TECH-CHRONICLE
AN INTERNATIONAL E-JOURNAL ON
EMERGING TRENDS IN SCIENCE, TECHNOLOGY AND MANAGEMENT

TENSOR ANALYSIS OF HIGHER ORDER DIMENSION

Kamini Shukla
Dept.of mathematics, General Engineering
Tulsiramji Gaikwad Patil College of Engineering & Technology,
Nagpur, India
kamini_shukla@yahoo.co.in

Abstract
The "volume" of a tensor is the product of the component dimensions n1, n2 . . . n d and Therein lies thecurse of dimensionality. Undoing the "curse of dimensionality" in any particular area of applications will undoubtedly lead to new technologies and have far-reaching implications throughout mathematics and computational science and engineering. In a very practical sense, multilinear algebra and an appropriate approximation theory are critical to the advancement of mathematics in applications where the curse of dimensionality is the main obstacle. In this paper we discuss algorithms for the reduced rank regression problem and algorithms for the computation of the best multilinear rank approximation of tensors. Tensor and multilinear algebra is an area that attracts more and more attention because of the multidimensional structure of the collected data in various applications. Two classification algorithms are given based on the higher order singular value decomposition (HOSVD).

Keywords— Tensorization;tensor decomposition; Tensor network; tensor matricization

I. INTRODUCTION

In many respects, the "tensor" grand challenge is to enable solutions to the grand challenge problems confronting data-deluged researchers in other fields. Funding initiatives for tensor related research should be considered a priority given current levels of support for information technology, biotechnology, climate modeling, and other critical areas that require sophisticated modeling and the analysis[11] of large, multidimensional datasets. By regarding tensors as tools for describing mathematical objects in high dimensions, it is clear that the development of computational multilinear[5] algebra should parallel the development of analytical tools for spaces of high dimension In many fields of science, engineering, and economics large amounts of data are stored and there is a need to analyze these data in order to extract information for various purposes**.** The development of mathematical models and algorithms is of key importance. Tensor algebra has many similarities but also many striking differences with matrix algebra - e.g., determining tensor rank is NP-hard, while low-rank tensor factorization is unique under mild conditions. Tensor factorizations have already found many applications in signal processing (speech, audio, communications, radar, signal intelligence, machine learning) and well beyond. Tensors are becoming increasingly important, especially for analyzing big data

II. TENSORS

Tensors are geometric objects that describe linear relations between geometric vectors, scalars, and other tensors. Elementary examples of such relations include the **dot product**, the **cross product a**nd maps. Euclidean, often used in physics and engineering **applications** and scalars themselves are also tensors.[1] The concept of a tensor of order two is often conflated with that of a matrix. Tensors of higher order[11] do however capture ideas important in science and engineering, as has been shown successively in numerous areas as they develop. A matrix is a dataset indexed by two indices, say (r, c) for (row, column). A tensor is a dataset indexed by three or more indices, say (i , j, k, · · ·). The term tensor has a different meaning in Physics, however it has been widely adopted in recent years to describe what was previously known as a multi-way array. Matrices are two-way tensors, and they are special because it turns out that there is an interesting dichotomy between two-way and three- or higher-way tensors, with the latter sharing common algebraic properties which are simply very different from those of matrices.

In numerical linear algebra, matrix computations and other mathematical sciences we mostly use quantities as vectors $X \epsilon R^{n}$ and matrices $A \epsilon R^{m \times n}$ for many different purposes. Vectors are usually elements of a vector space and matrices represent linear operators [8] with respect to some basis. Matrices also represent measured data, as a digital image or a collection of sensor signals. A vector is written as a one dimensional array of numbers and single index is used to address its entries. Similarly, a matrix is written as a two dimensional array of numbers whose entries are accessed with two indices. An order n tensor $A \epsilon R^{I_1 \times \ldots \times I_n}$ is a generalization of these algebraic objects to one with n indices. Vectors and matrices are in fact first and second order tensors, respectively. The dimension of A along the different modes

[10] is given by $\mathbf{I_i}$. Tensors are often found in differential geometry where they most of the time (if not exclusively) represent (abstract) multilinear [5] operators[8]. In this paper tensors will, most of the time, constitute multidimensional data arrays, which we want to analyze, model and extract information from. The main research topics are to analyze the basic properties of tensors, generalize the existing linear algebra theory to include tensors and construct efficient algorithms. The connection to applications is often very close.

