SPEEDING UP OF KERNEL-BASED LEARNING FOR HIGH-ORDER TENSORS

Ouafae Karmouda, Jeremie Boulanger and Remy Boyer

University of Lille, CNRS, CRIStAL, 59655 Lille, France,
firstname.lastname@univ-lille.fr

ABSTRACT

Supervised learning is a major task to classify datasets. In our context, we are interested into classification from high-order tensors datasets. The "curse of dimensionality" states that the complexities in terms of storage and computation grow exponentially with the order. As a consequence, the method from the state-of-art based on the Higher-Order SVD (HOSVD) works well but suffers from severe limitation in terms of complexities. In this work, we propose a fast Grassmannian kernel-based method for high-order tensor learning based on the equivalence between the Tucker and the tensor-train decompositions. Our solution is linked to the tensor network, where the aim is to break the initial high-order tensor into a collection of low-order tensors (at most 3-order). We show on several real datasets that the proposed method reaches a similar accuracy classification rate as the Grassmannian kernel-based method based on the HOSVD but for a much lower complexity.

Index Terms— Tensor classification, HOSVD, subspaces, Grassman manifold, Tensor Train.

1. INTRODUCTION

Nowadays, data needs more and more dimensions to be described [1]. A natural way to represent such data is to use multidimensional arrays called tensors [2]. Tensors of order Q [3] are multiway arrays of Q dimensions. They generalize the notions of vectors (first order tensors) and matrices (second order tensors). A prominent tensor decomposition is the High-Order SVD (HOSVD) [4] but this decomposition suffers from the well-known "curse of diemnsionality" meaning that the storage and computational costs grow exponentially with the order of the tensor. This is a severe limitation for tensors with $Q > 3$ [5, 6].

In the context of supervised classification, Support Vector Machines (SVMs) [7, 8] have been widely used due to their solid theoretical foundations, their performances and their easy implementation. Despite only processing linear classification, they can be modified to treat non linear problems via a kernel method. The main idea is to map data that are initially non linearly separable into a higher dimensional space (an RKHS) where it becomes linearly separable using a mapping ϕ. In practice (thanks to the kernel trick), the explicit computation is ϕ is not required as long as an expression for the kernel: $k(.,.) =< \phi(.), \phi(.) >$ exists. The exploitation of the HOSVD and SVMs to tensorial data has been introduced in [9]. The proposed method, denoted FAKSETT (Fast Kernel Subspace Estimation based on Tensor Train decomposition) shows good classification performance at the price of higher computational complexity due to the use of the HOSVD. The main idea of this work is to decrease this complexity using a recent theoretical result giving an algebraic link between the Tucker and Tensor-Train format [6].

For the remaining part of this work, scalar will be denoted by lower case letters (*e.g*: a), matrix will be denoted by upper case letters (*e.g*: A) while tensors will be denoted by calligraphic letters (*e.g*: $\mathcal{A}$)). The order of a tensor will generally be denoted by Q and we will consider the case when $Q > 3$. The q-th unfolding for $\mathcal{A}$ is a matrix and will be denoted $\mathcal{A}_{<q>}$ whose elements are given by:

$$\mathcal{A}_{<q>}(i_q\ ,\ i_1...i_{q-1}i_{q+1}...i_Q) = \mathcal{A}_{i_1\ ,\ \dots\ ,\ i_Q}.$$

The n-product between $\mathcal{A}$ and B will be denoted by $\times_n$ according to:

$$(\mathcal{A} \times_n B)_{i_1,\cdots,i_{q-1},i,i_{q+1},\cdots,i_Q} = \sum_i \mathcal{A}_{i_1,\cdots,i_Q} B_{i,i_q} \quad (1)$$

The contraction product $\times_n^m$ between $\mathcal{A}$ and $\mathcal{B}$ is a tensor constructed similarly by summing over the n-th index on $\mathcal{A}$ and over the m-th index from $\mathcal{B}$.

2. KERNEL FOR DATA TENSORS

We focus on supervised classification problems where data are high order tensors. Kernel-based classification methods [10, 11] require a similarity measure. It is standard [9] to consider a kernel between two tensors $\mathcal{X}$ and $\mathcal{Y}$ as for instance the RBF Gaussian kernel :

$$k(\mathcal{X}, \mathcal{Y}) = \exp(-\gamma ||\mathcal{X} - \mathcal{Y}||_F^2) \quad (2)$$

where $\gamma > 0$ is the bandwidth and $||.||_F$ is the Frobenius norm.

