Numer. Math. Theor. Meth. Appl.
doi: 10.4208/nmtma.2009.m9006s

Vol. **2**, No. 4, pp. 439-444
November 2009

Fast Revealing of Mode Ranks of Tensor in Canonical Form

Dmitry V. Savostyanov*

Institute of Numerical Mathematics, Russian Academy of Sciences, 119333 Moscow, Gubkina 8, Russia.

Received 15 March 2009; Accepted (in revised version) 24 June 2009

Abstract. Considering the problem of mode ranks revealing of d-dimensional array (tensor) given in canonical form, we propose fast algorithm based on cross approximation of Gram matrices of unfoldings.

AMS subject classifications: 15A21, 15A69, 19A90

Key words: Multidimensional array, canonical decomposition, Tucker approximation, fast recompression.

1. Introduction

Since d-dimensional array of size n at each dimension contains n^d elements, efficient algorithms working with multidimensional data should incorporate approximation of tensor in structured formats with much smaller number of data representation parameters. The most popular tensor formats now are canonical and Tucker. Canonical format [15] of tensor $\mathbf{F}$ with d indices $\mathbf{F} = [f_{ij\cdots k}]$ reads

$$\mathbf{F} = (A,B,\cdots,C) = \sum_{s=1}^{R} \mathbf{a}_s \otimes \mathbf{b}_s \otimes \cdots \otimes \mathbf{c}_s, \quad \text{or} \quad f_{ij\cdots k} = \sum_{s=1}^{R} a_{is} b_{js} \cdots c_{ks}, \tag{1.1}$$

where « ⊗ » denotes outer (Kronecker) product. Eq. (1.1) represents tensor $\mathbf{F}$ by dnR parameters and removes exponential dependence on d (so-called "curse of dimensionality"), that make canonical format very popular in computation practice, especially for large-dimensional problems. However, canonical decomposition/approximation with minimal number of summands R (referred to as *tensor rank*) is rather a complicated problem. Among several available algorithms [1–3, 6, 7, 14, 15, 18] none is known to be absolutely reliable, and many numerical packages (for example quantum chemistry package MOLPRO) compute (1.1) with very large R, what leads to excessive costs of storage and further computations.

*Corresponding author. *Email address:* `dmitry.savostyanov@gmail.com` (D. V. Savostyanov)

For more compressed data representation one can use Tucker format [21]

$$\mathbf{F} = \mathbf{G} \times_1 U \times_2 V \cdots \times_d W = \sum_{a=1}^{r_1} \sum_{b=1}^{r_2} \cdots \sum_{c=1}^{r_d} g_{ab\cdots c} \mathbf{u}_a \otimes \mathbf{v}_b \otimes \cdots \otimes \mathbf{w}_c, \tag{1.2}$$

where coefficient tensor $\mathbf{G} = [g_{ab\cdots c}]$ is referred to as *core* and matrices $U = [\mathbf{u}_a], V = [\mathbf{v}_b], \cdots, W = [\mathbf{w}_c]$ as *mode factors.* Here « $\times_k$ » denotes multiplication of tensor by matrix along k-th mode, for example, $\mathbf{F} = \mathbf{G} \times_2 V$ means $f_{ij\cdots k} = \sum_{j'} v_{jj'} g_{ij'\cdots k}$. Summation bounds $r_1, \cdots, r_d$ are called *mode ranks* of tensor. Compression in Tucker format can be performed by reliable SVD-based algorithm [4, 5], that computes (1.2) with optimal values of mode ranks, that often turn to be considerably smaller than tensor rank and 'practical' rank R of canonical form (1.1) computed by real algorithms.

If canonical format is given for tensor $\mathbf{F}$, it can be also utilised for core $\mathbf{G}$ and total number of parameters remains linear in d. In this case it is natural to develop a method of Tucker compression, which utilises canonical structure of input. In [17] we discuss algorithm, based on low-rank approximation of canonical factors by Cross2D method. Since factors are approximated independently, total complexity is linear by d. On the other hand, accuracy criteria for approximation are estimated by inexact bounds, and this leads to overrated values for mode ranks.

