2009 International Symposium on Nonlinear Theory and its Applications
NOLTA'09, Sapporo, Japan, October 18-21, 2009

Efficient and Stable Decompositions for Multidimensional Tensors

Eugene Tyrtyshnikov and Ivan Oseledets†

Institute of Numerical Mathematics, Russian Academy of Sciences†
Moscow, Russia
Email: tee@inm.ras.ru, ivan.oseledets@gmail.com

Abstract—Decompositions of d-dimensional tensors are crucial either in structure recovery problems and merely for a compact representation of tensors. However, the well-known decompositions have serious drawbacks: the Tucker decompositions suffer from exponential dependence on the dimensionality d while fixed-rank canonical decompositions are not stable. In this paper we present new decompositions that are stable and have the same number of representation parameters as canonical decompositions for the same tensor. Also we present a theoretical analysis of compression properties of the new approach and discuss some applications.

1. Introduction

In various problems we are naturally led to multi-index arrays (tensors) with entries $a(i_1, ..., i_d)$ determined by d indices, each attaining n values. The total number of the entries is n^d and exhibits exponential growth in d, which is known as the "curse of dimensionality". Even for $d \geq 5$ numerical computations with tensors are feasible only due to some special exact or approximate decompositions with a dramatically smaller number of defining parameters than n^d.

In higher dimensions one can consider the so-called *Tucker decomposition* [15] (a certain generalization of the Singular Value Decomposition for matrices)

$$a(i_1, ..., i_d) = \sum_{t_1=1}^{r} ... \sum_{t_d=1}^{r} g(t_1, ..., t_d)\, q_1(i_1,\, t_1) ... q_d(i_d,\, t_d),$$

which, besides the two-dimensional arrays $q_1, ..., q_d$, is determined by a d-dimensional array $g(t_1, ..., t_d)$ with r^d entries. Usually $r \ll n$, but the exponential growth in d remains. Apart from that, the skeleton decomposition for matrices transforms in tensors to the so-called *canonical decomposition* [3, 6]

$$a(i_1, ..., i_d) = \sum_{s=1}^{R} u_1(i_1,\, s) ... u_d(i_d,\, s) \qquad (1)$$

with dRn defining parameters. In contrast to the Tucker decomposition, the canonical decomposition with minimal possible values of R is an unstable and computationally hard problem [14]. In tensor operations in the canonical format the number of summands R grow rapidly, so we have to approximate the results using the same format with smaller value of R. The latter is called *recompression* and cannot be reliably and fastly done in the canonical format. Thus, if we want to overcome the "curse of dimensionality" then we need to look for some other decompositions.

2. Tensor tree decompositions

Recently we have proposed a *recursive decomposition* [8, 10]: given a d-dimensional tensor, we construct a tree in which the nodes of every next level are associated with tensors of lower dimensionality, the leafs are associated with the two or three-dimensional tensors, and in the end the tensor is defined by this tree and related two or three-dimensional tensors. First we called this decomposition a *Tree-Tucker decomposition* because the tree in [8] was suggested to construct for the Tucker core. Of course, a shorter hame is *TT decomposition*. Now we have to admit that the Tucker decomposition as a preliminary step is not mandatory. None the less, we keep the same name *TT* since anyway we construct some *Tensor Tree*.

By definition, the tensor tree is binary and arises by recursion. Each node is marked by an auxiliary tensor, the *node dimensionality* is the dimensionality of this tensor. The original tensor is a mark for the root node. If a node V_0 is of dimensionality d_0 then its children V_1 and V_2 have dimensionalities $d_1 + 1$ and $d_2 + 1$ with

$$d_1 + d_2 = d_0.$$

If $d_0 = 3$ then we can take $d_1 = 1$, $d_2 = 2$. Each node of dimensionality 2 is a leaf, each node of dimensionality 3 is a leaf or generates 2 leaf nodes of dimensionalities 2 and 3. A precise rule is below.

Let us refer to the original indices $i_1, ..., i_d$ as *spatial indices*. Each new auxilary tensor must be defined by some spatial indices (at least one) and some new indices $s_1, s_2, ...$ that will be called *auxiliary indices*. Each node that is not a leaf generates exactly one auxiliary index. Thus, the total number of auxiliary indices is equal to the total number of nodes minus the number of leaf nodes.