However, this kernel does not consider the multidimensional structure of data tensors.

2.1. HOSVD Decomposition

Definition: Tucker decomposition. A tensor $\mathcal{X}$ follows a Tucker Decomposition (TD) if it can be written as [4]:

$$\mathcal{X} = \mathcal{G} \times_1 U_1 \times_2 ... \times_Q U_Q \tag{3}$$

where U_q are of size $I \times R_q$, $1 \leq q \leq Q$ and $\mathcal{G}$ is the core tensor of size $R_1 \times \cdots \times R_Q$. The multi-linear ranks (m-ranks) of $\mathcal{X}$ is the Q-uplet $\{R_1, ..., R_Q\}$.

Definition: HOSVD. An important constrained format of the TD is the HOSVD. In the latter, the factors U_q are orthonormal and the core tensor $\mathcal{G}$ is all-orthogonal. In order to compute the HOSVD presented in Equation (3), [9] consider the R_q left dominant singular vectors from the q-th unfolding $\mathcal{X}_{<q>}$. The complexity of the HOSVD for a cubic Q-order tensor of size $I_1 \times \cdots \times I_Q$ is evaluated to $O(QRI^Q)$ where $I = \max_q\{I_q\}$ and $R = \max_q\{R_q\}$ is the maximal multi-linear rank. We can see that the HOSVD complexity grows linearly and exponentially with respect to the order Q. For low-order tensor [12, 13], this complexity remains acceptable but this limitation becomes rapidly severe for high-order tensors ($Q > 3$).

2.2. Tensor-based Kernel on HOSVD factors

In order to take account of the multidimensional structure of input data tensors, an idea presented in [9] consists in decomposing each tensor into its HOSVD:

$$\mathcal{X} = \mathcal{G} \times_1 U_1 \times_2 ... \times_Q U_Q \tag{4}$$

$$\mathcal{Y} = \mathcal{H} \times_1 V_1 \times_2 ... \times_Q V_Q \tag{5}$$

The kernel-based part of the proposed method is

$$k(\mathcal{X}, \mathcal{Y}) = \prod_q^Q k_q\left(U_q, V_q\right) \tag{6}$$

where $k_q(.,.)$ is a positive definite kernel defined on the matrices $\mathbb{R}^{I \times R_q} \times \mathbb{R}^{I \times R_q}$.

2.3. Kernel on a Grassmann manifold

It shall be noted that the decompositions from Equation (4) are not unique [3]. The output class will be affected by this lack of non-uniqueness. To mitigate this issue in the learning context, one can consider the subspaces spanned by the factors $\{U_q, ..., U_q\}$ and $\{V_1, ..., V_Q\}$. Indeed, the subspace spanned by the factor $U_q, q \leq N$ (respectively V_q) are invariant to any right multiplication by a non-singular matrix.
Therefore, let consider the sub-kernels k_q in the form:

$$k_q\left(U_q, V_q\right) = \tilde{k}_q\left(\text{span}(U_q), \text{span}(V_q)\right) \tag{7}$$

where $\tilde{k}_q$ is a kernel defined on the Grassman manifold $\mathcal{G}(R_q, I)$, *i.e* the subspaces from $\mathbb{R}^I$ with dimension R_q.

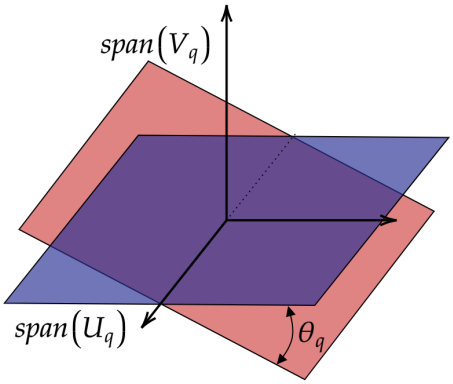

Fig. 1. Illustration of the angle θ_q from Equation (8)

