Université de Montréal

Analyse par apprentissage automatique des réponses fMRI du cortex auditif à des modulations spectro-temporelles.

par
Lysiane Bouchard

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté à la Faculté des arts et des sciences en vue de l'obtention du grade de Maître ès sciences (M. Sc.) en informatique

Décembre 2009

Université de Montréal
Faculté des arts et des sciences

Ce mémoire intitulé :
Analyse par apprentissage automatique des réponses fMRI du cortex auditif à des modulations spectro-temporelles.

présenté par :
Lysiane Bouchard

a été évalué par un jury composé des personnes suivantes :

Yoshua Bengio
président-rapporteur

Pascal Vincent
directeur de recherche

Douglas Eck
codirecteur

Marc Schönwiesner
codirecteur

Miklós Csürös
membre du jury

Résumé

L'application de classifieurs linéaires à l'analyse des données d'imagerie cérébrale (fMRI) a mené à plusieurs percées intéressantes au cours des dernières années. Ces classifieurs combinent linéairement les réponses des voxels pour détecter et catégoriser différents états du cerveau. Ils sont plus agnostics que les méthodes d'analyses conventionnelles qui traitent systématiquement les patterns faibles et distribués comme du bruit. Dans le présent projet, nous utilisons ces classifieurs pour valider une hypothèse portant sur l'encodage des sons dans le cerveau humain. Plus précisément, nous cherchons à localiser des neurones, dans le cortex auditif primaire, qui détecteraient les modulations spectrales et temporelles présentes dans les sons. Nous utilisons les enregistrements fMRI de sujets soumis à 49 modulations spectro-temporelles différentes. L'analyse fMRI au moyen de classifieurs linéaires n'est pas standard, jusqu'à maintenant, dans ce domaine. De plus, à long terme, nous avons aussi pour objectif le développement de nouveaux algorithmes d'apprentissage automatique spécialisés pour les données fMRI. Pour ces raisons, une bonne partie des expériences vise surtout à étudier le comportement des classifieurs. Nous nous intéressons principalement à 3 classifieurs linéaires standards, soient l'algorithme machine à vecteurs de support (linéaire), l'algorithme régression logistique (régularisée) et le modèle bayésien gaussien naïf (variances partagées).

mots clés : classifieur linéaire, neuroimagerie, modulation spectro-temporelle, cortex auditif, fMRI, modèle bayésien gaussien naïf, machine à vecteurs de support, régression logistique

Abstract

The application of linear machine learning classifiers to the analysis of brain imaging data (fMRI) has led to several interesting breakthroughs in recent years. These classifiers combine the responses of the voxels to detect and categorize different brain states. They allow a more agnostic analysis than conventional fMRI analysis that systematically treats weak and distributed patterns as unwanted noise. In this project, we use such classifiers to validate an hypothesis concerning the encoding of sounds in the human brain. More precisely, we attempt to locate neurons tuned to spectral and temporal modulations in sound. We use fMRI recordings of brain responses of subjects listening to 49 different spectro-temporal modulations. The analysis of fMRI data through linear classifiers is not yet a standard procedure in this field. Thus, an important objective of this project, in the long term, is the development of new machine learning algorithms specialized for neuroimaging data. For these reasons, an important part of the experiments is dedicated to studying the behaviour of the classifiers. We are mainly interested in 3 standard linear classifiers, namely the support vectors machine algorithm (linear), the logistic regression algorithm (regularized) and the naïve bayesian gaussian model (shared variances).

key words : linear classifier, neuroimaging, spectro-temporal modulation, auditory cortex, fMRI, naïve bayesian gaussian model, support vectors machine, logistic regression

Table des matières

Liste des tableaux

Table des figures

Liste des Abbréviations

fMRI :	Imagerie fonctionnelle par Résonance Magnétique (functional Magnetic Resonance Imaging)
MVS :	Machine à Vecteurs de Support
RL :	Régression Logistique
BGN :	modèle Bayésien Gaussien Naïf
FE :	Filet Élastique

Liste des Symboles

mm :	Millimètre
cm :	Centimètre
s :	Seconde
ms :	Milliseconde
Hz :	Hertz
H_0 :	Hypothèse nulle

Je dédie ce mémoire à Mme Colette Messier

Remerciements

J'aimerais d'abord remercier ma famille et mes amis pour leur soutient moral constant pendant ce parcours à la maîtrise.

