

# Tensor multiblock logistic regression

---

SELVESTREL Alexandre

January 13, 2025

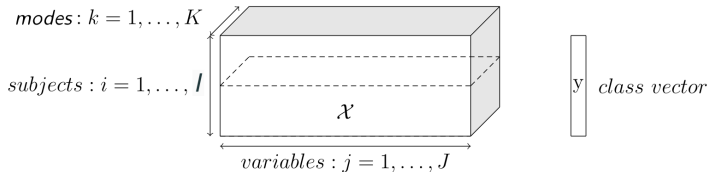
Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des signaux et systèmes

**Supervisors :** Arthur Tenenhaus, Laurent Lebrusquet

**Medical partner :** Henri Mondor hospital, radiologist: Sébastien Mulé

# Tensor data

Measuring the same variables across several modalities (position, time, etc...) → features with tensor structure

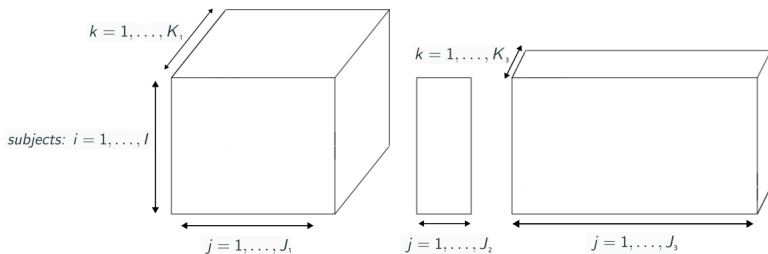


**Figure 1:** Type of data: tensorial

**Warning:** Often strong correlation between features of the same variable across different modalities → adapt the model to this structure

# Multiblock data

Each block is a tensor with its own structure



**Figure 2:** Type of data: multiblock

# Application example of real multiblock tensor data

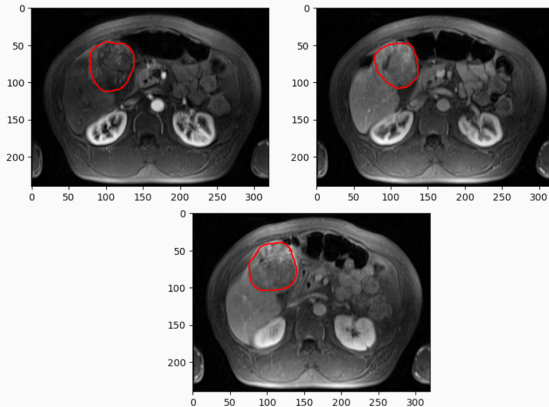
## classification of liver tumors through MRI:

Same variables extracted from each MRI image at 3 times (arterial, portal, late) → **tensor structure**

- MRI images in 3D of liver tumors. Information about grey levels, shape and correlations → **multiblock structure**
- Also include univariate features: gender and age

**Conclusion: Multiblock tensor data** → train a tensor multiblock logistic regression model

## Some MRI images



**Figure 3:** Example of MRI images of a liver tumor (arterial, portal, late)  
from From Henri Mondor hospital: the 3 images look quite similar

# Table of contents

New model for logistic regression

- Standard logistic regression

- Tensor logistic regression

- Tensor multiblock logistic regression

Fitting algorithm

Simulations and results

Application of proposed logistic regression to liver tumor classification

## New model for logistic regression

---

# Logistic regression

Generalized linear regression model for classification:

$\mathbf{x}$  : features vector (1)

$Y$  : binary response (explained variable) (2)

$$P(Y = 1|x) = \frac{\exp(\beta_0 + \mathbf{x}^T \boldsymbol{\beta})}{1 + \exp(\beta_0 + \mathbf{x}^T \boldsymbol{\beta})}$$

Defines a likelihood function  $\mathcal{L}(\boldsymbol{\beta}) = \prod_{i=1}^I P(Y_i = y_i | x_i)$



To many features (vs  $I$ )  $\rightarrow$  need to limit variance of prediction.

