Machine learning methods to help classify liver tumors from radiomics data

Alexandre SELVESTREL[1]

[1]Laboratoire des systèmes, Orsay France
[2]Centrale-Supélec, Gif-sur-Yvette France

Synthèse

L'objectif de mon stage était de réaliser une classification automatique (via du machine learning) de tumeurs du foie basée sur des radios et sur quelques données cliniques (âge, sexe du patient ...). Cette classification a permis de tester des modèles tensoriels [1,2] et de vérifier si ceux-ci donnaient de meilleures performances que les autres modèle. 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: Ces stage s'inscrit dans la cadre de la collaboration entre le l2S et l'AP-HP et vient plus généralement renforcer les liens entre le laboratoire et le monde médical. Du point de vue du l2S, il s'agit de mettre à l'épreuve une méthode de machine learning particulière, basée sur des modèles tensoriels et qui semble spécifiquement adaptée aux données étudiées. Or, cette méthode n'a pas encore été beaucoup utilisée, en particulier dans le domaine de la santé. Dans le cas où des avancées théoriques seraient proposées sur cette méthode, celles-ci pourraient faire l'objet d'une publication scientifique future. 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 radios 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 basics (régression logistique lasso et random forest) sur les données médicales. Cela nous a permis de prendre connaissance des données, d'établir un protocole pour comparer les différents modèles (en se basant sur l'area under curve: AUC) et 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 une tentative d'amélioration du modèle en séparant les variables en plusieurs blocs, aucun gain de perfrmance n'était observé.

Afin de vérifier que notre nouveau modèle était pertinent nous avons ensuite cherché à tester son efficacité sur des données simulées. Dans ce cas, notre modèle tensoriel a montré des performances bien meilleures (+ 0.1 d'AUC en moyenne) que les modèles non tensoriels. Cela nous a permis de conclure que notre modèle était pertinent et que le manque de performance observé sur les données médicales était probablement dû à un mauvais pré-traitement des données.

Enfin, nous avons approfondi les méthodes d'extraction des features à partir des radios et identifié des pistes d'amélioration pour les scripts pyradiomics utilisés. En changeant l'extraction des features nous sommes passé à une (AUC = 0.85) sur données médicales pour un modèle de régression logistique lasso. Notre modèle tensoriel quant à lui a permis d'obtenir une AUC de ...

Contents

1 Introduction

Il existe deux grands types de tumeur du foie: les carcinome hépatocellulaire (CHC) et les cholangiocarcinomes (CCK). Certaines tumeurs présentent même des caractéristiques CCK et CHC selon l'endroit du foie observé et sont alors dites mixtes. Pour les médecins, déterminer la classe d'une tumeur du foie est très important car le traitement à choisir dépend de cette classe. Pour l'instant, deux grandes approches sont à leur disposition: la microscopie et la radiographie avec injection de produit contrastant.

La microscopie est la méthode la plus fiable car elle permet de directement analyser les cellules tumorales. Toutefois, puisqu'elle nécessite de prélever un petit morceau de foie cancéreux, elle demande une opération et peut entraîner des complications chez le patient. De plus, étant donné qu'on a seulement accès à un fragment du foie, on peut généraliser à tort la nature de tumeur détectée à l'intégralité du foie. Cela peut être gênant dans le cas des tumeurs mixtes où certaines zones cancéreuses sont CHC et d'autres sont CCK.

La radiographie (par IRM ou scanner) avec injection de produit contrastant est au contraire non invasive. L'idée est d'observer les différents réhaussements lumineux du tissus tumoral en fonction du temps après injection du produit contrastant. En plus d'être non invasive, elle a l'avantage de donner accès à la tumeur en 3D dans son intégralité. C'est pour cela que les médecins (et l'AP-HP) cherchent à la développer davantage. Cependant, les images obtenues par radio ne permettent pas de déterminer avec certitude la nature de la tumeur. En effet, les caractéristiques des tumeurs CHC et CCK sont souvent très proches et les experts ne sont pas toujours d'accord entre eux lorsqu'ils analysent les images. Par ailleurs, de manière générale, il n'est pour l'instant pas possible de poser un diagnostique fiable sur une tumeur du foie à partir de méthodes non invasives seules [3].

