculated according to the expression:

$$\%\, LOF = 100\sqrt{\frac{\sum_{i,j} e_{ij}^2}{\sum_{i,j} d_{ij}^2}} \quad (3)$$

Where $d_{ij}$ designs an element of the input data matrix **D** and $e_{ij}$ is the related residual obtained from the difference between the input element and the ALS reproduction.

## IV. PROPOSED METHOD

---

Algorithm 1: 3-way Tucker-ALS.

---

Require: Tensor $\chi \in R^{I \times J \times K}$, desired core size: P x Q x R

Ensure: Core tensor $g_\chi \in R^{P \times Q \times R}$. and orthogonal factor matrices

$$A \in R^{1 \times P}, B \in R^{1 \times Q}, \text{ and } C \in R^{K \times R}$$

1: Initialize B, C;
2: repeat
3: $Y \leftarrow \chi \times_2 B^T \times_3 C^T$;
4: A ← P leading left singular vectors of $Y_{(1)}$;
5: $Y \leftarrow \chi \times_1 A^T \times_3 C^T$;
6: B ← Q leading left singular vectors of $Y_{(2)}$;
7: $Y \leftarrow \chi \times_1 A^T \times_2 B^T$;
8: C ← R leading left singular vectors of $Y_{(3)}$;
9: $g_\chi \leftarrow Y \times_3 C$;
10: until $\| g_\chi \|$ ceases to increase or the maximum number of outer iterations is exceeded.
11. After than we consider the Bisection (Binary search) Method which is based on the Intermediate Value Theorem (IVT).
12. Suppose a continuous function f, defined on [a, b] is given with f(a) and f(b) of opposite sign.
13. By the IVT, there exists a point p ∈ (a, b) for which f(p) = 0. In what follows, it will be assumed that the root in this interval is unique.

**Tucker model**

The mode-*n* product of a tensor $\boldsymbol{S} \in R^{J1\times J2\times\cdots\times Jn\times\cdots\times JN}$ by a matrix $\boldsymbol{A}(n) \in R^{In\times Jn}$ is defined by

$$[S\times n\boldsymbol{A}(n)]\, j_1\cdots j_{n-1} i n j_{n+1}\cdots j_N = \sum_{Jn-1=1}^{Jn} sj1\cdots jn-1jnjn+1\cdots jNainjn \quad (4)$$

leading to a tensor $\boldsymbol{S}\times n\boldsymbol{A}(n) \in R^{J1\times J2\times\cdots\times In\times\cdots\times JN}$ With the mode-*n* product, a familiar matrix factorization

$\boldsymbol{X} = \boldsymbol{USV}$ *is* written as $\boldsymbol{X} = \boldsymbol{S} \times_1 \boldsymbol{U} \times_2 \boldsymbol{V}$ in the tensor framework. The mode-*n* product has following two properties:

$(\boldsymbol{S} \times n\ \boldsymbol{U}) \times m\ \boldsymbol{V} = (\boldsymbol{S} \times m\ \boldsymbol{V}) \times n\ \boldsymbol{U}$(5)
$(\boldsymbol{S} \times n\ \boldsymbol{U}) \times n\ \boldsymbol{V} = \boldsymbol{S} \times n\ (\boldsymbol{V}\ \boldsymbol{U})$. (6)

The Tucker model seeks a decomposition of an *N*-way tensor $\boldsymbol{X} \in \mathrm{R}^{I1 \times I2 \times \cdots \times IN}$ as mode products of a core tensor $\boldsymbol{S} \in \mathrm{R}^{J1 \times J2 \times \cdots \times JN}$ and *N* mode matrices $\boldsymbol{A}(n) \in \mathrm{R}^{In \times Jn,}$

$\boldsymbol{X} \approx \boldsymbol{S} \times 1\ \boldsymbol{A}^{(1)} \times 2\ \boldsymbol{A}^{(2)} \cdot \cdot \cdot \times N\ \boldsymbol{A}^{(N)}$,(7)

which can be written in an element-wise form as

$$\sum_{J\,1,j\,2,\ldots jN} sj1\mathrm{j}2 \cdots jN\ aij1\ \ ai\ i2j2\ \ aiNjN..\quad (8)$$

The mode-*n* matricization of $\boldsymbol{X}$ in the Tucker model (7), is expressed by Kronecker products of the mode-*n* matricization of the core tensor and mode matrices:

$$\mathrm{X(n)} \approx A^{(n)}\ S_{(n)}\ [\ A^{(n-1)} \otimes \ldots\ldots \otimes A^{(2)} \otimes A^{(1)}$$

$$\otimes \mathrm{A}^{(\mathrm{N})} \otimes \ldots\ldots \otimes \mathrm{A}^{(\mathrm{n}+2)} \otimes \mathrm{A}^{(\mathrm{n}+1)}]^{\mathrm{T}}\ , \qquad (9)$$

Where $\boldsymbol{S}(n)$ is the mode-*n* matricization of the core tensor $\boldsymbol{S}$. The representation (9) plays a crucial role in deriving multiplicative updating algorithms for TD.