In this paper we propose a new fast algorithm for mode ranks revealing and Tucker approximation of tensor in canonical form. It is based on proper decomposition of Gram matrices of unfoldings, performed by cross approximation method with linear in n complexity. Unfortunately, our method can not be applied when desired accuracy is more precise than square root of machine precision.

2. Approximation in Tucker form

Suppose $\mathbf{F}$ is given in canonical form (1.1) with large R and we need to approximate it in Tucker form (1.2) with smaller values of mode ranks $r_1, \cdots, r_d$. Standard method of Tucker approximation involves singular decompositions of all *mode unfoldings,* i.e., matrices of all mode vectors. Considering $\mathbf{F} = [f_{ij\cdots k}]$ as $n \times n^{d-1}$ matrix $F = [f_{i(j\cdots k)}]$ with row index i and column 'long index' $(j \cdots k)$, we compute SVD $F = USV^T + \Delta F$ and truncate it, introducing error $||\Delta F||_F \le \sqrt{d}\varepsilon ||F||_F$. Number of dominant singular values gives mode rank r_1, and $n \times r_1$ matrix of corresponding singular vectors gives Tucker factor U. Computing factors $V, \cdots, W$ from SVD of other mode unfoldings, we write core tensor as

$$\mathbf{G} = \mathbf{F} \times_1 U^T \times_2 V^T \times_3 \cdots \times_d W^T = (U^T A, V^T B, \cdots, W^T C), \tag{2.1}$$

preserving canonical form for core and linear number of representation parameters for $\mathbf{F}$. Accuracy of approximation is given by

$$||\mathbf{F} - \mathbf{G} \times_1 U \times_2 V \times_3 \cdots \times_d W||_F \le \varepsilon ||\mathbf{F}||_F. \tag{2.2}$$

This method is reliable, but very expensive for large-scale tensors, because SVD of $n \times n^{d-1}$ matrix requires $\mathscr{O}(n^{d+1})$ operations. Some methods with linear in n complexity are available for $d = 3$ (see [16] and further development of these ideas in [8,9]), but they do not utilise canonical structure of input.

To take advantages of (1.1), consider Gram matrix $\hat{F} = [\hat{f}_{ii'}] \overset{\text{def}}{=} FF^T$.

$$\begin{aligned} \hat{f}_{ii'} \overset{\text{def}}{=} \sum_{j\cdots k} f_{i(j\cdots k)} f_{i'(j\cdots k)} &= \sum_{j\cdots k} \left(\sum_{s=1}^{R} a_{is} b_{js} \cdots c_{ks} \right) \left(\sum_{t=1}^{R} a_{i't} b_{jt} \cdots c_{kt} \right), \\ \hat{F} &= A \left[(B^T B) \odot \cdots \odot (C^T C) \right] A^T, \end{aligned} \tag{2.3}$$

where « $\odot$ » denotes element-by-element (Hadamard) product of Gram matrices of factors $B, \cdots, C$. Since singular values of F are square roots of eigenvalues of $\hat{F}$, mode rank r_1 and corresponding Tucker factor U can be found by truncated proper decomposition $\hat{F} = U \Lambda U^T$. Each Gram matrix $\hat{F}$ can be assembled in $\mathcal{O}(nR^2 + n^2 R)$ and eigenvalues can be found in $\mathcal{O}(n^3)$ operations, so total cost of evaluation of (1.2) is linear in d. To make the complexity linear also in mode size n, we propose a cross approximation method for Gram matrix $\hat{F}$.

