On Learning from Low Rank Tensor Data: A Random Tensor Theory Perspective

Anonymous Author
Anonymous Institution

Abstract

Building on recent advances in random tensor theory, this paper provides 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 with and without knowledge of the low rank structure of the data. Our analysis quantifies the performance gain when taking into account the low rank tensor structure of the data compared to treating them as simple vectors. We further show that this finding extends to unsupervised learning, demonstrating the importance of taking into account the low rank structure of the tensor data.

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-mode 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 [11]. 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 makes them increasingly important [21], leading to a growing interest in tensor data analysis in the statistical learning community.

Preliminary work. Under review by AISTATS 2023. Do not distribute.

A large part of previous works on tensor theory applied to machine learning problems assume a low-rank representation of input data [2, 8] and estimate this representation using as main ingredient the CANDECOMP/PARAFAC decomposition (CPD) [7]. Indeed, the low-rank tensor structure is a sparsity hypothesis that is natural in the modelling of real data seen through high-dimensional inputs [9]. 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 and such gain is theoretically quantified on a simple statistical data model*.

In the literature, the low-rank tensor structure has been exploited for example in tensor regression in a supervised setting [24] or clustering in an unsupervised setting [20]. 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 [12, 5], 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 [10].

On top of the performance gain shown by [10], 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

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 §2) and based on recent advances in random tensor theory [18], the main contributions brought by this paper are two-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 (§3.1).

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 (§3.2).

To the best of our knowledge, few works in the literature were focused on the exact estimation of the performance of ML methods when processing tensor data with low-rank structure. This paper suggests new directions to fill-in this gap leveraging on recent advances in random tensor theory (RTT). Outstandingly, we demonstrate that RTT 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 $\{1,\dots,n\}$. 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 $\mathbf{A}, \mathbf{B}, \mathbf{C}$. $T_{i_1,\dots,i_d}$ denotes the entry $(i_1,\dots,i_d)$ of the tensor $\mathbf{T}$. The inner product between two order-d tensors $\mathbf{A}$ and $\mathbf{B}$ is denoted $\langle \mathbf{A}, \mathbf{B}\rangle = \sum_{i_1,\dots,i_d} A_{i_1\dots i_d} B_{i_1\dots i_d}$. The ℓ_2-norm of $\mathbf{A}$ is denoted $\|\mathbf{A}\| = \sqrt{\langle \mathcal{A}, \mathcal{A}\rangle}$. For any vectors $\boldsymbol{u}_1,\dots,\boldsymbol{u}_d$, contractions of a tensor $\mathbf{A}$ are denoted by $\mathbf{A}(\boldsymbol{u}_1,\dots,\boldsymbol{u}_d) = \sum A_{i_1\dots i_d} u_{1i_1}\dots u_{di_d}$. The notation $\bigotimes_{i=1}^k \boldsymbol{v}_i$ stands for the tensor outer product between the vectors $\boldsymbol{v}_1,\dots,\boldsymbol{v}_k$ with $[\bigotimes_{i=1}^k \boldsymbol{v}_i]_{i_1\dots i_k} = \prod_{j=1}^k v_{ji_j}$. $Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^\infty e^{-\frac{t^2}{2}} dt$ 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. $\mathbb{S}^{d-1}$ stands for the unit sphere in dimension d.

2 Statistical data model

Let the training samples be n independent tensor-structured data $\mathbf{X}_1,\dots,\mathbf{X}_n$ 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 $\mathbf{X} = [\mathbf{X}_1,\dots,\mathbf{X}_n] \in \mathbb{R}^{p_1\times\cdots\times p_k\times n}$ 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 $\mathbf{Z} = [\mathbf{Z}_1,\dots,\mathbf{Z}_n] \in \mathbb{R}^{p_1\times\cdots\times p_k\times n}$. 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 a *rank-one spiked random tensor model* of order $k+1$, where the signal part is $\mathbf{M}\otimes\boldsymbol{y}$ and $\mathbf{Z}$ corresponds to the noise part.

Remark 2.1 (On the data model)**.** *Note that our present work can be easily extended to a more general (rank-r) data model of the form $\sum_{i=1}^r \bigotimes_{j=1}^d \boldsymbol{\mu}_j^{(i)} + \mathbf{Z}$ with the $\boldsymbol{\mu}_j^{(i)}$'s being orthogonal and r of order $O(1)$. However, for $\boldsymbol{\mu}_j^{(i)}$'s being arbitrary chosen, the extension of the random tensor theory in [18] is is still an interesting theoretical challenge for future work. See the end of Section 3.1 and supplementary material for the extension of our results to the orthogonal case.*

Throughout the following sections, 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 2.2 (Growth rate)**.** *For all $j \in [k]$, $\frac{p_j}{n} = \mathcal{O}_n(1)$ and $\|\boldsymbol{\mu}_j\| = \mathcal{O}_n(1)$*[1].

A classical assumption in learning theory and random matrix theory [16, 13, 1, 14, 23, 19] considers 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

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

large number of training samples n which might be difficult to achieve in practical settings. As such Assumption 2.2 is more realistic from the practical view point when dealing with tensor structured data.