Search for sparse easily interpretable model: choice of lasso

$$\text{penalization} = \lambda \|\beta\|_1 \quad (\lambda > 0)$$

Function to minimize :

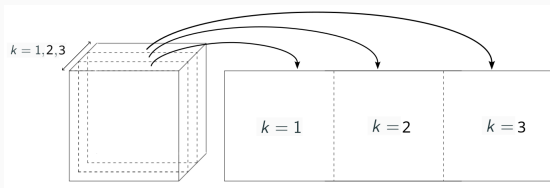
$$-\log(\mathcal{L}(\beta)) + \lambda \|\beta\|_1$$

Convex optimization problem

## Naive approach: unfolding

$\beta = (\beta_{j,k})_{j \in \llbracket 1, J \rrbracket, k \in \llbracket 1, K \rrbracket}$  so  $JK$  parameters to determine

$$x^T \beta \rightsquigarrow \sum_k \sum_j \beta_{j,k} x_{j,k} \quad \text{and} \quad \|\beta\|_1 = \sum_k \sum_j |\beta_{j,k}|$$



**Figure 4:** Unfolding a tensor

**Limitation of lasso:** Elimination of features without specific consideration for the same feature at other times/ other features at the same time

**Common solution:** grouping regression coefficients together in the penalization

$$\sum_{j,k} |\beta_{j,k}| \rightsquigarrow \sum_{g=1}^G \|\beta^g\|_2$$

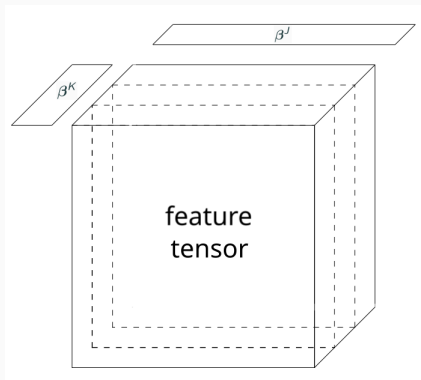
Tendency to set regression coefficients to zero by entire blocks.

### Downsides:

- grouping either by mode or by variable, not both.
- No tensor structure (or at least grouping of variables) in the likelihood term.

# Tensor regression models

**Idea:** each variable and mode has its own influence on the prediction (i.e. on  $\beta$ ) [2].



**Figure 5:** Tensor structure of  $\beta$

# Tensor regression models

**Idea:** each variable and mode has its own influence on the prediction (i.e. on  $\beta$ ) [2].

For  $J$  variables observed on  $K$  modalities (e.g. times)

$$\beta_{j,k} = \beta_j^J \beta_k^K$$

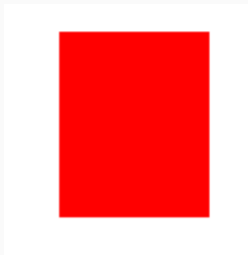
$\beta_j$  : impact of variable  $j$

$\beta_k$  : impact of modality  $k$

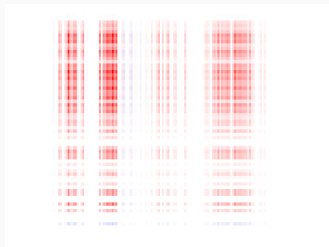
Only  $J + K$  parameters to determine (instead of  $JK$ )

# Limits of rank 1

$\beta_{j,k} = \beta_j^J \beta_k^K$  implies that  $\beta$  looks like:



**Figure 6:** Example of rank 1 pictogram (only 0 and 1)

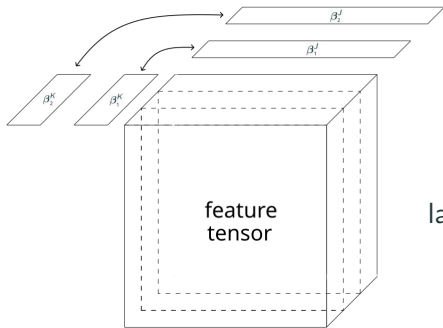


**Figure 7:** Example of rank 1 matrix (all values allowed)

This can be too simplistic

# Rank R tensor logistic regression [1]

Summing rank 1 together :  $\beta_{j,k} = \sum_{r=1}^R \beta_{j,r}^J \beta_{k,r}^K$

