SoTT: Greedy approximation of a tensor as a sum of tensor trains

Maria Fuente Ruiz, Damiano Lombardi, Virginie Ehrlacher
COMMEDIA, INRIA Paris, France & LJLL, Sorbonne Université, COMMEDIA, INRIA Paris, France & LJLL, Sorbonne Université, Ecole des Ponts ParisTech & INRIA (Matherials project-team)

Email : maria.fuente-ruiz@inria.fr

Mots Clés : Tensor methods, Canonical Polyadic, Tensor Train.

Biographie – I arrived to COMMEDIA group in INRIA Paris to do my master thesis from Polytechnic University in Madrid in June 2019. I continue the project with my PhD: Adaptive tensor methods for high dimensional problems, started in March 2020. It is supervised by Damiano Lombardi and Virginie Ehrlacher.

Resumé : Machine learning and data mining algorithms are becoming increasingly important in analyzing large volume, multi-relational and multi-modal datasets, which are often conveniently represented as multiway arrays or tensors. The main challenge in dealing with such data is the so called *curse of dimensionality*, that refers to the need of using a number of degrees of freedom exponentially increasing with the dimension.

This problem can be alleviated through various distributed and compressed tensor network formats, achieved by low-rank tensor network approximations. The concept of compression of multidimensional large-scale data by tensor network decompositions can be intuitively explained as follows. Consider the approximation of a d-variate function $F(x) = F(x_1, x_2, ..., x_d)$ by a finite sum of products of individual functions, each depending on only one or a very few variables. In the simplest scenario, the function $F(x)$ can be (approximately) represented in the following separable form

$$F(x_1, x_2, ..., x_d) \approx u^{(1)}(x_1)u^{(2)}(x_2)\cdots u^{(d)}(x_d)$$

In practice, when a d-variate function $F(x)$ is discretized into a dth-order array, or a tensor, the approximation above then corresponds to the representation by rank-1 tensors, also called elementary tensors. Let us denote by $\mathcal{N}_n$, $n = 1, 2, \cdots, d$, the size of the discretization grid associated to the n^{th} variate.

The separation of variables principle can be achieved trough different tensor formats. In the present work we will make use of two of them: the Canonical Polyadic and the Tensor Train decompositions.

The **Canonical Polyadic (CP)** decomposition of F (see [2],[3]):

$$F(x_1, x_2, ..., x_d) \approx \sum_{i=1}^{r} u_i^{(1)}(x_1)u_i^{(2)}(x_2)\cdots u_i^{(d)}(x_d)$$

where r is the CP rank.

The **Tensor Train (TT)** decomposition of F (see [4]):

$$F(x_1, x_2, ..., x_d) \approx \sum_{i_1=1}^{r_1} ... \sum_{i_{d-1}=1}^{r_{d-1}} u_{i_1}^{(1)}(x_1)u_{i_1,i_2}^{(2)}(x_2)u_{i_2,i_3}^{(3)}(x_3)\cdots u_{i_{d-2},i_{d-1}}^{(d-1)}(x_{d-1})u_{i_{d-1}}^{(d)}(x_d)$$

where $r_1, \ldots r_{d-1}$ are the TT ranks.

The main advantages of the CP decomposition are its intuitive expression and its storage scaling. Indeed, instead of the original cost of $\mathcal{O}(\mathcal{N}^d)$, the number of entries to store a CP representation reduces to $\mathcal{O}(d\mathcal{N}r)$, which scales linearly in the tensor order d and size $\mathcal{N}$. The compression of tensors using the CP is usually computed by means of the ALS (Alternating Least Squares) method, which sometimes has some performance issues [1].

The Tensor Train format is probably one of the most used tensor formats in realistic applications [5], due to a good trade off between optimality and numerical stability. However, the order of the variables is fixed *a priori*. If the order of the variables is not good, the results could be exponentially bad.

In this work, a method is proposed in order to compute an approximation of a given tensor as a sum of Tensor Trains (SoTT), without fixing any parameter *a priori*: the order of the variates and the values of the ranks can vary from one term to the other in an adaptive way, depending on each tensor.

The numerical scheme is based on a greedy approximation algorithm and an adaptation of the TT-SVD method. The intuition of the TT-SVD is that in every step a SVD is performed to detach one open mode from the tensor. The TT-SVD starts by calculating an SVD of the matricization of the initial tensor F, where all modes but the first one are combined. The resulting matrices are each dematricized and the procedure is continued for a total of $d-2$ steps.

The proposed approach in the particular case of a sum of rank-1 terms, can also be used in order to compute an approximation of a tensor in a CP format, as an alternative to standard fix-point based algorithms like Alternating Least Squares (ALS) or Alternating Singular Value Decomposition (ASVD) methods, that present some numerical instabilities in high dimension. Some numerical experiments are proposed, in which the method is compared to ALS and ASVD methods for the construction of a CP approximation of a given tensor and performs particularly well for certain families of high-order tensors. It also has been used as a recompression method for solution of parametric PDEs. In addition, the interest of approximating a tensor as a sum of Tensor Trains is illustrated in several numerical test cases, including some physical applications.

Références

[1] Gregory Beylkin and Martin J. Mohlenkamp. Algorithms for numerical analysis in high dimensions. *SIAM J. SCI. COMPUT*, 26:2133–2159, 2005.

[2] Frank L Hitchcock. The expression of a tensor or a polyadic as a sum of products. *Journal of Mathematics and Physics*, 6(1-4):164–189, 1927.

[3] T. Kolda and B. Bader. Tensor decompositions and applications. *SIAM Review*, 51:455–500, 2009.

[4] I. Oseledets. Tensor-train decomposition. *SIAM J. Sci. Comput.*, 33:2295–2317, 2011.

[5] Maxim Rakhuba and Ivan Oseledets. Calculating vibrational spectra of molecules using tensor train decomposition. *The Journal of Chemical Physics*, 145(12):124101, 2016.