3 Main results

3.1 On supervised learning

Given the training data tensor $\mathbf{X}$ in equation 2 and the corresponding labels vector $\boldsymbol{y}$, a basic learning approach [22] consists in reshaping $\mathbf{X}$ into a data matrix $\mathrm{Mat}_{k+1}(\mathbf{X}) \in \mathbb{R}^{n \times d}$ with $d = \prod_{j=1}^{k} p_j$, and then building a matched filter classifier[2] with parameters $\boldsymbol{w} \equiv \mathrm{vec}(\mathbf{W}) \in \mathbb{R}^d$ ($\mathbf{W} \in \mathbb{R}^{p_1 \times \cdots \times p_k}$) as[3]

$$\boldsymbol{w} = \frac{1}{\sqrt{np}} \mathrm{Mat}_{k+1}(\mathbf{X})^\top \boldsymbol{y}, \tag{3}$$

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 $g(\tilde{\mathbf{X}}_i) = \langle \boldsymbol{w}, \mathrm{vec}(\tilde{\mathbf{X}}_i) \rangle$ which is equivalent in tensor notations to

$$g(\tilde{\mathbf{X}}_i) = \langle \mathbf{W}, \tilde{\mathbf{X}}_i \rangle \underset{\mathcal{C}_2}{\overset{\mathcal{C}_1}{\lessgtr}} 0, \quad \mathbf{W} = \frac{1}{\sqrt{np}} \mathbf{X} \times_{k+1} \boldsymbol{y}. \tag{4}$$

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 a first result characterizing the theoretical performance of the matched filter for the data model in equation 2, which relies on classical high-dimensional statistics.

Proposition 3.1 (Performance of the matched filter classifier)**.** *Under Assumption 2.2, 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 supplementary material. □

Proposition 3.1 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 equation 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 (left) provides a histogram of the decision function of the matched filter classifier and its theoretical estimate through Proposition 3.1. Under Assumption 2.2, 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 exploited by performing a tensor decomposition of the weights tensor $\mathbf{W}$, since it is a noisy version of $\mathbf{M}$. Precisely, recall the definition of $\mathbf{W}$ in equation 4 and $\mathbf{X}$ in equation 2, thus

$$\mathbf{W} = \sqrt{\frac{n}{p}} \bigotimes_{j=1}^{k} \boldsymbol{\mu}_j + \frac{1}{\sqrt{p}} \tilde{\mathbf{Z}}, \tag{5}$$

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

As such, $\mathbf{W}$ has the form of a *spiked random tensor model* which has been studied in [18]. In order to extract the hidden low-rank structure of $\mathbf{W}$, one would consider the best rank-one CP approximation of $\mathbf{W}$ which yields estimates of the signal components $\boldsymbol{\mu}_j$'s (if the signal strength $\|\mathbf{M}\|$ is large enough) and then replace the weights $\mathbf{W}$ in the decision function by such rank-one approximation. Precisely, the best rank-one approximation of $\mathbf{W}$ can be obtained by solving the following objective

$$(\lambda^*, \{\boldsymbol{u}_i^*\}_{i=1}^k) = \underset{\lambda \in \mathbb{R}^+, \boldsymbol{u}_i \in \mathbb{S}^{p_i - 1}}{\arg\min} \|\mathbf{W} - \lambda \bigotimes_{i=1}^{k} \boldsymbol{u}_i\|_{\mathrm{F}}^2, \tag{6}$$

which corresponds to the maximum likelihood estimator (MLE). Computing the above MLE is NP-hard in the worst case [6]. However, it is possible to compute good estimates of the rank-one components of $\mathbf{W}$ in polynomial time, using tensor SVD[4] [4, 18] or tensor power iteration (Algorithm 1) initialized with tensor SVD [3] which yields more accurate estimation of the underlying components, provided that the signal strength $\|\mathbf{M}\|$ is larger than $\mathcal{O}(p^{\frac{k-2}{4}})$ as demonstrated in [3].

[2] Note that the matched filter classifier corresponds to a classical ridge regression classifier when the regularization parameter is set to ∞.

[3] The normalization by $\sqrt{np}$ is considered for convenience and does not affect the performances of the considered methods. Moreover, under Assumption 2.2 the quantities n and p are of the same order which is equivalent to the standard normalization by n.

[4] SVD applied to the unfolded tensor.