## B. BISECTION METHOD

Bisection method is the simplest among all the numerical schemes to solve the transcendental equations. This scheme is based on the intermediate value theorem for continuous functions

---

**Algorithm 2: Steps of BISECTION METHOD**

---

Step 1: Read a,b numbers between which the root is to be found
Step 2: Read e, error value
Step 3: If f (a)>0 and f (b) <0 then
w=a
a=b
b=w
Endif
Step 4: c=(a+b)/2
Step 5: If |f(c)|<e Goto Step 7
Step 6: If f(c)<0 Then
a=c
Else
b=c
Endif
Go to Step 4
Step 7: write c, the approximate root
Step 8: Stop

## V. RESULT

The method is guaranteed to converge. The error bound decreases by half with each iteration

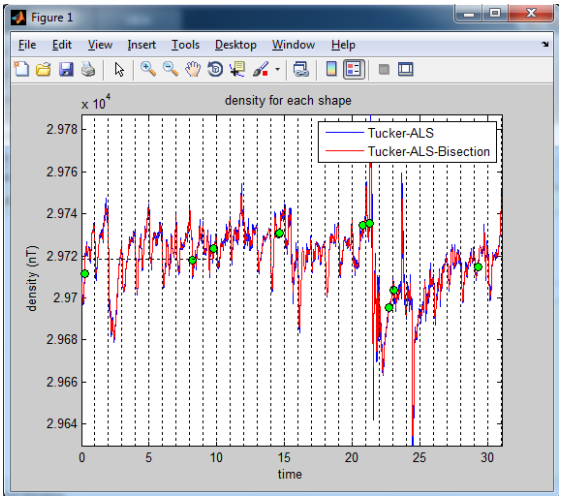


**Figure 4.1: Density for each shape of Tucker – ALS and Tucker-ALS bisection**

The figure 4.1 shows that, green dots for mean value of data, blue line for the Tucker-ALS and red line for the Tucker-ALS- bisection. The proposed method (Tucker-ALS-Bisection) is better as compare to existing Tucker-ALS for the calculation as we can see that proposed method able to mean value calculation from the time 20 to 25 getting four time independent calculation.

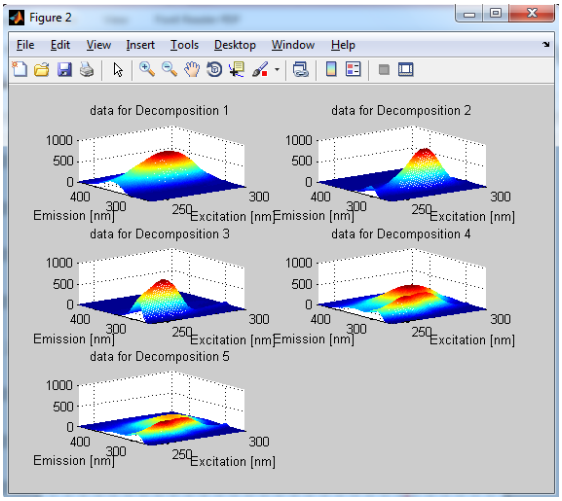


**Figure 4.2: Data normalization and decomposition**

In given figure we minimizing Density from the help of no of independent experiment.

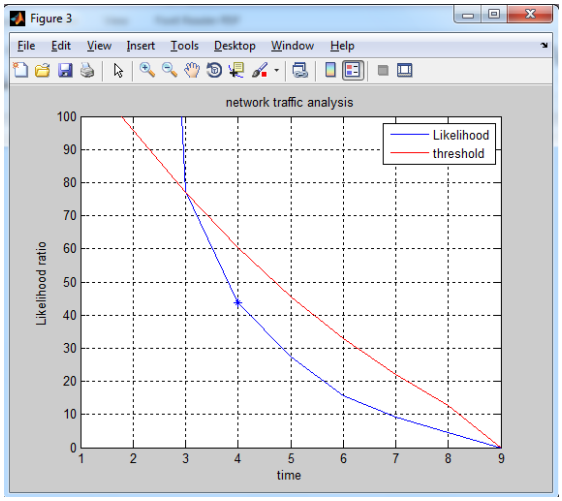


**Figure 4.3: Network traffic analysis of likelihood ratio and threshold**

The Likelihood ratios give us how much we should shift our network traffic particular calculation result. Likelihood ratio must be minimum of the threshold value.