III TENSOR MATRICIZATION

Another concept that is useful when working with tensors is the notion of matricizing, unfolding or flattening of a tensor. This is an operation for transforming a given multidimensional array into a matrix. For example a 3-tensor $A \in R^{I x J x K}$ can be reshaped to form matrices of dimensions I x JK, J x IK or K x IJ and each matrix has columns. Different matrices, with respect to column pivoting, will be obtained depending on which order the tensor fibers in each case are taken into the matricized forms.

A. Linear systems of equations and linear regression models

Linear systems of equations is one of the most common problems encountered in scientific computing
To solve

$$\mathbf{Ax = b}$$

Where $A \in R^{m x n}$ and $b \in R^{m}$ are given and $x \in R^{n}$ is unknown. Of course, if the vector b is not in the range space of A the equation will not have a solution. If this is the case, one computes a solution X such that AX is, in some measure, the best approximation to b. Mathematically we write

$$\mathbf{min \|AX - b\|}$$

where the norm is often the Euclidean. This problem occurs naturally when one wants to fit a linear model to measured observations. Then the number of measurements is larger than the number of unknowns, i.e. m > n and we have an over determined set of linear equations. The approach to minimize the residual r = AX - b is sound since it is interpreted as to minimize the influence of the errors in the measurements [11].
An alternative way to approach the problem is to relate the observation vector b to the unknown vector X using the linear statistical model

$$\mathbf{AX = b + \epsilon}$$

where we introduce the vector ϵ containing random errors. It is assumed in the standard model that the entries ϵi are uncorrelated, have zero mean and the same variance, i.e.
$E(\epsilon) = 0$; $V(\epsilon) = \sigma^2 I$; where $E(\epsilon)$ denotes the expected value and $V(\epsilon)$ denotes the variance.
A related model is linear reduced-rank regression,

$$\mathbf{b(t\ i) = X\ a(t\ i) + e(t\ i);\ i = 1; 2; : : : ;N}$$

where the unknown regression matrix $X \in R^{m x p}$ is constrained to have

$$\mathbf{rank\ (X) = k < min(m; p)}$$

The vectors $\mathbf{b(t_i) \in R^{m}}$ and $\mathbf{a(ti) \in R^{p}}$ are measurements at different time steps t_i and $e(t_i)$ are the corresponding errors. It is assumed that $e(t_i)$ is temporally white noise, normally distributed with unknown covariance matrix $E(e(t)e(t)^T)$. Gathering all the measurements we can write the regression model as

$$\mathbf{B = XA + E;\quad rank(X) = k;}$$

where **B = [b(t1) b(t2) : : : b(tN)], A = [a(t1) a(t2) : : : a(tN)]** and correspondingly for E. Without the rank constraint [3] on X, it would be straightforward to transform to the model by means of vectorization. Having the rank[3] constraint one can consider to minimize the difference in Frobenius norm, i.e.

$$\mathbf{min\ rank(X) = k \qquad \|B - XA\|_F :}$$

B. Determinant minimization problem

$$\mathbf{Min\quad det(B - XA)(B - XA)^T}$$

which, under certain circumstances, gives the maximum likelihood estimate for the reduced-rank regression model[14], In the derivation of this problem there are several assumptions on the data that are justified due to the existence of enough noise in the measurements. The effects of the assumptions are that the determinant cannot be made equal to zero no matter how we choose X. Thus, the most simple case where B = XA, which trivially gives **det (B - XA)(B - XA)^T = 0,** cannot be solved by this approach. Our objective was to determine the different scenarios where the determinant minimization criterion fails, in the sense that it does not give a well defined solution, and generalize the minimization criterion in order to obtain a well defined solution in all cases.
To clarify the problem with the determinant criterion, we recall that the determinant
of a matrix $F \in R^{m x m}$ can be written as