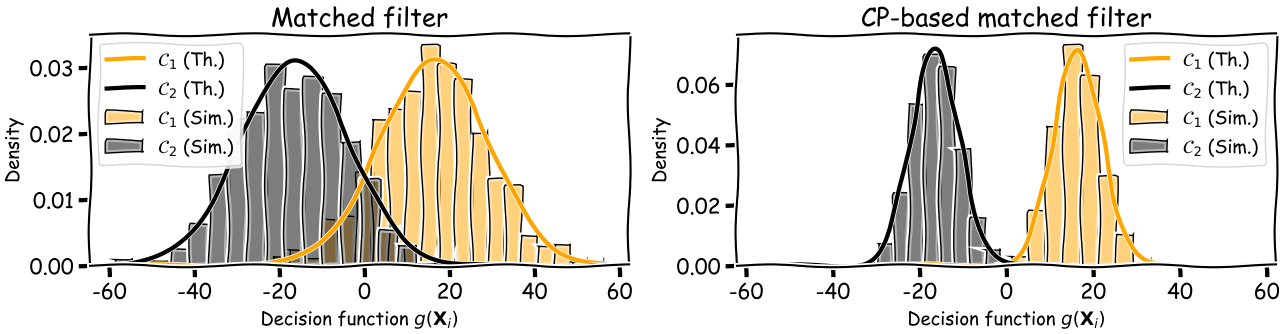

Figure 1: Theoretical versus empirical histogram of the decision function $g(\tilde{\mathbf{X}}_i)$ for the matched filter classifier as per Proposition 3.1 (left) and for the CP-based matched filter as per Proposition 3.3 (right). We considered $n = 200$ training data ($n_1 = n_2 = 100$) that are tensors of order 3 and of dimensions $p_1 = p_2 = p_3 = 20$, distributed as the rank-one tensor model in equation 1 with the $\boldsymbol{\mu}_j$'s being randomly sampled vectors from a sphere such that $\|\mathbf{M}\| = 3$.

Algorithm 1 Tensor Power Iteration [2]

Require: An order k tensor $\mathbf{W} \in \mathbb{R}^{p_1 \times \cdots \times p_k}$ and initialization components $\boldsymbol{u}_1^0, \cdots, \boldsymbol{u}_k^0$.
Output: Rank-one approximation of $\mathbf{W}$.
  $(\boldsymbol{u}_1, \cdots, \boldsymbol{u}_k) \leftarrow (\boldsymbol{u}_1^0, \cdots, \boldsymbol{u}_k^0)$
  while Not convergence **do**
    for $i \in [k]$ **do**
      $\boldsymbol{u}_i \leftarrow \frac{\mathbf{W}(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{i-1}, :, \boldsymbol{u}_{i+1}, \ldots, \boldsymbol{u}_k)}{\|\mathbf{W}(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{i-1}, :, \boldsymbol{u}_{i+1}, \ldots, \boldsymbol{u}_k)\|}$
    end for
  end while

In essence, extracting the rank-one component of $\mathbf{W}$ constitutes in a denoising scheme and allows to considerably reduce the variance of the decision function, thereby providing a better classification accuracy. Precisely, given the above MLE which we denote $\lambda^* \bigotimes_{i=1}^k \boldsymbol{u}_i^*$, the CP-based matched filter classifier is defined for a new datum $\tilde{\mathbf{X}}_i \in \mathcal{C}_a$ as

$$g_{CP}(\tilde{\mathbf{X}}_i) = \left\langle \lambda^* \bigotimes_{i=1}^k \boldsymbol{u}_i^*, \tilde{\mathbf{X}}_i \right\rangle \underset{\mathcal{C}_2}{\overset{\mathcal{C}_1}{\lessgtr}} 0. \tag{7}$$

We further introduce the quantities in equation 8 from [18] which describe the behavior of a k-order spiked random tensor model and shall be used subsequently. Therefore, our following result characterizes the theoretical performance of the CP-based matched filter classifier.

Proposition 3.3 (Performance of the CP-based matched filter classifier)**.** *Under Assumption 2.2, 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 $m_a = (-1)^a \sigma \|\mathbf{M}\| \prod_{j=1}^k q_j\left(\sigma, \|\mathbf{M}\|\sqrt{\frac{n}{p}}\right)$ and σ satisfies $f\left(\sigma, \|\mathbf{M}\|\sqrt{\frac{n}{p}}\right) = 0$ where q_j and f are defined in equation 8. Furthermore, the misclassification error verifies with probability one $\mathbb{P}\left((-1)^a g_{CP}(\tilde{\mathbf{X}}_i) < 0 \mid \tilde{\mathbf{X}}_i \in \mathcal{C}_a\right) - Q\left(\frac{|m_a|}{\sigma}\right) \to 0$.

Sketch of proof. The proof relies on estimating the expectation and the variance of the decision function $g_{\text{CP}}(\tilde{\mathbf{X}}_i)$ for some $\tilde{\mathbf{X}}_i \in \mathcal{C}_a$ with $a \in \{1, 2\}$ independent from the training set $\mathbf{X}$. Indeed, one finds that $\mathbb{E}\, g_{\text{CP}}(\tilde{\mathbf{X}}_i) = \mathbb{E}\left[(-1)^a \|\mathbf{M}\| \lambda^* \prod_{j=1}^k \langle \frac{\boldsymbol{\mu}_j}{\|\boldsymbol{\mu}_j\|}, \boldsymbol{u}_j^* \rangle\right]$ where the quantities λ^* and $\langle \frac{\boldsymbol{\mu}_j}{\|\boldsymbol{\mu}_j\|}, \boldsymbol{u}_j^* \rangle$ are estimated using equation 8 where $\lambda^* \to \sigma$ with σ satisfying $f(\sigma, \|\mathbf{M}\|\sqrt{\frac{n}{p}}) = 0$ and $\langle \frac{\boldsymbol{\mu}_j}{\|\boldsymbol{\mu}_j\|}, \boldsymbol{u}_j^* \rangle \to q_j(\sigma, \|\mathbf{M}\|\sqrt{\frac{n}{p}})$. The variance of $g_{\text{CP}}(\tilde{\mathbf{X}}_i)$ is computed similarly and we find $\text{Var}[g_{\text{CP}}(\tilde{\mathbf{X}}_i)] = \sigma^2$. See supplementary material for a detailled proof. □