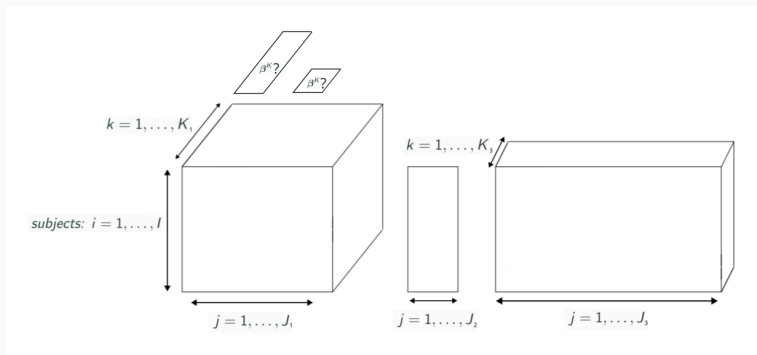


$$\text{lasso} \rightsquigarrow \lambda \sum_{r=1}^R \left( \|\beta_r^J\|_1 \|\beta_r^K\|_1 \right)$$

**Figure 8:** Tensor structure of  $\beta$

# Blocks of variables

**Problem:** Several groups of variables of different natures (first order, shape, texture). But  $\beta_r^K$  and  $\beta_r^J$  common to all groups.  
 $K_1 = K_2 = K_3$  needed or else:

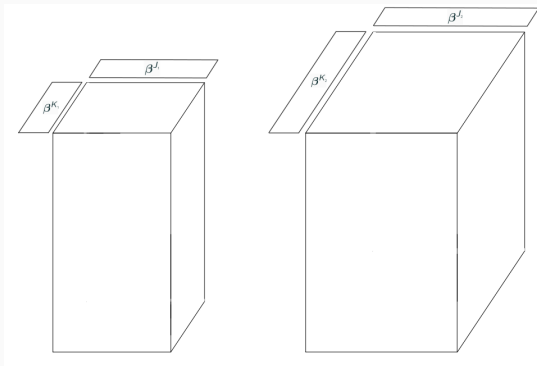


**Figure 9:** Problem if blocks have different orders or dimensions



# Tensor multiblock logistic regression

**Solution:** giving each block its own  $\beta^J$  and  $\beta^K$



**Figure 10:** Tensor multiblock model for rank 1

# Tensor multiblock logistic regression

Mathematically, this gives (for  $L$  blocks):

$$\mathbf{x}^T \boldsymbol{\beta} \rightsquigarrow \sum_{\ell=1}^L \sum_{j,k} x_{j,k}^{\ell} (\beta_{\ell})_{j,k}$$

With, for rank 1:  $(\beta_{\ell})_{j,k} = (\beta_{\ell}^J)_j (\beta_{\ell}^K)_k$

But each  $\beta_{\ell}$  can have a different rank  $R_{\ell}$ , which gives:

$$(\beta_{\ell})_{j,k} = \sum_{r=1}^{R_{\ell}} (\beta_{\ell,r}^J)_j (\beta_{\ell,r}^K)_k$$

Lasso penalty  $\rightsquigarrow \lambda \sum_{\ell,r} \left( \|\beta_{\ell,r}^K\|_1 \|\beta_{\ell,r}^J\|_1 \right)$

## Fitting algorithm

---

Likelihood term of the loss determined by

$$\mathbf{x}^T \boldsymbol{\beta} = \sum_{\ell, r, j} \left( \sum_k x_{j,k}^{\ell} (\beta_{\ell, r}^K)_k \right) (\beta_{\ell, r}^J)_j \quad (3)$$

$$= \sum_{\ell, r, k} \left( \sum_j x_{j,k}^{\ell} (\beta_{\ell, r}^J)_j \right) (\beta_{\ell, r}^K)_k \quad (4)$$

**Strategy:** alternating logistic regressions (to optimize respectively  $\beta^J$  and  $\beta^K$ )

If a mode is inexistent in a block, renumber the modes of that block to perform the optimization on another (existing) mode of this block

