On Learning from Tensor-structured Data: A Random Tensor Theory Approach

Anonymous Authors[1]

Abstract

Leveraging on recent advances in random tensor theory, this paper proposes a theoretical analysis of learning from data that have an underlying low-rank tensor structure, in both supervised and unsupervised settings. In particular, for the supervised setting, we provide an analysis of a matched filter classifier by considering or not the low-rank structure knowledge of the data. Our analysis shows a considerable performance gain when accounting for the tensor low-rankness of the data compared to treating them as mere vectors. Moreover, experiments are conducted on a brain activity measurements dataset, which confirm the superiority of low-rank tensor learning compared to an equivalent traditional ridge classifier in the presence of few training samples.

1. Introduction

The current era of artificial intelligence tackles learning tasks leveraging millions or even billions of data. These data lie in high-dimensional spaces and often come from multiple *modes*, such as multiple modalities, multiple sensors, multiple sources, multiple types, multiple (space, time, frequency, etc.) domains. In other words, these data can naturally be seen as tensors, in which vectors and matrices are simply the 1-mode and 2-modes versions.

Tensors are a natural way to store data and their inner geometric structure is richer than the one-dimensional and the two-dimensional algebra (Landsberg, 2012). In particular, unlike matrices, low-rank tensor factorization is essentially unique under mild assumptions when the number of modes is greater than three. Their ubiquity in numerous applications becomes therefore more and more important (Sun et al., 2014), leading to a growing interest in tensor data analysis in the statistical learning community.

A large part of previous works on tensor theory applied to machine learning problems assume a low-rank representation of input data (Anandkumar et al., 2014; Kadmon & Ganguli, 2019) and estimate this representation using as main ingredient the CANDECOMP/PARAFAC decomposition (CPD) (Hitchcock, 1927). Indeed, the low-rank tensor structure is a sparsity hypothesis that is natural in the modelling of real data seen through high-dimensional inputs (Kolda & Bader, 2019). However, faced with tensor-structured data, a simple and commonly used approach consists in neglecting the structure and reshaping them into a set of vectors, to which a classical machine learning algorithm is then applied. In this work, we challenge precisely this point by highlighting the fact that *a considerable gain can be obtained by taking advantage of the low-rank tensor structure of the processed data rather than treating them as mere vectors.*

In the literature, the low-rank tensor structure has been exploited for example in tensor regression in a supervised setting (Zhou et al., 2013) or clustering in an unsupervised setting (Sun & Li, 2019). The tensor structure has been shown to enhance the performance of learning models as a key ingredient of more complex learning architectures e.g. for multi-modal data or multi-spectral images (Liang et al., 2019; Chen et al., 2020), or in the design of advanced neural network architectures by replacing the flattening operation in fully connected layers of a Convolutional Neural Network by CP-based operations (Kossaifi et al., 2020).

On top of the performance gain shown by Kossaifi et al. (2020), the reduction of the number of parameters needed to describe the learned model is also significant. Indeed, the gain in the size of the parameter space can be seen when the data samples are order k tensors and have for example a rank-one underlying structure. In this case, if the tensors dimensions are $p_1 \times \cdots \times p_k$, the dimension of the parameter space can be significantly reduced from $\prod_{j=1}^k p_j$ to $\sum_{j=1}^k p_j$.

All this literature motivates the analysis of learning algorithms when processing low-rank tensor structured data. To do so, we consider a framework where the data are supposed to be low-rank tensors perturbed by some additive noise. Then, based on recent advances in random tensor theory, we characterize the theoretical performance of simple linear methods (in both supervised and unsupervised settings) with

[1] Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

and without incorporating the knowledge of the low-rank structure. We show analytically that the incorporation of this knowledge allows to considerably improve the performance of the studied methods, in particular, when a limited amount of training samples is at hand or equivalently when data are of high-dimension. Thus, exploiting the structure of the data allows to obtain equivalent performance with far fewer samples.

Considering a framework where data are generated as rank-one tensors with additive Gaussian noise (see §3) and based on recent advances in random tensor theory (Seddik et al., 2021a) recalled in §4, the main contributions brought by this paper are three-fold:

1. We first consider a supervised learning setting where we provide a theoretical analysis of a simple matched filter classifier with and without incorporating the low-rank tensor structure of the data (§5.1).
2. We further consider an unsupervised setting and characterize the theoretical performance of a simple linear clustering approach which consists in tensor unfolding which we compare to a low-rank tensor approximation clustering approach (§5.2).
3. We provide experiments using a brain activity measurements dataset where we find that the conclusions of the theoretical study extend to real data (§6).

To the best of our knowledge, this work constitutes the first analysis towards the theoretical understanding and improvement of machine learning algorithms when processing tensor-structured data. Outstandingly, we demonstrate that random tensor theory allows for the exact characterization of the studied methods while providing practical insights about learning from low-rank tensor data. In particular, *it takes fewer training samples to achieve better performances when relying on the low-rank tensor structure of the data.*

Notations: $[n]$ denotes the set $\{1, \ldots, n\}$. Scalars are denoted by lowercase letters as a, b, c. Vectors are denoted by bold lowercase letters as $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$. Matrices are denoted by bold uppercase letters as $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$. Tensors are denoted as $\mathbf{A}, \mathbf{B}, \mathbf{C}$. $T_{i_1,\ldots,i_d}$ denotes the entry $(i_1, \ldots, i_d)$ of the tensor $\mathbf{T}$. $\langle \boldsymbol{u}, \boldsymbol{v} \rangle = \sum_i u_i v_i$ denotes the scalar product between $\boldsymbol{u}$ and $\boldsymbol{v}$, the ℓ_2-norm of a vector $\boldsymbol{u}$ is denoted as $\|\boldsymbol{u}\|^2 = \langle \boldsymbol{u}, \boldsymbol{u} \rangle$. $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{t^2}{2}} dt$ corresponds to the Gaussian tail function. $\xrightarrow{\text{a.s.}}$ stands for the almost sure convergence and $\xrightarrow{\mathcal{D}}$ for the convergence in distribution.

2. Tensor operations

We briefly recall in this section some tensor notations and operations that shall be used throughout the paper.

Inner product and norm: The inner product of two same-sized order k tensors $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{p_1 \times \cdots \times p_k}$ is the sum of the products of their entries and is denoted as $\langle \mathbf{X}, \mathbf{Y} \rangle = \sum_{i_1,\ldots,i_k} X_{i_1 \cdots i_k} Y_{i_1 \cdots i_k}$. In particular, the norm $\|\mathbf{X}\|$ of $\mathbf{X} \in \mathbb{R}^{p_1 \times \cdots \times p_k}$ is $\|\mathbf{X}\|^2 = \langle \mathbf{X}, \mathbf{X} \rangle$.

Rank-one tensors An order k tensor $\mathbf{X} \in \mathbb{R}^{p_1 \times \cdots \times p_k}$ is said to be a *rank-one* tensor if it can be written as the outer product of k vectors $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_k$, i.e., $\mathbf{X} = \bigotimes_{j=1}^k \boldsymbol{a}_j = \boldsymbol{a}_1 \otimes \cdots \otimes \boldsymbol{a}_k$, where the outer product $\bigotimes_{i=1}^k \boldsymbol{a}_i$ is defined such that $\left(\bigotimes_{j=1}^k \boldsymbol{a}_j \right)_{i_1 \ldots i_k} = \prod_{j=1}^k (\boldsymbol{a}_j)_{i_j}$, i.e., each element of the rank-one tensor is the product of the corresponding vectors elements.

Tensor multiplication: The j-mode (matrix) product of a tensor $\mathbf{X} \in \mathbb{R}^{p_1 \times \cdots \times p_k}$ with a matrix $\boldsymbol{M} \in \mathbb{R}^{m \times p_j}$ is denoted $\mathbf{X} \times_j \boldsymbol{M}$ and is a tensor of size $p_1 \times \cdots \times p_{j-1} \times m \times p_{j+1} \times \cdots \times p_k$. Element-wise, the j-mode (matrix) product is defined as $(\mathbf{X} \times_j \boldsymbol{M})_{i_1 \cdots i_{j-1} k i_{j+1} \cdots i_k} = \sum_{i_j=1}^{p_j} X_{i_1 \cdots i_k} M_{k i_j}$. Similarly, the j-mode (vector) product or *contraction* of an order k tensor $\mathbf{X} \in \mathbb{R}^{p_1 \times \cdots \times p_k}$ with a vector $\boldsymbol{v} \in \mathbb{R}^{p_j}$ is also denoted as $\mathbf{X} \times_j \boldsymbol{v}$ and results in a tensor of order $k-1$ of dimension $p_1 \times \cdots \times p_{j-1} \times p_{j+1} \times \cdots \times p_k$. Element-wise, the j-mode contraction is defined as $(\mathbf{X} \times_j \boldsymbol{v})_{i_1 \cdots i_{j-1} i_{j+1} \cdots i_k} = \sum_{i_j=1}^{p_j} X_{i_1 \cdots i_k} v_{i_j}$, which basically consists in computing the inner product of each mode-j *fiber* with the vector $\boldsymbol{v}$.

