Rapport de stage de fin d'études sur l'analyse par machine learning de données médicales multivariées

Internship report on machine learning analysis of multivariate medical data

Alexandre SELVESTREL

Laboratoire des systèmes, Centrale-Supélec

Encadrants: Arthur Tenenhaus, Laurent Lebrusquet

Soutenance le 29 Novembre 2024

Synthèse (version française)

Présentation générale. L'objectif de mon stage était de réaliser une classification automatique (via du machine learning) de tumeurs du foie basée sur des IRMs et sur quelques données cliniques (âge, sexe du patient ...). Cette classification devait permettre de tester et améliorer des modèles tensoriels récents [1, 2] et de vérifier si ceux-ci donnaient de meilleures performances que les autres modèles. Ce stage était effectué au laboratoire des systèmes (l2S) en partenariat avec l'assistance publique des hôpitaux de Paris (AP-HP). Sur le versan médical, nous avons pu bénéficier de l'aide de Sébastien Mulé, Maître de conférence à la faculté de santé, Université Paris-Est Créteil (UPEC) et Radiologie, chef du département imagerie de l'hôpital Henri Mondor.

Enjeux. Ce stage s'inscrit dans la cadre de la collaboration entre le l2S et l'AP-HP. Du point de vue du l2S, il s'agit de mettre à l'épreuve des méthodes de machine learning particulières, basées sur des tenseurs et qui semblent spécifiquement adaptées aux données étudiées. Par ailleurs, en me formant au machine learning appliqué au domaine médical, le laboratoire s'assure dès le début du stage qu'en poursuivant en doctorat, je disposerai des compétences nécessaires pour être immédiatement opérationnel.

Pour l'AP-HP, l'enjeu est de faire progresser la recherche sur le cancer du foie. En effet, la détermination de la nature de la tumeur du foie d'un patient est un problème complexe auquel il n'existe pas de solution complètement satisfaisante à l'heure actuelle. Or, les médecins disposant des IRMs des patients malades, il serait dommage de ne pas les utiliser pour tenter de proposer un outil de diagnostic automatique. Même dans le cas où cet outil serait moins performant que ce qui existe déjà, il pourraît être utile aux médecins pour déterminer de nouveaux indices qui caractérisent la classe d'une tumeur.

Solutions et résultats:. Nous avons commencé par implémenter des modèles statistiques basiques (régression logistique lasso et random forest) sur les données de cancer du foie. Cela nous a permis d'établir une valeur de référence pour la performance de la classification (AUC = 0.68). Nous avons ensuite cherché à améliorer ce score en prorammant une régression logistique tensorielle (voir la section "Méthodes"). Mais malgré plusieurs tentatives d'amélioration du modèle (notamment en séparant les variables en plusieurs blocs), aucun gain de performance n'était observé.

Afin de vérifier que notre modèle était pertinent, nous avons alors cherché à tester son efficacité sur des données simulées. Sur ces données, notre modèle tensoriel a montré des performances bien meilleures que les modèles non tensoriels. Cela nous a permis de conclure que ce modèle était pertinent dans certains cas et que le manque de performance observé sur les données médicales était probablement dû à la mauvaise qualité de ces données. Après plus d'un mois de travail pour améliorer la qualité des données, je me suis rendu à l'Hôpital Henri Mondor afin de parler de mes résultats avec Sébastien Mulé sur son lieu de travail (et non dans mon laboratoire comme les fois précédente). Cette visite a permis de découvrir l'existence d'un autre jeu de données complètement ommises jusqu'à présent, beaucoup plus simples (seulement une quinzaine de variables par individu), donnant des résultats bien meilleures que les données précédentes quand on les traite par machine learning. Nous avons été surpris par l'arrivée au dernier moment de ces données qui, bien que de bonne qualité, ne sont pas adaptées aux modèles tensoriels. Nous ne les mentionnons donc pas dans la partie "article" du rapport mais nous les présentons juste après.

Synthesis (english version)

Overview. The aim of my internship was to carry out an automatic classification (via machine learning) of liver tumors based on MRI scans and some clinical data (patient age, sex, etc.). The goal of this classification was to test and improve recent tensor models [1, 2] and check whether they performed better than other models. The internship was carried out at the systems laboratory (l2S), in partnership with the Assistance Publique des Hôpitaux de Paris (AP-HP). On the medical side, we benefited from the help of Sébastien Mulé, Senior Lecturer at the Faculty of Health, Université Paris-Est Créteil (UPEC) and Radiology, Head of the Imaging Department at Henri Mondor Hospital.

Stakes. This internship is part of the collaboration between l2S and AP-HP. From the point of view of the l2S, the aim is to put to the test particular machine learning methods, based on tensors, which seem specifically adapted to the data under study. What's more, by training me in machine learning applied to the medical field, the laboratory has ensured from the start of the internship that if I go on to do a PhD, I'll have the necessary skills to be immediately operational.

For AP-HP, the challenge is to advance research into liver cancer. Determining the nature of a patient's liver tumor is a complex problem for which there is currently no completely satisfactory solution. Since doctors have access to MRI scans of patients with liver tumours, it would be a shame not to use them to try and offer an automatic diagnostic tool. Even if such a tool is less effective than what already exists, it could still be useful to doctors in determining new clues that characterize the class of a tumor.

Solutions and results:. We began by implementing basic statistical models (lasso logistic regression and random forest) on the liver cancer data. This enabled us to establish a benchmark for classification performance (AUC = 0.68). We then sought to improve this score by applying tensor logistic regression (see "Methods" section). But despite several attempts to improve the model (notably by separating the variables into several blocks), no gain in performance was observed. In order to verify the relevance of our model, we then sought to test its effectiveness on simulated data. On simulated data, our tensor model performed much better than the non-tensor models. This allowed us to conclude that the model was relevant in certain cases, and that the lack of performance observed on medical data was probably due to the poor quality of the data.

After more than a month's work to improve data quality, I went to Hôpital Henri Mondor to discuss my results with Sébastien Mulé at his workplace (and not in my laboratory as on previous occasions). This visit led to the discovery of another, hitherto completely omitted dataset, much simpler (only about fifteen variables per individual), giving much better results than the previous data when processed by machine learning. We were surprised by the last-minute arrival of this data, which, although of good quality, is not suitable for tensor models. We therefore do not mention them in the "article" section of the report, but present them immediately afterwards.

Contents

Multiway multiblock logistic regression to classify liver tumors from MRI images

Alexandre SELVESTREL

Laboratoire des systèmes, Centrale-Supélec, , Orsay, , Paris, France

Abstract

In this study, tensorial logistic regression [1, 2] is applied to the binary classification of liver tumors, distinguishing between hepatocellular carcinoma (HCC) and cholangiocarcinoma (CCK). This work extends the multiway logistic model presented in [2] by organizing features into separate blocks. The dataset consists of liver MRI images acquired at four distinct time points (arterial, portal, venous, and delayed phases), supplemented by clinical variables. The performance of tensor models is evaluated in comparison with classical logistic regression and group lasso [3], using both liver cancer data and simulated data.

Keywords: multiway data, multiblock data, MRI, tensor

1. Introduction

There are two main types of liver tumor: hepatocellular carcinoma (HCC) and cholangiocarcinoma (CCK). Some tumors even display both CCK and HCC characteristics, depending on the location of the liver observed, and are then said to be mixed. Since the treatment of liver tumors depends on their class, it is important to be able to distinguish them effectively. For the moment, there are two main approaches: microscopy and radiography with contrast injection.

Microscopy is the most reliable method, as it enables tumor cells to be analysed directly. However, since it requires the removal of a small piece of cancerous liver, it requires surgery and can lead to complications for the patient. What's more, it only gives access to a fragment of the liver, which is not necessarily representative of the tumor as a whole. Radiography (by MRI or scanner) with contrast injection, on the other hand, is non-invasive and give access to the entire 3D tumor. As the contrast medium diffuses into the liver, images are taken at four different times (arterial, portal, venous and late) to observe specific features of each phase. However, these images cannot be used to determine the nature of the tumor with the naked eye. Indeed, the characteristics of HCC and CCK tumors are often very similar, and experts do not always agree with each other when analyzing the images.

This article attempts to overcome the limitations of naked-eye analysis by using machine learning. Given the small number of patients studied (around 100) and the requirement for explicability in the medical field, classical machine learning is preferred to deep learning. In

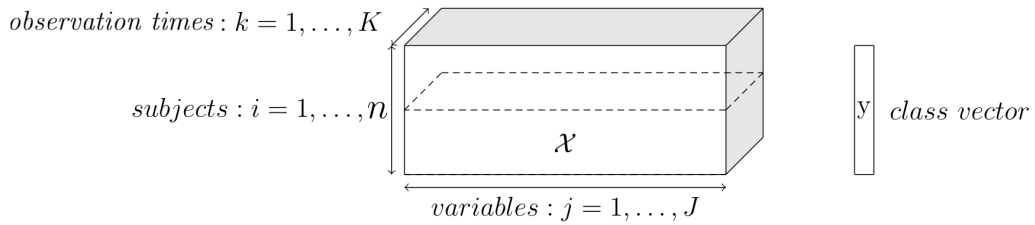


Fig. 1: Example of order 3 tensor from [2]

addition, again because of the small number of patients and in order to simplify the study, mixed tumors are not taken into account. Indeed, these are still poorly understood by doctors, who sometimes even prefer to categorize them as HCC or CCK depending on which aspect predominates in the tumor.

The features of each tumor are extracted from the RMI images using the pyradiomics Python library [4]. For each patient, the tumor is observed on four separate images, taken at specific times corresponding to the different phases of acquisition (arterial, portal, venous and late). The same variables are therefore measured for each of these phases. Thus, for each patient, the features are organized according to a matrix of size $J \times K$ where J is the number of features extracted by pyradiomics in each RMI image and K is the number of acquisition phases. By stacking these matrices one on top of the other, for each individual, we form a tensor of size $n \times J \times K$, where n is the number of individuals studied: we thus speak of tensorial data (Fig:1).

The model on which all the models presented in this article are based is lasso-penalized logistic regression. It allows parsimonious selection of explanatory variables, which is particularly useful when dealing with the large number of features extracted by pyradiomics. In order to take into account the tensorial structure of the data, several models specific to tensorial data, presented in section 2.2, have been implemented. These models are based on the assumption that the tensor structure of the data should be reflected in the regression parameter $\boldsymbol{\beta}$ of the logistic regression [1, 2]. Thus, if the data forms a 3-order tensor as shown in Fig 1, we assume that $\boldsymbol{\beta}$ or at least a portion of $\boldsymbol{\beta}$ can be written as

$$\sum_{r=1}^{R} \boldsymbol{\beta}_r^K \otimes \boldsymbol{\beta}_r^J \tag{1}$$

where "$\otimes$" is the kronecker product, R is the maximum rank allowed for the considered portion of $\boldsymbol{\beta}$ and $\boldsymbol{\beta}_r^K$ and $\boldsymbol{\beta}_r^J$ are vectors.