LR = probability of an individual without the condition having the test result

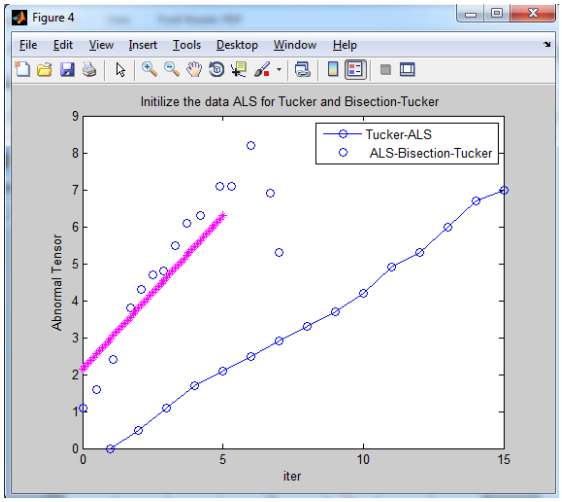


**Figure 4.4: Data initialization for ALS-Tucker and ALS-Bisection-Tucker**

Proposed method is identify a simulated scale in favour of all section that is theatrical scale is directly proportional to the X that conclude 3-way dimension convey to existing theatrical scale.

## VI. CONCLUSION

Tensor decompositions are capable tools for big data analytics as they carry multiple modes and features of data to an integrated framework, which permits us to discover complex internal structures and correlations of data. Unfortunately most existing methods are not designed to meet the major challenges posed by big data analytics. This paper attempted to improve the scalability of tensor decompositions and provides two contributions: A flexible and fast algorithm for the CP decomposition of tensors based on their Tucker compression; a distributed randomized Tucker decomposition approach for arbitrarily big tensors but with relatively low multi-linear rank. These two contributions can deal with huge tensors, even if they are dense. Extensive simulations provide empirical evidence of the validity and efficiency of the proposed algorithms.

## REFERENCE

[1] LeeSaela, InahJeonb, UKangb, "Scalable Tensor Mining" Big Data Research 2 (2015) 82–86.

[2] J. D. Carroll and J. J. Chang. Analysis of individualdifferences in multidimensional scaling via an N-way generalization of 'Eckart-Young' decomposition.*Psychometrika*, 35:283–319, 1970.

[3] A. Cichocki, R. Zdunek, S. Choi, R. J. Plemmons, and S. Amari.Non-negative tensor factorization using alpha and beta divergences.In *Proceedings of IEEE InternationalConference on Acoustics, Speech, and SignalProcessing*, Honolulu, Hawaii, 2007.

[4] L. de Lathauwer, B. de Moor, and J. Vandewalle.A multilinear singular value decomposition. *SIAM J.Matrix Anal. Appl.*, 21(4):1253–1278, 2000.

[5] R. A. Harshman. Foundations of the PARAFAC procedure: Models and conditions for an "Exploratory" multi-modal factor analysis. *UCLA Working Papersin Phonetics*, 16:1–84, 1970. 1

[6] R. A. Harshman. Determination and proof of minimum uniqueness conditions for PARAFAC1.*UCLAWorking Papers in Phonetics*, 22:111–117, 1972.

[7] T. Hazan, S. Polak, and A. Shashua.Sparse image coding using a 3D non-negative tensor factorization.In *Proceedings of International Conference on ComputerVision*, Beijing, China, 2005.

[8] P. O. Hoyer. Non-negative matrix factorization with sparseness constraints.*Journal of Machine LearningResearch*, 5:1457–1469, 2004. 6

[9] C. A. Andersson and R. Bro, The N-way toolbox for MATLAB, Chemometr. Intell. Lab., 52 (2000), pp. 1–4. See also http://www.models. kvl.dk/source/nwaytoolbox/.

[10] R. Bro and H. A. L. Kiers, A new efficient method for determining the number of componentsin PARAFAC models, Journal of Chemometrics, 17 (2003), pp. 274{286.