Remark 3.4 (On the assumptions)**.** *Proposition 3.3 requires additional technical assumptions (e.g., Assumption 3 from [18]). We highlight that this assumption is rather technical and needs the introduction of various notions (e.g., defining the block-wise contracted matrix introduced by [18]). However, note that this assumption is always satisfied by the maximum likelihood estimator when the SNR is large enough. In our notation the SNR corresponds to the quantity $\|\mathbf{M}\|$ which controls the difficulty of the classification problem; when $\|\mathbf{M}\| = 0$ the classification is impossible whereas when $\|\mathbf{M}\|$ is large the classification becomes trivial.*

Proposition 3.3 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 classifier in Proposition 3.1. *We particularly highlight that the variance σ for the CP-based classifier remains constant under Assumption 2.2 as depicted in Figure 1, thereby yielding a better classification accuracy for small values of the number of training samples n.*

$$f(z,\beta) = z + g(z) - \beta \prod_{i=1}^{k} q_i(z,\beta), \quad q_i(z,\beta) = \sqrt{1 - \frac{g_i^2(z)}{c_i}}, \quad c_i = \lim_{n\to\infty} \frac{p_i}{\sum_{j=1}^{k} p_j}$$
$$g_i(z) = \frac{g(z)+z}{2} - \frac{\sqrt{4c_i + (g(z)+z)^2}}{2}, \quad g(z) = \sum_{i=1}^{k} g_i(z) \tag{8}$$

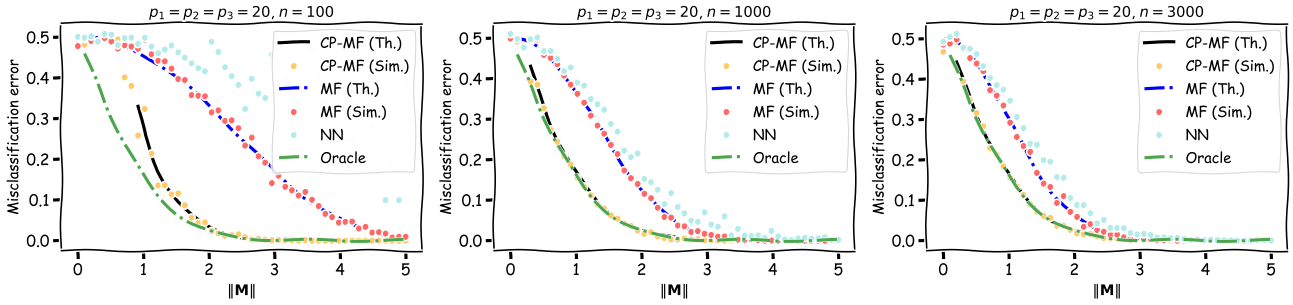

Figure 2: Theoretical versus empirical misclassification error of both matched filter (MF) and CP-based matched filter (CP-MF) classifiers. In cyan, the performance of a neural network with 10 hidden neurons and ReLU activation. We considered n training data as order 3 tensors of dimensions $p_1 = p_2 = p_3 = 20$ having a rank-one structure as in equation 1 with the $\boldsymbol{\mu}_j$'s being randomly sampled vectors. **"CP-MF performs better than MF when few training samples are available".**

Indeed, Figure 2 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. Note that the empirical curves for the CP-based matched filter classifier are obtained with tensor power iteration initialized with tensor SVD, and thus runs in polynomial time if $\|\mathbf{M}\|$ is larger than $\mathcal{O}(p^{\frac{k-2}{4}})$ as discussed previously. Figure 3 shows further experiments varying tensor dimensions and order which show the same conclusions.

Moreover, Figure 4 depicts the misclassification error of both methods varying the dimension p and $\|\mathbf{M}\|$, where we notice that the CP-based matched filter classifier performs better for large values of p in theory (middle plot). More interestingly, the right plot depicts the performance that can be achieved in polynomial time which corresponds to the algorithmic threshold $\|\mathbf{M}\| \geq \mathcal{O}(p^{\frac{k-2}{4}})$. 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.

Generalization to higher-rank data: Our results generalize to a more complex model of the following form. 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), such that for $\mathbf{X}_i \in \mathcal{C}_a$ with $a \in 1, 2$,

$$\mathbf{X}_i = \sum_{\ell=1}^{r_a} \bigotimes_{j=1}^{k} \boldsymbol{\mu}_{j,\ell}^{(a)} + \mathbf{Z}_i \in \mathbb{R}^{p_1 \times \cdots \times p_k}, \tag{9}$$

where $\mathbf{Z}_i$ is a random tensor with i.i.d. standard Gaussian entries, $\boldsymbol{\mu}_{j,\ell}^{(a)} \in \mathbb{R}^{p_j}$ are independent from $\mathbf{Z}_i$ such that $\langle \boldsymbol{\mu}_{j,\ell_1}^{(a)}, \boldsymbol{\mu}_{j,\ell_2}^{(a)} \rangle = \delta_{\ell_1 \ell_2}$. That is, the data tensors $\mathbf{X}_i$ have a rank-r_a (with r_a being small) structure with orthogonal components.