Tensor Rank and the CANDECOMP/PARAFAC Decomposition (CPD): The CP decomposition (Hitchcock, 1927; Landsberg, 2012) produces a decomposition of a tensor $\mathbf{X} \in \mathbb{R}^{p_1 \times \cdots \times p_k}$ into a sum of rank-one tensors, i.e., $\mathbf{X} = \sum_{i=1}^r \bigotimes_{j=1}^k \boldsymbol{a}_j^{(i)}$. The rank of $\mathbf{X}$ denoted $\operatorname{rank}(\mathbf{X})$ is defined as the smallest possible integer r for which $\mathbf{X}$ decomposes as above.

3. Data model: rank-one random tensors

Let the training samples be n independent tensor-structured data $\mathbf{X}_1, \ldots, \mathbf{X}_n$ each of order k and of dimension $p_1 \times \cdots \times p_k$. We denote the dimensions $p = \sum_{j=1}^k p_j$ and $d = \prod_{j=1}^k p_j$. We suppose that the $\mathbf{X}_i$'s are distributed in two classes $\mathcal{C}_1$ and $\mathcal{C}_2$ (of cardinality n_1 and n_2 respectively – that is $n = n_1 + n_2$), such that for $\mathbf{X}_i \in \mathcal{C}_a$ with $a \in \{1, 2\}$,

$$\mathbf{X}_i = (-1)^a \bigotimes_{j=1}^k \boldsymbol{\mu}_j + \mathbf{Z}_i \in \mathbb{R}^{p_1 \times \cdots \times p_k}, \tag{1}$$

where $\mathbf{Z}_i$ is a random tensor with i.i.d. standard Gaussian entries, $\boldsymbol{\mu}_j \in \mathbb{R}^{p_j}$ for $j \in [k]$ are independent from the $\mathbf{Z}_i$'s and $\mathbf{M} = \bigotimes_{j=1}^k \boldsymbol{\mu}_j$ stands for the outer product between all the $\boldsymbol{\mu}_j$'s. In the context of supervised binary classification, we are further given a vector of labels $\boldsymbol{y} \in \mathbb{R}^n$ such that $y_i = -1$ for $\mathbf{X}_i \in \mathcal{C}_1$ and $y_i = 1$ for $\mathbf{X}_i \in \mathcal{C}_2$.

We denote the training data tensor $\mathbf{X} = [\mathbf{X}_1, \ldots, \mathbf{X}_n] \in \mathbb{R}^{p_1 \times \cdots \times p_k \times n}$ by concatenating all the $\mathbf{X}_i$ along the $(k+1)$-th mode of dimension n. $\mathbf{X}$ expresses in tensor form as

$$\mathbf{X} = \mathbf{M} \otimes \boldsymbol{y} + \mathbf{Z}, \tag{2}$$

where $\mathbf{Z} = [\mathbf{Z}_1, \ldots, \mathbf{Z}_n] \in \mathbb{R}^{p_1 \times \cdots \times p_k \times n}$. Given the rank-one structure of the tensor mean $\mathbf{M}$, the outer product $\mathbf{M} \otimes \boldsymbol{y}$ results in a rank-one tensor of order $k+1$. As such, the data tensor $\mathbf{X}$ is typically called a *rank-one spiked random tensor model* of order $k+1$, where the signal part $\mathbf{M} \otimes \boldsymbol{y}$ is called the *spike* and $\mathbf{Z}$ corresponds to the noise part. We will show subsequently (in Section 5) that the characterization of several learning methods applied on the data model $\mathbf{X}$ boils down to the study of spiked random tensors. Henceforth, we present in the next section some recent advances on random tensor theory that precisely deals with such models.

4. Random tensor theory

Random tensor theory consists in generalizing classical random matrix theory (Marčenko & Pastur, 1967; Baik et al., 2005) to random tensor models. The first line of research in this topic was proposed by Montanari & Richard (2014) who introduced the concept of tensor PCA. Afterwards, many works have focused on the analysis of *symmetric* random tensors (Perry et al., 2020; Lesieur et al., 2017; Handschy, 2019; Jagannath et al., 2020; Goulart et al., 2021). However, symmetric random tensor models have limited applications in machine learning since real data structures do not necessarily have such symmetric properties. In a very recent work by Seddik et al. (2021a), a study of *asymmetric* spiked random tensors has been carried out. It considers an observed k-order tensor $\mathbf{T}$ of the form

$$\mathbf{T} = \beta \bigotimes_{j=1}^{k} \boldsymbol{u}_j + \frac{1}{\sqrt{\sum_{i=1}^k p_i}} \mathbf{Z} \in \mathbb{R}^{p_1 \times \cdots \times p_k}, \tag{3}$$

where $\boldsymbol{u}_j \in \mathbb{R}^{p_j}$ for $j \in [k]$ are unitary vectors, $\mathbf{Z}$ is a random tensor with i.i.d. $\mathcal{N}(0,1)$ entries and $\beta > 0$ is a parameter controlling the signal-to-noise ratio (SNR). The study has provided asymptotic evaluation of λ and $\langle \boldsymbol{u}_j, \boldsymbol{v}_j \rangle$ with $\lambda \bigotimes_{j=1}^k \boldsymbol{v}_j$ being the best rank-one approximation of $\mathbf{T}$ given by the maximum likelihood estimator (MLE) as

$$\operatorname*{arg\,min}_{\lambda > 0, \{\boldsymbol{v}_j \mid \|\boldsymbol{v}_j\| = 1, j \in [k]\}} \left\| \mathbf{T} - \lambda \bigotimes_{j=1}^{k} \boldsymbol{v}_j \right\|_{\mathrm{F}}^2. \tag{4}$$

This study was carried out in the high-dimensional regime, where $p_j \to \infty$ with $\frac{p_j}{\sum_{i=1}^k p_i} \to c_j \in [0,1]$. Precisely, Seddik et al. (2021a) provided the following results which will be subsequently applied in order to assess the performance of the learning algorithms studied in the present work.

4.1. k-order spiked random tensors

Theorem 4.1 (Theorem 8 in (Seddik et al., 2021a))**.** *As $p_j \to \infty$ with $\frac{p_j}{\sum_{i=1}^k p_i} \to c_j \in [0,1]$, for $k \geq 3$, there exists β_s such that for $\beta > \beta_s$,*

$$\begin{cases} \lambda \xrightarrow{a.s.} \lambda^\infty(\beta), \\ |\langle \boldsymbol{u}_j, \boldsymbol{v}_j \rangle| \xrightarrow{a.s.} q_j(\lambda^\infty(\beta)), \end{cases}$$

where $\lambda^\infty(\beta)$ satisfies[1] *$f(\lambda^\infty(\beta), \beta) = 0$ with $f(z,\beta) = z + g(z) - \beta \prod_{j=1}^k q_j(z)$, $q_j(z) = \left(\frac{\alpha_j(z)^{k-3}}{\prod_{i \neq j} \alpha_i(z)} \right)^{\frac{1}{2k-4}}$, $\alpha_j(z) = \frac{\beta}{z + g(z) - g_j(z)}$, $g_j(z) = \frac{g(z)+z}{2} - \frac{\sqrt{4c_j + (g(z)+z)^2}}{2}$ and $g(z)$ being the unique solution to $g(z) = \sum_{j=1}^k g_j(z)$.*

In essence, for a SNR β large enough, Theorem 4.1 predicts a non-zero correlation between the signal components (i.e., the $\boldsymbol{u}_j$'s) and their estimated counterparts (i.e., the $\boldsymbol{v}_j$'s) by the MLE. We refer the reader to (Seddik et al., 2021a) for a more detailed discussion.

4.2. Cubic spiked random tensors

In the case of cubic tensors, i.e., $k = 3$ and all the tensor dimensions are equal ($p_1 = p_2 = p_3$), λ^∞ and $q_j(\lambda^\infty)$ in Theorem 4.1 have closed form expressions in terms of β.

Corollary 4.2 (Corollary 3 in (Seddik et al., 2021a))**.** *As $p_j \to \infty$, for $\beta > \frac{2\sqrt{3}}{3}$,*

$$\begin{cases} \lambda \xrightarrow{a.s.} \lambda^\infty(\beta) = \sqrt{\frac{\beta^2}{2} + 2 + \frac{\sqrt{3}\sqrt{(3\beta^2-4)^3}}{18\beta}}, \\ |\langle \boldsymbol{u}_j, \boldsymbol{v}_j \rangle| \xrightarrow{a.s.} \bar{q}(\beta), \end{cases}$$

with $\bar{q}(\beta) = \dfrac{\sqrt{9\beta^2 - 12 + \frac{\sqrt{3}\sqrt{(3\beta^2-4)^3}}{\beta}} + \sqrt{9\beta^2 + 36 + \frac{\sqrt{3}\sqrt{(3\beta^2-4)^3}}{\beta}}}{6\sqrt{2}\beta}$.