$$\mathbf{det\ (F) = \prod \sigma_i}$$

where σ_i are the singular values[1]of F. If now the matrix F depends on some variables X, in particular if we set

$$\mathbf{F(X) = (B - XA)\ (B - XA)^T}$$

Then

$$\mathbf{det\ (F(X)) = \prod \sigma_i\ (F(X))}$$

The singular values[1]$\mathbf{\sigma_i\ (F(X))}$ now depend on X and zeroing one singular value of F(X) would zero the determinant. One can, in certain cases, zero the determinant by simply setting parts of the matrix X to zero. The rest of the matrix would be undetermined, nonetheless the determinant is minimized but with no solution of substance.

.

C ***Generalization to rank reduction and volume minimization***

Zeroing one singular value is not sufficient for computing a solution. We propose an approach to continue and zero as many singular values as possible. In addition one should also minimize the product of the rest of the singular values that cannot be zeroed. Mathematically we write the generalization as follows.

Min rank(X)=k

rank(F(X))=r

Min vol(F(X)) r min = min rank(X)=k rank(F(X));

where the volume of a matrix is defined as the product of the nonzero singular values .

If **rank (F) = r**

then

vol (F) = ∏ σ $_i$

Where **σ_1…. σ_r** are the nonzero singular values. The tools for analyzing, but also computing the solution to, the generalized minimization problem are the singular value decomposition [2] **(SVD).**

Theorem (SVD). Any given matrix A ϵ R $^{m\ x\ n}$ can be factorized as

A = UXV ^{T}

where **U ϵ R $^{m\ X\ m}$; V ϵ R ^{n} X ^{n}** are orthogonal matrices[7] and **Σ ϵ R $^{m\ x\ n}$** is a diagonal matrix with the nonnegative entries **σ1≥…σ $_{min\ (m\ n)}$**: The matrices U and V are called the left and right singular matrices, respectively, and σ $_i$ are the singular values.

Theorem (HOSVD). Any 3-tensor **A ϵ R $^{I\ x\ J\ x\ k}$** can be factorized **A = (U; V; W) x S;**

Where **U ϵ ^{I}R $^{x\ I}$, V ϵ R $^{J\ x\ J}$, and W ϵ R $^{K\ x\ K}$,** are orthogonal matrices, and the tensor **S ϵ R $^{I\ x\ J\ x\ K}$** is all-orthogonal: the matrices [7]<S, S>$_{-I}$, i = 1; 2; 3; are diagonal, and

||S(1; :; :)|| ≥ ||S(2; :; :)|| … ≥ 0

||S(:; 1; :)||≥||S(:; 2; :)||…≥ 0

||S(:; :; 1)|| ≥ ||S(:; :; 2)||… ≥ 0

are the 1-mode, 2-mode, and 3-mode[10] singular values, also denoted **$\sigma_i^{(1)}$, $\sigma_i^{(2)}$ $\sigma_i^{(3)}$**

.

IV AS MULTIDIMENSIONAL ARRAYS

Just as avector in an *n*-dimensinal space is represented by a one-dimensional array of length *n* with respect to a given basis , any tensor with respect to a basis is represented by a multidimensional array. For example, a linear transformation is represented in a basis as a two-dimensional square $n \times n$ array. The numbers in the multidimensional array are known as the scalar components of the tensor or simply its components. They are denoted by indices giving their position in the array, as subscript and superscript, following the symbolic name of the tensor. For example, the components of an order 2 tensor *T* could be denoted T_{ij} where*i* and *j* are indices running from 1 to *n*, or also by T_i^j. Whether an index is displayed as a superscript or subscript depends on the transformation properties of the tensor, described below. The total number of indices required to uniquely select each component is equal to the dimension of the array, and is called the order,degree or rank of the tensor.

