TENSORLAB 4.0 – A PREVIEW

Michiel Vandecappelle[*†], *Martijn Boussé*[†], *Nico Vervliet*[†], *Matthieu Vendeville*[†], *Rob Zink*[†] *and Lieven De Lathauwer*[*†]

Department of Electrical Engineering (ESAT), KU Leuven[†]
Group Science, Engineering and Technology, KU Leuven Kulak[*]

ABSTRACT

Since its initial release in 2013, Tensorlab has evolved into a powerful Matlab toolbox for the analysis of tensors and the computation of tensor decompositions. This upcoming release of Tensorlab, version 4.0, widens the applicability of the toolbox to a larger range of real-world applications. New β-divergence and low-rank weighted least squares (WLS) cost-functions are introduced for the canonical polyadic decomposition (CPD), offering higher flexibility to the user. Further, updating algorithms for the CPD allow both the tracking of streaming data and the incremental computation of the CPD of a large tensor. An LS-CPD algorithm is included to compute the CPD of a tensor that is only implicitly available as the solution of an underdetermined linear system.

1. TENSORLAB 3.0

The current 3.0 version of Tensorlab [1] (available online at `www.tensorlab.net`), features a large range of methods for the manipulation, analysis and visualization of tensors, as well as algebraic and optimization-based algorithms for the computation of tensor decompositions [2]. These range from the CPD and multilinear singular value decomposition (MLSVD) to block term decompositions (BTD) such as the decomposition in multilinear rank-$(L_r, L_r, 1)$ and more general variants. Specialized routines handle sparse, incomplete and large-scale tensors. Structured tensors (defined implicitly in Hankel, Löwner, CPD, or other form) are also supported, and their structure is strongly exploited during the computation of their decompositions. Additionally, the structured data fusion (SDF) framework provides an intuitive way to compute structured and/or coupled decompositions of tensors. The numerical philosophy behind the algorithms in Tensorlab is explained in [3]. An extensive user guide and a series of demos are available online.

Funding: Michiel Vandecappelle is supported by an SB Grant from the Research Foundation – Flanders (FWO). Research furthermore supported by: (1) Flemish Government: FWO: EOS Project no. 30468160 (SeLMA); (2) EU: The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC Advanced Grant: BIOTENSORS (n° 339804). This paper reflects only the authors' views and the Union is not liable for any use that may be made of the contained information; (3) KU Leuven Internal Funds C16/15/059.

KU Leuven, Department of Electrical Engineering ESAT/STADIUS, Kasteelpark Arenberg 10, bus 2446, B-3001 Leuven, Belgium; Group Science, Engineering and Technology, KU Leuven - Kulak, E. Sabbelaan 53, 8500 Kortrijk, Belgium.

2. NEW FEATURES IN TENSORLAB 4.0

With the upcoming release of version 4.0 of Tensorlab, its functionality will again be increased significantly. One of the main new features is the increased flexibility for computing the CPD of a tensor. The CPD of a tensor decomposes a rank-R tensor $\mathcal{T} \in \mathbb{R}^{I\times J\times K}$ as a linear combination of R rank-1 terms: $\mathcal{T} = \sum_{r=1}^{R} \mathbf{a}_r \otimes \mathbf{b}_r \otimes \mathbf{c}_r = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ with $\otimes$ the outer product of vectors, $\mathbf{A} = [\mathbf{a}_1 \dots \mathbf{a}_R] \in \mathbb{R}^{I\times R}$, and similarly for $\mathbf{B}$ and $\mathbf{C}$. A wider range of cost functions will be supported by allowing β-divergence and WLS cost functions. Additionally, incremental computation of the decomposition will become possible with CPD updating which will allow both the tracking of a CPD in real-time as well as the piecewise computation of the CPD of a large tensor. Furthermore, one will be able to compute the CPD of a tensor that is only available as the solution of a linear system of equations with the new LS-CPD algorithm.

2.1. β-divergence cost functions for the CPD

Tensorlab 3.0 only uses the Euclidean distance as metric for the cost function when approximating a tensor. This leads to good results in a lot of applications, but there are numerous situations in which the choice of a different metric can be beneficial. The Euclidean distance can be generalized to the family of β-divergences, where β is a parameter. This more general family of cost functions allows tensors to also be used for the analysis of for example counts, concentrations and spectra, for which the Euclidean distance may yield inferior results [4]. It includes, aside from the Euclidean distance ($\beta = 2$), a whole range of divergences of which the well-known Kullback–Leibler (KL) and Itakura–Saito (IS) divergences, with $\beta = 1$ and $\beta = 0$ respectively, are also part. The first- and second-order optimization algorithms for the CPD in Tensorlab were generalized so that they also support these β-divergence cost functions.