4.3. Spiked random matrices

For $k = 2$, the model in Eq. (3) becomes a so-called *spiked random matrix* which has been extensively studied using random matrix theory (Baik et al., 2005; Benaych-Georges & Nadakuditi, 2011; Capitaine et al., 2009; Péché, 2006; Ben Arous et al., 2021). Theorem 4.1 covers also such models by not letting all tensor dimensions go to infinity which yields the following corollary.

Corollary 4.3 (Corollary 5 in (Seddik et al., 2021a))**.** *As $p_1, p_2 \to \infty$ with $\frac{p_1}{p_1+p_2} \to c \in [0,1]$, for $\beta > \sqrt[4]{c(1-c)}$,*

$$\begin{cases} \lambda \xrightarrow{a.s.} \lambda^\infty(\beta) = \sqrt{\beta^2 + 1 + \frac{c(1-c)}{\beta^2}}, \\ |\langle \boldsymbol{u}_1, \boldsymbol{v}_1 \rangle| \xrightarrow{a.s.} \frac{1}{\kappa(\beta,c)}, \quad |\langle \boldsymbol{u}_2, \boldsymbol{v}_2 \rangle| \xrightarrow{a.s.} \frac{1}{\kappa(\beta,1-c)}, \end{cases}$$

where $\kappa(\beta,c) = \beta\sqrt{\frac{\beta^2(\beta^2+1)-c(c-1)}{(\beta^4+c(c-1))(\beta^2+1-c)}}$.

[1] We will sometimes omit the dependence on β for simplicity.

Remark 4.4 (On the high-dimensional regime). Although the results have been derived in an asymptotic regime where p_j tends to infinity, the rates of convergence are of order $\mathcal{O}(p_j^{-1/2})$ which allows their application even for non-asymptotic regimes (i.e., finite tensor dimensions).

5. Main results

In this section, we present both supervised and unsupervised applications of the random tensor tools presented previously. Throughout the following subsections, we assume a high-dimensional regime, i.e., the number of training samples n scales linearly with the tensor dimensions p_j while $\|\boldsymbol{\mu}_j\|$ remains constant.

Assumption 5.1 (Growth rate)**.** For all $j \in [k]$, $\frac{p_j}{n} = \mathcal{O}_n(1)$ and $\|\boldsymbol{\mu}_j\| = \mathcal{O}_n(1)$[2].

When studying machine learning methods with random matrix theory (Pennington & Worah, 2017; Louart et al., 2018; Ali & Couillet, 2017; Mai & Couillet, 2018; Tiomoko et al., 2020; Seddik et al., 2021b), it is commonly assumed that the feature size scales linearly with the number of samples, which yields that $\prod_{j=1}^{k} p_j$ must scale linearly with n in the supposed case of tensor data. However, for $k \geq 2$, this requirement imposes a large number of training samples n which might be difficult to achieve in practical settings. As such Assumption 5.1 is more realistic from the practical view point.

5.1. On supervised learning

Given the training data tensor $\mathbf{X}$ in Eq. (2) and the corresponding labels vector $\boldsymbol{y}$, a basic learning approach (Tiomoko et al., 2021) consists in reshaping $\mathbf{X}$ into a data matrix $\operatorname{Mat}(\mathbf{X}) \in \mathbb{R}^{d\times n}$ with $d = \prod_{j=1}^{k} p_j$, and then learning a matched filter classifier with parameters $\boldsymbol{w} \equiv \operatorname{vec}(\mathbf{W}) \in \mathbb{R}^d$ ($\mathbf{W} \in \mathbb{R}^{p_1\times\cdots\times p_k}$) as[3]

$$\boldsymbol{w} = \frac{1}{\sqrt{np}} \operatorname{Mat}(\mathbf{X})\boldsymbol{y}, \tag{5}$$

where we recall that $p = \sum_{j=1}^{k} p_j$, for which the decision function (for a new datum $\tilde{\mathbf{X}}_i \in \mathcal{C}_a$) is given by $g(\tilde{\mathbf{X}}_i) = \langle \boldsymbol{w}, \operatorname{vec}(\tilde{\mathbf{X}}_i)\rangle$ which is equivalent in tensor notations to

$$g(\tilde{\mathbf{X}}_i) = \langle \mathbf{W}, \tilde{\mathbf{X}}_i\rangle \underset{\mathcal{C}_2}{\overset{\mathcal{C}_1}{\lessgtr}} 0, \quad \mathbf{W} = \frac{1}{\sqrt{np}} \mathbf{X} \times_{k+1} \boldsymbol{y}. \tag{6}$$

As such, the matched filter classifier does not consider the low-rank tensor structure of the underlying data model and treats the data as mere vectors. For now, we first provide our first result characterizing the theoretical performance of the matched filter for the data model in Eq. (2).

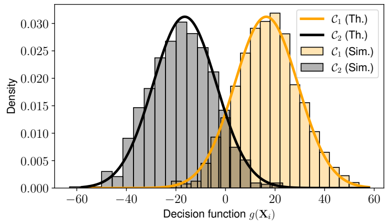

Figure 1. Theoretical versus empirical histogram of the decision function $g(\tilde{\mathbf{X}}_i)$ for the matched filter classifier as per Proposition 5.2. We considered $n = 200$ training data ($n_1 = n_2 = 100$) that are tensors of order 3 and of dimensions $p_1 = p_2 = p_3 = 20$, distributed as the rank-one tensor model in Eq. (1) with the $\boldsymbol{\mu}_j$'s being randomly sampled vectors from a sphere such that $\|\mathbf{M}\| = 3$.

Proposition 5.2 (Performance of the matched filter classifier)**.** *Under Assumption 5.1, for $\tilde{\mathbf{X}}_i \in \mathcal{C}_a$ with $a \in \{1, 2\}$ independent from the training set $\mathbf{X}$,*

$$\frac{1}{\sigma}\left(g(\tilde{\mathbf{X}}_i) - m_a\right) \xrightarrow{\mathcal{D}} \mathcal{N}(0, 1),$$

where $m_a = (-1)^a \|\mathbf{M}\|^2 \sqrt{\frac{n}{p}}$ and $\sigma = \sqrt{\frac{n}{p}\|\mathbf{M}\|^2 + \frac{d}{p}}$. Moreover, the misclassification error verifies with probability one $\mathbb{P}\left((-1)^a g(\tilde{\mathbf{X}}_i) < 0 \mid \tilde{\mathbf{X}}_i \in \mathcal{C}_a\right) - Q\left(\frac{|m_a|}{\sigma}\right) \to 0$.

Proof. See Appendix A.1. □

Proposition 5.2 states that the performance of the matched filter classifier depends solely on $\|\mathbf{M}\|$ and the dimension ratios $\frac{n}{p}$ and $\frac{d}{p}$. Moreover, since the data are mean-wise centred as per Eq. (1), the optimal classification is obtained by taking the sign of the decision function which is also suggested theoretically since the optimal threshold is $\frac{m_1+m_2}{2} = 0$. Figure 1 provides a histogram of the decision function of the matched filter classifier and its theoretical estimate through Proposition 5.2. Under Assumption 5.1, the mean m_a remains constant while the variance σ increases due to the term $\frac{d}{p}$ as the dimension of data increases. This phenomenon highlights the drawback of treating the input data as simple vectors and not exploiting the low-rank structure of the mean tensor $\mathbf{M}$.

CP-based approach: However, such low-rank structure can be recovered by performing a tensor decomposition on the weights tensor $\mathbf{W}$, since it is a noisy version of $\mathbf{M}$. Precisely, recall the definition of $\mathbf{W}$ in Eq. (6) and $\mathbf{X}$ in Eq. (2), thus

$$\mathbf{W} = \sqrt{\frac{n}{p}} \bigotimes_{j=1}^{k} \boldsymbol{\mu}_j + \frac{1}{\sqrt{p}} \tilde{\mathbf{Z}}, \tag{7}$$

[2] The notation $a = \mathcal{O}_n(1)$ means that a converges to a constant not depending on n if $n \to \infty$.

[3] The normalization by $\sqrt{np}$ is considered for convenience and does not affect the performances of the considered methods. Moreover, under Assumption 5.1 the quantities n and p are of the same order which is equivalent to the standard normalization by n.