Just as the components of a vector change when we change the basis of the vector space, the components of a tensor also change under such a transformation. Each tensor comes equipped with a transformation law that details how the components of the tensor respond to a change of basis. The components of a vector can respond in two distinct ways to a change of basis where the new basis vectors $\hat{\mathbf{e}}_i$ are expressed in terms of the old basis vectors $\mathbf{e}_j$ as,

$$\hat{\mathbf{e}}_i = \sum_{j=1}^{n} R_i^j \mathbf{e}_j = R_i^j \mathbf{e}_j .$$

Here $R_i{}^j$ are the entries of the change of basis matrix, and in the second expression the summation sign was suppressed: this is the Einstein summation convention, which will be used throughout this article The components v^i of a column vector **v** transform with the inverse of the matrix *R*,

$$\hat{v}^i = \left(R^{-1}\right)_j^i v^j ,$$

where the hat denotes the components in the new basis. This is called a *contravariant* transformation law, because the vector transforms by the *inverse* of the change of basis. In contrast, the components, w_i, of a covector (or row vector), **w** transform with the matrix R itself,

$$\hat{w}_i = R_i^j w_j .$$

This is called a *covariant* transformation law, because the covector transforms by the *same matrix* as the change of basis matrix. The components of a more general tensor transform by some combination of covariant and contravariant transformations, with one transformation law for each index. If the transformation matrix of an index is the inverse matrix of the basis transformation, then the index is called *contravariant* and is traditionally denoted with an upper index (superscript). If the transformation matrix of an index is the basis transformation itself, then the index is called *covariant* and is denoted with a lower index (subscript).

V TENSOR NETWORKS

Emerging technology is Tensor Decompositions (TDs) [4] and Tensor Networks (TNs) via low-rank matrix/tensor approximations. The challenge is how to analyze large-scale, multiway data sets[13]. Data explosion creates deep research challenges that require new scalable, TD and TN algorithms. Tensors, which are multi-dimensional generalizations of matrices provide often a useful representation for such data. Tensor decompositions (TDs)[9] decompose data tensors in factor matrices, while tensor networks (TNs) represent higher-order tensors by interconnected lower-order tensors. A tensor network aims to represent or decompose a higher-order tensor

into a set of lower-order tensors and 3rd-order tensors called cores or components which are sparsely interconnected. In other words, in contrast to TDs, TNs represent decompositions of the data tensors into a set of sparsely (weakly) interconnected lower-order tensors. Recently, the curse of dimensionality for higher-order tensors has been considerably alleviated or even completely avoided through the concept of tensor networks (TN) .

If a tensor network is a tree, i.e., it does not contain any cycle, each of its edges splits the modes of the data tensor into two groups, which is related to the suitable matricization of the tensor. If, in such a tree tensor network, all nodes have degree 3 or less, it corresponds to anHierarchical Tucker (HT) decomposition.

VI. TENSORIZATION AND QUANTIZATION

Curse of Dimensionality - The term curse of dimensionality, in the context of tensors, refers to the fact that the number of elements of an Nth-order (I X I X… X I) tensor, I^N, grows exponentially with the tensor order N. Tensors can easily become really big for very high order tensors since the size is exponentially growing with the number of dimensions (ways, or modes). For example, for the Tucker decomposition the number of entries of a original data tensor but also a core tensor scales exponentially in the tensor order, for instance, the number of entries of an Nth-order (R X R X…X R) core tensor is R^N. If all computations are performed on a CP tensor format and not on the raw data tensor itself, then instead of the original I^N raw data entries, the number of parameters in a CP representation reduces to NRI, which scales linearly in N and I . This effectively bypasses the curse of dimensionality, however the CP approximation may involve numerical problems, since existing algorithms are not stable for high-order tensors. At the same time, existing algorithms for tensor networks, especially TT/HT ensure very good numerical properties (in contrast to CPD algorithms), making it possible to control an error of approximation i.e., to achieve a desired accuracy of approximation The curse of dimensionality can be overcome through quantized tensor networks, which represents a tensor of possibly very high-order as a set of sparsely interconnected low-order and very low dimensions cores.