These models are compared to the group lasso [3], in order to check whether the simple grouping of variables into distinct packets can suffice to capture the relevant relationships between features (or whether it is essential to take full account of the tensor aspect of the data to obtain the best results). They are also compared to classical logistic regression.

Furthermore, in order to assess the benefits of the models presented independently of medical data, their performance is first compared on simulated data.

2. Methodology

2.1. Tensorial data and notations

We designate as tensorial data any data where the explanatory variables are structured along several dimensions. To avoid confusion with the notion of dimension of a vector space we call these dimensions modes in the following. For example, if like in our real data, we measure the same quantities at several fixed times and depths, we say that time and depth are modes in our data. Then, instead of having a matrix of explanatory variables $\mathbf{X} = (x_{ij})_{i\in[\![1,n]\!],j\in[\![1,J]\!]}$ (where i is the individual and j is the quantity of interest), we get a tensor of explanatory variables $\underline{\mathbf{X}} = (x_{ijk_1k_2\ \ldots\ k_M})_{i\in[\![1,n]\!],j\in[\![1,J]\!],k_1\in[\![1,K_1]\!]\ \ldots\ k_m\in[\![1,K_M]\!]}$ (where i is the individual, j is the quantity of interest and where for $m \in [\![1, M]\!]$, k_m is the k_m-th modality of the m-th mode of the data). In terms of notations, we use those of Kolda and Bader [5], especially concerning matricization (see section 2.4 of [5]). However, as some details need to be precised, we do this here:

- The concatenation of two matrices $\mathbf{A}$ and $\mathbf{B}$ by juxtaposing their columns side by side is denoted $[\mathbf{A}\ \mathbf{B}]$.
- To avoid overuse of the symbol ^{T}, we also define a notation to designate the juxtaposition of two matrices one below the other. Thus, the matrix defined by block with $\mathbf{A}$ above $\mathbf{B}$ is denoted $[\mathbf{A}; \mathbf{B}]$. It can also be written $[\mathbf{A}^T\ \mathbf{B}^T]^T$ but this multiplies the ^{T} symbols, which impairs legibility.
- Since vectors are column matrices, using the same notation, we write the concatenation of two vectors $\mathbf{u}$ and $\mathbf{v}$ as follows: $[\mathbf{u}; \mathbf{v}]$.
- The vector (column) whose elements are $(u_i)_{i\in[\![1,I]\!]}$ is denoted $(u_1, u_2, \ \cdots\ u_I)$.
- If $\mathbf{X}$ is a matrix of explanatory variables, $\mathbf{x}_i$ is the vector (column) composed of the i-th row of $\mathbf{X}$.
- The vector of length I filled with 1 is denoted by $\mathbb{1}_I$.
- We denote $\mathrm{Diag}(\mathbf{u})$ the diagonal matrix whose diagonal is the vector $\mathbf{u}$.

2.2. Machine learning models

In this section, we describe all the machine learning methods that we used and compared in order to get our results. We start briefly by non tensorial methods and then we describe in details the tensorial methods that we used. For the sake of simplicity, we only describe the situation where $\underline{\mathbf{X}}$ is a tensor of order 3. However, all the methods described here can be generalized to tensors of any order.

2.2.1. Non tensorial methods

For these methods, we start by unfolding the tensorial data $\underline{\mathbf{X}}$ into the matrix $\mathbf{X}_{(1)} = [\mathbf{X}_{::1} \ \dots \ \mathbf{X}_{::K}]$. We then complete this matrix by concatenating (along the columns) the matrix of non tensorial data $\mathbf{X}_{\text{tab}}$ (where "tab" stands for "tabular"). By doing so we obtain $\mathbf{X}_{\text{tot}} = [\mathbf{X}_{(1)} \ \mathbf{X}_{\text{tab}}]$.

We first train a penalized logistic regression lasso on $\mathbf{X}_{\text{tot}}$. Then, still based on the matrix $\mathbf{X}_{\text{tot}}$, we train a group lasso [3]. In order to make a comparison with tensorial models, we group by variable name or by mode. When the data is structure according to variable blocks, we finally group by block.

2.2.2. Multiway logistic regression with lasso

We now turn to tensor approaches. We start by studying a multiway logistic regression penalized by lasso. This model is described for rank 1 in Le Brusquet et al. [1] and in Girka et al. [2] for its extension to rank $R \in \mathbb{N}^*$. In this report, we directly describe the generalization to rank $R \in \mathbb{N}^*$, rank 1 being a special case of this model.

The fundamental idea of the model is to decompose the parameter $\boldsymbol{\beta}_{\text{tens}} \in \mathbb{R}^{JK}$ associated with the tensor explanatory variables of the logistic regression as:

$$\boldsymbol{\beta}_{\text{tens}} = \sum_{r=1}^{R} \boldsymbol{\beta}_r^K \otimes \boldsymbol{\beta}_r^J \tag{2}$$

with for all $r \in [\![1, R]\!]$, $\boldsymbol{\beta}_r^J \in \mathbb{R}^J$ and $\boldsymbol{\beta}_r^K \in \mathbb{R}^K$. To take account of the M tabular variables (non tensorial), we associate them with a coefficient $\boldsymbol{\beta}_{\text{tab}} \in \mathbb{R}^M$. In this way, the parameter $\boldsymbol{\beta}$ of the logistic regression is written: $[\boldsymbol{\beta}_{\text{tens}}; \boldsymbol{\beta}_{\text{tab}}]$.
As usual with logistic regressions, we consider that each realization of the explained variable y_i ($i \in [\![1, n]\!]$) follows an independent Bernoulli law conditionally on $\mathbf{x_i}$. For logistic regression, this proba is parametrzied by $\boldsymbol{\beta}$ and defined as

$$\mathbb{P}(y_i = 1 \,|\, \mathbf{x}_i) = \frac{1}{1 + \exp(-\mathbf{x}_i^T \boldsymbol{\beta} - \beta_0)} \tag{3}$$

where $\beta_0 \in \mathbb{R}$ is the intercept

We set $\boldsymbol{\beta}^J = \left[\boldsymbol{\beta}_1^J; \ \cdots \ ; \boldsymbol{\beta}_R^J\right]$ and $\boldsymbol{\beta}^K = \left[\boldsymbol{\beta}_1^K; \ \cdots \ ; \boldsymbol{\beta}_R^K\right]$. In order to simplify the calculations, while ensuring that the penalty continues to promote sparse models, we adapt the definition of the lasso penalty. The new penalty defines the following optimization problem:

$$\beta_0, \boldsymbol{\beta}^J, \boldsymbol{\beta}^K, \boldsymbol{\beta}_{\text{tab}} = \underset{\beta_0, \boldsymbol{\beta}^J, \boldsymbol{\beta}^K, \boldsymbol{\beta}_{\text{tab}}}{\text{argmin}} \left[-\sum_{i=1}^{N} \log(\mathbb{P}(y_i = 1 \,|\mathbf{x}_i)) + \lambda \left(\sum_{r=1}^{R} \|\boldsymbol{\beta}_r^K \otimes \boldsymbol{\beta}_r^J\|_1 + \|\boldsymbol{\beta}_{\text{tab}}\|_1 \right) \right] \tag{4}$$

Optimization is performed by alternating directions between $\left[\beta_0; \boldsymbol{\beta}^J; \boldsymbol{\beta}_{\text{tab}}\right]$ and $\left[\beta_0; \boldsymbol{\beta}^K; \boldsymbol{\beta}_{\text{tab}}\right]$.

The stopping criterion is defined by the relative difference between the value of the objective function before optimization in the first direction and the value of the same function after optimization in the second direction. We note that optimizing the loss function in each of these directions is tantamount to performing a simple logistic regression with a lasso penalty. Indeed, if we denote C the loss function of classical logistic regression penalized by lasso (for any $K_0 \in \mathbb{N}^*$):

$$C : \begin{cases} \mathbb{R} \times \mathbb{R}^{K_0} \times \mathbb{R}^{N \times K_0} \times \mathbb{R}^N \times \mathbb{R} & \longrightarrow & \mathbb{R} \\ (\beta_0, \boldsymbol{\beta}, \mathbf{X}, \mathbf{y}, \lambda) & \longmapsto & -\sum_{i=1}^{N} [y_i(\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta}) - \log(1 + \exp(\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta}))] + \lambda \|\boldsymbol{\beta}\|_1 \end{cases} \tag{5}$$

optimizing the overall loss function with respect to $\left[\beta_0; \boldsymbol{\beta}^J; \boldsymbol{\beta}_{\text{uni}}\right]$ amounts to solve

$$\underset{(\beta_0, \boldsymbol{\beta}) \in \mathbb{R} \times \mathbb{R}^{JR+M}}{\operatorname{argmin}} C(\beta_0, (\mathbf{Q}^J)^{-1} \boldsymbol{\beta}, \mathbf{Z}^J \mathbf{Q}^J, \mathbf{y}, \lambda) \tag{6}$$

Where $\mathbf{Q}^J$ and $\mathbf{Z}^J$ are defined as follows:

$$\mathbf{Z}^J = [\mathbf{Z}_1^J \ \cdots \ \mathbf{Z}_R^J \ \ \mathbf{X}_{\text{tab}}] \tag{7}$$

$$\text{where } \forall r \in [\![1, R]\!], \quad \mathbf{Z}_{\mathbf{r}}^{\mathbf{J}} = \sum_{k=1}^{K} (\beta_r^K)_k \, \mathbf{X}_{::k} \quad (\mathbf{Z}_r^J \in \mathbb{R}^{N \times J}) \tag{8}$$

$$\mathbf{Q}^J = \text{Diag}([\|\boldsymbol{\beta}_1^K\|_1^{-1} \mathbb{1}_J; \ \cdots \ ; \|\boldsymbol{\beta}_R^K\|_1^{-1} \mathbb{1}_J; \ \mathbb{1}_M]) \tag{9}$$

Girka et al. [2] demonstrate this result by noting that for $i \in [\![1, n]\!]$,

$$\mathbf{x}_{(1)i}^T \left(\sum_{r=1}^{R} \boldsymbol{\beta}_r^K \otimes \boldsymbol{\beta}_r^J \right) = \sum_{r=1}^{R} \left[(\mathbf{x}_{(1)i}^T \left(\boldsymbol{\beta}_r^K \otimes \mathbf{I}_J \right) \right] \boldsymbol{\beta}_r^J \tag{10}$$

$$= \sum_{r=1}^{R} (\mathbf{z}_r^J)_i^T \boldsymbol{\beta}_r^J \tag{11}$$

and that

$$\sum_{r=1}^{R} \|\boldsymbol{\beta}_r^K \otimes \boldsymbol{\beta}_r^J\|_1 = \|\mathbf{R}_{\text{tens}}^J \boldsymbol{\beta}^J\|_1 \tag{12}$$

$$\text{with} \quad \mathbf{R}_{\text{tens}}^J = \text{Diag}([\|\boldsymbol{\beta}_1^K\|_1 \mathbb{1}_J; \ \cdots \ ; \|\boldsymbol{\beta}_R^K\|_1 \mathbb{1}_J]) \tag{13}$$