Algorithm 1 CP-based matched filter classifier

```
Input: Tensor data $\mathbf{X}$, labels vector $\boldsymbol{y}$ and test datum $\tilde{\mathbf{X}}_i$
Output: Label $\tilde{y}_i = \operatorname{sign}(g_{\mathrm{CP}}(\tilde{\mathbf{X}}_i))$
Compute the matched filter $\mathbf{W} = \frac{1}{\sqrt{np}} \mathbf{X} \times_{k+1} \boldsymbol{y}$
Extract rank-1 approx of $\mathbf{W}$: $\bigotimes_{j=1}^k \hat{\boldsymbol{\mu}}_j = \mathrm{CPD}(\mathbf{W}, 1)$
Set the decision function as $g_{\mathrm{CP}}(\tilde{\mathbf{X}}_i) = \langle \bigotimes_{j=1}^k \hat{\boldsymbol{\mu}}_j, \tilde{\mathbf{X}}_i \rangle$
```

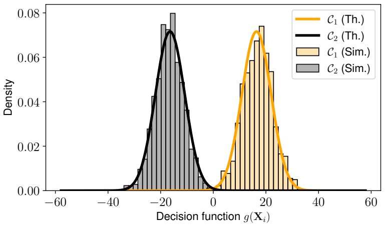

Figure 2. Theoretical versus empirical histogram of the decision function $g_{\mathrm{CP}}(\tilde{\mathbf{X}}_i)$ for the CP-based matched filter classifier as per Proposition 5.4. We considered the same parameters as Figure 1.

where $\tilde{\mathbf{Z}} = \frac{1}{\sqrt{n}} \mathbf{Z} \times_{k+1} \boldsymbol{y} = \frac{1}{\sqrt{n}} \sum_{i=1}^n y_i \mathbf{Z}_i$. Since $\tilde{\mathbf{Z}}$ is a sum of n i.i.d. random tensors normalized by $\sqrt{n}$, then $\tilde{\mathbf{Z}}$ is also a random tensor with i.i.d. standard Gaussian entries.

Remark 5.3. Note that for the actual supervised learning setting, the Gaussianity assumption on the $\mathbf{Z}_i$ might be relaxed to any symmetric distribution with zero mean and unit variance, for which $\tilde{\mathbf{Z}}$ remains a random tensor with i.i.d. standard Gaussian entries by the central limit theorem.

As such, $\mathbf{W}$ is precisely a spiked random tensor model of the same form as the model in Eq. (3). In order to leverage the low-rank structure of $\mathbf{W}$, we consider applying a rank-one CP approximation which yields estimates of the $\boldsymbol{\mu}_j$'s and then replace the weights $\mathbf{W}$ in the decision function by their rank-one approximation. Algorithm 1 provides a pseudo-code for the CP-based matched filter classifier. In essence, extracting the rank-one component constitutes in a denoising scheme and allows to considerably reduce the variance of the decision function, thereby providing a better classification accuracy. Our following result characterizes theoretically the performance of the CP-based matched filter classifier, by means of the random tensor theory results described in Section 4.

Proposition 5.4 (Performance of the CP-based matched filter classifier)**.** *Under Assumption 5.1, for $\tilde{\mathbf{X}}_i \in \mathcal{C}_a$ with $a \in \{1, 2\}$ independent from the training set $\mathbf{X}$,*

$$\frac{1}{\sigma}\left(g_{CP}(\tilde{\mathbf{X}}_i) - m_a\right) \xrightarrow{\mathcal{D}} \mathcal{N}(0,1),$$

where $m_a = (-1)^a \sigma \|\mathbf{M}\| \prod_{j=1}^k q_j \left(\lambda^\infty\left(\|\mathbf{M}\|\sqrt{\frac{n}{p}}\right)\right)$ and $\sigma = \lambda^\infty\left(\|\mathbf{M}\|\sqrt{\frac{n}{p}}\right)$, where $\lambda^\infty(\cdot)$ and $q_j(\cdot)$ are defined in Theorem 4.1. Furthermore, the misclassification error verifies with probability one $\mathbb{P}\left((-1)^a g_{CP}(\tilde{\mathbf{X}}_i) < 0 \mid \tilde{\mathbf{X}}_i \in \mathcal{C}_a\right) - Q\left(\frac{|m_a|}{\sigma}\right) \to 0$.

Proof. See Appendix A.2. □

Proposition 5.4 states that the performance of the CP-based matched filter classifier depends on $\|\mathbf{M}\|$ and the ratio $\frac{p}{n}$, but not on the ratio $\frac{d}{p}$ as was the case for the matched filter case in Proposition 5.2. *We particularly highlight that the variance σ for the CP-based classifier remains constant under Assumption 5.1, thereby yielding a better classification accuracy for small values of the number of training samples n.* Indeed, Figure 3 depicts the theoretical versus empirical misclassification error for both methods, from which we notice that the CP-based matched filter classifier yields drastically better performances (almost closer to the oracle which assumes perfect knowledge of $\mathbf{M}$) when n is not too large, or alternatively when the dimension of data is high. Indeed, Figure 4 depicts the misclassification error of both methods as a function of the ratio $\frac{n}{p}$ and $\|\mathbf{M}\|$, where we notice that the CP-based matched filter classifier performs better when $\frac{n}{p}$ is not large. This toy example clearly demonstrates that one can take benefit of the underlying data structure, if such information is available. We will see that these conclusions also extend to an unsupervised setting, where no labels are provided.

5.2. On unsupervised learning

In a setting where only n training samples $\mathbf{X}_1, \ldots, \mathbf{X}_n$ are provided without their corresponding labels, one would rely on unsupervised learning to estimate their classes. Given the data model in Eq. (2), a simple unsupervised learning approach (Ng et al., 2002) consists in reshaping $\mathbf{X}$ as

$$\boldsymbol{X} = \mathrm{Mat}(\mathbf{X}) = \mathrm{vec}(\mathbf{M})\boldsymbol{y}^\top + \mathrm{Mat}(\mathbf{Z}) \in \mathbb{R}^{d\times n}, \qquad (8)$$

then estimating the labels $\boldsymbol{y}$ through the dominant eigenvector of the Gram matrix $\boldsymbol{X}^\top \boldsymbol{X}$ denoted $\hat{\boldsymbol{y}}$, which coincides with the dominant right singular vector of $\boldsymbol{X}$. Therefore, Corollary 4.3 applies here to assess the performance of this *linear spectral method*. We precisely have the following proposition. Without loss of generality, we further assume that the data are ordered following their class order.

Proposition 5.5 (Performance of linear spectral clustering)**.** *Let $\hat{\boldsymbol{y}}$ be the right singular vector of $\boldsymbol{X}$ corresponding to its largest singular value. The estimated class for the datum $\mathbf{X}_i$ is given as $\hat{\mathcal{C}}_i = \operatorname{sign}(\hat{y}_i)$. Then under Assumption 5.1,*

$$\frac{1}{\sigma}\left(\sqrt{n}\hat{y}_i - \alpha y_i\right) \xrightarrow{\mathcal{D}} \mathcal{N}(0,1),$$

where $\alpha = \kappa\left(\|\mathbf{M}\|\sqrt{\frac{n}{d+n}}, \frac{n}{d+n}\right)^{-1}$, $\sigma = \sqrt{1-\alpha^2}$ and $\kappa(\cdot,\cdot)$ is defined in Corollary 4.3. Furthermore, the misclassification error is given with probability one by $Q\left(\frac{\alpha}{\sqrt{1-\alpha^2}}\right)$.

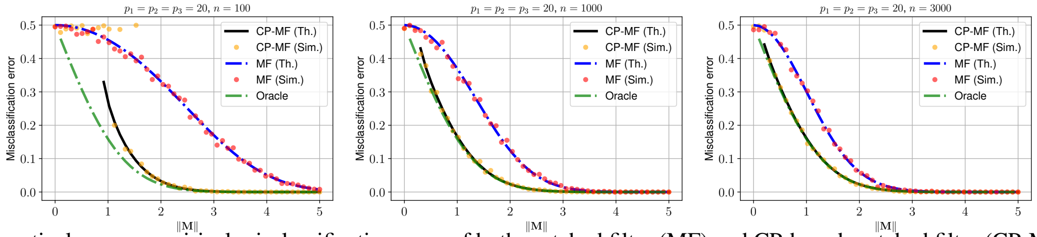

Figure 3. Theoretical versus empirical misclassification error of both matched filter (MF) and CP-based matched filter (CP-MF) classifiers. We considered n training data as order 3 tensors of dimensions $p_1 = p_2 = p_3 = 20$ having a rank-one structure as in Eq. (1) with the $\boldsymbol{\mu}_j$'s being randomly sampled vectors. **"CP-MF performs better than MF when few training samples are available".**

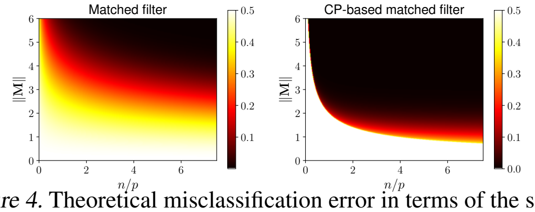

Figure 4. Theoretical misclassification error in terms of the signal strength $\|\mathbf{M}\|$ and the ratio $\frac{n}{p}$ for both MF and CP-MF as per Propositions 5.2 and 5.4 respectively.

