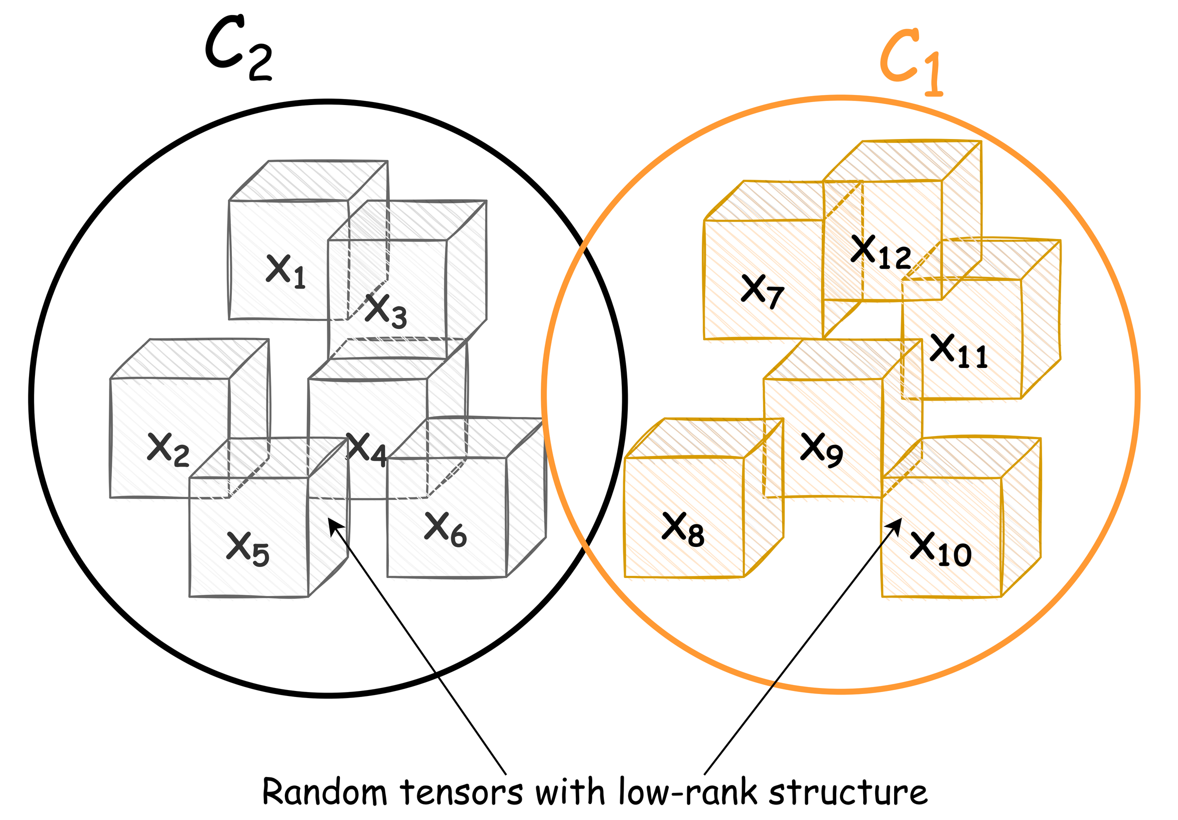
Learning from Low-Rank Tensor Data: A Random Tensor Theory Perspective

Mohamed El Amine Seddik[1] Malik Tiomoko[2] Alexis Decurninge[2] Maxim Panov[1] Maxime Guillaud[3]

[1]Technology Innovation Institute, United Arab Emirates [2]Huawei Technologies, France [3]Inria, France

Abstract



- Theoretical analysis of learning from data with hidden **low-rank tensor** structure.
- Quantification of **performance gain** between considering the low-rank **tensor structure** versus treating data as **vectors**.

Setting & Data Model

We consider n data points: $(\boldsymbol{x}_1 \otimes \boldsymbol{x}_2 \otimes \boldsymbol{x}_3)_{ijk} = x_{1i}x_{2j}x_{3k}$

$$\mathsf{X}_i \in \mathcal{C}_a \quad \Leftrightarrow \quad \mathsf{X}_i = (-1)^a \boldsymbol{\mu}_1 \otimes \cdots \otimes \boldsymbol{\mu}_k + \mathsf{Z}_i \in \mathbb{R}^{p_1 \times \cdots \times p_k}$$

where $[\mathsf{Z}_i]_{i_1 \ldots i_k} \sim \mathcal{N}(0,1)$ i.i.d. and denote $\mathsf{M} = \boldsymbol{\mu}_1 \otimes \cdots \otimes \boldsymbol{\mu}_k$.