Thus,

$$(\mathbf{x}_{\text{tot}})_i^T \boldsymbol{\beta} = (\mathbf{z}_i^J)^T [\boldsymbol{\beta}^J; \boldsymbol{\beta}_{\text{tab}}] \tag{14}$$

$$\text{and} \quad \sum_{i=1}^{N} \|\boldsymbol{\beta}_r^K \otimes \boldsymbol{\beta}_r^J\|_1 + \|\boldsymbol{\beta}_{\text{tab}}\|_1 = \|(\mathbf{Q}^J)^{-1}\boldsymbol{\beta}\|_1 \tag{15}$$

This justifies the previous results

For optimization with respect to $\left[\beta_0; \boldsymbol{\beta}^K; \boldsymbol{\beta}_{\text{tab}}\right]$, the method follows the same steps. The only difference concerns the definition of $\mathbf{Z}^K$. It is:

$$\mathbf{Z}^K = [\mathbf{Z}_1^K \ \cdots \ \mathbf{Z}_R^K \ \ \mathbf{X}_{\text{tab}}] \tag{16}$$

$$\text{with} \forall r \in [\![1, R]\!] \quad \mathbf{Z}_r^K = \sum_{j=1}^{J} (\beta_r^J)_j \mathbf{X}_{:j:} \tag{17}$$

This is justified by:

$$\mathbf{x}_{(1)_i}^T \left(\sum_{r=1}^{R} \boldsymbol{\beta}_r^K \otimes \boldsymbol{\beta}_r^J \right) = \sum_{r=1}^{R} \left[(\mathbf{x}_{(1)_i}^T \left(I_K \otimes \boldsymbol{\beta}_r^J \right) \right] \boldsymbol{\beta}_r^K \tag{18}$$

$$= \sum_{r=1}^{R} (\mathbf{z}_r^K)_i^T \boldsymbol{\beta}_r^K \tag{19}$$

2.2.3. Multiway and multibloc logistic regression with lasso

We now present the lasso-penalized multiway and multiblock logistic regression. This model draws heavily on the multiway logistic regression we have just presented, while also taking into account a block structure of tensor data. More precisely, each of these blocks will have its own independent coefficient $\boldsymbol{\beta}_l$, which was not the case in the previous model. We also allow each block to have its own rank R_l. As tabular quantities are not measured according to several modalities, they are not placed in any particular block. They will be included in the model in the same way as in the multiway case. Mathematically, we define the model as follows:

Let $L \in \mathbb{N}^*$ denote the number of blocks of variables. For any $l \in [\![1, L]\!]$, let d_l be the number of tensorial variables in block l. Thus we have :

$$\sum_{l=1}^{L} d_l = J$$

We reorganize $\underline{\mathbf{X}}$ by grouping together slices $\mathbf{X}_{:j:}$ associated with variables from the same block. More precisely, for all $l \in [\![1, L]\!]$, we call $\underline{\mathbf{X}}^l$ the tensor constituted by the slices $\mathbf{X}_{:j:}$ associated with the l-th block. We then concatenate all these tensors along their second mode (which is the variable name) to obtain the new tensor of explanatory variables: $\underline{\mathbf{X}}'$.

The new $\boldsymbol{\beta}$ structure is defined by blocks. It is:

$$\boldsymbol{\beta} = \left[\sum_{r_1=1}^{R_1} \boldsymbol{\beta}^K_{(1,r_1)} \otimes \boldsymbol{\beta}^J_{(1,r_1)};\ \cdots\ ;\ \sum_{r_L=1}^{R_L} \boldsymbol{\beta}^K_{(L,r_L)} \otimes \boldsymbol{\beta}^J_{(L,r_L)}\ ;\ \boldsymbol{\beta}_{\text{tab}} \right] \tag{20}$$

With for all $l \in [\![1, L]\!]$, we have $r_l \in [\![1, R_l]\!]$, $\boldsymbol{\beta}^J_{(l,r_l)} \in \mathbb{R}^{d_l}$ and $\boldsymbol{\beta}^K_{(l,r_l)} \in \mathbb{R}^K$

We call $\boldsymbol{\beta}^J$ and $\boldsymbol{\beta}^K$ the vectors

$$\boldsymbol{\beta}^J = \left[\boldsymbol{\beta}^J_{(1,1)};\ \cdots\ ;\ \boldsymbol{\beta}^J_{(1,R_1)};\ \cdots\ \cdots\ ;\ \boldsymbol{\beta}^J_{(L,1)}\ \cdots\ ;\ \boldsymbol{\beta}^J_{(L,R_L)} \right] \tag{21}$$

$$\boldsymbol{\beta}^K = \left[\boldsymbol{\beta}^K_{(1,1)};\ \cdots\ ;\ \boldsymbol{\beta}^K_{(1,R_1)};\ \cdots\ \cdots\ ;\ \boldsymbol{\beta}^K_{(L,1)}\ \cdots\ ;\ \boldsymbol{\beta}^K_{(L,R_L)} \right] \tag{22}$$

In a similar way to what is done in the multiway model, we adapt the lasso penalty, so that the new optimization problem becomes:

$$\beta_0, \boldsymbol{\beta}^J, \boldsymbol{\beta}^K, \boldsymbol{\beta}_{\text{tab}} = \underset{\beta_0, \boldsymbol{\beta}^J, \boldsymbol{\beta}^K, \boldsymbol{\beta}_{\text{tab}}}{\operatorname{argmin}} \left(-\sum_{i=1}^{N} \log(\mathbb{P}(y_i = 1 \,|\mathbf{x}_i)) + \sum_{l=1}^{L} \sum_{r_l=1}^{R_l} \|\boldsymbol{\beta}^K_{(l,r_l)} \otimes \boldsymbol{\beta}^J_{(l,r_l)}\|_1 + \|\boldsymbol{\beta}_{\text{tab}}\|_1 \right) \tag{23}$$

Once again, this problem is solved by alternating optimization directions $\left[\beta_0; \boldsymbol{\beta}^J; \boldsymbol{\beta}_{\text{tab}}\right]$ and $\left[\beta_0; \boldsymbol{\beta}^K; \boldsymbol{\beta}_{\text{tab}}\right]$. Each of these two problems can be reduced to a lasso-penalized classical logistic regression

Indeed, optimizing according to $\left[\beta_0; \boldsymbol{\beta}^J; \boldsymbol{\beta}_{\text{tab}}\right]$ is equivalent to searching

$$\underset{(\beta_0, \boldsymbol{\beta})}{\operatorname{argmin}}\ C(\beta_0, (\mathbf{Q}^J)^{-1}\boldsymbol{\beta}, \mathbf{Z}^J\mathbf{Q}^J, \mathbf{y}, \lambda) \tag{24}$$

Where $\mathbf{Q}^J$ and $\mathbf{Z}^J$ are defined as follows:

$$\mathbf{Z}^J = [\mathbf{Z}^J_{(1,1)}\ \cdots\ \mathbf{Z}^J_{(1,R_1)}\ \cdots\ \cdots\ \mathbf{Z}^J_{(L,1)}\ \cdots\ \mathbf{Z}^J_{(L,R_L)}\ \mathbf{X}_{\text{tab}}] \tag{25}$$

$$\text{where } \forall r_l \in [\![1, R_l]\!], \quad \mathbf{Z}^J_{(l,r_l)} = \sum_{k=1}^{K} \left(\beta^K_{(l,r_l)}\right)_k \mathbf{X}^l_{::k} \quad \left(\mathbf{Z}^J_{(l,r_l)} \in \mathbb{R}^{n \times d_l}\right) \tag{26}$$

$$\mathbf{Q}^J = \text{Diag}([\|\boldsymbol{\beta}^K_{(1,1)}\|_1^{-1}\mathbb{1}_{d_1};\ \cdots\ ;\|\boldsymbol{\beta}^K_{(1,R_1)}\|_1^{-1}\mathbb{1}_{d_1};\ \cdots\ \cdots\ ;\ \|\boldsymbol{\beta}^K_{(L,1)}\|_1^{-1}\mathbb{1}_{d_L};\ \cdots\ ;\ \|\boldsymbol{\beta}^K_{(L,R_L)}\|_1^{-1}\mathbb{1}_{d_L};\ \mathbb{1}_M]) \tag{27}$$

The demonstration of this result is similar to that of the multiway case. Indeed, we note

that

$$(\mathbf{x}'_{(1)})_i^T \left[\sum_{r_1=1}^{R_1} \boldsymbol{\beta}^K_{(1,r_1)} \otimes \boldsymbol{\beta}^J_{(1,r_1)};\ \cdots\ ; \sum_{r_L=1}^{R_L} \boldsymbol{\beta}^K_{(L,r_L)} \otimes \boldsymbol{\beta}^J_{(L,r_L)} \right] = \sum_{l=1}^{L} \sum_{r_l=1}^{R_l} \left(\mathbf{x}^l_{(1)}\right)_i^T \left(\boldsymbol{\beta}^K_{(l,r_l)} \otimes \boldsymbol{\beta}^J_{(l,r_l)}\right) \tag{28}$$

$$= \sum_{l=1}^{L} \sum_{r_l=1}^{R_l} \left[\left(\mathbf{x}^l_{(1)}\right)_i^T \left(\boldsymbol{\beta}^K_{(l,r_l)} \otimes I_{d_l}\right) \right] \boldsymbol{\beta}^J_{(l,r_l)} \tag{29}$$

$$= \sum_{l=1}^{L} \sum_{r_l=1}^{R_l} \left(\mathbf{z}^J_{(l,r_l)}\right)_i^T \boldsymbol{\beta}^J_{(l,r_l)} \tag{30}$$

And that

$$\sum_{l=1}^{L} \sum_{r_l=1}^{R_l} \|\boldsymbol{\beta}^K_{(l,r_l)} \otimes \boldsymbol{\beta}^J_{(l,r_l)}\|_1 = \|\mathbf{R}^J_{\text{tens}} \boldsymbol{\beta}^J\|_1 \tag{31}$$

$$\text{with} \quad \mathbf{R}^J_{\text{tens}} = \text{Diag}([\|\boldsymbol{\beta}^K_{(1,1)}\|_1 \mathbb{1}_{d_1};\ \cdots\ ; \|\boldsymbol{\beta}^K_{(1,R_1)}\|_1 \mathbb{1}_{d_1};\ \cdots\ \cdots\ ; \|\boldsymbol{\beta}^K_{(L,1)}\|_1 \mathbb{1}_{d_L};\ \cdots\ ; \|\boldsymbol{\beta}^K_{(L,R_L)}\|_1 \mathbb{1}_{d_L};\ \mathbb{1}_M]) \tag{32}$$

We deduce that

$$[\mathbf{x}'_{(1)_i};\ \mathbf{x}_{\text{tab}_i}]\, \boldsymbol{\beta} = (\mathbf{z}^J_i)^T [\boldsymbol{\beta}^J;\ \boldsymbol{\beta}_{\text{tab}}] \tag{33}$$

$$\text{and} \quad \sum_{l=1}^{L} \sum_{r_l=1}^{R_l} \|\boldsymbol{\beta}^K_{(l,r_l)} \otimes \boldsymbol{\beta}^J_{(l,r_l)}\|_1 + \|\boldsymbol{\beta}_{\text{uni}}\|_1 = \|(\mathbf{Q}^J)^{-1} \boldsymbol{\beta}\|_1 \tag{34}$$

Wich justifies the previous results.