Proof. See Appendix A.3. □

Proposition 5.5 states that the entries of the estimated right singular vector corresponding to the largest singular value of $\boldsymbol{X}$ is a Gaussian random variable, whose mean and variance depend on $\|\mathbf{M}\|$ and the ratio $c = \frac{n}{d+n}$. Essentially, in order to obtain a non-zero correlation between $\hat{\boldsymbol{y}}$ and $\boldsymbol{y}$, the signal strength $\|\mathbf{M}\|$ must be greater than $\frac{\sqrt[4]{c(1-c)}}{\sqrt{c}}$ (see Corollary 4.3). However, under Assumption 5.1, the ratio $\frac{n}{d+n} \to 0$ if $n \to \infty$, thereby yielding a high misclassification error. Figure 5-(a) depicts the estimated singular vector along with its theoretical mean and fluctuations as per Proposition 5.5. In contrast, extracting the low-rank structure of the data tensor allows to improve the classification performance. Indeed, given the data model in Eq. (2), computing a rank-1 approximation of $\mathbf{X}$ and extracting the corresponding $(k+1)$-th mode component yields an estimation of the labels vector $\boldsymbol{y}$. We precisely have the following proposition characterizing the performance of *the CP-based clustering method.*

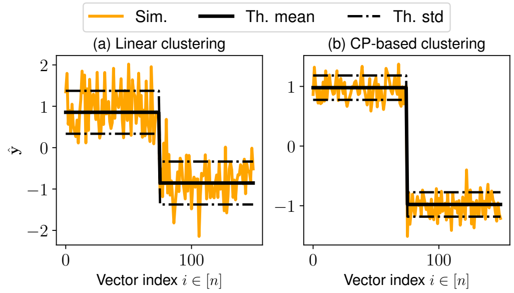

Figure 5. In orange: (a) the right singular vector of $\mathrm{Mat}(\mathbf{X})$ corresponding to its largest singular value, (b) the $(k+1)$-th mode vector of a rank-one CP decomposition of $\mathbf{X}$. We considered $k = 2$ and $n_1 = n_2 = 75$ square matrices $\mathbf{X}_i$ of size $p = 150$ generated as the model in Eq. (1) with $\|\mathbf{M}\| = 5$. In black, the theoretical means and fluctuations according to Propositions 5.5 and 5.6 respectively.

Proposition 5.6 (Performance of CP-based clustering)**.** *Let $\hat{\boldsymbol{y}}$ be the $(k+1)$-th mode component of the rank-1 tensor approximation of $\mathbf{X}$. The estimated class for the datum $\mathbf{X}_i$ is given as $\hat{\mathcal{C}}_i = \operatorname{sign}(\hat{y}_i)$. Then under Assumption 5.1,*

$$\frac{1}{\sigma}\left(\sqrt{n}\hat{y}_i - \alpha y_i\right) \xrightarrow{\mathcal{D}} \mathcal{N}(0,1),$$

where $\alpha = q_{k+1}\left(\lambda^\infty\left(\|\mathbf{M}\|\sqrt{\frac{n}{p+n}}\right)\right)$, $\sigma = \sqrt{1-\alpha^2}$ and $q_{k+1}(\cdot)$ and $\lambda^\infty(\cdot)$ are defined in Theorem 4.1. Furthermore, the misclassification error is given with probability one by $Q\left(\frac{\alpha}{\sqrt{1-\alpha^2}}\right)$.

Proof. See Appendix A.4. □

As for the linear clustering approach, the estimated labels vector $\hat{\boldsymbol{y}}$ with CP decomposition has Gaussian entries centered on the scaled labels $\boldsymbol{y}$ with a scaling factor α and fluctuations depending on such α. However, now the clustering performance depends on $\|\mathbf{M}\|$ and the ratio $\frac{n}{p+n}$, thereby yielding the same clustering performance as n increases and p being of at least of the same order as n. Figure 5-(b) depicts the estimated labels with CP decomposition with its theoretical mean and fluctuations as per Proposition 5.6. From Figure 5 we note that CP decomposition yields lower variance compared to a classical linear approach, thereby allowing better clustering performance.

To best illustrate the comparison between linear clustering and CP-based clustering, let us suppose that the training data are matrices of dimension $p_1 = p_2 = n$, hence $\frac{n}{p+n} = \frac{1}{3}$. In this case, the performance of CP-based clustering is given in closed form thanks by 4.2. Essentially, in order to have a correlation between $\hat{\boldsymbol{y}}$ and $\boldsymbol{y}$, the signal strength $\|\mathbf{M}\|$ must be greater than 2, i.e., there is a phase transition independent of n. On the other hand, for linear clustering, the minimal signal strength writes as $\frac{\sqrt[4]{c(1-c)}}{\sqrt{c}}$ with $c = \frac{1}{n+1}$, i.e., an

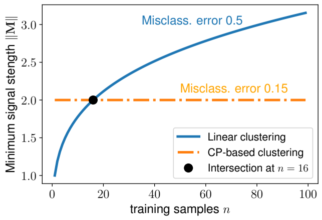

Figure 6. Minimum signal strength $\|\mathbf{M}\|$ corresponding to the theoretical phase transition above which $\hat{\boldsymbol{y}}$ starts to correlate with $\boldsymbol{y}$, in terms of n. The data are considered to be n matrices of size $n \times n$.

increasing function of n. Figure 6 depicts the minimum signal strength for both methods in terms of n.

In terms of classification error, when $\|\mathbf{M}\|$ is set to the minimum value for which it is theoretically possible to obtain a correlation between $\hat{\boldsymbol{y}}$ and $\boldsymbol{y}$. The obtained misclassification error for linear clustering is $Q(0) = 0.5$ while CP-based clustering yields a misclassification error of $Q\left(\frac{\alpha}{\sqrt{1-\alpha^2}}\right) \approx 0.15$ with $\alpha = \frac{\sqrt{2}}{2}$ (corresponding to the minimal possible alignment as per Corollary 4.2), that is a lower misclassification error with CP-based clustering. Figure 7 depicts the theoretical misclassification error varying the tensor dimensions, the number of samples n and the signal strength $\|\mathbf{M}\|$ where we highlight the superiority of CP-based clustering in theory.

However, for $k+1 \geq 3$ performing CP decomposition is NP-hard (Montanari & Richard, 2014; Biroli et al., 2020) when the signal strength is not large enough. Typically under Assumption 5.1, the signal strength must satisfy $\|\mathbf{M}\| > Cn^{\frac{k}{2}}$ for some constant C independent of n, in order to ensure recovery of the signal $\boldsymbol{y}$ in polynomial time. On the other hand, applying linear clustering has a polynomial time complexity since it consists in applying SVD. We refer the reader to (Lesieur et al., 2017; Jagannath et al., 2020; Huang et al., 2020) concerning the algorithmic phase transitions.

6. Experiments on real data

In the section, we report experiments to highlight the superiority of a CP-based learning approach compared to a standard learning method for which tensor data are reshaped as vectors. Precisely, given some tensor training samples $\mathbf{X}_1, \ldots, \mathbf{X}_n \in \mathbb{R}^{p_1 \times \cdots \times p_k}$ with their labels $y_1, \ldots, y_n \in \{-1, 1\}$, we consider a *CP-based ridge-type regressor* (Zhou et al., 2013) which consists in optimizing[4]

$$\begin{aligned} &\operatorname*{arg\,min}_{\mathbf{W}, b} \sum_{i=1}^{n} (y_i - \langle \mathbf{W}, \mathbf{X}_i \rangle - b)^2 + \lambda \|\mathbf{W}\|^2, \\ &\text{subject to} \quad \mathbf{W} = \sum_{\ell=1}^{r} \bigotimes_{j=1}^{k} \hat{\boldsymbol{\mu}}_j^{(\ell)}, \end{aligned} \tag{9}$$

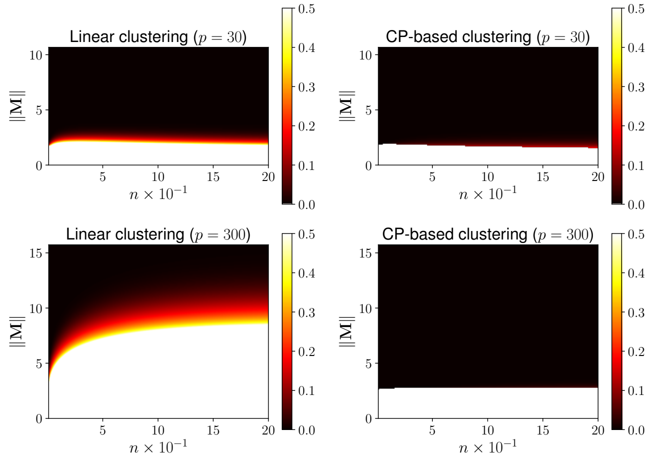

Figure 7. Theoretical misclassification error in terms of the signal strength $\|\mathbf{M}\|$ and n for both linear clustering and CP-based clustering as per Propositions 5.5 and 5.6 respectively.

where $\lambda > 0$ is a regularization parameter (set to $\lambda = 1$ in all experiments). The CP regressor is a simple ridge regressor with an additional rank-r tensor constraint on the weights tensor $\mathbf{W}$. Specifically, when $\lambda \gg 1$ the corresponding ridge regressor (without the rank-r constraint) boils down to the matched filter classifier of Section 5.1. In essence, the matched filter is an optimal classifier for the data model in Eq. (1) (Tiomoko et al., 2021), while for more realistic data, which might have an additional covariance profile, the above regularized CP regressor model is more suitable and yields better performances as we will see in the sequel.