3. Cross approximation for Gram matrix

Truncated singular/proper decomposition is used in cases where low-rank approximation is required. This problem can be solved by faster methods, for example, those based on *skeleton approximation* $A \approx \tilde{A} = UGV^T$, where factors U, V^T consist of columns and rows of $n \times n$ matrix A, and core $G = B^{-1}$, where B is $r \times r$ submatrix on the intersection of *cross* formed by selected columns and rows. Accuracy of this approximation depends on choice of B crucially. In [11–13] it is shown that a good choice for B is *maximum volume* submatrix, i.e., the one with maximum modulus of determinant among all $r \times r$ submatrices. It is known that search of this submatrix is NP-complexity problem, that is not feasible even for quite moderate values of n and r. A good practical remedy is search of 'good enough' submatrix instead of maximum volume one. Such a method, called *cross approximation,* was first introduced in [22], and then developed with implementation details in [10], where some properties of arising *dominant* submatrices are also discussed.

If supported cross is iteratively widened at each step by one row and column that intersect on element where residual is maximum in modulus, cross approximation method is equivalent to Gauss decomposition with full pivoting. If A is symmetrical and positive definite (that means it is Gram matrix), this element always belongs to diagonal, and thus pivoting requires linear in matrix size number of operations. This remarkable property remains valid (in exact arithmetics) on all steps of cross elimination. Finally we come to the following Algorithm 3.1, that is equivalent to unfinished Choletsky decomposition.

Note that total $n \times n$ matrix never appears during the computations, and number of used memory cells is linear in n. The proposed algorithm includes computation of diagonal of matrix and r columns and also $\mathcal{O}(nr^3 + r^4)$ additional operations.[†] If Gram matrices

[†] The complexity of method can still be reduced, if special methods are applied for rediagonalization of symmetric diagonal plus rank one matrix.

Algorithm 3.1. Cross approximation for Gram matrix

```
Require: Function $\mathsf{a}_{ij}$ to compute elements of SPD $n \times n$ matrix $A$.
Ensure: Approximation $\tilde{A} = U \Lambda U^T$.
 1. Set $p = 0, \tilde{A} = 0$, compute $a = \mathrm{diag}(A)$                    $n\mathsf{a}_{ij}$
    {Compute diagonal of matrix}
 2. $i := \arg\max_j |a_j|$                                                  $n$
    {Find maximum element of residual}
 3. $u := a_{:,i} - U\Lambda(u_{i,:})^T$,                                    $n\mathsf{a}_{ij} + \mathcal{O}(np)$
    {Compute active column of residual}
 4. $u := u/\sqrt{u_i}$ {Pivot should be positive (in exact arithmetics)}
 5. $a_: := a_: - |u_:|^2$ {Update diagonal of residual}                     $n$
 6. $u =: [Uu']x$. {Orthogonalize $u$ to subspace $U$}                       $\mathcal{O}(np)$
    This decomposition is evaluated as follows:
    $x_{1:p} := U^T u$, $u' := (I - UU^T)u$, $x_{p+1} = \|u'\|_F$, $u' := u'/\|u'\|_F$
    and can require reorthogonalization step in machine arithmetics.
 7. $U := [Uu']$, and approximation writes $\tilde{A} = U(\Lambda + x^T x)U^T$   $\mathcal{O}(p^3)$
    {New approximation is exact on positions of all evaluated crosses}
 8. $\Lambda + x^T x := VDV^T$ {Re-diagonalize decomposition}
 9. $U := UV$, $\Lambda := D$, and approximation writes $\tilde{A} = U\Lambda U^T$   $\mathcal{O}(np^2)$
10. Check stopping criterion.
    If stopping criterion is satisfied, return $\tilde{A}$, otherwise set $p := p+1$ and repeat from
    step 2.
```

$A^T A, B^T B, \cdots, C^T C$ are computed (that requires $\mathcal{O}(nR^2)d$ operations), the diagonal of (2.3) is found in $\mathcal{O}(nR^2)$ operations, and every column require $\mathcal{O}(R^2 + nR)$ operations. The total complexity of mode ranks revealing by the method based on Algorithm 3.1 is $\mathcal{O}(nR^2 + nRr + R^2 r + nr^3 + r^4)d$, that is much smaller than $\mathcal{O}(n^2 R + nR^2 + n^3)d$ complexity of proper decomposition of full Gram matrix. In the next chapter we will illustrate this by a numerical example.