Cependant, les limitations que nous avons présentées pour la radiographie ne concernent que les analyses d'images effectuées à l'oeuil nu. On peut donc espérer s'en affranchir en utilisant du machine learning. Il s'agit précisément de l'enjeu de ce stage. L'objectif est d'élaborer un modèle capable de classer le mieux possible les tumeurs à partir de radios et d'informations cliniques sur les patients (âge et sexe). On souhaite que le modèle soit facilement interprétable. En effet, nous devons aider les médecins à déterminer quels critères sont clefs dans la classification, ce qu'un modèle "boîte noire" ne permettrait pas. On préfère donc utiliser du machine learning classique par rapport au deep learning.

Cependant, avant même de parler de modèle de Machine Learning, il faut choisir une méthode d'extraction des données. En effet, les modèles de machine learning classique sont généralement assez mauvais en traitement d'images brutes et il faut donc choisir une manière d'extraire les features intéressantes des radios. Ce choix n'a rien d'évident et il n'y a à priori pas de "bonne manière de faire". Par exemple, on peut extraire des grandeurs "globales", qui décrivent la tumeur dans son intégralité, en 3D, (volume, luminosité globale etc...) ou au contraire extraire des informations coupe par coupe dans la tumeur (surface de tumeur dans la coupe, luminosité de la tumeur dans la coupe etc...). Cette deuxième méthode permet de récupérer davantage de données par individu mais celles-ci risquent d'être redondantes d'une coupe à l'autre pour une même tumeur. On peut aussi ajouter qu'il y a un nombre non négligeable de données manquantes, par exemple liées au fait que le patient a pu bouger durant une radio -la rendant inutilisable- et qu'il faut s'y adapter. Pour réaliser l'extraction des features, des scripts Python utilisant la bibliothèque pyradiomics [4] avaient déjà été écrits par mes encadrants avant mon stage mais ils n'étaient pas optimaux. Ainsi, une partie importante de mon travail a consisté à les améliorer.

Concernant le modèle en lui-même, nous avons décidé qu'il devrait tenir compte du fait qu'à chaque radio, pour un même patient, ce sont les mêmes features qui sont observées à des temps différents. Dans le cas d'une extraction coupe par coupe, on veut aussi qu'il tienne compte du fait que ce sont les mêmes grandeurs qui sont évaluées à différents niveaux de "profondeur" d'une même tumeur. Ce type de modèles est appelé

modèle tensoriel [1, 2]. Cependant, afin de coller au mieux à la structure des données, nous introduisons une extension d'un de ces modèles, en regroupant les variables par blocs. Cette extension sera développée dans la partie "Méthode" de ce rapport. Le but de ce stage est donc double: d'une part il s'agit de chercher à améliorer la classification des tumeurs du foie à partir de radios et d'informations cliniques, et d'autre part il s'agit de vérifier si l'extension du modèle tensoriel que nous proposons peut réellement apporter une plus-value par rapport aux modèles déjà existants.

Le problème posé étant déjà très complexe, nous avons décidé par simplicité d'exclure les tumeurs mixtes de notre étude. En effet, il n'existe pas à ce jour de traitement standardisé pour ces tumeurs et elles sont encore assez mal connues. Nous avons donc préféré nous concentrer sur un modèle à deux classes (CHC et CCK) pour lequel les médecins ont déjà des critères de classification bien établis. Toutefois, les méthodes utilisées pourraient théoriquement facilement être étendues à des modèles à trois classes (en changeant le modèle linéaire généralisé sous-jascent).

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 $\underline{\mathbf{X}} = (x_{ijk_1k_2\ \dots\ k_M})_{i\in[\![1,n]\!],j\in[\![1,J]\!],k_1\in[\![1,K_1]\!]\ \dots\ k_m\in[\![1,K_M]\!]}$ (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 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 methodes 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 any order of tensor.

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 [6]. 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 blocs, we finally group by bloc.

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:

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

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 $\mathbf{x_i}$. This law is defined by:

$$\mathbb{P}(y_i = 1 \,|\, \mathbf{x}_i) = \frac{1}{1 + \exp(-\mathbf{x}_i^T \boldsymbol{\beta} - \beta_0)} \tag{2}$$

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:

$$\beta_0, \boldsymbol{\beta}^J, \boldsymbol{\beta}^K, \boldsymbol{\beta}_{\text{tab}} = \underset{\beta_0, \boldsymbol{\beta}^J, \boldsymbol{\beta}^K, \boldsymbol{\beta}_{\text{tab}}}{\operatorname{argmin}} \left(-\sum_{i=1}^{N} \log(\mathbb{P}(y_i = 1 \,|\, \mathbf{x}_i)) + \sum_{r=1}^{R} \|\boldsymbol{\beta}_r^K \otimes \boldsymbol{\beta}_r^J\|_1 + \|\boldsymbol{\beta}_{\text{tab}}\|_1 \right) \tag{3}$$