## Lasso penalty in fitting algorithm

$$\text{penalty} \propto \sum_{\ell,r} \left( \|\beta_{\ell,r}^K\|_1 \|\beta_{\ell,r}^J\|_1 \right) \quad (5)$$

$$= \sum_{\ell,r} \left( \left\| \|\beta_{\ell,r}^K\|_1 \beta_{\ell,r}^J \right\|_1 \right) = \sum_{\ell,r} \left( \left\| \|\beta_{\ell,r}^J\|_1 \beta_{\ell,r}^K \right\|_1 \right) \quad (6)$$

## Lasso penalty in fitting algorithm

$$\text{penalty} \propto \sum_{\ell,r} \left( \|\beta_{\ell,r}^K\|_1 \|\beta_{\ell,r}^J\|_1 \right) \quad (5)$$

$$= \sum_{\ell,r} \left( \left\| \|\beta_{\ell,r}^K\|_1 \beta_{\ell,r}^J \right\|_1 \right) = \sum_{\ell,r} \left( \left\| \|\beta_{\ell,r}^J\|_1 \beta_{\ell,r}^K \right\|_1 \right) \quad (6)$$

**Strategy:** dilate  $\beta_{\ell,r}^J$  by  $\|\beta_{\ell,r}^K\|_1$  and  $x_{j,k}^\ell$  by  $\|\beta_{\ell,r}^K\|_1^{-1}$ , so

$$\mathbf{x}^T \beta = \sum_{j,\ell,r} \left( \sum_k x_{j,k}^\ell (\beta_{\ell,r}^K)_k \right) (\beta_{\ell,r}^J)_j$$

does not change but

$$\|\beta^J\|_1 \rightsquigarrow \sum_{\ell,r} \left( \|\beta_{\ell,r}^K\|_1 \|\beta_{\ell,r}^J\|_1 \right)$$

## Lasso penalty in fitting algorithm

$$\text{penalty} \propto \sum_{\ell,r} \left( \|\beta_{\ell,r}^K\|_1 \|\beta_{\ell,r}^J\|_1 \right) \quad (5)$$

$$= \sum_{\ell,r} \left( \left\| \|\beta_{\ell,r}^K\|_1 \beta_{\ell,r}^J \right\|_1 \right) = \sum_{\ell,r} \left( \left\| \|\beta_{\ell,r}^J\|_1 \beta_{\ell,r}^K \right\|_1 \right) \quad (6)$$

**Strategy:** dilate  $\beta_{\ell,r}^J$  by  $\|\beta_{\ell,r}^K\|_1$  and  $x_{j,k}^\ell$  by  $\|\beta_{\ell,r}^K\|_1^{-1}$ , so

$$\mathbf{x}^T \beta = \sum_{j,\ell,r} \left( \sum_k x_{j,k}^\ell (\beta_{\ell,r}^K)_k \right) (\beta_{\ell,r}^J)_j$$

does not change but

$$\|\beta^J\|_1 \rightsquigarrow \sum_{\ell,r} \left( \|\beta_{\ell,r}^K\|_1 \|\beta_{\ell,r}^J\|_1 \right)$$

Then, do logistic lasso regression with  $\mathbf{x} = (x_{ijk}^J \|\beta_{\ell,r}^K\|_1^{-1})$  and  $\lambda$  penalty coefficient to find  $\left( (\beta_{\ell,r}^J)_j \|\beta_{\ell,r}^K\|_1 \right)$

## Stopping criterion

Before each optimization cycle, calculate the loss:

$$C = -\log(\mathcal{L}(\beta)) + \text{penalty}$$

Before the  $t$ -th optimization cycle, its value is  $C^t$  and after this cycle it becomes  $C^{t+1}$

End the optimization when:

$$|C^{t+1} - C^t| < \epsilon |C^t|$$

(typically  $\epsilon = 10^{-4}$ )