A popular choice for $\tilde{k}_q$ that gives rise to a positive definite kernel and used in [9] is given by:

$$k_q\left(U_q, V_q\right) = \exp\left(-\gamma \sin^2\left(\theta_q\right)\right) \tag{8}$$

where θ_q is the principal angle between span(U_q) and span(V_q). It should be noted that despite θ_q being the geodesic distance in the Grassman manifold between the two subspaces, the expression $\sin(\theta_q)$ is considered instead, making the kernel k_q definite positive (therefore, SVM methods can be used for classification) [14]. Readers can refer to [14] for explicit ways of computing the principal angles. In our case, it is possible to directly use the projectors:

$$\sin^2\left(\theta_q\right) = 2\left|\left|U_q U_q^T - V_q V_q^T\right|\right|_F^2. \tag{9}$$

The principal angle θ_q is illustrated in Figure 1.

3. THE FAKSETT METHOD : A FAST ALTERNATIVE TO THE METHOD OF [9]

In order to decrease the complexity of the HOSVD discussed in Section 2.2, we propose to use a fast multiLinear projection method proposed in [6]. The theoretical foundations of this method are based on the Tensor Network theory [5] and in particular on the equivalence between the Tucker and Tensor-Train formats introduced in [6] and described in the following. We will first begin with a definition of Tensor Train decomposition (TTD).

Definition: Tensor-Train Decomposition (TTD). A Tensor $\mathcal{X}$ admits a TTD with TT-ranks $(R'_1, \ldots, R'_{Q-1})$ if it can be expressed as:

$$\mathcal{X} = G_1 \times_2^1 \mathcal{G}_2 \times_{Q-1}^1 \mathcal{G}_{Q-1} \times_Q^1 G_Q \tag{10}$$

where the size of each core is:

- $G_1 \in \mathbb{R}^{I \times R'_1}$
- $\mathcal{G}_q \in \mathbb{R}^{R'_{q-1} \times I \times R'_q},\ \forall q : 1 < q < Q$

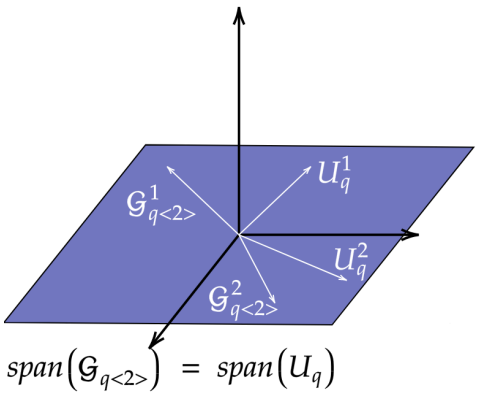

Fig. 2. In the case of $R_q = 2$ with $U_q = [U_q^1, U_q^2]$: Despite giving different factors, HOSVD and FAKSETT gives factors that span the same subspace.

- $G_Q \in \mathbb{R}^{R'_{Q-1} \times I}$

and where $(R'_1, R'_2, ..., R'_{Q-1})$ are the TT-ranks. To estimate the core tensors, we can use the TT-SVD algorithm [15] or its generalization [16].
Assume that a tensor $\mathcal{X}$ follows a TD of m-ranks $\{R_1, \cdots R_Q\}$ with orthonormal factor U_q. An equivalence between TD and TTD is presented in [6]. Each core extracted from the TD given by Equation (10) follows a 3-order Tucker model with two latent matrices in its first and third dimensions. In the second dimension, we have the interesting property that R_q left dominant singular vectors from the second unfolding spans the same subspace as U_q. As a consequence, we have for $2 \leq q \leq Q-1$:

$$\text{span}(U_q) = \text{span}\left(\mathcal{G}_{q<2>}\right). \tag{11}$$

This property is illustrated on Figure 2 and $\text{span}(U_1) = \text{span}(G_1)$, $\text{span}(U_Q) = \text{span}(G_Q^T)$. Therefore, the expression from Equation (8) can be obtained from a the computation of the left dominant singular vectors thanks to the SVD associated to $Q-2$ cores.
The m-ranks and TT-ranks verify the following relation:

$$R'_q = \min\left(\prod_{p=1}^{q} R_p, \prod_{p=q+1}^{Q} R_p\right).$$

The last step of the FAKSETT method is to compute the kernel defined in Equations (6) and (8).