Optimization is performed by alternating directions between $\left[\beta_0; \boldsymbol{\beta}^J; \boldsymbol{\beta}_{\text{uni}}\right]$ and $\left[\beta_0; \boldsymbol{\beta}^K; \boldsymbol{\beta}_{\text{uni}}\right]$. The stopping criterion is defined by the relative difference between the value of the objective function after 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 $K_0 \in \mathbb{N}^*$):

$$C : \begin{cases} \mathbb{R} \times \mathbb{R}^{K_0} \times \mathbb{R}^{N \times K_0} \times \mathbb{R}^N \times \mathbb{R} & \longrightarrow & \mathbb{R} \\ (\beta_0, \boldsymbol{\beta}, \mathbf{X}, \mathbf{y}, \lambda) & \longmapsto & -\sum_{i=1}^{N} [y_i(\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta}) - \log(1 + \exp(\beta_0 + \mathbf{x}_i^T \boldsymbol{\beta}))] + \lambda \|\boldsymbol{\beta}\|_1 \end{cases} \tag{4}$$

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

$$\underset{(\beta_0, \boldsymbol{\beta}) \in \mathbb{R} \times \mathbb{R}^{JR+M}}{\operatorname{argmin}} C(\beta_0, (\mathbf{Q}^J)^{-1} \boldsymbol{\beta}, \mathbf{Z}^J \mathbf{Q}^J, \mathbf{y}, \lambda) \tag{5}$$

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

$$\mathbf{Z}^J = [\mathbf{Z}_1^J \ \cdots \ \mathbf{Z}_R^J \ \ \mathbf{X}_{\text{tab}}] \tag{6}$$

$$\text{where } \forall r \in [\![1, R]\!], \quad \mathbf{Z}_{\mathbf{r}}^{\mathbf{J}} = \sum_{k=1}^{K} \mathbf{X}_{::k} (\beta_r^K)_k \quad (\mathbf{Z}_r^J \in \mathbb{R}^{N \times J}) \tag{7}$$

$$\mathbf{Q}^J = \text{Diag}([\|\boldsymbol{\beta}_1^K\|_1^{-1} \mathbb{1}_J; \ \cdots \ ; \|\boldsymbol{\beta}_R^K\|_1^{-1} \mathbb{1}_J; \ \mathbb{1}_M]) \tag{8}$$

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

$$\mathbf{x}_{(1)_i}^T \left(\sum_{r=1}^{R} \boldsymbol{\beta}_r^K \otimes \boldsymbol{\beta}_r^J \right) = \sum_{r=1}^{R} \left[(\mathbf{x}_{(1)_i}^T \left(\boldsymbol{\beta}_r^K \otimes \mathbf{I}_J \right) \right] \boldsymbol{\beta}_r^J \tag{9}$$

$$= \sum_{r=1}^{R} (\mathbf{z}_r^J)_i^T \boldsymbol{\beta}_r^J \tag{10}$$

and that

$$\sum_{r=1}^{R} \|\boldsymbol{\beta}_r^K \otimes \boldsymbol{\beta}_r^J\|_1 = \|\mathbf{R}_{\text{tens}}^J \boldsymbol{\beta}^J\|_1 \tag{11}$$

$$\text{with} \quad \mathbf{R}_{\text{tens}}^J = \text{Diag}([\|\boldsymbol{\beta}_1^K\|_1 \mathbb{1}_J; \ \cdots \ ; \|\boldsymbol{\beta}_R^K\|_1 \mathbb{1}_J]) \tag{12}$$

Thus,

$$(\mathbf{x}_{\text{tot}})_i^T \boldsymbol{\beta} = (\mathbf{z}_i^J)^T [\boldsymbol{\beta}^J; \boldsymbol{\beta}_{\text{tab}}] \tag{13}$$

$$\text{and} \quad \sum_{i=1}^{N} \|\boldsymbol{\beta}_r^K \otimes \boldsymbol{\beta}_r^J\|_1 + \|\boldsymbol{\beta}_{\text{tab}}\|_1 = \|(\mathbf{Q}^J)^{-1} \boldsymbol{\beta}\|_1 \tag{14}$$