- Generalizes the classical model ($k=1$), i.e. $\boldsymbol{x}_i = (-1)^a \boldsymbol{\mu}_1 + \boldsymbol{z}_i$.
- Even for $k \geq 2$, the standard approach consists in **flattening** the data.
- What is the **optimal** classifier? Theoretical misclassification?

Supervised Setting

Given $\mathsf{X} = [\mathsf{X}_1, \ldots, \mathsf{X}_n] \in \mathbb{R}^{p_1 \times \cdots p_k \times n}$ and $\boldsymbol{y} = [y_1, \ldots, y_n] \in \{-1, 1\}^n$

Denote $\boldsymbol{X} = \mathsf{X}_{(k+1)} \in \mathbb{R}^{n \times P}$ with $P = \prod_{i=1}^k p_i$ and consider the *Ridge* classifier:

$$\min_{\boldsymbol{w}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|^2 + \gamma \|\boldsymbol{w}\|^2 \quad \Leftrightarrow \quad \boldsymbol{w}^* = \left(\boldsymbol{X}^\top \boldsymbol{X} + \gamma \boldsymbol{I}\right)^{-1} \boldsymbol{X}^\top \boldsymbol{y}$$

For some $\gamma \gg \|\boldsymbol{X}^\top \boldsymbol{X}\|$ (optimal for the above data model):

$$\boldsymbol{w} = \frac{1}{\sqrt{np}} \boldsymbol{X}^\top \boldsymbol{y}$$

where $p = \sum_{i=1}^k p_i$. In tensor notations, the decision function is:

$$f_{\mathsf{R}}(\tilde{\mathsf{X}}_i) = \langle \mathsf{W}, \tilde{\mathsf{X}}_i \rangle \underset{\mathcal{C}_2}{\overset{\mathcal{C}_1}{\gtrless}} 0 \qquad \mathsf{W} \equiv \frac{1}{\sqrt{np}} \mathsf{X} \times_{k+1} \boldsymbol{y}$$

with $\tilde{\mathsf{X}}_i$ a test datum independent of X.

Assumption: $p_i = \mathcal{O}(n)$ and $\|\mathsf{M}\| = \mathcal{O}(1)$.

Data Flattening Performance

Theorem: For $\tilde{\mathsf{X}}_i$ independent of X:

$$\frac{1}{\sigma}\left(f_{\mathsf{R}}(\tilde{\mathsf{X}}_i) - m_a\right) \xrightarrow{\mathcal{D}} \mathcal{N}(0,1) \quad \Rightarrow \quad \mathcal{E} = Q\left(\frac{|m_a|}{\sigma}\right)$$

where $m_a = (-1)^a \|\mathsf{M}\|^2 \sqrt{\frac{n}{p}}$ and $\sigma = \sqrt{\frac{n}{p}\|\mathsf{M}\|^2 + \frac{P}{p}}$.

Tensor-based Classification

The weight tensor W is a **spiked random tensor**:

$$\mathsf{W} = \sqrt{\frac{n}{p}} \bigotimes_{i=1}^k \boldsymbol{\mu}_i + \frac{1}{\sqrt{p}} \mathsf{Z}$$

with $\mathsf{Z} = \frac{1}{\sqrt{n}} \sum_{i=1}^n y_i \mathsf{Z}_i$ (*Universality with CLT*).

Tensor-Ridge classifier is defined as:

$$f_{\mathsf{TR}}(\tilde{\mathsf{X}}_i) = \left\langle \lambda^* \bigotimes_{i=1}^k \boldsymbol{u}_i^*, \tilde{\mathsf{X}}_i \right\rangle \underset{\mathcal{C}_2}{\overset{\mathcal{C}_1}{\gtrless}} 0$$

where (best rank-one approximation of W):

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

Remark: The above MLE is **NP-hard** but feasible if $\|\mathsf{M}\| \geq \mathcal{O}(P^{1/4}/p^{1/2})$.

Tensor-based Performance

Theorem: For $\tilde{\mathsf{X}}_i$ independent of X:

$$\frac{1}{\sigma}\left(f_{\mathsf{TR}}(\tilde{\mathsf{X}}_i) - m_a\right) \xrightarrow{\mathcal{D}} \mathcal{N}(0,1) \quad \Rightarrow \quad \mathcal{E} = Q\left(\frac{|m_a|}{\sigma}\right)$$

where $m_a = (-1)^a \sigma \|\mathsf{M}\| \prod_{j=1}^k q_j(\sigma)$ and $f\left(\sigma, \|\mathsf{M}\|\sqrt{\frac{n}{p}}\right) = 0$ with q_j and f in [1].