## Simulations and results

---

# Aim of the simulations

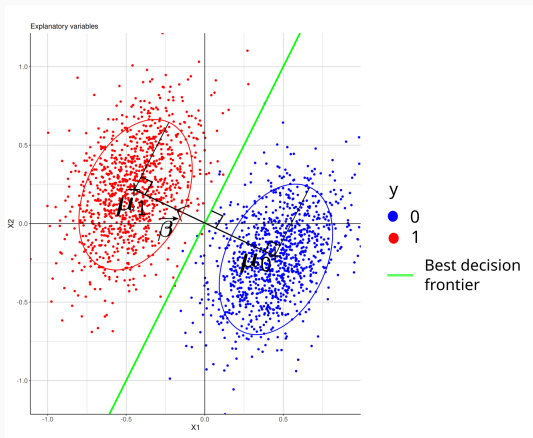
Allow to choose

- Difficulty of the classification (overlap between classes, distance between means of classes etc ...)
- Balance between classes
- Structure of the regression parameter  $\beta$  (several blocks)

Allow to evaluate

- Quality of the classification (Area Under the ROC Curve : AUC)
- Quality of the reconstruction of  $\beta$

## Illustration in 2D



**Figure 11:** Example of explanatory variables for  $\beta = (-2, 1)$

Choose the  $\beta$  to be reconstructed (pictograms)

Generate the  $(\mathbf{x}_i)_{i \in \llbracket 1, I \rrbracket}$  with 2 multivariate normal laws of means  $\mu_0$  and  $\mu_1$  and common covariance matrix  $\Sigma$  such that:

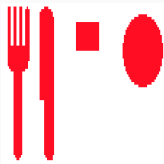
- $\mu_1 - \mu_0$  colinear to  $\beta$
- One of the principal axis of  $\Sigma$  colinear to  $\beta$

Separation of classes linked with eigenvalues of  $\Sigma$  (to be compared with  $\|\mu_1 - \mu_0\|$ )

# AUC simulated data

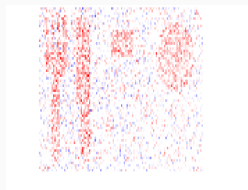
**Table 1:** Cross validated AUC for each model on simulated data for 3000 individuals

$(\sigma_{\beta}, \sigma_{\text{noise}})$	lasso	g.l. (blocks)	g.l. (mode)	g.l. (var)	tensor	tensor blocks
(0.1,0.5)	0.83	0.86	0.94	0.94	0.99	0.99
(0.1,0.8)	0.63	0.64	0.68	0.68	0.93	0.99



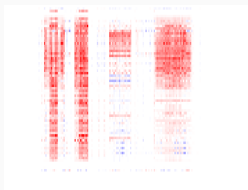
**Figure 12:** Pictogram of shape  $66 \times 117$

# Reconstructed $\beta$



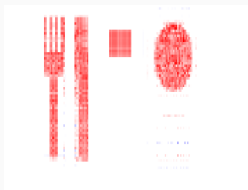
**(a)** lasso

$(\sigma_\beta, \sigma_{\text{noise}}) = (0.1, 0.5)$



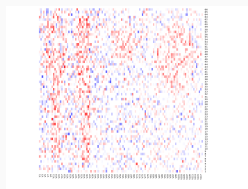
**(b)** tensor  $R : 10$

$(\sigma_\beta, \sigma_{\text{noise}}) = (0.1, 0.5)$



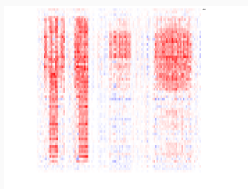
**(c)** T.M.  $R : (12, 1, 10)$

$(\sigma_\beta, \sigma_{\text{noise}}) = (0.1, 0.5)$



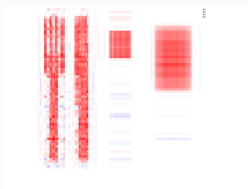
**(d)** lasso

$(\sigma_\beta, \sigma_{\text{noise}}) = (0.1, 0.8)$



**(e)** tensor  $R : 10$

$(\sigma_\beta, \sigma_{\text{noise}}) = (0.1, 0.8)$



**(f)** T.M.  $R : (6, 1, 1)$

$(\sigma_\beta, \sigma_{\text{noise}}) = (0.1, 0.8)$

## Interpretation of the results

Better reconstruction by tensor multiblock logistic model

Performance (AUC) in adequation with quality of reconstruction (because small changes in  $\beta \rightsquigarrow$  big changes in the classification, when  $\sigma_{\text{noise}}$  increases)

Efficient sparse (low rank) approximation of pictograms by tensor multiblock logistic model in contrast with simple tensor model.