This 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 definition of $\mathbf{Z}^K$. It is:

$$\mathbf{Z}^K = [\mathbf{Z}_1^K \ \cdots \ \mathbf{Z}_R^K \ \ \mathbf{X}_{\text{tab}}] \tag{15}$$

$$\text{with} \forall r \in [\![1, R]\!] \quad \mathbf{Z}_r^K = \sum_{j=1}^{J} (\beta_r^J)_j \mathbf{X}_{:j:} \tag{16}$$

This is justified by:

$$\mathbf{x}_{(1)_i}^T \left(\sum_{r=1}^{R} \boldsymbol{\beta}_r^K \otimes \boldsymbol{\beta}_r^J \right) = \sum_{r=1}^{R} \left[(\mathbf{x}_{(1)_i}^T \left(I_K \otimes \boldsymbol{\beta}_r^J \right) \right] \boldsymbol{\beta}_r^K \tag{17}$$

$$= \sum_{r=1}^{R} (\mathbf{z}_r^K)_i^T \boldsymbol{\beta}_r^K \tag{18}$$

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 blocs 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 bloc. 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:

$$\boldsymbol{\beta} = \left[\sum_{r_1=1}^{R_1} \boldsymbol{\beta}_{(1,r_1)}^K \otimes \boldsymbol{\beta}_{(1,r_1)}^J ; \ \cdots \ ; \sum_{r_L=1}^{R_L} \boldsymbol{\beta}_{(L,r_L)}^K \otimes \boldsymbol{\beta}_{(L,r_L)}^J \ ; \boldsymbol{\beta}_{\text{tab}} \right] \tag{19}$$

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

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

$$\boldsymbol{\beta}^J = \left[\boldsymbol{\beta}_{(1,1)}^J ; \ \cdots \ ; \boldsymbol{\beta}_{(1,R_1)}^J ; \ \cdots \ \cdots \ ; \boldsymbol{\beta}_{(L,1)}^J \ \cdots \ ; \boldsymbol{\beta}_{(L,R_L)}^J \right] \tag{20}$$

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

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

$$\beta_0, \boldsymbol{\beta}^J, \boldsymbol{\beta}^K, \boldsymbol{\beta}_{\text{tab}} = \underset{\beta_0, \boldsymbol{\beta}^J, \boldsymbol{\beta}^K, \boldsymbol{\beta}_{\text{tab}}}{\operatorname{argmin}} \left(-\sum_{i=1}^{N} \log(\mathbb{P}(y_i = 1 \,|\mathbf{x}_i)) + \sum_{l=1}^{L} \sum_{r_l=1}^{R_l} \|\boldsymbol{\beta}^K_{(l,r_l)} \otimes \boldsymbol{\beta}^J_{(l,r_l)}\|_1 + \|\boldsymbol{\beta}_{\text{tab}}\|_1 \right) \tag{22}$$

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

$$\underset{(\beta_0, \boldsymbol{\beta})}{\operatorname{argmin}} \, C(\beta_0, (\mathbf{Q}^J)^{-1}\boldsymbol{\beta}, \mathbf{Z}^J\mathbf{Q}^J, \mathbf{y}, \lambda) \tag{23}$$

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

$$\mathbf{Z}^J = [\mathbf{Z}^J_{(1,1)} \;\cdots\; \mathbf{Z}^J_{(1,R_1)} \;\cdots\cdots\; \mathbf{Z}^J_{(L,1)} \;\cdots\; \mathbf{Z}^J_{(L,R_L)} \;\mathbf{X}_{\text{tab}}] \tag{24}$$

$$\text{where } \forall r_l \in [\![1, R_l]\!], \qquad \mathbf{Z}^J_{(l,r_l)} = \sum_{k=1}^{K} \mathbf{X}^l_{::k} \left(\beta^K_{(l,r_l)}\right)_k \qquad \left(\mathbf{Z}^J_{(l,r_l)} \in \mathbb{R}^{n \times d_l}\right) \tag{25}$$

$$\mathbf{Q}^J = \text{Diag}([\|\boldsymbol{\beta}^K_{(1,1)}\|_1^{-1}\mathbb{1}_{d_1};\; \cdots;\|\boldsymbol{\beta}^K_{(1,R_1)}\|_1^{-1}\mathbb{1}_{d_1};\; \cdots\cdots;\|\boldsymbol{\beta}^K_{(L,1)}\|_1^{-1}\mathbb{1}_{d_L};\; \cdots;\|\boldsymbol{\beta}^K_{(L,R_L)}\|_1^{-1}\mathbb{1}_{d_L};\; \mathbb{1}_M]) \tag{26}$$