For optimization with respect to $\left[\beta_0;\ \boldsymbol{\beta}^K;\ \boldsymbol{\beta}_{\text{tab}}\right]$, the method is analogous. The only difference concerns the form of $\mathbf{Z}^K$. It is written as:

$$\mathbf{Z}^K = [\mathbf{Z}^K_{(1,1)}\ \cdots\ \mathbf{Z}^K_{(1,R_1)}\ \cdots\ \cdots\ \mathbf{Z}^K_{(L,1)}\ \cdots\ \mathbf{Z}^K_{(L,R_L)}\ \mathbf{X}_{\text{tab}}] \tag{35}$$

$$\text{where } \forall r_l \in [\![1, R_l]\!], \quad \mathbf{Z}^K_{(l,r_l)} = \sum_{j=1}^{d_l} \mathbf{X}^l_{:j:} \left(\beta^J_{(l,r_l)}\right)_j \quad \left(\mathbf{Z}^K_{(l,r_l)} \in \mathbb{R}^{n \times K}\right) \tag{36}$$

The justification of that last result is analogous to the one used in the multiway case.

Pseudo-code:

In order to clarify the algorithm that we use, we give here the pseudo-code of our implementation

Inputs

- $\epsilon > 0$, $\lambda > 0$, $R \in \mathbb{N}^*$
- $\boldsymbol{\beta}^{K(0)} \in \mathbb{R}^{LRK}$

Treatment

- $q \leftarrow 0$

Repeat

- Construct $\mathbf{Z}^J$ according to eqs. (25) and (26)
- Construct $\mathbf{Q}^J$ according to eq. (27)
- $(\beta_0^{(q)}, \boldsymbol{\beta}^{J(q)}) \longleftarrow \underset{(\beta_0,\boldsymbol{\beta})\in\mathbb{R}\times\mathbb{R}^{RJ+M}}{\operatorname{argmin}} (C(\beta_0, (\mathbf{Q}^J)^{-1}\boldsymbol{\beta}, \mathbf{Z}^J\mathbf{Q}^J, \mathbf{y}, \lambda))$
- Construct $\mathbf{Z}^K$ according to eqs. (35) and (36)
- Construct $\mathbf{Q}^K$ by adapting eq. (27)
- $(\beta_0^{(q)}, \boldsymbol{\beta}^{K(q)}) \longleftarrow \underset{(\beta_0,\boldsymbol{\beta})\in\mathbb{R}\times\mathbb{R}^{LRK+M}}{\operatorname{argmin}} (C(\beta_0, (\mathbf{Q}^K)^{-1}\boldsymbol{\beta}, \mathbf{Z}^K\mathbf{Q}^K, \mathbf{y}, \lambda))$
- $q \leftarrow q + 1$

until $|C^K - C^J| < \epsilon |C^J|$

Return $(\beta_0^{(q)}, \boldsymbol{\beta}^{K(q)}, \boldsymbol{\beta}^{J(q)})$

Notes:

- The worst-case complexity of the tensor algorithms presented here is $O(J+K+R)$, compared with $O(JK)$ for non-tensor algorithms. So, if J and K are large and supposing $R \ll \min(J, K)$ then tensor algorithms are more efficient, as shown in [2].
- With the multiway multiblock model, we can deal with the case where each block is a tensor of different order. All we need to do is optimize several times according to the same $\boldsymbol{\beta}$ mode in blocks with fewer modes than the others.
- We decided to optimize the loss function completely in one direction before turning to the other one instead of alternating one step in each direction because the first procedure was more stable and could be implemented efficiently using the glmnet package in R [6].

2.3. Simulated data generation

To test our multiway, multiblock model, we perform tests on simulated data. In this section, we explain how we generate this data.

2.3.1. Regression parameter structure

We have structured our simulated data into several blocks and modes. This enables us to compare the performance of the multiblock multiway model with other logistic models in a

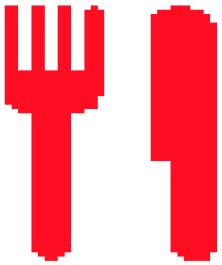


Fig. 2: Example of the pictogram used to generate $\boldsymbol{\beta}$.

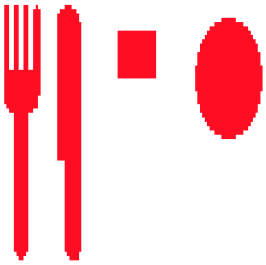


Fig. 3: Example of pictogram concatenation used to generate β.

setting where the data has exactly the form predicted by the multiblock multiway model.

The multiway and multiblock aspect of our most advanced model is reflected in its regression parameter $\boldsymbol{\beta}$. This is why we have chosen to generate our data in such a way that the optimal regression parameter $\boldsymbol{\beta}_{\text{opti}}$ (i.e. minimizing the classification error) has a multiblock multiway structure. To make the reconstruction of the regression parameter as visual as possible, we reused the method presented in [7]. Thus, $\boldsymbol{\beta}_{\text{opti}}$ is in fact composed exclusively of 0 and 1. The 1 are arranged to form simple geometric patterns when the beta vector is split into several lines (Fig: 2). The result is $\boldsymbol{\beta}_{\text{opti}}$ in the form of a second-order tensor, each column of which is associated with a different explanatory variable and each row with a different observation modality. As pictograms are simple, the rank of the tensor is expected to be low in relation to the number of variables and modalities.

To add a multiblock aspect to $\boldsymbol{\beta}_{\text{opti}}$, instead of choosing just one pictogram, we consider the columnar concatenation of several pictograms (Fig: 3). Thus, each pictogram, seen as a 2nd-order tensor, is of low rank, but the concatenation of several pictograms produces a tensor of higher rank. It is this concatenation which, after being unfolded into a single line, constitutes $\boldsymbol{\beta}_{\text{opti}}$. This renders the single multiway logistic regression model less relevant (which will need to have a high rank to correctly reconstruct $\boldsymbol{\beta}_{\text{opti}}$), without putting the multiway multiblock model at a disadvantage (which will be able to separate $\boldsymbol{\beta}_{\text{opti}}$ into several tensors of lower rank: one per pictogram).

2.3.2. Generation of explanatory variables

The method generally used to simulate explanatory variables in regression models is to use a simple probability distribution (often the standardized normal distribution), identical for all individuals. The explained variable is then obtained by applying the regression model with $\boldsymbol{\beta} = \boldsymbol{\beta}_{\text{opti}}$ to the explanatory variables. This is, for example, what is proposed in [7]. However, this method poses a problem in binary classification, as we have no control over the number of individuals in each class. It is always possible to work by trial and error (generating one $\boldsymbol{\beta}_{\text{opti}}$ and then verifying whether it is possible to extract a balanced subset of the generated data of the desired size. If not, generating a new $\boldsymbol{\beta}_{\text{opti}}$ and so on), but it is inefficient.

To overcome this difficulty, we decided to generate the explanatory variables differently, by correlating them with the individual's class. More precisely, for each individual class, we

chose to generate the explanatory variables according to a multivariate normal distribution. The two classes have the same covariance matrix, but different means. These means and covariance matrices are chosen to ensure that $\boldsymbol{\beta}_{\text{opti}}$ is indeed the normal vector to the best class-separation hyperplane. To prove this, we will demonstrate that the method used ensures that this hyperplane is the Bayes classifier minimizing the classification error for the simulated data.

Proposition 1. *Noting respectively $\boldsymbol{\mu}_0$ and $\boldsymbol{\mu}_1$ the mean vectors of the n explanatory variables of the two classes and $\boldsymbol{\Sigma}$ the covariance matrix of these same variables, if we impose*

$$\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0 /\!\!/ \beta_{\text{opti}} \tag{37}$$

$$\boldsymbol{\Sigma} = \mathbf{PDP}^T \quad \text{with } \mathbf{P} \in \mathcal{O}(n) \tag{38}$$

$$\text{the first column of } \mathbf{P} \text{ is colinear to } \boldsymbol{\beta}_{\text{opti}} \tag{39}$$

then the decision frontier of the Bayes estimator minimizing the classification error is a hyperplane with normal vector β_{opti}.

Proof.
In a binary classification, the g^* Bayes estimator that minimizes the error is:

$$g^* : \begin{cases} \mathbb{R}^n \longrightarrow \{0,1\} \\ \mathbf{x} \longmapsto \begin{cases} 1 & \text{if } E[Y|X=\mathbf{x}] \geq 0.5 \\ 0 & \text{else} \end{cases} \end{cases} \tag{40}$$

Given that X and Y admit densities with respect to the lebesgue measure and the counting measure respectively, we have:

$$E(Y|X=\mathbf{x}) = \frac{1}{f_X(\mathbf{x})} \int y f_{(X,Y)}(\mathbf{x}, y) dy \tag{41}$$

Since Y admits a density with respect to the counting measure, this integral can be rewritten:

$$E(Y|X=\mathbf{x}) = \frac{1}{f_X(\mathbf{x})} \sum_{y \in \{0,1\}} y f_{(X,Y)}(\mathbf{x}, y) \tag{42}$$

And therefore

$$E(Y|X=\mathbf{x}) = \frac{f_{(X,Y)}(\mathbf{x}, y=1)}{f_X(\mathbf{x})} \tag{43}$$

Which means

$$E(Y|X=\mathbf{x}) = \frac{f_{(X|Y)}(\mathbf{x}|y=1)P(Y=1)}{f_{X|Y}(\mathbf{x}|y=1)P(Y=1) + f_{X|Y}(\mathbf{x}|y=0)P(Y=0)} \tag{44}$$

Now, by hypothesis, we know that for $y \in \{0,1\}$, $f_{X|Y}(\,.\,|y)$ is the density of $\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma})$. Also, $P(Y=1)$ and $P(Y=0)$ correspond exactly to the proportion of individuals generated in each class and are therefore known. For the sake of simplicity, let's note: $P(Y=1) = p_1$ and $P(Y=0) = p_0$. Consequently

$$E(Y|X=\mathbf{x}) \geq \frac{1}{2} \tag{45}$$

$$\iff \frac{p_1 \exp\left(-\frac{(\mathbf{x}-\boldsymbol{\mu}_1)\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_1)}{2}\right)}{p_1 \exp\left(-\frac{(\mathbf{x}-\boldsymbol{\mu}_1)\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_1)}{2}\right) + p_0 \exp\left(-\frac{(\mathbf{x}-\boldsymbol{\mu}_0)\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_0)}{2}\right)} \geq \frac{1}{2} \tag{46}$$

$$\iff \frac{1}{1+\frac{p_0}{p_1}\exp\left(-\frac{(\mathbf{x}-\boldsymbol{\mu}_0)\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_0)}{2} + \frac{(\mathbf{x}-\boldsymbol{\mu}_1)\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_1)}{2}\right)} \geq \frac{1}{2} \tag{47}$$

$$\iff \frac{(\mathbf{x}-\boldsymbol{\mu}_0)\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_0)}{2} - \frac{(\mathbf{x}-\boldsymbol{\mu}_1)\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_1)}{2} \geq \log\left(\frac{p_0}{p_1}\right) \tag{48}$$