Let us denote $\mathbf{M}_a = \sum_{\ell=1}^{r_a} \bigotimes_{j=1}^{k} \boldsymbol{\mu}_{j,\ell}^{(a)}$. In a supervised setting, it is convenient to center the data by subtracting[5] $\frac{1}{2}(\mathbf{M}_1 + \mathbf{M}_2)$ from each data sample which yields tensors of the form

$$\mathbf{X}_i = (-1)^a (\mathbf{M}_1 - \mathbf{M}_2) + \mathbf{Z}_i, \tag{10}$$

where $\mathbf{M}_1 - \mathbf{M}_2$ is clearly a low-rank tensor (of rank $r_1 + r_2$) with orthogonal components. Stacking all the data samples $\mathbf{X}_i$ in a data tensor $\mathbf{X} \in \mathbb{R}^{p_1 \times \cdots \times p_k \times n}$, the matched filter classifier has weights tensor of the form

$$\mathbf{W} = \frac{1}{\sqrt{np}} \mathbf{X} \times_{k+1} \boldsymbol{y} = \sqrt{\frac{n}{p}} \mathbf{M} + \frac{1}{\sqrt{p}} \tilde{\mathbf{Z}}, \tag{11}$$

where $\tilde{\mathbf{Z}} = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} y_i \mathbf{Z}_i$ and $\mathbf{M} = \mathbf{M}_1 - \mathbf{M}_2 = \sum_{\ell=1}^{r_1+r_2} \bigotimes_{j=1}^{k} \boldsymbol{\mu}_{j,\ell}$ is a rank-$(r_1 + r_2)$ tensor. Therefore,

[5] In real scenarios one would first estimate the $\mathbf{M}_a$'s with their empirical estimates through tensor decomposition.