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

$$(\mathbf{x}'_{(1)})^T_i \left[\sum_{r_1=1}^{R_1} \boldsymbol{\beta}^K_{(1,r_1)} \otimes \boldsymbol{\beta}^J_{(1,r_1)};\; \cdots;\; \sum_{r_L=1}^{R_L} \boldsymbol{\beta}^K_{(L,r_L)} \otimes \boldsymbol{\beta}^J_{(L,r_L)}\right] = \sum_{l=1}^{L}\sum_{r_l=1}^{R_l} \left(\mathbf{x}^l_{(1)}\right)^T_i \left(\boldsymbol{\beta}^K_{(l,r_l)} \otimes \boldsymbol{\beta}^J_{(l,r_l)}\right) \tag{27}$$

$$= \sum_{l=1}^{L}\sum_{r_l=1}^{R_l} \left[\left(\mathbf{x}^l_{(1)}\right)^T_i \left(\boldsymbol{\beta}^K_{(l,r_l)} \otimes I_{d_l}\right)\right] \boldsymbol{\beta}^J_{(l,r_l)} \tag{28}$$

$$= \sum_{l=1}^{L}\sum_{r_l=1}^{R_l} \left(\mathbf{z}^J_{(l,r_l)}\right)^T_i \boldsymbol{\beta}^J_{(l,r_l)} \tag{29}$$

An that

$$\sum_{l=1}^{L}\sum_{r_l=1}^{R_l} \|\boldsymbol{\beta}^K_{(l,r_l)} \otimes \boldsymbol{\beta}^J_{(l,r_l)}\|_1 = \|\mathbf{R}^J_{\text{tens}}\boldsymbol{\beta}^J\|_1 \tag{30}$$

$$\text{with} \quad \mathbf{R}^J_{\text{tens}} = \text{Diag}([\|\boldsymbol{\beta}^K_{(1,1)}\|_1\mathbb{1}_{d_1};\; \cdots;\|\boldsymbol{\beta}^K_{(1,R_1)}\|_1\mathbb{1}_{d_1};\; \cdots\cdots;\|\boldsymbol{\beta}^K_{(L,1)}\|_1\mathbb{1}_{d_L};\; \cdots;\|\boldsymbol{\beta}^K_{(L,R_L)}\|_1\mathbb{1}_{d_L};\; \mathbb{1}_M]) \tag{31}$$

We deduce that

$$[\mathbf{x}'_{(1)\,i};\, \mathbf{x}_{\text{tab}_i}]\boldsymbol{\beta} = (\mathbf{z}^J_i)^T[\boldsymbol{\beta}^J; \boldsymbol{\beta}_{\text{tab}}] \tag{32}$$

$$\text{and} \quad \sum_{l=1}^{L}\sum_{r_l=1}^{R_l} \|\boldsymbol{\beta}^K_{(l,r_l)} \otimes \boldsymbol{\beta}^J_{(l,r_l)}\|_1 + \|\boldsymbol{\beta}_{\text{uni}}\|_1 = \|(\mathbf{Q}^J)^{-1}\boldsymbol{\beta}\|_1 \tag{33}$$

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:

$$\mathbf{Z}^K = [\mathbf{Z}^K_{(1,1)} \ \cdots \ \mathbf{Z}^K_{(1,R_1)} \ \cdots \ \cdots \ \mathbf{Z}^K_{(L,1)} \ \cdots \ \mathbf{Z}^K_{(L,R_L)} \ \mathbf{X}_{\text{tab}}] \tag{34}$$

$$\text{where } \forall r_l \in [\![1, R_l]\!], \quad \mathbf{Z}^K_{(l,r_l)} = \sum_{j=1}^{d_l} \mathbf{X}^l_{:j:} \left(\beta^J_{(l,r_l)}\right)_j \quad \left(\mathbf{Z}^K_{(l,r_l)} \in \mathbb{R}^{n \times K}\right) \tag{35}$$

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

Notes:

- 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 [7].

Pseudo-code:

In order to clarify the algorithm that we use, we give in annexe A the pseudo-code of our implementation.