Unsupervised Setting

- **Linear clustering:** compute the **left singular vector** of:

$$\boldsymbol{X} = \mathsf{X}_{(k+1)} = \boldsymbol{y} \otimes \mathtt{flatten}(\mathsf{M}) + \boldsymbol{Z} \in \mathbb{R}^{n \times P} \quad \rightarrow \quad \hat{\boldsymbol{y}}$$

- **Tensor-based clustering:** compute the **best rank-one approximation** of:

$$\mathsf{X} = \mathsf{M} \otimes \boldsymbol{y} + \mathsf{Z} \in \mathbb{R}^{p_1 \times \cdots \times p_k \times n} \quad \rightarrow \quad \hat{\boldsymbol{y}}$$

Theorem: The estimated class for X_i is given by $\operatorname{sign}(\hat{y}_i)$:

$$\frac{1}{\sqrt{1-\alpha^2}}\left(\sqrt{n}\hat{y}_i - \alpha y_i\right) \xrightarrow{\mathcal{D}} \mathcal{N}(0,1) \quad \Rightarrow \quad \mathcal{E} = Q\left(\frac{\alpha}{\sqrt{1-\alpha^2}}\right)$$

- **Linear:** $\alpha = \kappa\left(\|\mathsf{M}\|\sqrt{\frac{n}{P+n}}, \frac{n}{P+n}\right)^{-1}$ with κ in [1].
- **Tensor:** $\alpha = q_{k+1}(\lambda^*)$ with $f\left(\lambda^*, \|\mathsf{M}\|\sqrt{\frac{n}{p+n}}\right) = 0$ (λ^* spectral norm of X).

Simulations

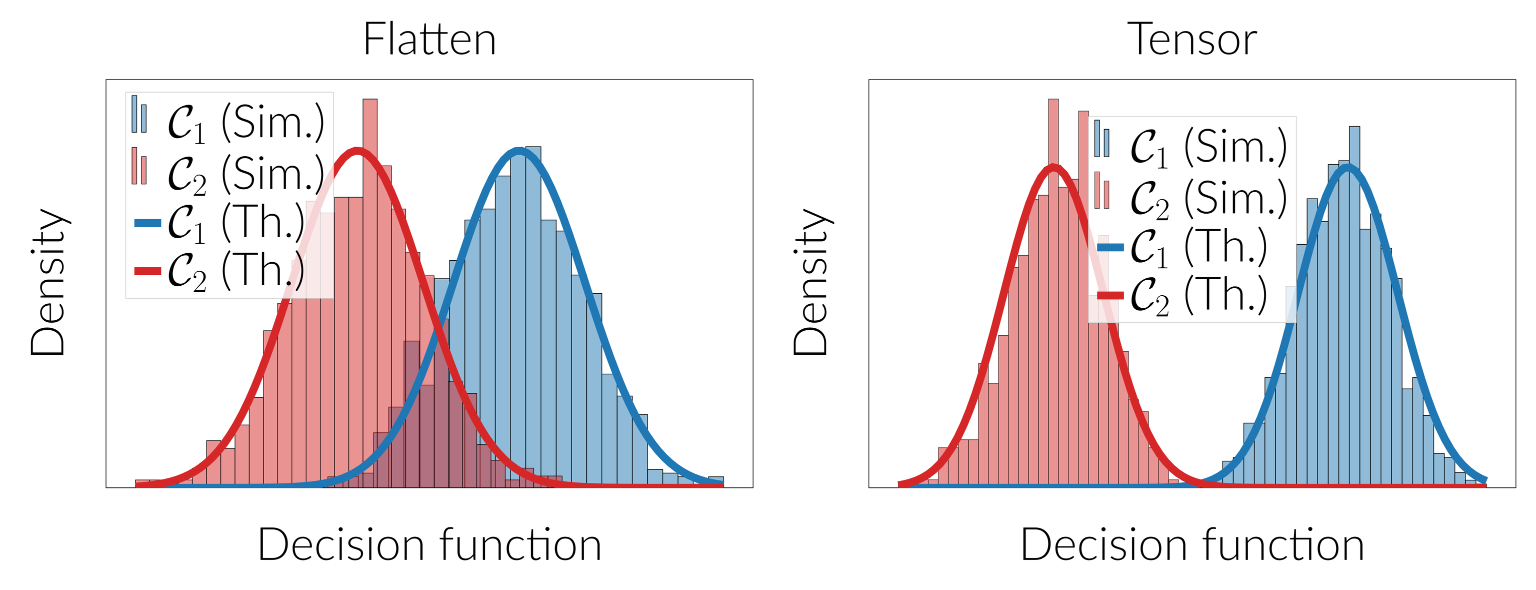


Figure 1. $n = 200$, tensor shape $(15, 30, 20)$ and $\|\mathsf{M}\| = 3$.