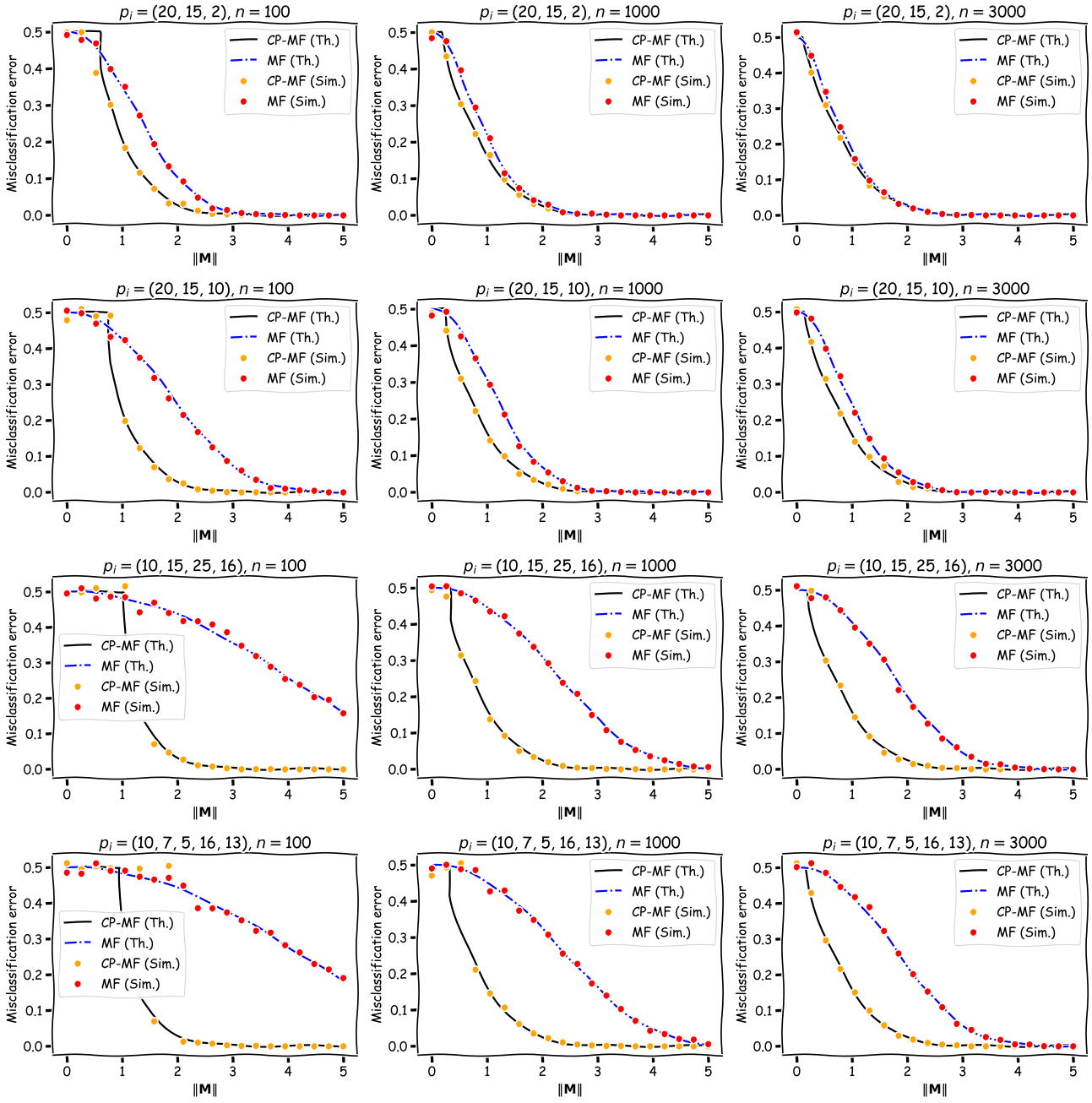

Figure 3: Theoretical versus empirical misclassification error of both matched filter (MF) and CP-based matched filter (CP-MF) classifiers. We considered n training data as k-order tensors with $k \in \{3, 4, 5\}$ of dimensions p_i's having a rank-one structure as in equation 1 with the $\boldsymbol{\mu}_j$'s being randomly sampled vectors.

the CP-based matched filter classifier for this case relies on low-rank approximation of $\mathbf{W}$ of rank $r_1 + r_2$ which might performed using tensor power iteration with deflation procedure. We therefore have the following proposition characterizing the performance of the CP-based matched filter classifier in this case.

Proposition 3.5 (Performance of the CP-based matched filter classifier for data model in equation 10)**.** *Under Assumption 2.2, for* $\tilde{\mathbf{X}}_i \in \mathcal{C}_a$ *with* $a \in \{1, 2\}$ *independent from the training set* $\mathbf{X}$*,*

$$\frac{1}{\sqrt{\sum_{\ell=1}^{r_1+r_2} \sigma_\ell^2}} \left(g_{CP}(\tilde{\mathbf{X}}_i) - m_a \right) \xrightarrow{\mathcal{D}} \mathcal{N}(0, 1),$$

where $m_a = (-1)^a \sum_{\ell=1}^{r_1+r_2} \sigma_\ell \mu_\ell \prod_{j=1}^{k} q_j(\sigma_\ell, \mu_\ell \sqrt{\frac{n}{p}})$ *where* $\mu_\ell = \| \bigotimes_{j=1}^{k} \boldsymbol{\mu}_{j,\ell} \|$ *and* σ_ℓ *satisfies* $f(\sigma_\ell, \mu_\ell \sqrt{\frac{n}{p}}) =$

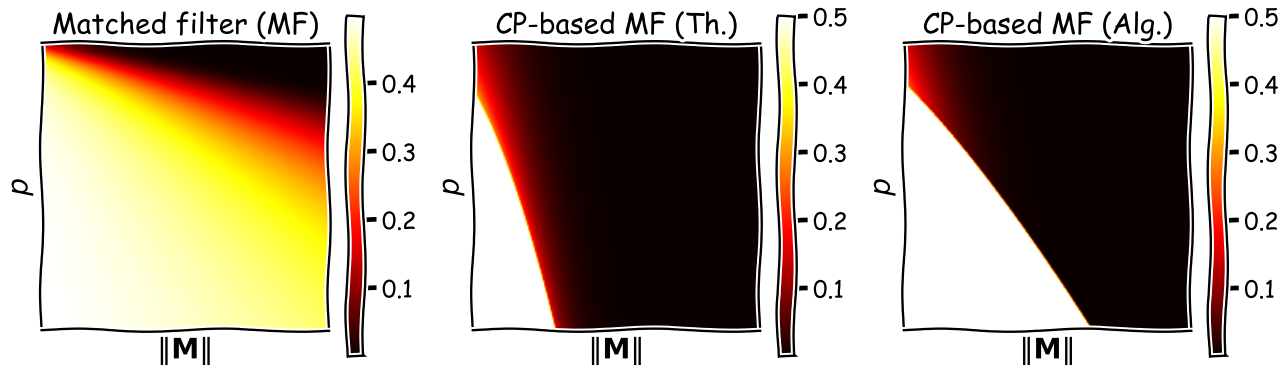

Figure 4: Theoretical misclassification error in terms of the signal strength $\|\mathbf{M}\|$ and the tensor dimension p with $n = 200$ fixed. For both MF and CP-MF as per Propositions 3.1 and 3.3 respectively. The right plot corresponds to polynomial time CP-based MF which is possible for $\|\mathbf{M}\|$ larger than $\mathcal{O}(p^{\frac{k-2}{4}})$ using tensor power iteration initialized with tensor SVD [3].

0. *q_j and f are defined in equation 8. Furthermore, the misclassification error verifies with probability one* $\mathbb{P}\left((-1)^a g_{CP}(\tilde{\mathbf{X}}_i) < 0 \mid \tilde{\mathbf{X}}_i \in \mathcal{C}_a\right) - Q\left(\frac{|m_a|}{\sqrt{\sum_{\ell=1}^{r_1+r_2}\sigma_\ell^2}}\right) \to 0$.