Let a node V_0 be marked by a tensor $a_0(i'_1, ..., i'_{d_0})$ and generate an auxiliary index s. Then its children nodes V_1 and V_2 are marked by tensors $a_1(i''_1, ..., i''_{d_1}, s)$ and $a_2(i''_{d_1+1}, ..., i''_{d_0}, s)$, where the indices $i''_1, ..., i''_{d_0}$ constitute a permutation of the indices $i'_1, ..., i'_{d_0}$. If there is exactly one

auxilary index among the latters, then it is given to one of the tensors a_1 or a_2. In the case of two auxiliary indices one is given to a_1 and the other is given to a_2. It is important to note that *this rule guarantees that every tensor that is not a leaf receives at most two auxiliary indices*.

By definition, the leaf nodes are of dimensionality 2 or 3. Every node of dimensionality 2 is a leaf and has exactly one spatial index. Then we consider alternative options (A) and (B) as below, choose one of them before constructing a tree and during the construction demand that a node of dimesionality 3 is set to be a leaf if and only if it has:

(A) two auxiliary indices;

(B) three auxiliary indices.

If we opt for the rule (A), then a binary tree constructed is called a (basic) *tensor tree*, if (B) then it is an *extended tensor tree*. The corresponding decompositions represent the given tensor as the sum of products of the leaf tensors over all auxiliary indices and are called *TT* (*basic TT*) or *extended TT* decompositions.

Besides the above formalism, the tensors a_0, a_1, a_2 are to be related as follows:

$$a_0(i'_1, ..., i'_{d_0}) = \sum_{s=1}^{r} a_1(i''_1, ..., i''_{d_1}, s)\, a_2(i''_{d_1+1}, ..., i''_{d_0}, s). \quad (2)$$

It is required that the number of summands r for a_0 is minimal possible among all splittings of the form (2). The number $r = r(V_0)$ is called a *splitting rank* or *compression rank* of the node V, and its maximal value for all the nodes is the *TT rank* of the given tensor. If r is the TT rank, then each of auxiliary indices attains at most r values.

A complete description of the TT decomposition includes:

- a binary tree;
- two and three-dimensional tensors associated with the leafs;
- a distribution of spatial and auxilary indices for every node.

Theorem 1. *The total number of entries of the leaf tensors of an extended TT decomposition of rank r does not exceed* $drn + (d-2)r^3$.

Theorem 2. *Assume that a tensor* $a(i_1, ..., i_d)$ *possesses a canonical decomposition (1) with R terms. Then* $a(i_1, ..., i_d)$ *admits a TT decomposition of rank R or less.*

It is proved in [8] that a basic tensor tree has at most $d-2$ leafs with two auxiliary indices, and the splitting of any of them can give exactly one leaf with three auxiliary indices. Thus we have at most $d-2$ leafs with three auxiliary indices. Besides that, there should be exactly d leafs with spatial indices, and each contains exactly one auxiliary index, which makes the claim obvious. Theorem 2 is proved in [8].

3. Effective rank of a tensor

The minimal possible value $\mathtt{RANK}(a)$ of R in decompositions of the form (1) is called *canonical rank* of the tensor $a(i_1, ..., i_d)$. Its properties are known to differ a lot from the properties of the matrix rank. First, if all the entries are obliged to belong to a subfield $\mathbb{F}$ of complex numbers $\mathbb{C}$ then the canonical rank may depend on $\mathbb{F}$. Second, a tensor of rank R can be a limit of tensors of rank $r < R$.

We propose a certain way to avoid the both difficulties. Define the *effective rank* as follows:

$$\mathrm{ERank}(a) = \limsup_{\varepsilon \to +0} \min_{\substack{|b-a| \le \varepsilon \\ b \in \mathbb{C}(n_1,...,n_d)}} \mathrm{RANK}(b), \quad (3)$$

where $\mathbb{C}(n_1, ..., n_d)$ is a set of all arrays (tensors) of size $n_1 \times ... \times n_d$ with entries from $\mathbb{C}$. Similarly, $\mathbb{F}(n_1, ..., n_d)$ denotes the set of all tensors of size $n_1 \times ... \times n_d$ with entries from $\mathbb{F}$. If a tensor a belongs to $\mathbb{F}(n_1, ..., n_d)$ then we can be naturally interested to calculate its *canonical rank over* $\mathbb{F}$ (when minimizing R we use the constraint $u_1(i_1, s), ..., u_d(i_d, s) \in \mathbb{F}$).

Despite the fact that canonical rank depends on $\mathbb{F}$, by the very definition the effective rank (3) does not, for ε-perturbations of the tensor are now *all those* that belong to $\mathbb{C}$ and no longer obliged to stay in $\mathbb{F}$. A close concept is that of *border rank*[2]. However, the border rank assumes that $b \in \mathbb{F}(n_1, ..., n_d)$ and keeps dependence on $\mathbb{F}$. An important theorem is as follows [10].