We also have to define stopping criterion for our method. It should be computed in linear time and thus direct check of residual norm $\|A - U\Lambda U^*\|_F \le \varepsilon \|A\|_F$ is unaffordable. We propose two options:

- check residual norm of diagonal $\|\,\mathrm{diag}(A - U\Lambda U^*)\|_F \le \varepsilon \|\,\mathrm{diag}(A)\|_F$;
- check convergence of dominant eigenvalues in Λ. More precisely, on each step split

all p eigenvalues in Λ in 'dominant' and 'smaller' part, the latter defined by

$$\left(\sum_{i=q+1}^{p} \lambda_i^2\right)^{1/2} \leq \varepsilon \|\Lambda\|_F. \tag{3.1}$$

Stop if all new eigenvalue during 3 successive iterations fall into 'smaller' part.

Both criteria lead to similar results in our experiments. In any case, after completion of Algorithm 3.1 we should remove 'smaller' part of eigenvalues according to (3.1).

4. Recompression of electron density

We apply the proposed method to recompress the three-dimensional electron density computed by MOLPRO quantum chemical package from canonical format with very large rank to Tucker format. Our method is compared to the algorithm, based on independent low-rank approximation of canonical factors [17] (further development is given in [20]). As shown in [19], approximation of canonical factor can introduce a large error to the whole tensor, and in order to avoid this, individual approximation bounds for every factor should be computed, which results in overrated values of ranks ρ_1, ρ_2, ρ_3 for approximated factors. To find "real" mode ranks r_1, r_2, r_3, algorithm based on individual factor filtering should include post-compression step, reducing size of core tensor from $\rho_1 \times \rho_2 \times \rho_3$ to $r_1 \times r_2 \times r_3$. We also compare new algorithm to the one based on full computation of proper decomposition for (2.3).

In Table 1 we show time T_1 of individual factor filtering method and overrated ranks ρ_1, ρ_2, ρ_3. Then we show time T_2 of proposed algorithm based on cross approximation 3.1 and time T_3 of algorithm based on full proper decomposition, together with 'true' mode ranks r_1, r_2, r_3. We see that individual filtration is sufficiently faster than algorithm proposed in this paper, but it is more tricky in implementation, especially for large d. On the other hand, the method of cross approximation provides considerable speedup in comparison with full proper decomposition method.

Table 1: Time for electron density compression. Mode size $n = 5121$, relative approximation accuracy $\varepsilon = 10^{-6}$. Time (mm:ss) is measured on Core2Duo T5300 processor with frequency 1.33 GHz. We use GNU Fortran 4.3.3 compiler and GotoBLAS-1.26 library.

		individual filtering		eigenvalues of Gram matrices		
molecule	R	ρ_1, ρ_2, ρ_3	T_1	r_1, r_2, r_3	T_2	T_3
methane	1334	$77 \times 77 \times 81$	0:06	$34 \times 34 \times 34$	0:10	23:00
ethane	3744	$78 \times 92 \times 121$	0:15	$24 \times 44 \times 35$	1:06	25:30
ethanol	6945	$134 \times 123 \times 166$	0: 23	$53 \times 55 \times 54$	4:00	31:10
glycine	9208	$103 \times 182 \times 229$	1:00	$30 \times 79 \times 82$	8:10	35:30

Acknowledgments This work was partially supported by RFBR grants 08-01-00115, 09-01-12058, RFBR/DFG grant 09-01-91332 and Priority Research Program of Dep. Math. RAS No. 3 and 5. The author is grateful to Heinz-Juergen Flad and Rao Chinnamsettey for providing input data for electron density used in numerical experiments.