Proof. The proof strategy is the same as for Proposition 3.3. □

3.2 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 equation 2, without loss of generality, we further assume that the data are ordered following their class order, a simple unsupervised learning approach [15] consists in unfolding $\mathbf{X}$ as

$$\boldsymbol{X} = \mathrm{Mat}_{k+1}(\mathbf{X}) = \boldsymbol{y}\,\mathrm{vec}(\mathbf{M})^\top + \mathrm{Mat}_{k+1}(\mathbf{Z}) \in \mathbb{R}^{n\times d}, \tag{12}$$

then estimating the labels $\boldsymbol{y}$ through the dominant eigenvector of the Gram matrix $\boldsymbol{X}\boldsymbol{X}^\top$ denoted $\hat{\boldsymbol{y}}$, which coincides with the dominant left singular vector of $\boldsymbol{X}$. The theoretical performance of this *linear spectral method* is given by the following proposition.

Proposition 3.6 (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 $\hat{\mathcal{C}}_i = \mathrm{sign}(\hat{y}_i)$. Then under Assumption 2.2,*

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

where $\alpha = \kappa\left(\|\mathbf{M}\|\sqrt{\frac{n}{d+n}}, \frac{n}{d+n}\right)^{-1}$, $\sigma = \sqrt{1-\alpha^2}$ and $\kappa(\beta, c) = \beta\sqrt{\frac{\beta^2(\beta^2+1)-c(c-1)}{(\beta^4+c(c-1))(\beta^2+1-c)}}$. Furthermore, the misclassification error is given with probability one by $Q\left(\frac{\alpha}{\sqrt{1-\alpha^2}}\right)$.

Proof. See supplementary material. □

Proposition 3.6 states that the entries of the estimated left 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 5 of [18]). However, under Assumption 2.2, the ratio $\frac{n}{d+n} \to 0$ if $n \to \infty$, thereby yielding a high misclassification error. Figure 5 (left) depicts the $2D$ pro-

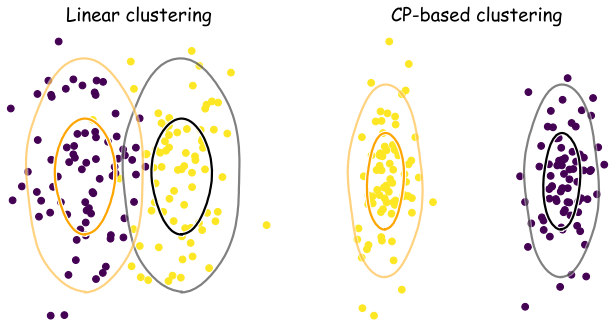

Figure 5: Left: the $2D$ projection space obtained by linear clustering. Right: the $2D$ projection space by CP-based clustering obtained through a rank-two CP decomposition of $\mathbf{X}$. We considered $k = 2$ and $n_1 = n_2 = 75$ square matrices $\mathbf{X}_i$ of size $p = 150$ generated as the model in equation 1 with $\|\mathbf{M}\| = 5$. The ellipses correspond to the theoretical means and fluctuations according to Propositions 3.6 and 3.7 respectively.

jection space corresponding to the two largest eigenvectors of $\boldsymbol{X}\boldsymbol{X}^\top$ along with its theoretical mean and fluctuations as per Proposition 3.6. In contrast, extracting the low-rank structure of the data tensor allows to improve the classification performance. Indeed, given the data model in equation 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*.

Proposition 3.7 (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 2.2,*

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

where $\alpha = q_{k+1}\left(\lambda^\infty, \|\mathbf{M}\|\sqrt{\frac{n}{p+n}}\right)$, $\sigma = \sqrt{1-\alpha^2}$ with $q_{k+1}(\cdot,\cdot)$ defined by equation 8 for a $(k+1)$-th order tensor and λ^∞ is the unique solution to $f\left(\lambda^\infty, \|\mathbf{M}\|\sqrt{\frac{n}{p+n}}\right) = 0$. Furthermore, the misclassification error is given with probability one by $Q\left(\frac{\alpha}{\sqrt{1-\alpha^2}}\right)$.

Proof. See supplementary material. □

Remark 3.8. *In the actual unsupervised setting, the generalization to the data model in equation 9 is more challenging since the data tensor $\mathbf{X}$ in this case does not follow a CP decomposition but rather a block-term decomposition [17] which is more challenging to analyse theoretically and is therefore left for a future investigation.*

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 and p increase at the same rate. Figure 5 (right) depicts the $2D$ projection space obtained by a rank-two CP decomposition of $\mathbf{X}$ with its theoretical mean and fluctuations as per Proposition 3.7. From Figure 5 we clearly 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 by Corollary 3 from [18]. Essentially, in order to have a correlation between $\hat{\boldsymbol{y}}$ and $\boldsymbol{y}$, the signal strength $\|\mathbf{M}\|$ must be greater than 2 in theory. However, in order to estimate the label signal in practice in polynomial time, $\|\mathbf{M}\|$ must be greater than $\frac{\sqrt[4]{c(1-c)}}{\sqrt{c}}$ with $c = \frac{1}{n+1}$, which