2.3 Simulated data generation

To test our multi-channel, multi-block 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 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 β. This is why we have chosen to generate our data in such a way that the optimal regression parameter β_{opti} (i.e. minimizing the classification error) has a multiblock multiway structure. And to make the reconstruction of the regression parameter as visual as possible, we reused the method presented in [8]. Thus, β_{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: 1). The result is β_{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 β_{opti}, instead of choosing just one pictogram, we consider the columnar concatenation of several pictograms (Fig: 2). 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. And it's this concatenation

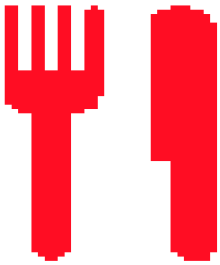

Fig. 1: Example of the pictogram used to generate β.

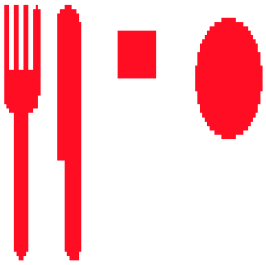

Fig. 2: Example of pictogram concatenation used to generate β.

which, after being unfolded into a single line, constitutes β_{opti}. This renders the single multi-way logistic regression model less relevant (which will need to have a high rank to correctly reconstruct β_{opti}), without putting the multiway multiblock model at a disadvantage (which will be able to separate β_{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 reduced normal distribution), identical for all individuals. The explained variable is then obtained by applying the regression model with $\beta = \beta_{\text{opti}}$ to the explanatory variables. This is, for example, what is proposed in [8]. However, this method poses a problem in binary classification, as we have no control over the number of individuals in each class. And given the size of our β_{opti}, the time needed to obtain a balanced set of individuals through trial and error is likely to be long.

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 expectations. These expectations and covariance matrices are chosen to ensure that β_{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 expectation vectors of the n explanatory variables of the two classes and $\boldsymbol{\Sigma}$ the covariance matrix of these same variables, if we impose*

$$\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0 /\!\!/ \beta_{\text{opti}} \tag{36}$$

$$\boldsymbol{\Sigma} = \mathbf{PDP}^T \quad \text{with } \mathbf{P} \in \mathcal{O}(n) \tag{37}$$

$$\text{the first column of } \mathbf{P} \text{ is colinear to } \boldsymbol{\beta}_{\text{opti}} \tag{38}$$

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:

$$g^* : \begin{cases} \mathbb{R}^n \longrightarrow \{0,1\} \\ \mathbf{x} \longmapsto \begin{cases} 1 & \text{if } E[Y|X=\mathbf{x}] \geq 0.5 \\ 0 & \text{else} \end{cases} \end{cases} \tag{39}$$

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

$$E(Y|X=\mathbf{x}) = \frac{1}{f_X(\mathbf{x})} \int y f_{(X,Y)}(\mathbf{x}, y) dy \tag{40}$$

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

$$E(Y|X=\mathbf{x}) = \frac{1}{f_X(\mathbf{x})} \sum_{y \in \{0,1\}} y f_{(X,Y)}(\mathbf{x}, y) \tag{41}$$

And therefore

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

Which means

$$E(Y|X=\mathbf{x}) = \frac{f_{(X|Y)}(\mathbf{x}|y=1)P(Y=1)}{f_{X|Y}(\mathbf{x}|y=1)P(Y=1) + f_{X|Y}(\mathbf{x}|y=0)P(Y=0)} \tag{43}$$

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

$$E(Y|X=\mathbf{x}) \geq \frac{1}{2} \tag{44}$$

$$\iff \frac{p_1 \exp\left(-\frac{(\mathbf{x}-\boldsymbol{\mu}_1)\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_1)}{2}\right)}{p_1 \exp\left(-\frac{(\mathbf{x}-\boldsymbol{\mu}_1)\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_1)}{2}\right) + p_0 \exp\left(-\frac{(\mathbf{x}-\boldsymbol{\mu}_0)\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_0)}{2}\right)} \geq \frac{1}{2} \tag{45}$$

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

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

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:

$$E(Y|X=\mathbf{x}) \geq \frac{1}{2} \tag{48}$$

$$\iff \langle \mathbf{x}-\boldsymbol{\mu}_0, \mathbf{x}-\boldsymbol{\mu}_0 \rangle_{\Sigma^{-1}} + \langle -\mathbf{x}+\boldsymbol{\mu}_1, \mathbf{x}-\boldsymbol{\mu}_1+\boldsymbol{\mu}_0-\boldsymbol{\mu}_0 \rangle_{\Sigma^{-1}} \geq 2\log\left(\frac{p_0}{p_1}\right) \tag{49}$$