Since $\boldsymbol{\Sigma}^{-1}$ is positive symmetric, we can associate it with the positive semidefinite bilinear form it induces, which we denote $\langle\,.\,,\,.\,\rangle_{\Sigma^{-1}}$. Thus:

$$E(Y|X=\mathbf{x}) \geq \frac{1}{2} \tag{49}$$

$$\iff \langle \mathbf{x}-\boldsymbol{\mu}_0, \mathbf{x}-\boldsymbol{\mu}_0\rangle_{\Sigma^{-1}} + \langle -\mathbf{x}+\boldsymbol{\mu}_1, \mathbf{x}-\boldsymbol{\mu}_1+\boldsymbol{\mu}_0-\boldsymbol{\mu}_0\rangle_{\Sigma^{-1}} \geq 2\log\left(\frac{p_0}{p_1}\right) \tag{50}$$

$$\iff \langle \mathbf{x}-\boldsymbol{\mu}_0, \mathbf{x}-\boldsymbol{\mu}_0\rangle_{\Sigma^{-1}} + \langle -\mathbf{x}+\boldsymbol{\mu}_1, \mathbf{x}-\boldsymbol{\mu}_0\rangle_{\Sigma^{-1}} + \langle -\mathbf{x}+\boldsymbol{\mu}_1, \boldsymbol{\mu}_0-\boldsymbol{\mu}_1\rangle_{\Sigma^{-1}} \geq 2\log\left(\frac{p_0}{p_1}\right) \tag{51}$$

$$\iff \langle \boldsymbol{\mu}_1-\boldsymbol{\mu}_0, \mathbf{x}-\boldsymbol{\mu}_0\rangle_{\Sigma^{-1}} - \langle \boldsymbol{\mu}_1-\mathbf{x}, \boldsymbol{\mu}_1-\boldsymbol{\mu}_0\rangle_{\Sigma^{-1}} \geq 2\log\left(\frac{p_0}{p_1}\right) \tag{52}$$

$$\iff \langle 2\mathbf{x}-\boldsymbol{\mu}_0-\boldsymbol{\mu}_1, \boldsymbol{\mu}_1-\boldsymbol{\mu}_0\rangle_{\Sigma^{-1}} \geq 2\log\left(\frac{p_0}{p_1}\right) \tag{53}$$

$$\iff \mathbf{x}^T\mathbf{P}\mathbf{D}^{-1}\mathbf{P}^T(\boldsymbol{\mu}_1-\boldsymbol{\mu}_0) \geq \log\left(\frac{p_0}{p_1}\right) + \frac{1}{2}\langle \boldsymbol{\mu}_0+\boldsymbol{\mu}_1, \boldsymbol{\mu}_1-\boldsymbol{\mu}_0\rangle_{\Sigma^{-1}} \tag{54}$$

$$\tag{55}$$

By hypothesis, the first column of $\mathbf{P}$ is collinear with $\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0$. We denote $\mathbf{v}$ this column and λ the real such that $\mathbf{v} = \lambda(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)$. Since $\mathbf{P}$ is orthogonal, all its other columns are

orthogonal to $\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0$. We therefore have, noting d_1 the first real of the diagonal of $\mathbf{D}$:

$$\mathbf{x}^T\mathbf{P}\mathbf{D}^{-1}\mathbf{P}^T(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) = (\mathbf{x}^T\mathbf{v} \;\; 0 \;\; 0 \;\; \cdots \;\; 0)\mathbf{D}^{-1}\begin{pmatrix}\mathbf{v}^T(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)\\ 0\\ 0\\ \vdots\\ 0\end{pmatrix} \tag{56}$$

$$= \lambda^2\mathbf{x}^T(\boldsymbol{\mu_1} - \boldsymbol{\mu_0})\, d_1^{-1}(\boldsymbol{\mu_1} - \boldsymbol{\mu_0})^T(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) \tag{57}$$

And therefore

$$E(Y|X = \mathbf{x}) \geq \frac{1}{2} \tag{58}$$

$$\iff \mathbf{x}^T(\boldsymbol{\mu_1} - \boldsymbol{\mu_0}) \geq \frac{d_1}{\lambda^2\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0\|^2}\log\left(\frac{p_0}{p_1}\right) + \frac{d_1}{2\lambda^2\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0\|^2}\langle\boldsymbol{\mu}_0 + \boldsymbol{\mu}_1, \boldsymbol{\mu}_1 - \boldsymbol{\mu}_0\rangle \tag{59}$$

$$\tag{60}$$

By hypothesis, $\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0 /\!\!/ \beta_{\text{opti}}$. Since the term on the right is independent of $\mathbf{x}$, the decision frontier of the Bayes classifier is indeed a hyperplane with normal vector $\boldsymbol{\beta}_{\text{opti}}$.

The hyperparameters used to generate the simulated variables are presented in appendix Appendix A. These are chosen experimentally to enable our models to reconstruct $\boldsymbol{\beta}_{\text{opti}}$ without a simple 2-means algorithm being able to separate them (see appendix Appendix A). The projection onto the plane of the first two principal components of the explanatory variables is shown in figure 4. It shows that the classes are difficult to separate with the naked eye.

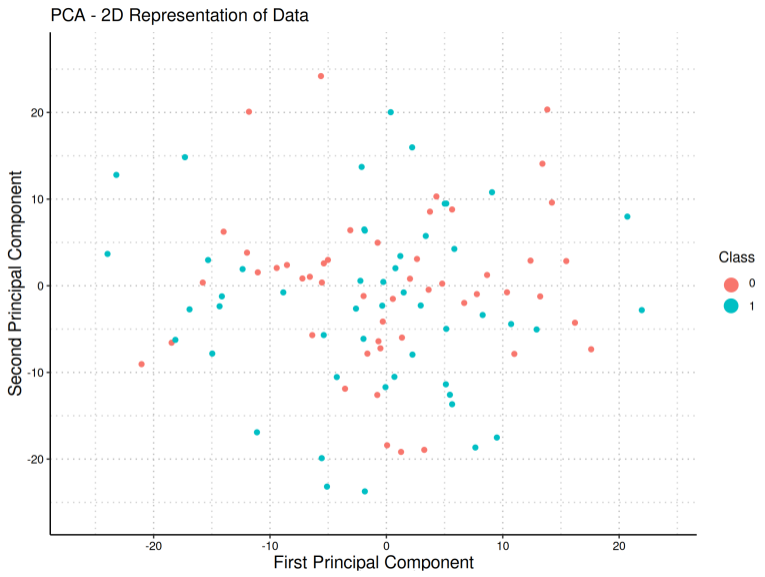


Fig. 4: Plane projection of the first two principal components of the explanatory variables simulated for 100 individuals when $\boldsymbol{\beta}_{\text{opti}}$ is given by the concatenation of pictograms in Fig: 3

3. Real dataset

3.1. Presentation of real data

The actual data on which we are working comes from a cohort of 145 patients with liver tumors. 86 of them have HCC tumors, 22 have CCK tumors and 37 have mixed tumors. These proportions reflect the actual proportions of the different tumor classes in liver cancer patients. Each patient underwent four MRI radiographs of the liver, one at each time point after contrast injection. These were arterial, portal, venous and late. However, not all MRIs are usable. The patient may move during the MRI, rendering it unusable. A summary table (Table 1) is provided in order to specify the number of usable MRIs by temporality.

Clinical data are also available: age at tumor detection, gender and patient alpha feto-protein (AFP) levels. However, since the AFP levels of 22% patients are missing from the data, we decided to exclude this clinical variable. As the gender of some patients (one with a HCC tumor, the other with a CCK tumor) was unknown, they were previously excluded from the study. The figures presented here and in the summary table Table 1 show only those patients for whom we know the age at which the tumor was diagnosed and the gender.

On each of the MRI, the tumor area is displayed and saved as a mask superimposed on the MRI. The MRIs and masks are in .nii format. Although taken at four different times, the four MRIs are very similar. In particular, the MRIs at venous and late time are extremely similar and often redundant in the eyes of radiologists. We'll take this opportunity to eliminate the venous time MRIs, as this is the time for which there are the most missing MRIs. We propose two possible extractions for features. A 3D extraction, where features are extracted from the entire tumor volume, and a 2D extraction, where features are extracted from each tumor section. These extractions are the result of a calibration in which we used

Table 1: Number of patients with usable MRI at the times indicated in the column for each tumor class. The total number of patients with each tumor class is entered in the total column.

class	Arterial	Portal	Venous	Late	All times	all times except venous	total
HCC	84	81	83	78	72	74	86
CCK	18	18	14	18	12	16	19
Mixtes	35	36	32	34	29	31	37

the performance of a lasso-penalized logistic model as a reference (to know which features to add or remove).

As previously mentioned, we will only study the distinction between HCC and CCK tumors, which allows us to directly use the binary classification models described in the "Machine learning models" section.(2.2).

3.2. feature extraction in 3D

We use the pyradiomics package [4] to extract an array of 3D features for each tumor. Only the original (unfiltered) image is used to extract these features. We extract all the first-order parameters (relative to gray levels), 3D shape parameters (volume, surface, etc.), and texture parameters (based on co-occurrence matrix, gradient matrix, etc.) proposed by the package (except those considered deprecated or duplicative: for example, we eliminate glcm joint average as it is redundant with glcm sum average). The result is 106 features for each radio. Shape parameters are averaged over all extracted temporalities, as we consider that the shape of a tumor has no reason to change between different MRIs.

The exact parameters used for pyradiomics extraction are given in appendix Appendix B. They were recommended by ... . To ensure that the extraction is consistent from one tumor to the next, all tumors have been resampled to the same scale. On each (x, y, z) axis, the spacing used is half the median spacing on that axis (calculated over all available MRIs). The idea behind this spacing is to avoid losing too much information by increasing the voxel size of higher-resolution MRIs without having to completely interpolate lower-resolution MRIs. Image interpolations are performed using cubic splines, while mask interpolations are based on the closest interpolation method (to guarantee mask connectivity).

The advantage of 3D extraction is that each MRI image is summarized in a relatively small number of features (compared with 2D extraction). What's more, since the parameters are calculated on the tumor in its entirety, they do not omit any part of it. This idea is confirmed by the lack of improvement in the performance of the logistic lasso regression when features from the 2D extraction are added to the 3D parameters: the 3D features seem to stand on their own. The weakness of the 3D extraction is that it requires a complete segmentation by the radiologist of every tumor in the training database, which is very time-consuming.