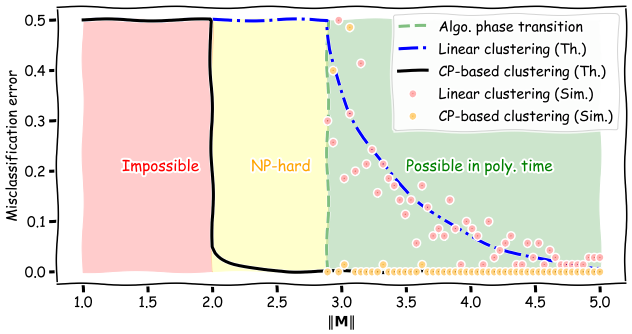

Figure 6: Theoretical versus empirical misclassification errors in terms of the signal strength $\|\mathbf{M}\|$ for both linear clustering and CP-based clustering as per Propositions 3.6 and 3.7 respectively. We considered data to be matrices such that $p_1 = p_2 = n = 70$.

corresponds to the phase transition of linear clustering from Proposition 3.6. Figure 6 depicts the theoretical versus empirical misclassification errors along with the different thresholds for $\|\mathbf{M}\|$. Importantly, it shows three different regions; (i) impossible: where it is informationally impossible to recover the clusters or even detect them, (ii) NP-hard: where there is no polynomial time algorithm that can recover the labels signal, and (iii) possible: where recovery is possible in polynomial time (e.g., using tensor power iteration initialized with tensor SVD as we discussed previously). Figure 6 clearly highlights the benefit of CP-based clustering upon linear clustering if the data has an underlying low-rank structure. Notably, the performances of the different approaches are accurately estimated by Propositions 3.6 and 3.7.

4 Concluding remarks

This paper has brought a 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 and such gain is accurately quantified for the considered statistical model in equation 1. As such, 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.

References

[1] Hafiz Tiomoko Ali and Romain Couillet. Improved spectral community detection in large heterogeneous

networks. *The Journal of Machine Learning Research*, 18(1):8344–8392, 2017.

[2] Animashree Anandkumar, Rong Ge, Daniel Hsu, Sham M Kakade, and Matus Telgarsky. Tensor decompositions for learning latent variable models. *Journal of machine learning research*, 15:2773–2832, 2014.

[3] Arnab Auddy and Ming Yuan. On estimating rank-one spiked tensors in the presence of heavy tailed errors. *arXiv preprint arXiv:2107.09660*, 2021.

[4] Gérard Ben Arous, Daniel Zhengyu Huang, and Jiaoyang Huang. Long random matrices and tensor unfolding. *arXiv preprint arXiv:2110.10210*, 2021.

[5] Wanli Chen, Xinge Zhu, Ruoqi Sun, Junjun He, Ruiyu Li, Xiaoyong Shen, and Bei Yu. Tensor low-rank reconstruction for semantic segmentation. In *European Conference on Computer Vision*, pages 52–69. Springer, 2020.

[6] Christopher J Hillar and Lek-Heng Lim. Most tensor problems are np-hard. *Journal of the ACM (JACM)*, 60(6):1–39, 2013.

[7] 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.

[8] Jonathan Kadmon and Surya Ganguli. Statistical mechanics of low-rank tensor decomposition. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124016, 2019.

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

[10] Jean Kossaifi, Zachary C Lipton, Arinbjörn Kolbeinsson, Aran Khanna, Tommaso Furlanello, and Anima Anandkumar. Tensor regression networks. *Journal of Machine Learning Research*, 21:1–21, 2020.

[11] Joseph M Landsberg. Tensors: geometry and applications. *Representation theory*, 381(402):3, 2012.

[12] Paul Pu Liang, Zhun Liu, Yao-Hung Hubert Tsai, Qibin Zhao, Ruslan Salakhutdinov, and Louis-Philippe Morency. Learning representations from imperfect time series data via tensor rank regularization. *arXiv preprint arXiv:1907.01011*, 2019.

[13] Cosme Louart, Zhenyu Liao, and Romain Couillet. A random matrix approach to neural networks. *The Annals of Applied Probability*, 28(2):1190–1248, 2018.

[14] Xiaoyi Mai and Romain Couillet. 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.

[15] Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856, 2002.

[16] Jeffrey Pennington and Pratik Worah. Nonlinear random matrix theory for deep learning. 2017.

[17] Athanasios A Rontogiannis, Eleftherios Kofidis, and Paris V Giampouras. Block-term tensor decomposition: Model selection and computation. *IEEE Journal of Selected Topics in Signal Processing*, 15(3):464–475, 2021.

[18] Mohamed El Amine Seddik, Maxime Guillaud, and Romain Couillet. When random tensors meet random matrices. *arXiv preprint arXiv:2112.12348*, 2021.

[19] Mohamed El Amine Seddik, Cosme Louart, Romain Couillet, and Mohamed Tamaazousti. The unexpected deterministic and universal behavior of large softmax classifiers. In *International Conference on Artificial Intelligence and Statistics*, pages 1045–1053. PMLR, 2021.

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

[21] Will Wei Sun, Botao Hao, and Lexin Li. Tensors in modern statistical learning. *Wiley StatsRef: Statistics Reference Online*, pages 1–25, 2014.

[22] Malik Tiomoko, Romain Couillet, and Frédéric Pascal. Pca-based multi task learning: a random matrix approach. *arXiv preprint arXiv:2111.00924*, 2021.

[23] Malik Tiomoko, Romain Couillet, and Hafiz Tiomoko. Large dimensional analysis and improvement of multi task learning. *arXiv preprint arXiv:2009.01591*, 2020.

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