4. EXPERIMENTS

For the following datasets, a classification task is realized via SVM [7]. This relies on the similarity matrix obtained using the kernel defined previously. The kernel is computed with FAKSETT and is compared to the native method of [9].

Fig. 3. Three classes from Extended Yale Database.

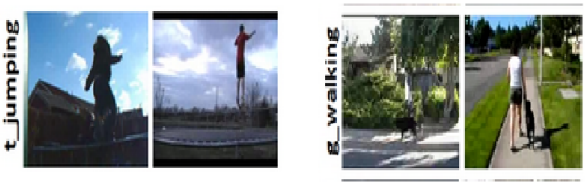

Fig. 4. Two classes from UCF11 Database.

4.1. Datasets

- **UCF11 dataset**: This dataset [17] contains 1600 video clips belonging to 11 human actions such as: *diving*, *trampoling jumping*, *walking*, *shooting*... Two human actions are chosen: *trampoling jumping* and *walking*, presented in Figure 4. They will represents 2 classes for the classification. Sequence that contains the first 240 frames from each clip video where the resolution of each RGB frame is 320×240 are considered. These clip videos can be interpreted as tensors of order 4 with dimensions $240 \times 240 \times 320 \times 3$. A total of 109 tensors are present in each class. Randomly selecting 60% of them constitutes the training set. The rest is left for the test.

- **Extended Yale dataset B**: This dataset [18] contains 28 human subjects. For each subject, there are 576 images of size 480×640 taken under 9 poses. Each pose is taken under 64 different illuminations. In that case, 3 subjects, represented in Figure 3 represents 3 classes for a classification problem. In order to construct the training and the test set, we break the tensor of each subject in 16 tensors by considering each 4 illuminations in a tensor of size $9 \times 480 \times 640 \times 4$.

4.2. Classification performance

In this section, we report on numerical experiments where we use accuracy as a performance measure.

Both the SVM regularization parameter and the kernel bandwidth γ from Equation (8) are selected from the grid of values $\{2^{-9}, 2^{-8}, ..., 2^8, 2^9\}$ by a 5 fold cross validation. All the experiments are conducted on a computer with Intel Core i7 9th generation 2.6 GHZ processor and 32 Go RAM

memory running Windows 10. Computations of SVDs are realised using optimised TensorLy (Tensor Learning in Python) library.

- Table 1 and Table 2 show very close accuracy scores between FAKSETT and method of [9] for classification tasks on both real database. This indicates that the FAKSETT method operates as efficiently as the state-of-art method. Reducing the size of the training data set (*i.e* training with less data) does not impact significantly the performances.
- However, it is noticeable from Table 3 that FAKSETT reduces significantly the running time for the computation of the factors, despite working with only $Q = 4$ order tensors. Higher orders would lead to an even higher running time gain between the two methods.

s%	m-ranks	FAKSETT	method of [9]
%50	[2,2,2,2]	$0.72(10^{-2})$	$\mathbf{0.73(10^{-2})}$
%60	[3,3,3,3]	$\mathbf{0.7(10^{-2})}$	$\mathbf{0.7(10^{-2})}$
%80	[3,3,3,3]	$0.76(10^{-2})$	$\mathbf{0.77(10^{-2})}$

Table 1. Mean accuracy (standard deviation) on test data for UCF11 database

s%	m-ranks	FAKSETT	method of [9]
%50	[1,3,2,1]	$0.98(10^{-2})$	$\mathbf{0.99(10^{-2})}$
%60	[1,2,2,1]	$\mathbf{0.99(10^{-2})}$	$\mathbf{0.99(10^{-2})}$

Table 2. Mean accuracy (standard deviation) on test data for Extended Yale database

Database	m-ranks	FAKSETT	method of [9]
UCF11	[2,2,2,2]	$\mathbf{14(0.42)}$	$69(3)$
	[3,3,3,3]	$\mathbf{15(0.63)}$	$104(5)$
Extended Yale	[1,2,2,1]	$\mathbf{2.56(0.09)}$	$9.47(0.1)$

Table 3. Mean time (standard deviation) on seconds consumed to compute HOSVD for different databases *w.r.t* to different values of multi-linear ranks.