3.3. Feature extraction in 2D

The first step in this extraction process is to determine the slices we wish to extract from the tumor. We choose the axial plane for the slices, as this is the one used by radiologists

when analyzing a tumor. As for the extraction parameters, they are again given in Appendix Appendix B. However, we can't simply extract slices at regular intervals along the vertical axis, for two reasons:

Firstly, tumor size varies from patient to patient. Thus, a certain spacing between slices will lead to the extraction of 3 slices of tumors in some patients and 10 slices in others. However, the machine learning models we use need to compare the same features in all patients. Secondly, slices with a very small piece of tumor are not very significant for our analysis. However, extracting at regular intervals will lead to the extraction of such slices at the beginning and end of certain elongated tumors (along vertical axis). We'd therefore like to give more importance to slices where the tumor is most present (without completely ignoring slices with less tumor on them).

We therefore propose an extraction where we first specify the number of slices n_{slices} to be extracted from each tumor. We begin by interpolating the cumulative distribution of tumor volume by depth (along the vertical axis) for each tumor (see Fig. 5). This curve is then inverted to obtain the depth distribution as a function of the cumulative tumor volume covered. A slice is then extracted at each of the following depths:

$$(i - 0.5)\frac{\text{area}_{\max}}{n_{\text{slices}}} \qquad \text{for i } \in [\![1, n_{\text{slices}}]\!] \tag{61}$$

We have experimented other extraction methods, in particular trying to extract precisely the same depths for each MRI of the same tumor, while taking into account the fact that the patient may have moved slightly between two MRIs. However, as the results were of lesser quality, we will not develop these approaches here.

The feature extraction used for each slice is almost the same as the one used for the 3D tumor (in the previous section), except for shape parameters. In fact, 2D shape parameters (instead of 3D shape parameters) are now extracted by pyradiomics. 2D shape parameters are always averaged over all MRIs of the same tumor (variations in tumor shape between MRIs result solely from changes in the way radiologists cut masks, and therefore do not provide information on the tumor itself).

The advantage of this type of extraction is that the radiologist may only needs to segment a limited number of slices that are "representative" of the tumour, instead of the whole tumor. Indeed, the work carried out to find the right slices to extract could be replaced by an estimate made by the radiologist's naked eye. Although it would be necessary to check that the results are not affected by this change, this method seems simpler to generalize to large datasets (the segmentation of a couple of slices being less time consuming for a radiologist than the segmentation of the whole tumor). The disadvantage of this extraction method is that it loses the information contained in the slices that were not selected, as well as that concerning the overall shape of the tumor (which cannot be reduced to the shape observed on a few slices). This results in a slightly poorer performance of all models on these data 4.

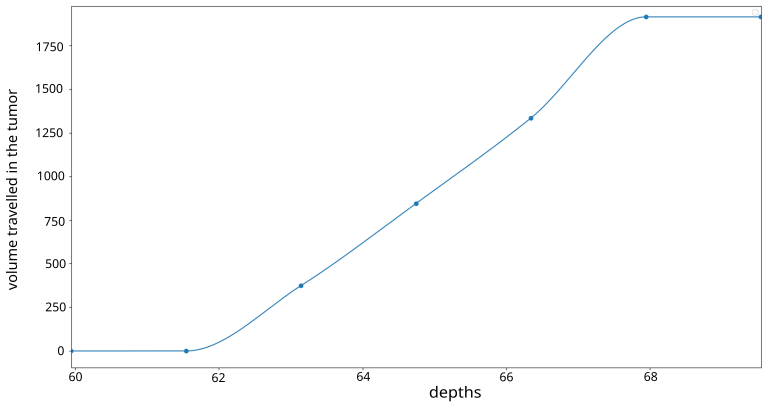


Fig. 5: graph of the cumulative volume distribution (in mm^3) of the third CCK patient's tumor according to depth (in mm) for a given tumor. The points correspond to the slices recorded in the sitk image (with its initial spacing). The curve is obtained by interpolating these points using Hermite cubic splines.

3.4. Extraction of healthy liver parts

We wanted to add the features obtained by performing the extraction on portions of healthy liver. Radiologists generally compare the luminosity of the tumor area with the rest of the liver, so it seemed appropriate to do the same with our model.

To do this, a small strip of tissue was extracted around the tumor area. To ensure that no area outside the liver or crossed by a blood vessel was included, we decided to extract only areas of low local variance and whose luminosity was greater than that of the black background. By adding a 3D connectivity criterion, we can extract a 3D area of healthy liver large enough to perform a 3D extraction of firstorder and texture features (the shape of the extracted area being of no interest).

To ensure that the same area of healthy tissue was extracted from each MRI of the same tumor, we decided to crop the healthy tissue on the late MRI only (this was when our extraction method was most visually successful). We then applied the same trimming to the other MRIs, shifting the extracted area slightly to take account of the patient's movements. These movements were estimated by comparing the tumor areas on each slice and trying to increase as much as possible the intercorrelation of the area curves between each MRI and the late MRI. This procedure is performed along all three spatial axes. Finally, the extracted zone can be visualized, as in Fig. 6 to check that the extraction is proceeding correctly.

However, we did not perceive any improvement in the performance of our models by adding these features. We therefore decided not to include them in the rest of our study.

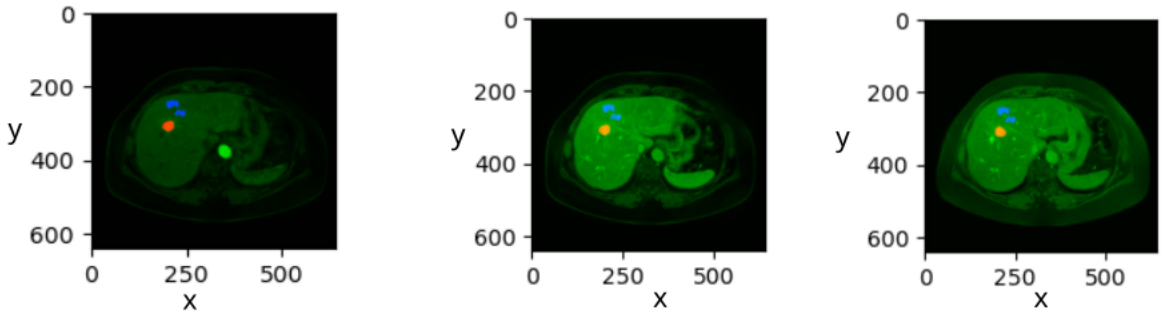


Fig. 6: MRIs of a CCK tumor slice with the tumor area in red and the peripheral area of extracted healthy liver in blue. From left to right, arterial, portal and late MRIs. Axes are graduated in mm.

4. results

4.1. Simulated data

We performed tests on simulated data generated with the parameters presented in Appendix A. The pictogram to retrieve is the one in figure 3. We evaluated the performance in two settings: one with a lot of individuals to classify (3000 individuals in the training data set) and another with fewer individuals to classify (500 individuals in the training dataset). The testing dataset is always composed of 1000 individuals. We consider the area under curve to determine the performance of each model as it is a robust and fine grained indicator of the efficience of each model to separate the two classes. We show the results in Table 2. For each model, the hyperparameters used during cross-validation are provided in Appendix A. In each case, we present the pictogram reconstructed in Appendix C

We can see that, in the presence of a large number of individuals (3,000), the multiway multiblock model is the most successful at reconstructing pictograms. In terms of performance, measured by the area under curve, its performance in this setting is similar to that of the multiway model. The other models (group lasso and classical logistic regression with lasso) perform much less well than the two tensor models. In the presence of a small number of individuals (500), the multiway model performed best. In fact, it captures the pictogram structure in a very small number of parameters (the rank chosen by cross validation is equal to 1 in this configuration). Even imposing a rank of 1 on the multiway multiblock model requires the calculation of more coefficients (around 2 times more in the case of our pictograms), which can lead to greater over-interpretation.

These simulations clearly demonstrate the usefulness of tensor models for classifying high-dimensional data. They show that the structure of the β coefficient is more easily found by the multiway multiblock model (when β has a block structure, as is the case for our pictograms). However, results also indicate that the multiway model performs better than the multiway multiblock model in classification tasks when working with small datasets.

Table 2: Area under curve for each model on simulated data. For the group lasso model, it is possible to group variables by block, mode or variable. The type of grouping used is indicated in brackets

number of individuals	lasso	goup lasso (by block)	goup lasso (by mode)	group lasso (by variable)	multiway	multiway multibloc
3000	0.83	0.86	0.94	0.94	0.99	0.99
500	0.59	0.65	0.64	0.63	0.92	0.63

Table 3: Average area under curve obtained with each model on real data for 50 different trainings (with different partition between training set and testing set). For the group lasso model, it is possible to group variables by block, mode or variable. The type of grouping used is indicated in brackets. The confidence intervals provided are normal-based at a 95% confidence level.

Type of data	lasso	goup lasso (by block)	goup lasso (by time)	group lasso (by variable)	multiway	multiway multibloc
3D	0.74 ± 0.04	0.78 ± 0.03	0.76 ± 0.03	0.73 ± 0.03	0.77 ± 0.03	0.77 ± 0.03

Area under curve (AUC) on 3D real data

Type of data	lasso	goup lasso (by block)	goup lasso (by slice)	group lasso (by time)	group lasso (by variable)	multiway	multiway multibloc
2D	0.73 ± 0.03	0.71 ± 0.03	0.70 ± 0.04	0.71 ± 0.03	0.71 ± 0.03	0.66 ± 0.04	0.71 ± 0.03

Area under curve (AUC) on 2D real data

4.2. real data

We carried out tests on simulated data both using the 3D extraction (as described in 3.2) and the 2D extraction (as described in 3.3). Results are presented for each model in table 3. All results are averaged over 50 different training sessions. An analysis of the importance of each feature of the data studied is also proposed in Appendix D

The performance obtained on medical data are not good enough for our models to be used in real conditions. In particular, in the course of our tests, we found that no model achieved an accuracy of better than 50% in the detection of HCC tumors. Overall, 3D data give better results than 2D data. This may be explained by the fact that 2D data do not take into account the entire tumor. Experiments combining 2D and 3D parameters have been carried out, but have never exceeded the performance of 3D data alone.

With both extraction procedures, the multiway multiblock model performs well compared to other models, but never better than all the studied non-tensorial models. This indicates that the structure of the optimal $\boldsymbol{\beta}$ parameter is likely not tensorial. Indeed, a simple grouping of features into groups (enabled by the group lasso), is sufficient to obtain similar results. In terms of computation time, the multiblock model is the most time-consuming, as it requires cross-validation on the rank to be used. In particular, as the number of times and slices in the data is close to 1, there is no gain in computation time compared with non-tensor models.

5. Conclusion of the article

The results obtained from simulated data show a clear advantage of tensor-based methods over non-tensor methods, attributable to the fact that the regression coefficient $\boldsymbol{\beta}_{\text{opt}}$ possesses a tensor structure. The multiway multiblock model is particularly well suited to finding the structure of the $\boldsymbol{\beta}_{\text{opt}}$ coefficient efficiently when the data are separated into blocks. But in any case, the classic multiway model offers the best classification performance. However, on the liver cancer data, tensor methods offer no particular advantage over other models. We can therefore assume that, while the features are well structured in tensor form, this is not the case for the $\boldsymbol{\beta}_{\text{opt}}$ coefficient. What's more, the results obtained on these data are far too weak to be exploitable in a medical context. These poor results can be explained by the lack of training data: there are only 16 CCK tumors whose MRI images contain all the times studied.