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

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

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

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

$$\tag{54}$$

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 $\mathbf{D}$:

$$\mathbf{x}^T\mathbf{P}\mathbf{D}^{-1}\mathbf{P}^T(\boldsymbol{\mu}_1-\boldsymbol{\mu}_0) = (\mathbf{x}^T\mathbf{v} \;\; 0 \;\; 0 \;\; \cdots \;\; 0)\mathbf{D}^{-1}\begin{pmatrix} \mathbf{v}^T(\boldsymbol{\mu}_1-\boldsymbol{\mu}_0) \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{55}$$

$$= \lambda^2\mathbf{x}^T(\boldsymbol{\mu_1}-\boldsymbol{\mu_0})\, d_1^{-1}(\boldsymbol{\mu_1}-\boldsymbol{\mu_0})^T(\boldsymbol{\mu}_1-\boldsymbol{\mu}_0) \tag{56}$$

And therefore

$$E(Y|X=\mathbf{x}) \geq \frac{1}{2} \tag{57}$$

$$\iff \mathbf{x}^T(\boldsymbol{\mu_1}-\boldsymbol{\mu_0}) \geq \frac{d_1}{\lambda^2\|\boldsymbol{\mu}_1-\boldsymbol{\mu}_0\|^2}\log\left(\frac{p_0}{p_1}\right) + \frac{d_1}{2\lambda^2\|\boldsymbol{\mu}_1-\boldsymbol{\mu}_0\|^2}\langle \boldsymbol{\mu}_0+\boldsymbol{\mu}_1, \boldsymbol{\mu}_1-\boldsymbol{\mu}_0 \rangle_{\Sigma^{-1}} \tag{58}$$

$$\tag{59}$$

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 β_{opti}.

The hyperparameters used to generate the simulated variables are presented in appendix B. These are chosen experimentally to enable our models to reconstruct β_{opti} without a simple 2-means algorithm being able to separate them (see appendix B). The projection onto the plane of the first two principal components of the explanatory variables is shown in figure 3. It shows that the classes are difficult to separate with the naked eye.

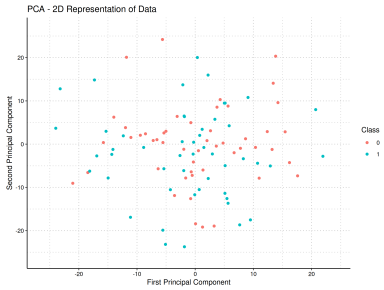

Fig. 3: Plane projection of the first two principal components of the explanatory variables simulated for 100 individuals when $\boldsymbol{\beta}_{\text{opti}}$ is given by the concatenation of pictograms in Fig: 2

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
CHC	84	81	83	78	72	74	86
CCK	21	21	17	21	15	19	22
Mixtes	35	36	32	34	29	31	37

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 CHC 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 fetoprotein (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 CHC 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 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 CHC 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 C. They were recommended by ... . To ensure that the extraction is consistent from one tumor to the next, all tumors have been re-sampled 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 spaing 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).

3.3 Feature extraction in 2D

The first step in this extraction process is to determine the slices we wish to extract from your tumour. 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 C. 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. 4). This curve is then inverted to obtain the depth distribution as a function of the cumulative tumour volume covered. A slice is then extracted at each of the following depths:

$$(i - 0.5)\frac{\text{area}_{\max}}{n_{\text{slices}}} \qquad \text{for i} \in [\![1, n_{\text{slices}}]\!] \tag{60}$$

We have tried other extraction methods, in particular trying to extract precisely the same depths for each MRI of the same tumor, and 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 features extraction used for each slice is almost the same as the one used for the 3D tumor (in

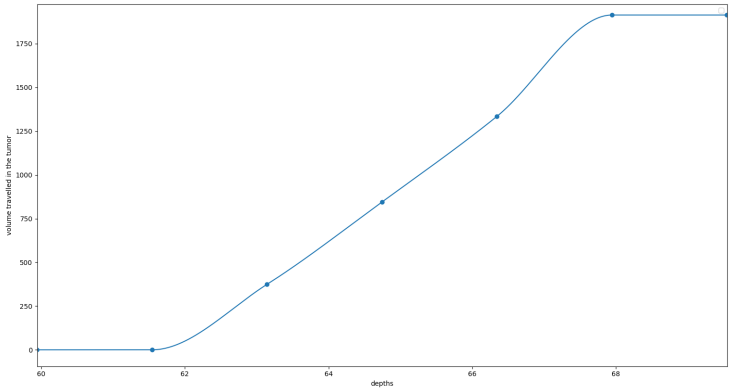

Fig. 4: graph of the cumulative volume distribution (in mm^3) of the third CCK patient's tumor according to depth (in mm) for a given tumor. The points correspond to the slices recorded in the sitk image (with its initial spacing). The curve is obtained by interpolating these points using Hermite cubic splines.