Theorem 3. *Let* $a \in \mathbb{F}(n_1, ..., n_d)$. *Then for this tensor there exists a TT decomposition of rank* $r \le \mathtt{ERank}(a)$ *with entries of all tensors belonging to* $\mathbb{F}$.

Example 1. Let a d-dimensional tensor is a representation of a matrix A of the following form:

$$A = \Lambda \otimes I \otimes ... \otimes I + I \otimes \Lambda \otimes ... \otimes I + .. + I \otimes ... \otimes I \otimes \Lambda.$$

Then

$$P(h) \equiv \otimes_{s=1}^{d} (I + h\Lambda) = I + hA + O(h^2),$$

and it follows that

$$A = \frac{1}{h} P(h) - \frac{1}{h} P(0) + O(h).$$

Hence, $\mathtt{ERank}(A) = 2$.

Example 2. Consider a real-valued tensor F defined by the function

$$f(x_1, ..., x_d) = \sin(x_1 + ... + x_d)$$

on some grids for $x_1, ..., x_d$. In [1] it is shown that the real canonical rank of F does not exceed d (it is likely to be exactly d). Using the equality

$$\sin x = \frac{\exp(\imath x) - \exp(-\imath x)}{2\imath}$$

we deduce that $\mathtt{ERank}(F) = 2$.

By Theorem 2, any canonical decomposition can be transformed to a TT decomposition. By Theorem 3, the latter may have (and frequently has!) an advantage in the number of representation parameters as the TT rank may be (and frequently is!) significantly less than the canonical rank.

Moreover, in contrast to the canonical decomposition, the optimal TT approximation to a given tensor with prescribed compression rank bounds always exists [9] and is stable by the following reason: if a sequence of tensors converges to a tensor then the compression ranks of the limit tensor cannot be less than the lower limits of sequences of the corresponding compression ranks of these tensors. In [9] we present a *full-to-TT* compression algorithm that computes a quasi-optimal approximation to a given array (tensor) of dimensionality d with a deterioration factor that does not exceed $\sqrt{d-1}$.

4. TT algorithms

A nice paradigm for the new generation of numerical algorithms for tensor operations is as follows: assume that all initial tensors are given in the TT format, then tensor operations can be performed approximately within a wanted accuracy and the result of any operation should be a tensor in the same TT format with a prescribed bound on the TT rank. Remark that either the Tucker and the canonical formats do not suit this paradigm, though by different reasons: the former does not because it suffers from the curse of dimensionality, the latter since a good recompression procedure is lacking.

It is certainly most importnat that the TT format admits a fast and reliable recompression algorithm proposed in [7]. It is based on a sequence of SVDs and in all respects inherits the perfection of the SVD algorithm. In effect we are able to compute even exact singular value decompositions for all auxiliary unfolding matrices of a tensor in the TT format. These matrices can be of huge size, but their SVDs receive orthogonal (unitary) matrices in a structured factorized form (see an exposition example in [11]).

The recompression algorithm is in no way trivial, but, despite that, it has a simple and lucid logical structure. Formally it applies to very special tensor trees where at every splitting of indices we put exactly one spatial index into one of the two groups. Such a tree generates a representation of the tensor in the form of a *tensor train*

$$a(i_1,\dots,i_d) = \sum_{s_1,\dots,s_{d-1}} g_1(i_1,s_1)\, g_2(s_1,i_2,s_2)\, \dots \\ \dots\, g_{d-1}(s_{d-2},i_{d-1},s_{d-1})\, g_d(s_{d-1}i_d)$$

with *tensor carriages* $g_1,\dots,g_d$. This particular TT representation was put forth in [7] and found so convenient that certainly ought to be the main case in numerical practice. Since the associated tree is trivial, it remains only as kind of artefact. In this special case all constructions and descriptions need not to refer to a tree. Thus, this TT decomposition involves *neither tree, nor Tucker* anymore. The name *tensor trains* was proposed in [9, 11] and seems just natural as any two neighbor tensor carriages are "attached" each to other by one common auxiliary index.

In the light of the above paradigm, when performing operations with tensors we can approximate a TT tensor by another TT tensor of lower rank in a fast way. The tensor-train recompression procedure has complexity $O(dnr^3)$ [7]. The influence of such approximations can be studied within a general framework suggested in [5].

5. Applications and perspectives

It is especially interesting to acquire TT approximations using a small portion of entries of a given tensor, as in the incomplete cross approximation methods [4, 13] that interpolate a matrix on the entries of some cross of its columns and rows. These matrix constructions have recently been extended to tensors and provided us with a new interpolation formula in d dimensions on the entries of a certain *tensor-train cross* [9]. The total amount of entries is $O(dnr^2)$, the complexity of the TT-cross approximation algorithm is $O(dnr^3)$. Thus, the curse of dimensionality is not there.