Tendency to choose rank 1 for both tensor model when too few data (or too high noise)

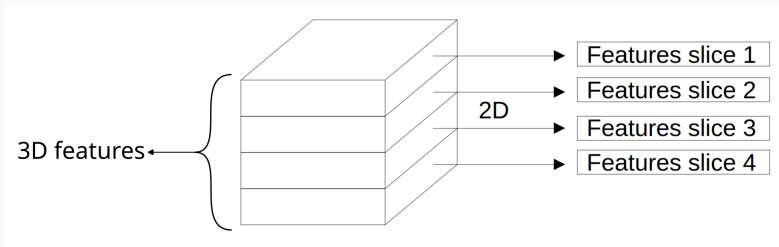
# **Application of proposed logistic regression to liver tumor classification**

---



## Feature extraction with pyradiomics [4]

Extraction of  $\simeq 100$  features (about intensities, shape, texture) for each 2D or 3D image.



**Figure 14:** Feature extraction with pyradiomics for an MRI image composed of 4 slices

## Results

Type of data	lasso	g.l. (block)	g.l. (time)	g.l. (var)	tensor	tensor blocks
3D	$0.74 \pm 0.04$	$0.78 \pm 0.03$	$0.76 \pm 0.03$	$0.73 \pm 0.03$	$0.77 \pm 0.03$	$0.77 \pm 0.03$

Cross validated AUC on 3D real data

Type of data	lasso	g.l. (block)	g.l. (slice)	g.l. (time)	g.l. (var)	tensor	tensor blocks
2D	$0.73 \pm 0.03$	$0.71 \pm 0.03$	$0.70 \pm 0.04$	$0.71 \pm 0.03$	$0.71 \pm 0.03$	$0.66 \pm 0.04$	$0.71 \pm 0.03$

Cross validated AUC on 2D real data

# Conclusion

State of the art performances

Better than state of the art in simulated data when few overlapping between classes

Scales better than regular logistic regression for high order tensors

Lowers the complexity of the regression model and therefore reduces overfitting

Good interpretability (sparse + displays importance of each block, mode and variable in  $\beta$ )

## Further work

Testing on other real datasets whether the performances on the simulated dataset can be replicated

Testing other penalizations (group lasso, elastic net)

Extending the multiblock approach to other classical machine learning algorithms (other GLMs, SVM etc...). Comparing it to CNN.

Improving the optimization, by using coordinate descent (as done in glmnet [3] in R)

# Bibliography

---



Fabien Girka, Pierrick Chevaillier, Arnaud Gloaguen, Giulia Gennari, Ghislaine Dehaene-Lambertz, Laurent Le Brusquet, and Arthur Tenenhaus.

### **Rank-R Multiway Logistic Regression.**

In *52èmes Journées de Statistique*, Nice, France, 2021.

les 52èmes journées de Statistique 2020 sont reportées ! Elles auront lieu du 7 au 11 Juin 2021.



Laurent Le Brusquet, Gisela Lechuga, and Arthur Tenenhaus.

### **Régression Logistique Multivoie.**

In *JdS 2014*, page 6 pages, Rennes, France, June 2014.



Rob Tibshirani, Trevor Hastie, and Jerome Friedman.

### **Regularized paths for generalized linear models via coordinate descent.**

*Journal of Statistical Software*, 33, 02 2010.



Joost J.M. van Griethuysen, Andriy Fedorov, Chintan Parmar, Ahmed Hosny, Nicole Aucoin, Vivek Narayan, Regina G.H. Beets-Tan, Jean-Christophe Fillion-Robin, Steve Pieper, and Hugo J.W.L. Aerts.

**Computational Radiomics System to Decode the Radiographic Phenotype.**

*Cancer Research*, 77(21):e104–e107, 10 2017.



Hua Zhou, Lexin Li, and Hongtu Zhu.

**Tensor regression with applications in neuroimaging data analysis.**

*Journal of the American Statistical Association*, 108:540–552, 06 2013.