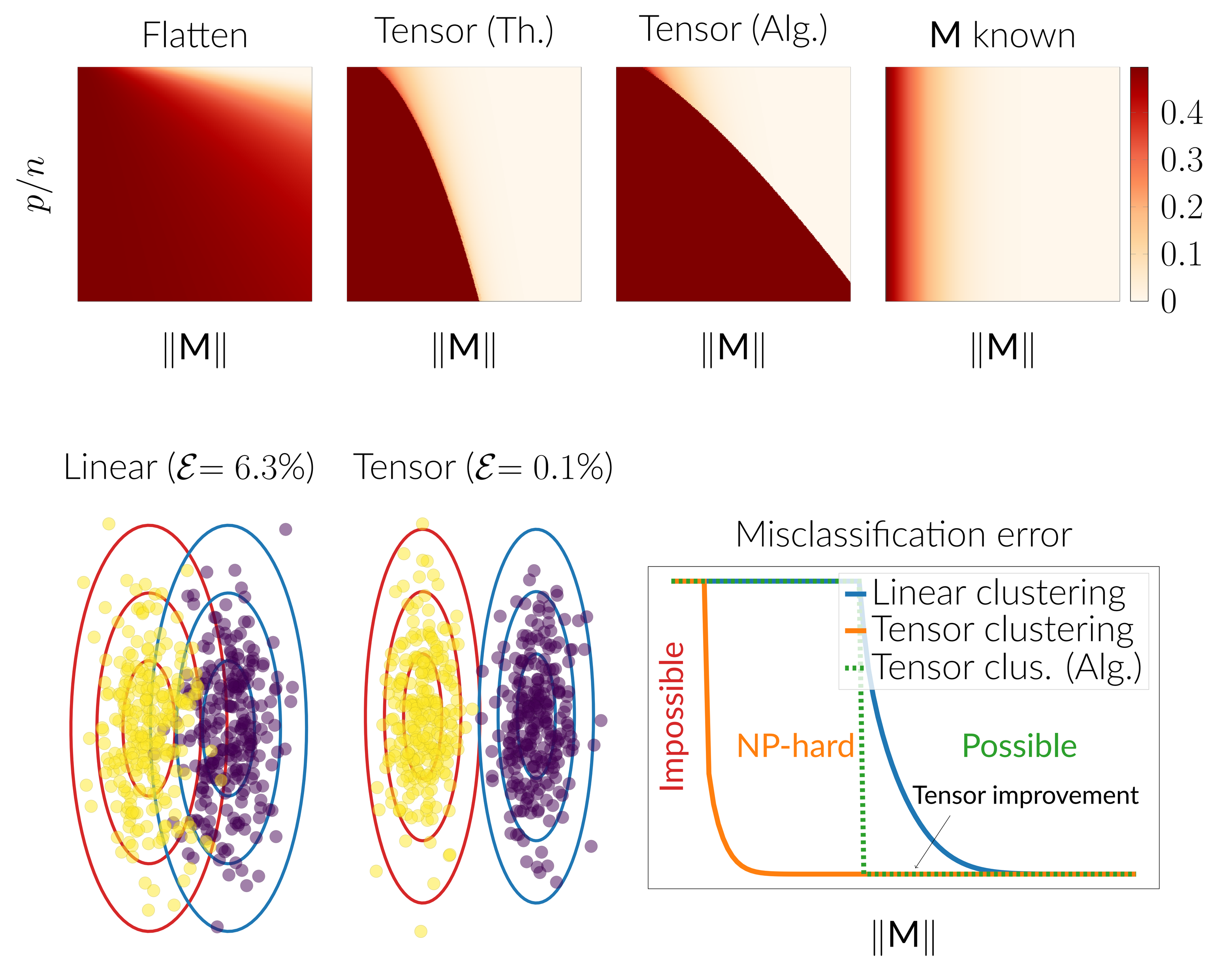


Figure 2. $n = 200$, tensor shape $(15, 30, 20)$ and $\|\mathsf{M}\| = 3$.

Conclusion

- This work analyzes learning from **low-rank tensor** data and shows **performance gains**.
- It applies **random tensor theory** to evaluate simple learning methods.
- This paves the way for **improving machine learning** algorithms for tensor-structured data.

[1] MEA.Seddik, M.Guillaud, R.Couillet, "When Random Tensors meet Random Matrices", *Annals of Applied Probability 2023*.

Uncertainty in Artificial Intelligence - Pittsburgh, PA, USA - 2023

Link to article: `https://openreview.net/pdf?id=uBqcJ8QEe1`

Get in touch: `melaseddik.github.io`