5. CONCLUSION

Recently, supervised kernel-based learning for tensors based on a Grassmannian metric between subspaces extracted from HOSVD has been proposed. Despite a good classification accuracy, this method suffers from a high complexity cost in particular for datasets associated to Q-order tensors when $Q > 3$. In this work, we exploit some recent algebraic link between the HOSVD and the TTD to speed up the native method. We call this new supervised leaning scheme FAKSETT for Kernel-based Fast Multilinear Projection. On real datasets, we show that the FAKSETT scheme reaches a very similar classification accuracy as the state-of-art method but for a running time considerably reduced on real datatsets.

6. REFERENCES

[1] Andrzej Cichocki, Danilo Mandic, Lieven De Lathauwer, Guoxu Zhou, Qibin Zhao, Cesar Caiafa, and Huy Anh Phan, "Tensor decompositions for signal processing applications: From two-way to multiway component analysis," *IEEE Signal Processing Magazine*, vol. 32, no. 2, pp. 145–163, 2015.

[2] Nicholas D. Sidiropoulos, Lieven De Lathauwer, Xiao Fu, K. Huang, Evangelos E. Papalexakis, and Christos Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3551–3582, 2017.

[3] Tamara G. Kolda and Brett W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, September 2009.

[4] Lieven Lathauwer and Bart De Moor, "A multi-linear singular value decomposition," *Society for Industrial and Applied Mathematics*, vol. 21, pp. 1253–1278, 03 2000.

[5] Andrzej Cichocki, Namgil Lee, Ivan Oseledets, Anh-Huy Phan, Qibin Zhao, and Danilo P Mandic, "Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions," *Foundations and Trends® in Machine Learning*, vol. 9, no. 4-5, pp. 249–429, 2016.

[6] Yassine Zniyed, Rémy Boyer, André L.F. De Almeida, and Gérard Favier, "High-order tensor estimation via trains of coupled third-order cp and tucker decompositions," *Linear Algebra and its Applications*, vol. 588, pp. 304 – 337, 2020.

[7] Christopher JC Burges, "A tutorial on support vector machines for pattern recognition," *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[8] Alex J Smola and Bernhard Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.

[9] Marco Signoretto, Lieven De Lathauwer, and Johan A. K. Suykens, "A kernel-based framework to tensorial data analysis," *Neural networks : the official journal of the International Neural Network Society*, vol. 24 8, pp. 861–74, 2011.

[10] Elizabeth Newman, Misha Kilmer, and Lior Horesh, "Image classification using local tensor singular value

decompositions," *2017 IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, 2017.

[11] Tim Peeters, Anna Vilanova, and Bart ter Haar Romeny, *Analysis of Distance/Similarity Measures for Diffusion Tensor Imaging*, pp. 113–136, Springer, 01 2008.

[12] Thomas Papastergiou, Evangelia Zacharaki, and Vasileios Megalooikonomou, "Tensor decomposition for multiple-instance classification of high-order medical data," *Complexity*, vol. 2018, pp. 1–13, 12 2018.

[13] Konstantinos Makantasis, Anastasios D. Doulamis, Nikolaos D. Doulamis, and Antonis Nikitakis, "Tensor-based classification models for hyperspectral data analysis," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 12, pp. 6884–6898, 2018.

[14] Sadeep Jayasumana, Richard Hartley, Mathieu Salzmann, Hongdong Li, and Mehrtash Harandi, "Kernel methods on riemannian manifolds with gaussian rbf kernels," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 12, pp. 2464–2477, 2015.

[15] Ivan V Oseledets, "Tensor-train decomposition," *SIAM Journal on Scientific Computing*, vol. 33, no. 5, pp. 2295–2317, 2011.

[16] Yassine Zniyed, Remy Boyer, André LF de Almeida, and Gérard Favier, "A tt-based hierarchical framework for decomposing high-order tensors," *SIAM Journal on Scientific Computing*, vol. 42, no. 2, pp. A822–A848, 2020.

[17] Jingen Liu, Jiebo Luo, and Mubarak Shah, "Recognizing realistic actions from videos "in the wild"," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1996–2003.

[18] Athinodoros S. Georghiades, Peter N. Belhumeur, and David J. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. Pattern Anal. Mach. Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.