.VII.CONCLUSIONS

In this paper we generalize the determinant minimization criterion for the reduced rank regression problem into dimensionality reduction[6] of the objective matrix and then volume minimization, where volume of a matrix is defined as the product of its nonzero singular values.

In this paper we discuss two classification algorithms based on higher order[11] singular value decomposition (HOSVD)

Tensor networks can be considered as a generalization and extension of TDs and are promising tools for the analysis [11]of big data due to their extremely good compression abilities and distributed and parallel processing. Overall, the benefits of multiway[12] tensor analysis methods can be summarized as follows:

"Super" compression of huge multidimensional, structured data which admits a low-rank approximation via TNs of high-order tensors by extracting factor matrices and/or core tensors of low-rank and low-order and perform all mathematical manipulations in tensor formats (especially, TT and HT formats). A compact and very flexible approximate representation of structurally rich data by accounting for their spatio-temporal and spectral dependencies. Opportunity to establish statistical links between cores, factors, components or hidden latent variables for blocks of data. Possibility to operate with noisy, incomplete, missing data by using powerful low-rank tensor/matrix approximation techniques. A framework to incorporate various diversities or constraints in different modes and thus naturally extend the standard (2-way) CA methods to largescale multidimensional data.

REFERENCES

[1] O. Alter, P.O. Brown, and D. Botstein. Generalized singular value decomposition for comparative analysis of genome-scale expression data sets of two different organisms. PNAS, 100(6):3351_3356, March 2003.

[2] Å. Björck. Numerical Methods for Least Squares Problems. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1996.23

[3] S. R. Chinnamsetty, M. Espig, B. N. Khoromskij, and W. Hackbusch. Tensor product approximation with optimal rank in quantum chemistry. The Journal of Chemical Physics, 127(084110):14, 2007.

[4] P. Comon. Tensor decompositions. In J. G. McWhirter and I. K. Proudler, editors, Mathematics in Signal Processing V, pages 1_24. Clarendon Press, Oxford, UK, 2002.

[5] L. De Lathauwer, B. De Moor, and J. Vandewalle. A multilinear singular value decomposition. SIAM Journal on Matrix Analysis and Applications, 21(4):1253_1278, 2000.

[6] M. Ishteva, L. De Lathauwer, P.-A. Absil, and S. Van Hu_el. Dimensionality reduction for higher-order tensors: algorithms and applications. Internal Report 07-187, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), 2007. Submitted to: International Journal of Pure and Applied Mathematics.

[7] T. G. Kolda. Orthogonal tensor decompositions. SIAM Journal on Matrix Analysis and Applications, 23(1):243_255, 2001.

[8] T. G. Kolda. Multilinear operators for higher-order decompositions. Technical Report SAND2006-2081, Sandia National Laboratories, April 2006.

[9] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. Technical Report SAND2007-6702, Sandia National Laboratories, 2007.

[10] P. M. Kroonenbeg. Principal component analysis of three-mode data by means of alternating least squares algorithms. Psychometrika, 45:69_97, 1980.

.

[11] L. Omberg, G. H. Golub, and O. Alter. A tensor higher-order singular value decomposition for integrative analysis of dna microarray data from different studies. Proceedings of the National Academy of Sciences, 104(47):18371_18376, 2007.

[12] Acar and B. Yener (2009). "Unsupervised Multiway Data Analysis: A Literature Survey," *IEEE Transactions on Knowledge and Data Engineering*, 21(1), 6–20.

[13] E. Acar, C.A. Bingol, H. Bingol, R. Bro and B. Yener (2007). "Multiway Analysis of Epilepsy Tensors," *Bioinformatics*, 23(13), i10–i18.

[14] E. Acar, T.G. Kolda and D.M. Dunlavy (2009). "An Optimization Approach for Fitting Canonical Tensor Decompositions," Technical Report Number SAND2009-0857, Sandia National Laboratories, Albuquerque, NM and Livermore, CA.