However, our approach to the real data studied has another limitation. We rely exclusively on pyradiomics to extract features of interest from MRI images of tumors. However, these features (gray levels, co-occurrence matrix, etc.) are more a matter of image processing than of medicine. Thus, they do not necessarily correspond to what radiologists would look at to classify liver tumors. It would therefore be interesting to train machine learning models on indicators constructed by radiologists and compare their performance with that obtained in this article. Finally, the results shown on simulated data were obtained with a test set of only 1000 individuals. On real data, we carried out only 50 successive simulations for each model before taking the average of the performances obtained. The results obtained could therefore be refined by increasing the size of the simulations. However, in view of the performance obtained, it seems impossible that larger simulations would drastically change the conclusions obtained.

6. Latest results: not mentioned in the article

As indicated in the summary of this internship report, two weeks ago I went to visit Sébastien Mulé at Henri Mondor Hospital to talk to him about the pre-processing of liver cancer data. It was then that he remembered the existence of another database concerning the liver cancers of the patients studied in the article. In fact, for each individual, the radiologist had also indicated the presence or absence on the MRI images of 13 markers (presence of necrosis, presence of luminal enhancement in the late phase, etc.) which are usually used by radiologists to determine the class (HCC or CCK) of the tumor. This translates into 13 binary variables in the database, to which we add the patient's gender. Unlike the features extracted by pyradiomics, the features in this database are based exclusively on medical criteria. Furthermore, each marker indicated by the radiologist takes into account all 4 MRI images, whereas features extracted by pyradiomics only took into account one image at a time and were therefore extracted image by image. This explains why, unlike the features extracted by pyradiomics, those in the new database are not tensorial.

In order to compare these data with those studied in the article, a lasso logistic regression was performed on these data. Averaging the area under curve over 50 trainings, we find:

$$\text{AUC} = 0.96 \pm 0.02 \qquad \text{balanced accuracy} = 0.85 \pm 0.05$$

where confidence intervals are of the normal type and calculated at the 95% threshold. We can see that the two most important features in the classification (where feature importances are calculated as in Appendix D) are, in descending order, late luminal enhancement and non-peripheral washout. The first of these two features is considered the most important by radiologists, which is consistent with our model. These results are much better than those obtained with data extracted by pyradiomics. The quality of these results is particularly high given the low number of patients in the training database. Moreover, lasso logistic regression may not be the best model on these data, and we can hope to improve these results still further by using models more suited to binary data (such as random forest). These results are left out of the article section for the following reasons:

- They have no connection with tensors, whereas tensors are at the heart of the article. In order to practice writing reports in article format, it was therefore decided that I should write the section on actual data as if the latest data had not been communicated to me.
- These results have not been studied in depth, as they were obtained too late in the course. They cannot therefore be developed in the article.

I'm well aware, however, that the latest results completely exceed those obtained in the article, and that, with a view to publication, it would be necessary to study another set of real data, more suited to tensor models.

7. Pipot

7.1. Prolongements possibles

The work carried out on tensor models during this course shows that these models can be genuinely useful when the regression parameter has a tensor structure, as in the simulated data. However, the multiblock approach chosen in this internship may not be the most promising. Indeed, the primary objective of classification models is often to achieve optimal performance, rather than precisely reconstructing the structure of the regression parameters. In this area, the multiway model already developped in [2] offers a real advantage over the multiway mulktiblock model presented in this report. But other approaches remain unexplored. We could change the penalty used to, for example, mix the L1 and L2 penalties, or mix the group lasso with tensor models. We could also try to adapt variable selection procedures such as those proposed in [8], to improve computation speed in very high-dimensional tensors.

With regard to the liver cancer data, it would be interesting to train several tabular machine learning models on the data studied (random forest, group lasso, boosting etc...) in order to propose the most accurate model possible and compare it with the methods already used by radiologists. Indeed, while the performance of the algorithms on the data presented in the article section of this report left no doubt as to their inferiority to that of radiologists, on the new data, this becomes more uncertain. Finally, it would also be interesting to add mixed tumors to the analysis, to see whether the models are also capable of distinguishing them. In practice, they make up around a fifth of tumours, and even if they are less well understood than HCC or CCK tumours, it is useful to be able to distinguish them.

7.2. Bilan et prise de recul

Ce que j'ai appris (données de dernière minute notamment), la gestion des risques, l'éthique.

References

[1] L. Le Brusquet, G. Lechuga, A. Tenenhaus, Régression Logistique Multivoie, in: JdS 2014, Rennes, France, 2014, p. 6 pages.
URL https://centralesupelec.hal.science/hal-01056558

[2] F. Girka, P. Chevaillier, A. Gloaguen, G. Gennari, G. Dehaene-Lambertz, L. Le Brusquet, A. 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.
URL https://centralesupelec.hal.science/hal-03051752

[3] L. Meier, S. Van De Geer, P. Bühlmann, The Group Lasso for Logistic Regression, Journal of the Royal Statistical Society Series B: Statistical Methodology 70 (1) (2008) 53–71. arXiv:https://academic.oup.com/jrsssb/article-pdf/70/1/53/49796502/jrsssb_70_1_53.pdf, doi:10.1111/j.1467-9868.2007.00627.x.
URL https://doi.org/10.1111/j.1467-9868.2007.00627.x

[4] J. J. van Griethuysen, A. Fedorov, C. Parmar, A. Hosny, N. Aucoin, V. Narayan, R. G. Beets-Tan, J.-C. Fillion-Robin, S. Pieper, H. J. Aerts, Computational Radiomics System to Decode the Radiographic Phenotype, Cancer Research 77 (21) (2017) e104–e107. arXiv:https://aacrjournals.org/cancerres/article-pdf/77/21/e104/2934659/e104.pdf, doi:10.1158/0008-5472.CAN-17-0339.
URL https://doi.org/10.1158/0008-5472.CAN-17-0339

[5] T. G. Kolda, B. W. Bader, Tensor decompositions and applications, SIAM Review 51 (3) (2009) 455–500. arXiv:https://doi.org/10.1137/07070111X, doi:10.1137/07070111X.
URL https://doi.org/10.1137/07070111X

[6] R. Tibshirani, T. Hastie, J. Friedman, Regularized paths for generalized linear models via coordinate descent, Journal of Statistical Software 33 (02 2010). doi:10.1163/ej.9789004178922.i-328.7.

[7] H. Zhou, L. Li, H. Zhu, Tensor regression with applications in neuroimaging data analysis, Journal of the American Statistical Association 108 (2013) 540–552. doi:10.1080/01621459.2013.776499.

[8] J. Fan, J. Lv, Sure independence screening for ultra-high dimensional feature space (2008). arXiv:math/0612857.
URL https://arxiv.org/abs/math/0612857

Appendix A. Hyperparameters for simulated data

Appendix A.1. Data generation

In our simulations, we unfold the β^l of each pictogram line by line into a vector (rather than a matrix) and concatenate these vectors to obtain $\boldsymbol{\beta} = [\boldsymbol{\beta}^1; \cdots \boldsymbol{\beta}^L]$. Let N be the size of $\boldsymbol{\beta}$. We then use the following parameters to generate the simulated data:

- $\boldsymbol{\mu}_0 = \mathbb{0}_N$
- $\boldsymbol{\mu}_1 = \boldsymbol{\beta}/\|\boldsymbol{\beta}\|$
- $\mathbf{P}$ is obtained by completing in orthonormal basis $\boldsymbol{\beta}/\|\boldsymbol{\beta}\|$
- $d_1 = 0.01$
- For $i \in [\![2, N]\!]$, $d_i = 0.25$ (where (d_i) are the diagonal elements of $\mathbf{D}$)

On 1000 individuals generated, with 500 in each class, the accuracy obtained by the 2-means algorithm is 0.48: in other words, it doesn't do better than chance (even slightly worse in our case). However, the section 4 shows far better performance from our models

Appendix A.2. Cross validation of models

For the lasso and group lasso models, we cross-validate on 20 values of λ distributed according to a logarithmic scale between 10^{-5} and 10^{-13}.

For the muliway and multiway mulibloc models, we cross-validate on 5 values of λ distributed on a logarithmic scale between 10^{-3} and 10^{-6}. We also cross-validate on the rank used. The rank of $\boldsymbol{\beta}_{\text{opti}}$ is bounded above by the sum of the ranks of the individual pictograms (around 15, 1 and 10 respectively). So we won't exceed 27 for rank in our cross validation. In fact, we hope to approximate β with a matrix of rank lower than 26 to ensure a certain sparsity in the model. We therefore cross-validate on ranks 1, 10 and 20 in the multiway model (to propose a model parameterization with a minimal rank, an intermediate rank and a high rank).

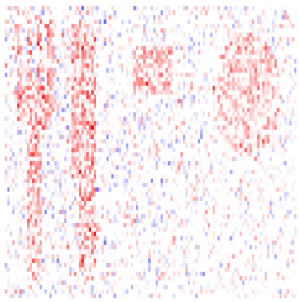
In the multiblock model, you can choose a rank for each pictogram. In order not to overload the cross-validation, we limit ourselves to cross-validating on the rank of the first pictogram. As this is the highest-ranking pictogram, there are more different "choices" possible for the multiblock model, depending on the desired sparsity. We therefore propose the ranks 1, 6 and 12 for this pictogram in the cross validation. For the other pictograms, we impose 1 and 10 respectively. Here again, we also cross-validate on 5 values of λ, distributed on a logarithmic scale between 10^{-3} and 10^{-6}.