References

[1] R. Bro, *PARAFAC: Tutorial and applications*, Chemometrics and Intelligent Lab. Syst. 38 (1997), no. 2, pp. 149–171.

[2] J. D. Caroll and J. J Chang, *Analysis of individual differences in multidimensional scaling via n-way generalization of Eckart-Young decomposition*, Psychometrika 35 (1970), pp. 283–319.

[3] P. Comon, *Tensor decomposition: state of the art and applications*, IMA Conf. Math. in Sig. Proc., Warwick, UK, 2000.

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

[5] ———, *On best rank-1 and rank-(R_1,R_2,$\cdots$,R_N) approximation of high-order tensors*, SIAM J. Matrix Anal. Appl. 21 (2000), pp. 1324–1342.

[6] ———, *Computing of Canonical decomposition by means of a simultaneous generalized Schur decomposition*, SIAM J. Matrix Anal. Appl. 26 (2004), pp. 295–327.

[7] M. Espig, L. Grasedick, and W. Hackbusch, *Black box low tensor rank approximation using fibre-crosses*, Constr. Appr. (2009). To appear.

[8] H.-J. Flad, B. N. Khoromskij, D. V. Savostyanov, and E. E. Tyrtyshnikov, *Verification of the cross 3D algorithm on quantum chemistry data*, Rus. J. Numer. Anal. Math. Model. 23 (2008), no. 4, pp. 329–344.

[9] S. A. Goreinov, *On cross approximation of multi-index array*, Doklady Math. 420 (2008), no. 4, pp. 404–406.

[10] S. A. Goreinov, I. V. Oseledets, D. V. Savostyanov, E. E. Tyrtyshnikov, and N. L. Zamarashkin, *How to find a good submatrix*, Research Report 08-10, ICM HKBU, Kowloon Tong, Hong Kong, 2008, `www.math.hkbu.edu.hk/ICM/pdf/08-10.pdf`.

[11] S. A. Goreinov and E. E. Tyrtyshnikov, *The maximal-volume concept in approximation by low-rank matrices*, Contemporary Mathematics 208 (2001), pp. 47–51.

[12] S. A. Goreinov, E. E. Tyrtyshnikov, and N. L. Zamarashkin, *Pseudo–skeleton approximations of matrices*, Reports of Russian Academy of Sciences 342 (1995), no. 2, pp. 151–152.

[13] ———, *A theory of pseudo–skeleton approximations*, Lin. Algebra Appl. 261 (1997), pp. 1–21.

[14] L. Grasedyck, *Existence and computation of low kronecker-rank approximations for large systems in tensor product structure*, Computing 72 (2004), pp. 247–265.

[15] 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.

[16] I. V. Oseledets, D. V. Savostyanov, and E. E. Tyrtyshnikov, *Tucker dimensionality reduction of three-dimensional arrays in linear time*, SIAM J. Matrix Anal. Appl. 30 (2008), no. 3, pp. 939–956.

[17] ———, *Cross approximation in tensor electron density computations*, J. Numer. Lin. Alg. Appl. (2009), submitted, `www.math.hkbu.edu.hk/ICM/pdf/09-04.pdf`.

[18] ———, *Fast simultaneous orthogonal reduction to triangular matrices*, SIAM J. Matrix Anal. Appl. 31 (2009), no. 2, pp. 316–330.

[19] ———, *Linear algebra for tensor problems*, Computing 85 (2009), no. 3, pp. 169–188.

[20] D. V. Savostyanov and E. E. Tyrtyshnikov, *Approximate multiplication of tensor marices by individual filtration of factors*, J. Comp. Math. Math. Phys. 49 (2009), no. 10. To appear.

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

[22] E. E. Tyrtyshnikov, *Incomplete cross approximation in the mosaic–skeleton method*, Computing 64 (2000), no. 4, pp. 367–380.