the previous section), except for shape parameters. In fact, 2D shape parameters (instead of 3D shape parameters) are now extracted. However, in order to retain important information on the overall shape of the tumor, the 3D shape parameters from the previous extraction are retained. Indeed, these cannot be reduced to slice-by-slice extracted parameters. Both 2D and 3D 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).

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 tumour, 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. 5 to check that the extraction is proceeding correctly.

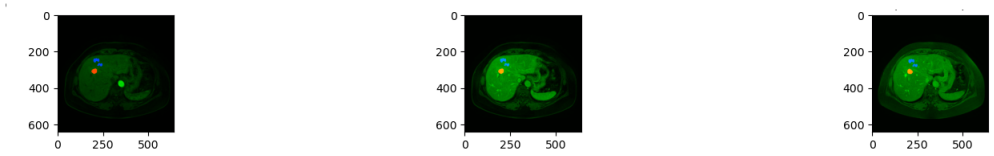

Fig. 5: MRIs of a CCK tumor slice with the tumor area in red and the peripheral area of extracted healthy liver in blue. From left to right, arterial, portal and late MRIs. Axes are graduated in mm.

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.

4 results

Conclusions

References

[1] L. Le Brusquet, G. Lechuga, and A. Tenenhaus, "Régression Logistique Multivoie," in *JdS 2014*, (Rennes, France), p. 6 pages, June 2014.

[2] F. Girka, P. Chevaillier, A. Gloaguen, G. Gennari, G. Dehaene-Lambertz, L. Le Brusquet, and 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.

[3] M. Jacquemin, "Performance of imaging in the diagnosis of hepatocellular carcinoma: a single-centre, retrospective series of 167 patients," *HAL Open Science*, 2021.

[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, and H. J. Aerts, "Computational Radiomics System to Decode the Radiographic Phenotype," *Cancer Research*, vol. 77, pp. e104–e107, 10 2017.

[5] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

[6] L. Meier, S. Van De Geer, and P. Bühlmann, "The Group Lasso for Logistic Regression," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 70, pp. 53–71, 01 2008.

[7] R. Tibshirani, T. Hastie, and J. Friedman, "Regularized paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, vol. 33, 02 2010.

[8] H. Zhou, L. Li, and H. Zhu, "Tensor regression with applications in neuroimaging data analysis," *Journal of the American Statistical Association*, vol. 108, pp. 540–552, 06 2013.

A Pseudo-code for multiblock multivariate logistic regression with lasso

In order to be more readable, we keep the notations that were used during the presentation of the model.

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. (24) and (25)
- Construct $\mathbf{Q}^J$ according to eq. (26)
- $(\beta_0^{(q)}, \boldsymbol{\beta}^{J(q)}) \longleftarrow \underset{(\beta_0,\boldsymbol{\beta})\in\mathbb{R}\times\mathbb{R}^{RJ+M}}{\operatorname{argmin}} \left(C(\beta_0, (\mathbf{Q}^J)^{-1}\boldsymbol{\beta}, \mathbf{Z}^J\mathbf{Q}^J, \mathbf{y}, \lambda)\right)$
- Construct $\mathbf{Z}^K$ according to eqs. (34) and (35)
- Construct $\mathbf{Q}^K$ by adapting eq. (26)
- $(\beta_0^{(q)}, \boldsymbol{\beta}^{K(q)}) \longleftarrow \underset{(\beta_0,\boldsymbol{\beta})\in\mathbb{R}\times\mathbb{R}^{LRK+M}}{\operatorname{argmin}} \left(C(\beta_0, (\mathbf{Q}^K)^{-1}\boldsymbol{\beta}, \mathbf{Z}^K\mathbf{Q}^K, \mathbf{y}, \lambda)\right)$
- $q \leftarrow q + 1$

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

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

B Génération des données simulées

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). By comparison, a lasso-penalized logistic regression on these data achieves an accuracy of

C 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