It should be noted that the theoretical analysis of the the CP regressor is more challenging compared to the studied CP-based matched filter. However, we will see that the theoretical insights extend to the CP regressor on real data.

6.1. Dataset Description

The conducted experiments rely on a brain activity measurements dataset (Huang et al., 2021). The dataset consists of multivariate fNIRS time series recordings from 68 participants for four possible mental workload intensity levels depending on the difficulty of the task assigned to the subjects (0-back, 1-back, 2-back and 4-back). For illustration, we considered the binary classification distinguishing low workload (0-back) from high workload (2-back).

Each fNIRS recording is a multivariate time series representing brain activity across the sequence of 8 features: (i) 2 blood chemical concentration changes (oxygenated hemoglobin and deoxygenated hemoglobin) (ii) 2 optical data types used for the measurement (intensity and phase) (iii) 2 spatial locations on the forehead. For each subject, the data collection produces an fNIRS recording lasting over 20 minutes (task duration). A sliding-window approach

[4] For experiments, we used the tensorly implementation in http://tensorly.org/.

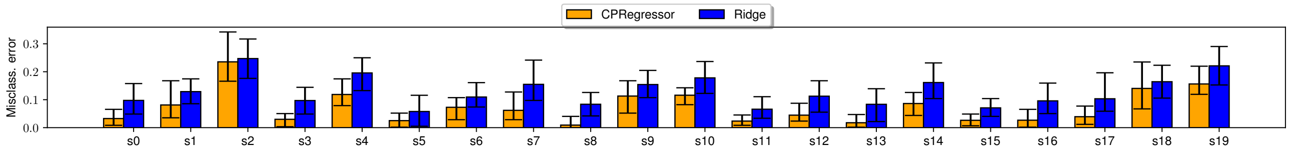

Figure 8. Misclassification error of CP regressor versus ridge regression across different subjects. Results are averaged over 50 uniformly random splitting of the data for each subject with 20% of data used for models training.

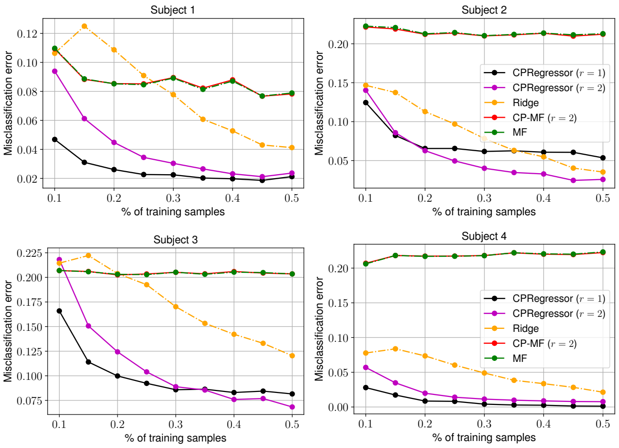

Figure 9. Misclassification error of CP regressor versus ridge regressor across different subjects in terms of % of training samples. The curves are averaged over 50 uniformly random splitting of the data across the different subjects.

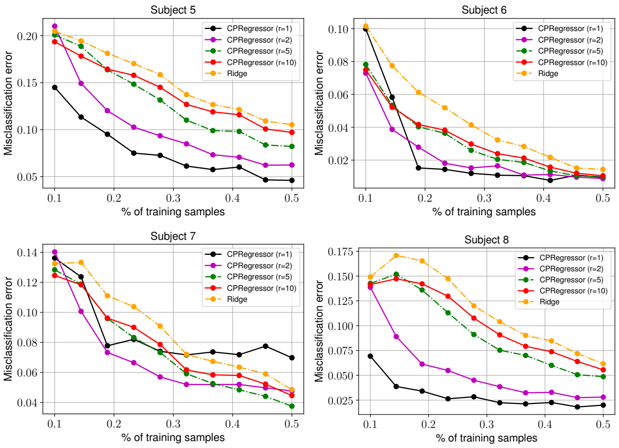

Figure 10. Misclassification error of CP regressor for different ranks decomposition versus ridge regressor across different subjects averaged over 50 random train test splitting.

is used to extract 30 seconds overlapping windows with a stride of 0.6 seconds labeled as n-back depending on the task assigned to the subject. As such each data sample is constructed as a matrix of dimension 150×8 and each subject corresponds to around 700 data samples.

6.2. Comments on the experiments

Figure 9 depicts the performance of the CP regressor (with ranks 1 and 2), ridge regressor, CP-based matched filter and the matched filter classifier, varying the proportion of training samples for each subject samples of the fNIRS dataset. We note that the performance of the matched filter (with and without CP) are sub-optimal since the FNIRS data features exhibit more complex correlations compared to the simple theoretical model in Eq. (1). However, although our theoretical study does not apply for the CP regressor, our conclusions seem to extend to real data with the CP regressor. Indeed, *we highlight the superiority of the CP regressor compared to the ridge regressor when the proportion of training samples is low*. Moreover, Figure 8 shows a constant gain in performance of the CP regressor over the ridge regressor on more subjects, where 20% of each subject data is considered for training.

We further investigate the effect of varying the rank parameter of the CP regressor. Indeed, as suggested by Figure 10, the higher the rank of the decomposition, the more the performance degrades towards the performance of the ridge regressor. In essence this observation is in line with the theory we developed. Indeed, using the first informative low rank structure of the data brings more discrimination to the data while additional components induces more noisy information which may harm the classification up to the performance of the classical ridge regressor.

7. Concluding remarks

This paper has brought a first theoretical analysis on learning from tensor data that have a hidden low-rank structure. Both analytical and empirical assessments suggest that a considerable performance gain can be achieved by exploiting such low-rank tensor structure when few training samples are available. In addition, the paper explicitly demonstrates the application of *random tensor theory* to evaluate the performance of simple learning methods (such as the CP-based matched filter classifier), whose behavior was not so far theoretically understood. This paves the way for more systematic theoretical analysis and improvement of sophisticated machine learning algorithms when dealing with tensor-structured data. For instance, the analysis of the CP regressor could be performed using random tensor theory and is left for future investigation.

References

Ali, H. T. and Couillet, R. Improved spectral community detection in large heterogeneous networks. *The Journal of Machine Learning Research*, 18(1):8344–8392, 2017.

Anandkumar, A., Ge, R., Hsu, D., Kakade, S. M., and Telgarsky, M. Tensor decompositions for learning latent variable models. *Journal of machine learning research*, 15:2773–2832, 2014.

Baik, J., Arous, G. B., and Péché, S. Phase transition of the largest eigenvalue for nonnull complex sample covariance matrices. *The Annals of Probability*, 33(5):1643–1697, 2005.

Ben Arous, G., Huang, D. Z., and Huang, J. Long random matrices and tensor unfolding. *arXiv preprint arXiv:2110.10210*, 2021.

Benaych-Georges, F. and Nadakuditi, R. R. The eigenvalues and eigenvectors of finite, low rank perturbations of large random matrices. *Advances in Mathematics*, 227(1):494–521, 2011.

Billingsley, P. *Probability and measure*. John Wiley & Sons, 2008.

Biroli, G., Cammarota, C., and Ricci-Tersenghi, F. How to iron out rough landscapes and get optimal performances: averaged gradient descent and its application to tensor pca. *Journal of Physics A: Mathematical and Theoretical*, 53(17):174003, 2020.

Capitaine, M., Donati-Martin, C., and Féral, D. The largest eigenvalues of finite rank deformation of large wigner matrices: convergence and nonuniversality of the fluctuations. *The Annals of Probability*, 37(1):1–47, 2009.

Chen, W., Zhu, X., Sun, R., He, J., Li, R., Shen, X., and Yu, B. Tensor low-rank reconstruction for semantic segmentation. In *European Conference on Computer Vision*, pp. 52–69. Springer, 2020.

Couillet, R. and Benaych-Georges, F. Kernel spectral clustering of large dimensional data. *Electronic journal of statistics*, 10(1):1393–1454, 2016.

Goulart, J. H., Couillet, R., and Comon, P. A random matrix perspective on random tensors. *stat*, 1050:2, 2021.

Handschy, M. C. *Phase Transition in Random Tensors with Multiple Spikes*. PhD thesis, University of Minnesota, 2019.

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

Huang, J., Huang, D. Z., Yang, Q., and Cheng, G. Power iteration for tensor pca. *arXiv preprint arXiv:2012.13669*, 2020.