2.2. Low-rank WLS cost functions for the CPD

When using standard least squares cost functions to compute the CPD of a tensor, one inherently assumes that the residuals have equal variances and are uncorrelated. These assumptions may not hold in practice, however, for example when sensors of different quality are used in array processing applications. It therefore makes sense to assign different weights to the different residuals of the decomposition and take their correlation into account. Using a full weight tensor would be too computationally demanding, but a low-rank weight tensor can already offer much more flexibility compared to standard least squares (where the weight tensor is rank-1, with vectors $\mathbf{a}, \mathbf{b}, \mathbf{c}$ of which all entries are equal to 1 in the third-order case) while its CPD structure can be exploited during the computations. This leads to the following low-rank WLS cost function for the CPD of a tensor $\mathcal{T}$ in the third-order case:

$$\min_{\mathbf{A},\mathbf{B},\mathbf{C}} \frac{1}{2} \left\| [\![\mathbf{X}, \mathbf{Y}, \mathbf{Z}]\!] * ([\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] - \mathcal{T}) \right\|_F^2,$$

where $[\![\mathbf{X}, \mathbf{Y}, \mathbf{Z}]\!]$ is a low-rank approximation of the weight tensor $\mathcal{W}$. Note that $\mathcal{W}$ and $\mathcal{T}$ can have different rank. Tensorlab 4.0 offers first- and second-order optimization algorithms to efficiently compute a CPD of a tensor in the WLS sense with a low-rank weight tensor [5].

2.3. CPD updating

Batch tensor methods can easily become too slow to continually recompute a CPD of a tensor when new data is added to or removed from the tensor. Also, for very large tensors, storing the full tensor to be decomposed can already be problematic. CPD updating methods solve these problems by applying efficient updates to a previously computed CPD instead of recomputing the entire decomposition when new data is added to the tensor [6, 7]. In Tensorlab, the structure of the decomposition is exploited to achieve fast updates for the CPD when new slices are added to the tensor in certain modes [3, 6]. In addition, only the previous CPD and the new tensor slices have to be stored in every updating step. This enables both the tracking of the CPD of a fast-changing tensor and the piecewise computation of the CPD of a large tensor.

2.4. LS-CPD

Algebraic and first- and second-order optimization-based algorithms have been added to Tensorlab to compute the CPD of tensors that are only available as the solution of a system of equations [8]. These problems can for instance appear in tensor-based classification, or signal processing applications such as blind deconvolution of constant modulus signals. In the third-order case, one tries to find $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$, such that

$$\mathbf{M}\mathbf{x} = \mathbf{p}, \text{ with } \mathbf{x} = \operatorname{vec}([\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]),$$

where $\operatorname{vec}(.)$ denotes the vectorization of a tensor. The algorithms exploit the structure of $\mathbf{x}$ to directly obtain the factor matrices of the CPD, while being computationally efficient. This is especially beneficial when the system to be solved is (highly) underdetermined.

3. CONCLUSION

Tensorlab 4.0 will introduce a number of new algorithms that widen its applicability significantly. The focus of this release lies on the generalization of the CPD algorithms so that they offer a wider range of cost functions and can be used when the tensor to decompose is not static or is only given as the solution of a linear system.

4. REFERENCES

[1] N. Vervliet, O. Debals, and L. De Lathauwer, “Tensorlab 3.0 — numerical optimization strategies for large-scale constrained and coupled matrix/tensor factorization,” in *2016 50th Asilomar Conference on Signals, Systems and Computers (ASILOMAR 2016)*, pp. 1733–1738, November 2016.

[2] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, “Tensor decomposition for signal processing and machine learning,” *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3551–3582, 2017.

[3] N. Vervliet and L. De Lathauwer, “Numerical optimization based algorithms for data fusion,” in *Data Handling in Science and Technology vol:33* (M. Cocchi, ed.), ch. 4, pp. 1–41, Elsevier, 2018.

[4] A. Cichocki, R. Zdunek, S. Choi, R. Plemmons, and S. Amari, “Non-negative tensor factorization using alpha and beta divergences,” in *IEEE International Conference on Acoustics, Speech and Signal Processing, 2007. (ICASSP 2007).*, vol. 3, pp. 1393–1396, 2007.

[5] M. Boussé and L. De Lathauwer, “Nonlinear least squares algorithm for canonical polyadic decomposition using low-rank weights,” in *IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP17)*, pp. 39–43, December 2017.

[6] M. Vandecappelle, N. Vervliet, and L. De Lathauwer, “Nonlinear least squares updating of the canonical polyadic decomposition,” in *Proceedings of the 2017 25th European Signal Processing Conference (EUSIPCO2017)*, pp. 693–697, August 2017.

[7] D. Nion and N. D. Sidiropoulos, “Adaptive algorithms to track the parafac decomposition of a third-order tensor,” *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2299–2310, 2009.

[8] M. Boussé, N. Vervliet, I. Domanov, O. Debals, and L. De Lathauwer, “Linear systems with a canonical polyadic decomposition constrained solution: Algorithms and applications,” *Technical Report 17-99, ESAT-STADIUS, KU Leuven, Belgium*, 2017.