The TT-cross approximation algorithm proposed in [9] is bound to make essential progress in many applications. For instance, since the TT cross algorithm gets on input only a procedure for computation of tensor entries and then carefully applies it to compute only a small portion of all entries, this yields a new method for computation of d-dimensional integrals without Monte Carlo ideas [9]. Using appropriate one-dimensional quadrature rules, we need then to find a scalar value

$$\gamma = \sum_{i_1} \dots \sum_{i_d} a(i_1,\dots,i_d)\; x_1(i_1) \dots x_d(i_d).$$

And this can be done in $O(dnr^2)$ operations so long as the tensor $a(i_1,\dots,i_d)$ is represented by a tensor train with the compression rank r. For more perspectives for application of tensor trains we refer to our recent work [9].

Last but not least, tensor trains *only look* as a particular case of tensor-tree decompositions. In fact, binary tensor trees such as considered above *always* reduce to some tensor trains [12].

Acknowledgments

This work was supported by the RFBR grants 08-01-00115, 09-01-91332 (RFBR/DFG), 09-01-12058, 09-01-00565 and by Priority Research Programme of the Department of Mathematical Sciences of Russian Academy of Sciences.

References

[1] G. Beylkin and M. J. Mohlenkamp, "Numerical operator calculus in higher dimensions", *Proc. Natl. Acad. Sci. USA*, vol. 99, 16, pp. 10246–10251, 2002.

[2] D. Bini and M. Capovani, "Tensor rank and border rank of band Toeplitz matrices", *SIAM J. Comput.*, 2, pp. 252-258, 1987.

[3] J. D. Caroll and J. j. Chang, "Analysis of individual differences in multidimensional scaling via n-way generalization of Eckart-Young decomposition", *Psychometrica*, vol. 35, 1970, pp. 283-319, 1970.

[4] G. A. Goreinov, E. E. Tyrtyshnikov, N. L. Zamarashkin, "A theory of pseudo-skeleton decompositions", *Linear Algebra Appl.*, vol. 261, pp. 1–21, 1997.

[5] W. Hackbush, B. N. Khoromskij, E. E. Tyrtyshnikov, "Approximate iterations for structured matrices", *Numer. Math.*, vol. 109, no. 3, pp. 365–383, 2008.

[6] R. A. Harshman, "Foundations of the Parafac procedure: models and conditions for an explanatory multimodal factor analysis", *UCLA Working Papers in Phonetics*, 16 (1970), pp. 1–84.

[7] I. V. Oseledets, "On a new tensor decomposition", *Doklady Math.*, vol. 97 (2009), to appear.

[8] I. V. Oseledets and E. E. Tyrtyshnikov, "Breaking the curse od dimensionality, or how to use SVD in many dimensions", *Research Report 09-03, Hong Kong: ICM HKBU, 2009* (www.math.hkbu.edu.hk/ICM/pdf/09-03.pdf), to appear in SIAM J. Sci. Comput.

[9] I. V. Oseledets and E. E. Tyrtyshnikov, "TT-cross approximation for multidimensional arrays", *Research Report 09-10, Hong Kong: ICM HKBU, 2009* (www.math.hkbu.edu.hk/ICM/pdf/09-10.pdf), to appear in Linear Algebra Appl.

[10] I. V. Oseledets and E. E. Tyrtyshnikov, "Recursive decomposition of multidimensional tensors", *Doklady Math.*, vol. 97 (2009), to appear.

[11] I. V. Oseledets and E. E. Tyrtyshnikov, "Recursive and Tensor-Train Decompositions in Higher Dimensions", HERCMA 2009 Proceedings, 2009.

[12] I. V. Oseledets and E. E. Tyrtyshnikov, "Binary tensor trees reduce to tensor trains", in preparation.

[13] I. Oseledets, D. Savostyanov D., E. Tyrtyshnikov, "Tucker dimensionality reduction of three-dimensional arrays in linear time", *SIAM J. Matrix Anal. Appl.*, 30 (3), pp. 939–956, 2008.

[14] V. De Silva, L.-H. Lim, "Tensor rank and the ill-posedness of the best low-rank approximation problem", *SIAM J. Matrix Anal. Appl.*, 2008, vol. 30, no. 3, pp. 1084–1127, 2008.

[15] L. R. Tucker, "Some mathematical notes on three-mode factor analysis", *Psychometrika*, vol. 31, pp. 279–311, 1966.