Appendix B. Parameters used for feature extraction with pyradiomics

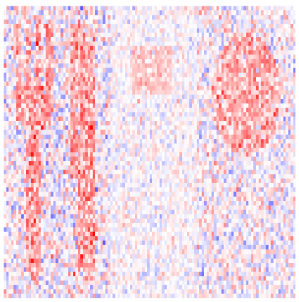
List of parameters used for feature extraction by pyradiomics:

- Bin width : 25
- Resampled Pixel Spacing : $[2, 2, 2]$ si l'extraction est en 3D, $[2, 2]$ si elle est en 2D
- interpolator : sitkBSpline
- force2D : True
- force2Ddimension : 2

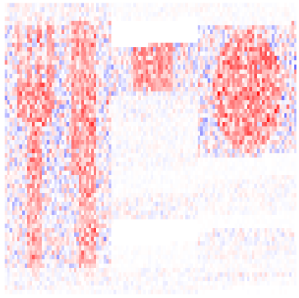
Appendix C. Reconstructed pictograms



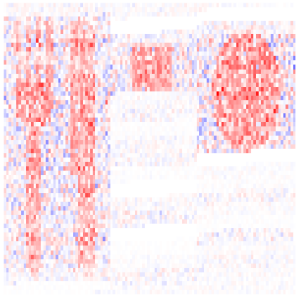
(a) lasso



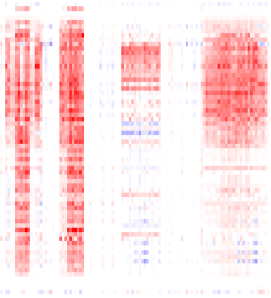
(b) group lasso (block)



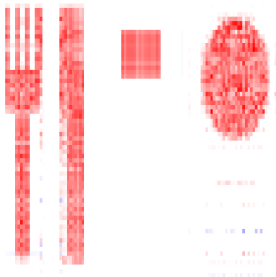
(c) group lasso (variable)



(d) group lasso (mode)

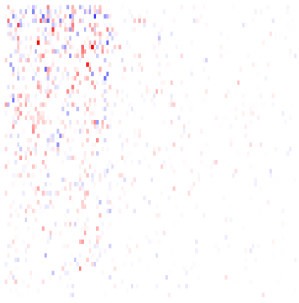


(e) multiway

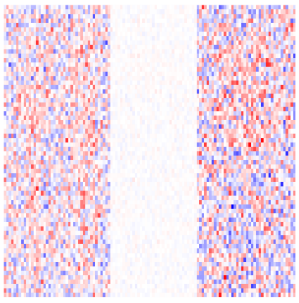


(f) multiway multibloc

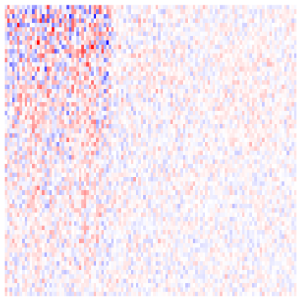
Fig. C.7: Pictograms reconstructed by the different models for 3000 individuals in the training dataset. The name of the model used is indicated in the legend of each figure.



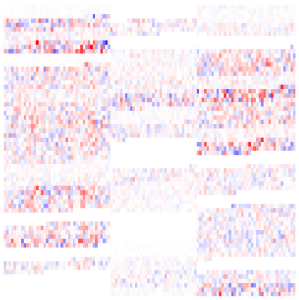
(a) lasso



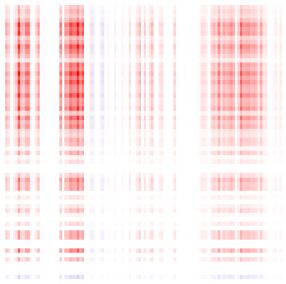
(b) group lasso (block)



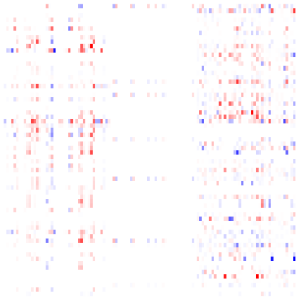
(c) group lasso (variable)



(d) group lasso (mode)



(e) multiway



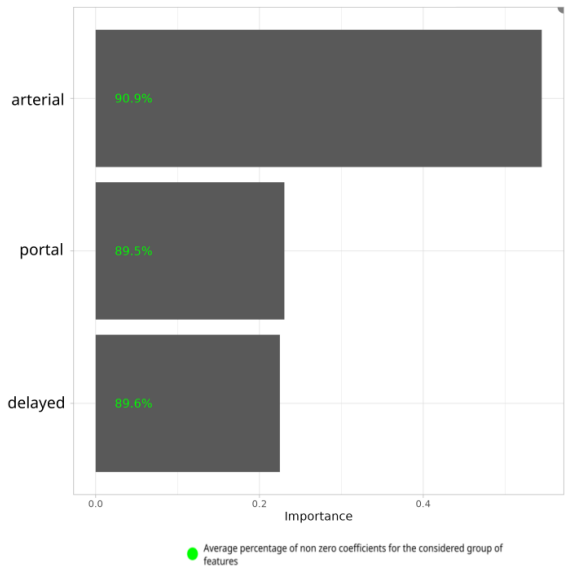
(f) multiway multibloc

Fig. C.8: Pictograms reconstructed by the different models for 500 individuals in the training dataset. The name of the model used is indicated in the legend of each figure.

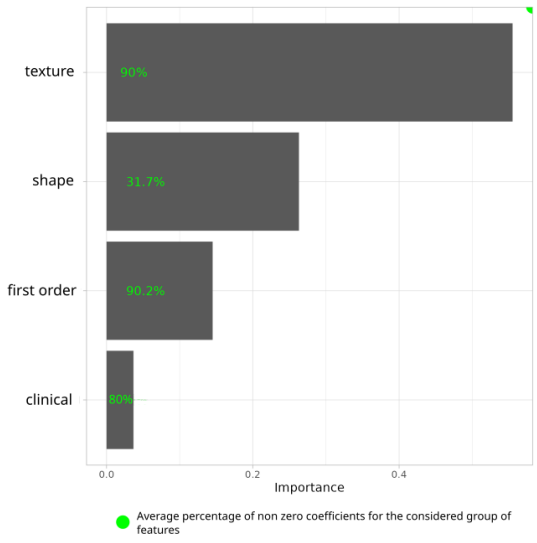
Appendix D. Importance of features

Importance graphs are given here for the best-performing models (in terms of AUC, see Table 3), namely the group lasso model with block grouping for 3D data and the classic lasso for 2D data. The importance of a feature is measured as the absolute value of the β coefficient in front of it. This value is then averaged over all simulations to find the feature's importance. Given the large number of features, we group them by block, mode and/or variable name. The importance of a group is given as the sum of the importances of its features. All group importances are finally renormalized so that their sum is 1 (we're interested in the relative importance of features in relation to each other).

On each stick of each bar chart, in addition to the importance, we can read a percentage in green. This provides information on the number of times the coefficient in front of the features associated with the stick has been non-zero in the simulations. As all our models are penalized by the lasso, they tend to set the coefficients of the least important variables to zero. We can therefore calculate for each variable, over the 50 training sessions carried out, the percentage of times this coefficient was non-zero. The average of these percentages over all the features in a feature group (block, mode or variable) is shown in green on the graph. This average reflects the number of times the features in this group were deemed important by the model (i.e. their regression coefficient was non-zero).

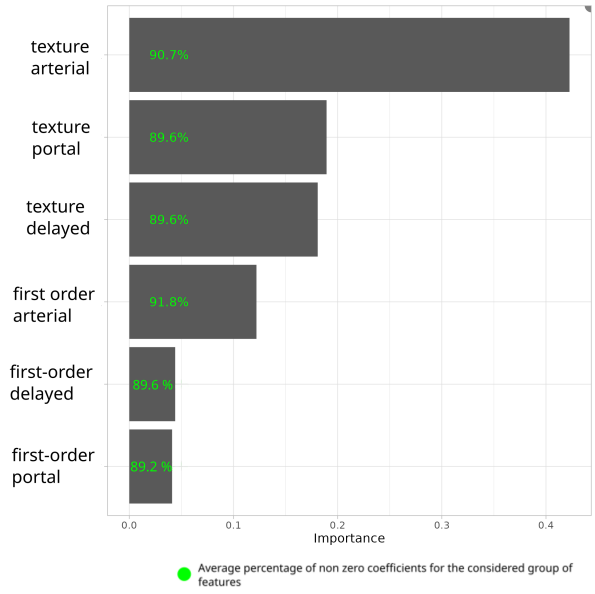


(a) Relative importance of time on 3D data (excluding clinical data)

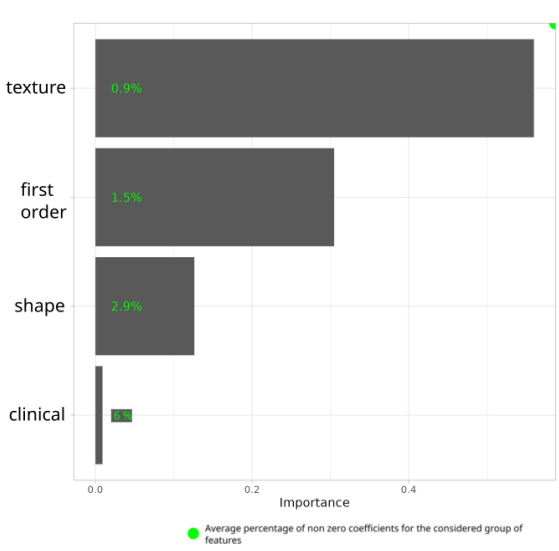


(b) relative importance of blocks on 3D data

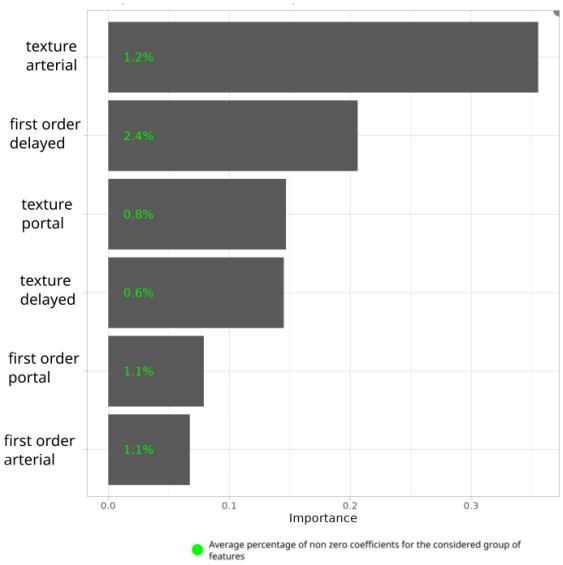
Fig. D.9: Relative importance of features for 3D data with the best-performing model for this data: group lasso with block grouping (part 1)



(c) relative importance of block times on 3D data (excluding clinical data)

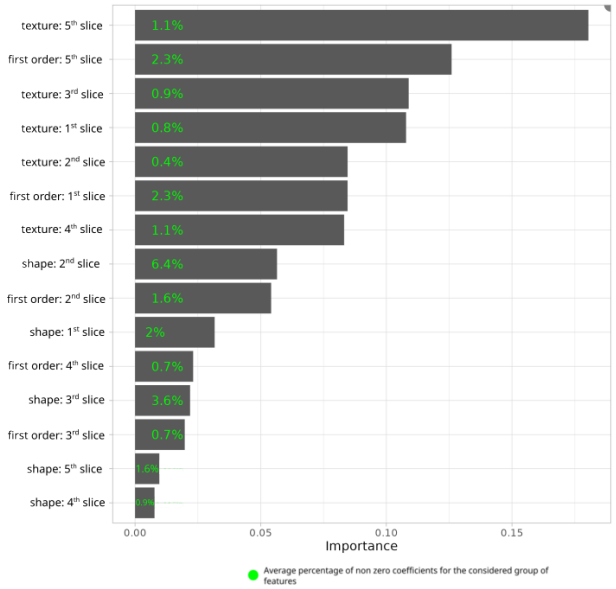


(a) Relative importance of blocks on 2D data



(b) relative importance of times per block on 2D data (excluding clinical data and shape featrues because they do not depend on time)

Fig. D.10: Relative importance of features for 2D data with the best-performing model on these data: lasso logistic regression



(c) relative importance of slices per block on 2D data (excluding clinical data)