Huang, Z., Wang, L., Blaney, G., Slaughter, C., McKeon, D., Zhou, Z., Jacob, R. J. K., and Hughes, M. C. The tufts fnirs mental workload dataset and benchmark for brain-computer interfaces that generalize. In *Proceedings of the Neural Information Processing Systems (NeurIPS) Track on Datasets and Benchmarks*, 2021. URL https://openreview.net/pdf?id=QzNHE7QHhut.

Jagannath, A., Lopatto, P., and Miolane, L. Statistical thresholds for tensor PCA. *The Annals of Applied Probability*, 30(4):1910–1933, 2020.

Kadmon, J. and Ganguli, S. Statistical mechanics of low-rank tensor decomposition. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124016, 2019.

Kolda, T. G. and Bader, B. W. Tensor decompositions and applications. *SIAM Review*, 51:455–500, 2019.

Kossaifi, J., Lipton, Z. C., Kolbeinsson, A., Khanna, A., Furlanello, T., and Anandkumar, A. Tensor regression networks. *Journal of Machine Learning Research*, 21: 1–21, 2020.

Landsberg, J. M. Tensors: geometry and applications. *Representation theory*, 381(402):3, 2012.

Lesieur, T., Miolane, L., Lelarge, M., Krzakala, F., and Zdeborová, L. Statistical and computational phase transitions in spiked tensor estimation. In *Proc. IEEE International Symposium on Information Theory (ISIT)*, pp. 511–515, 2017.

Liang, P. P., Liu, Z., Tsai, Y.-H. H., Zhao, Q., Salakhutdinov, R., and Morency, L.-P. Learning representations from imperfect time series data via tensor rank regularization. *arXiv preprint arXiv:1907.01011*, 2019.

Louart, C., Liao, Z., and Couillet, R. A random matrix approach to neural networks. *The Annals of Applied Probability*, 28(2):1190–1248, 2018.

Mai, X. and Couillet, R. A random matrix analysis and improvement of semi-supervised learning for large dimensional data. *The Journal of Machine Learning Research*, 19(1):3074–3100, 2018.

Marčenko, V. A. and Pastur, L. A. Distribution of eigenvalues for some sets of random matrices. *Mathematics of the USSR-Sbornik*, 1(4):457, 1967.

Montanari, A. and Richard, E. A statistical model for tensor PCA. *arXiv preprint arXiv:1411.1076*, 2014.

Ng, A. Y., Jordan, M. I., and Weiss, Y. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pp. 849–856, 2002.

Péché, S. The largest eigenvalue of small rank perturbations of hermitian random matrices. *Probability Theory and Related Fields*, 134(1):127–173, 2006.

Pennington, J. and Worah, P. Nonlinear random matrix theory for deep learning. 2017.

Perry, A., Wein, A. S., and Bandeira, A. S. Statistical limits of spiked tensor models. In *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, volume 56, pp. 230–264. Institut Henri Poincaré, 2020.

Seddik, M. E. A., Guillaud, M., and Couillet, R. When random tensors meet random matrices. *arXiv preprint arXiv:2112.12348*, 2021a.

Seddik, M. E. A., Louart, C., Couillet, R., and Tamaazousti, M. The unexpected deterministic and universal behavior of large softmax classifiers. In *International Conference on Artificial Intelligence and Statistics*, pp. 1045–1053. PMLR, 2021b.

Sun, W. and Li, L. Dynamic tensor clustering. *Journal of the American Statistical Association*, 114:1894–1907, 2019.

Sun, W. W., Hao, B., and Li, L. Tensors in modern statistical learning. *Wiley StatsRef: Statistics Reference Online*, pp. 1–25, 2014.

Tiomoko, M., Couillet, R., and Tiomoko, H. Large dimensional analysis and improvement of multi task learning. *arXiv preprint arXiv:2009.01591*, 2020.

Tiomoko, M., Couillet, R., and Pascal, F. Pca-based multi task learning: a random matrix approach. *arXiv preprint arXiv:2111.00924*, 2021.

Zhou, H., Li, L., and Zhu, H. Tensor regression with applications in neuroimaging data analysis. *Journal of the American Statistical Association*, 108, 2013.

A. Proofs

A.1. Poof of Proposition 5.2

Recall $\boldsymbol{w} = \operatorname{vec}(\mathbf{W})$, $\boldsymbol{X} = \operatorname{Mat}(\mathbf{X})$, $p = \sum_{j=1}^k p_j$ and $d = \prod_{j=1}^k p_j$, hence $\boldsymbol{w} = \frac{1}{\sqrt{np}}\boldsymbol{X}\boldsymbol{y}$. Denoting $\tilde{\boldsymbol{x}}_i = \operatorname{Mat}(\tilde{\mathbf{X}}_i)$ for some $\tilde{\mathbf{X}}_i \in \mathcal{C}_a$ with $a \in \{1, 2\}$ independent of the training data $\mathbf{X}$, the decision function write as $g(\tilde{\boldsymbol{x}}_i) = \boldsymbol{w}^\top \tilde{\boldsymbol{x}}_i = \sum_{j=1}^d w_j \tilde{x}_{ij}$. Thus, by Lyapunov's central limit theorem (Billingsley, 2008), the decision function has a Gaussian distribution for large n, we therefore need to compute its expectation and variance.

Computation of $\mathbb{E}[g(\tilde{\boldsymbol{x}}_i)]$: Let $\boldsymbol{\mu} = \operatorname{vec}(\mathbf{M})$, then $\tilde{\boldsymbol{x}}_i = (-1)^a \boldsymbol{\mu} + \boldsymbol{z}_i$ with $\boldsymbol{z}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$ and

$$\mathbb{E}[g(\tilde{\boldsymbol{x}}_i)] = \frac{1}{\sqrt{np}} \mathbb{E}\left[\boldsymbol{y}^\top \boldsymbol{X}^\top \tilde{\boldsymbol{x}}_i\right] = \frac{1}{\sqrt{np}} \boldsymbol{y}^\top \boldsymbol{y} \boldsymbol{\mu}^\top (-1)^a \boldsymbol{\mu} = (-1)^a \sqrt{\frac{n}{p}} \|\boldsymbol{\mu}\|^2 = (-1)^a \sqrt{\frac{n}{p}} \|\mathbf{M}\|^2.$$

Computation of $\mathbb{E}[g(\boldsymbol{x}_i)^2]$:

$$\mathbb{E}\left[g(\boldsymbol{x}_i)^2\right] = \mathbb{E}\left[\frac{1}{np} \boldsymbol{y}^\top \boldsymbol{X}^\top \tilde{\boldsymbol{x}}_i \tilde{\boldsymbol{x}}_i^\top \boldsymbol{X} \boldsymbol{y}\right] = \mathbb{E}\left[\frac{1}{np} \boldsymbol{y}^\top \boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{y}\right] + \mathbb{E}\left[\frac{1}{np} \boldsymbol{y}^\top \boldsymbol{X}^\top \boldsymbol{\mu}\boldsymbol{\mu}^\top \boldsymbol{X} \boldsymbol{y}\right] = E_1 + E_2.$$

Since $\boldsymbol{X} = \boldsymbol{\mu}\boldsymbol{y}^\top + \boldsymbol{Z}$ with $\boldsymbol{Z} = [\boldsymbol{z}_1, \dots, \boldsymbol{z}_n] = \operatorname{Mat}(\mathbf{Z}) \in \mathbb{R}^{d \times n}$, we have

$$\begin{aligned} E_1 &= \frac{1}{np} \|\boldsymbol{\mu}\|^2 \|\boldsymbol{y}\|^4 + \frac{1}{np} \mathbb{E}\left[\boldsymbol{y}^\top \boldsymbol{Z}^\top \boldsymbol{Z} \boldsymbol{y}\right] = \frac{n}{p} \|\mathbf{M}\|^2 + \frac{d}{p}, \\ E_2 &= \frac{1}{np} \boldsymbol{y}^\top \boldsymbol{y} \boldsymbol{\mu}^\top \boldsymbol{\mu}\boldsymbol{\mu}^\top \boldsymbol{\mu} \boldsymbol{y}^\top \boldsymbol{y} + \frac{1}{np} \mathbb{E}\left[\boldsymbol{y}^\top \boldsymbol{Z}^\top \boldsymbol{\mu}\boldsymbol{\mu}^\top \boldsymbol{Z} \boldsymbol{y}\right] = \frac{1}{np} \|\boldsymbol{y}\|^4 \|\boldsymbol{\mu}\|^4 + \frac{1}{np} \operatorname{tr}\left(\mathbb{E}\left[\boldsymbol{Z}\boldsymbol{y}\boldsymbol{y}^\top \boldsymbol{Z}^\top\right] \boldsymbol{\mu}\boldsymbol{\mu}^\top\right), \end{aligned}$$

where $\mathbb{E}\left[\boldsymbol{Z}\boldsymbol{y}\boldsymbol{y}^\top \boldsymbol{Z}^\top\right] = \mathbb{E}\left[\left(\sum_{i=1}^n y_i \boldsymbol{z}_i\right)\left(\sum_{i=1}^n y_i \boldsymbol{z}_i^\top\right)\right] = \sum_{i=1}^n y_i^2 \,\mathbb{E}\left[\boldsymbol{z}_i \boldsymbol{z}_i^\top\right] = n\boldsymbol{I}_d$. Therefore,

$$\mathbb{E}\left[g(\tilde{\boldsymbol{x}}_i)^2\right] = \frac{n}{p} \|\mathbf{M}\|^2 + \frac{d}{p} + \frac{n}{p} \|\mathbf{M}\|^4 + \frac{1}{p} \|\mathbf{M}\|^2,$$

and the term $\frac{1}{p}\|\mathbf{M}\|^2$ vanishes for large values of p under Assumption 5.1. In particular, the variance of $g(\boldsymbol{x}_i)$ is given by $\mathbb{E}\left[g(\tilde{\boldsymbol{x}}_i)^2\right] - \mathbb{E}\left[g(\tilde{\boldsymbol{x}}_i)\right]^2 = \frac{n}{p}\|\mathbf{M}\|^2 + \frac{d}{p}$ for large values of p.

A.2. Poof of Proposition 5.4

Denote $\mathbf{M} = \gamma \bigotimes_{j=1}^k \boldsymbol{u}_j$ where $\boldsymbol{u}_j = \frac{\boldsymbol{\mu}_j}{\|\boldsymbol{\mu}_j\|}$, as such $\|\mathbf{M}\| = \gamma$. Therefore, from Eq. (6) and further denoting $\beta = \|\mathbf{M}\|\sqrt{\frac{n}{p}}$, $\mathbf{W}$ expresses as

$$\mathbf{W} = \beta \bigotimes_{j=1}^k \boldsymbol{u}_j + \frac{1}{\sqrt{p}} \tilde{\mathbf{Z}}.$$

The best rank-one approximation $\lambda \bigotimes_{j=1}^k \boldsymbol{v}_j$ (with the $\boldsymbol{v}_j$'s being unitary vectors) of $\mathbf{W}$ is given by the MLE as

$$\operatorname*{arg\,min}_{\lambda > 0, \{\boldsymbol{v}_j \mid \|\boldsymbol{v}_j\| = 1,\, j \in [k]\}} \left\| \mathbf{W} - \lambda \bigotimes_{j=1}^k \boldsymbol{v}_j \right\|_{\mathrm{F}}^2.$$

As in Appendix A.1, for a new test datum $\tilde{\mathbf{X}}_i = (-1)^a \mathbf{M} + \tilde{\mathbf{Z}}_i$, the decision function $g_{CP}(\tilde{\mathbf{X}}_i)$ is a Gaussian random variable, the mean of which expresses as follows.

$$\mathbb{E}\left[g_{CP}(\tilde{\mathbf{X}}_i)\right] = \mathbb{E}\left[\left\langle \lambda \bigotimes_{j=1}^k \boldsymbol{v}_j, \tilde{\mathbf{X}}_i \right\rangle\right] = \mathbb{E}\left[(-1)^a \|\mathbf{M}\| \lambda \prod_{j=1}^k \langle \boldsymbol{u}_j, \boldsymbol{v}_j \rangle\right] \to (-1)^a \|\mathbf{M}\| \lambda^\infty(\beta) \prod_{j=1}^k q_j(\lambda^\infty(\beta)),$$

by Theorem 4.1. Moreover, the variance of $g_{CP}(\tilde{\mathbf{X}}_i)$ expresses as

$$\begin{aligned}\mathbb{V}\mathrm{ar}\left[g_{CP}(\tilde{\mathbf{X}}_i)\right] &= \mathbb{E}\left[\left\langle \lambda \bigotimes_{j=1}^k \boldsymbol{v}_j, \tilde{\mathbf{Z}}_i \right\rangle^2\right] = \mathbb{E}\left[\lambda^2 \left(\sum_{i_1,\dots,i_k} \prod_{j=1}^k (\boldsymbol{v}_j)_{i_j} (\tilde{\mathbf{Z}}_i)_{i_1,\dots,i_k}\right)^2\right] \\ &= \mathbb{E}\left[\lambda^2 \sum_{i_1,\dots,i_k,i'_1,\dots,i'_k} \prod_{j=1}^k (\boldsymbol{v}_j)_{i_j} (\tilde{\mathbf{Z}}_i)_{i_1,\dots,i_k} \prod_{j=1}^k (\boldsymbol{v}_j)_{i'_j} (\tilde{\mathbf{Z}}_i)_{i'_1,\dots,i'_k}\right] \\ &= \mathbb{E}\left[\lambda^2 \sum_{i_1,\dots,i_k} \prod_{j=1}^k (\boldsymbol{v}_j)^2_{i_j} (\tilde{\mathbf{Z}}_i)^2_{i_1,\dots,i_k}\right] = \mathbb{E}\left[\lambda^2 \sum_{i_1,\dots,i_k} \prod_{j=1}^k (\boldsymbol{v}_j)^2_{i_j}\, \mathbb{E}[(\tilde{\mathbf{Z}}_i)^2_{i_1,\dots,i_k} \mid \mathbf{Z}]\right] = \mathbb{E}[\lambda^2] \to \lambda^\infty(\beta)^2,\end{aligned}$$

since $\mathbb{E}[(\tilde{\mathbf{Z}}_i)^2_{i_1,\dots,i_k} \mid \mathbf{Z}] = 1$ and $\sum_{i_1,\dots,i_k} \prod_{j=1}^k (\boldsymbol{v}_j)^2_{i_j} = \prod_{j=1}^k \|\boldsymbol{v}_j\|^2 = 1$.

A.3. Poof of Proposition 5.5

The equivalent random matrix model writes as

$$\tilde{\boldsymbol{X}} = \sqrt{\frac{n}{d+n}} \operatorname{vec}(\mathbf{M}) \bar{\boldsymbol{y}}^\top + \frac{1}{\sqrt{d+n}} \operatorname{Mat}(\mathbf{Z}) \in \mathbb{R}^{d \times n},$$

where $\bar{\boldsymbol{y}} = \boldsymbol{y}/\sqrt{n}$ and the normalization by $\sqrt{d+n}$ is considered for convenience. Let $\hat{\boldsymbol{y}}$ be the right singular vector of $\tilde{\boldsymbol{X}}$ corresponding to its largest singular value. Then evoking Corollary 4.3, the asymptotic alignment under Assumption 5.1 is given as

$$|\langle \hat{\boldsymbol{y}}, \bar{\boldsymbol{y}} \rangle| \xrightarrow{\text{a.s.}} \alpha = \kappa \left(\|\mathbf{M}\| \sqrt{\frac{n}{d+n}}, \frac{n}{d+n} \right)^{-1}.$$

Moreover, $\hat{\boldsymbol{y}}$ decomposes as

$$\hat{\boldsymbol{y}} = \alpha \bar{\boldsymbol{y}} + \sigma \boldsymbol{w},$$

where $\boldsymbol{w} \in \mathbb{R}^n$ is a random vector, orthogonal to $\bar{\boldsymbol{y}}$ and of unit norm. Since $\hat{\boldsymbol{y}}$ is of unit norm, σ satisfies $1 = \alpha^2 + \sigma^2$, as such $\sigma = \sqrt{1-\alpha^2}$. Finally, the Gaussianity of the entries of $\hat{\boldsymbol{y}}$ is obtained thanks to similar arguments as in (Couillet & Benaych-Georges, 2016).

A.4. Poof of Proposition 5.6

The equivalent random tensor model writes as

$$\tilde{\mathbf{X}} = \sqrt{\frac{n}{p+n}} \mathbf{M} \otimes \bar{\boldsymbol{y}} + \frac{1}{\sqrt{p+n}} \mathbf{Z} \in \mathbb{R}^{p_1 \times \cdots \times p_k \times n},$$

where $\bar{\boldsymbol{y}} = \boldsymbol{y}/\sqrt{n}$. As such $\tilde{\mathbf{X}}$ is a spiked random tensor of order $k+1$. As in Appendix A.3, we need to express the asymptotic alignment between $\hat{\boldsymbol{y}}$ and $\bar{\boldsymbol{y}}$ with $\hat{\boldsymbol{y}}$ being the $(k+1)$-th mode component of the rank-one tensor approximation of $\tilde{\mathbf{X}}$, which is straightforwardly obtained thanks to Theorem 4.1, applied to a $(k+1)$-th order tensor of dimensions $p_1 \times \cdots \times p_k \times n$, yielding

$$|\langle \hat{\boldsymbol{y}}, \bar{\boldsymbol{y}} \rangle| \xrightarrow{\text{a.s.}} \alpha = q_{k+1}\left(\lambda^\infty \left(\|\mathbf{M}\| \sqrt{\frac{n}{p+n}} \right) \right),$$

where $q_{k+1}(\cdot)$ and $\lambda^\infty(\cdot)$ are defined in Theorem 4.1.