Certaines personnes ont été d'une aide précieuse dans ce projet par leurs conseils, écoute et assistance. Je nomme Robert Zatorre, Hugo Larochelle, Nicolas Chapados, Sean Wood, Frederic Bastien, Patrick Bermudez, Aaron Courville et Dumitru Erhan.

Finalement, pour leur appui financier et leur investissement personnel dans ce projet, j'aimerais remercier chaleureusement mes superviseurs, soit Pascal Vincent, Douglas Eck et Marc Schönwiesner.

Introduction

La technologie d'**imagerie fonctionnelle par résonance magnétique** (fMRI) compte parmi les techniques les mieux adaptées pour mesurer l'activité cérébrale chez l'humain. Bien qu'imprécise par rapport à d'autres techniques de mesure, elle a l'avantage d'être **non-invasive**, i.e. ne nécessite pas l'ouverture du crâne. Elle permet de mesurer la consommation d'oxygène dans les petits capillaires situés à proximité des populations de neurones, en plongeant la région du cerveau ciblée dans un champ magnétique de haute intensité.

La résolution spatiale des voxels des images peut aller jusqu'à l'ordre du millimètre, ce qui est précis parmi les techniques d'imagerie non-invasives, mais très imprécis comparativement à l'échelle du neurone, qui relève davantage du micromètre. Cela pose évidemment des limites quant aux études qui peuvent être menées. Lorsque que les structures neuronales qu'on cherche à étudier sont petites par rapport à la résolution fMRI, elles induisent au mieux des réponses très faibles et difficiles à distinguer du bruit par les techniques d'analyse standard.

Récemment, de nouvelles percées au niveau des techniques d'analyse des données fMRI [8] ont permis de repousser en partie les limites liées à la résolution spatiale. En utilisant des techniques d'analyse plus souples, basées sur des **classifieurs linéaires**, il semble qu'on puisse différencier les réponses faibles induites par les petites structures du bruit. Ces classifieurs combinent linéairement les différentes unités de mesure fMRI pour catégoriser différents états du cerveau. De bonnes performances de prédiction indiquent la présence de patterns significatifs dans les réponses fMRI.

Les classifieurs linéaires font partie d'une grande famille d'algorithmes appelée **apprentissage automatique**. Ces algorithmes se veulent une imitation du processus d'apprentissage par exemple chez l'humain et consistent à extraire automatiquement des patterns pertinents présents dans des ensemble de données (i.e. ensemble d'exemples). Dans le cas des classifieurs linéaires, les patterns repérés sont des combinaisons linéaires de voxels.

L'usage des classifieurs linéaires sur données fMRI est devenu de plus en plus populaire ces dernières années. Notamment, assez récemment, ils ont permis une avancée importante dans l'étude de la perception visuelle. En les employant, Kamitani et Tong (2005)[8] sont arrivés à localiser les filtres d'orientation dans le cortex visuel primaire humain. Ces filtres font partie des premiers stades de la perception visuelle. Ils permettent de capturer les modulations spatiales de luminosité dans les images projetées sur la rétine.

Les données fMRI recueillies pour cette étude se prêtaient mal aux méthodes d'analyse standard. En effet, les filtres d'orientation sont des structures neuronales beaucoup plus petites que la résolution fMRI et, de surcroît, sont distribués aléatoirement à travers les différentes unités de mesure fMRI. Ils induisent de faibles réponses éparpillées dans le cortex visuel primaire qui, isolément, sont statistiquement peu significatives. En combinant ces réponses à l'aide d'un classifieur linéaire, Kamitani et Tong arrivent à prédire le stimulus perçu par le sujet avec un haut pourcentage d'efficacité.

Dans le présent projet, nous nous proposons de répliquer l'analyse de Kamitani et Tong au niveau

de la perception auditive. L'existence de filtres analogues aux filtres d'orientation dans le cortex auditif primaire humain a été postulée (Shamma [19]). Ceux-ci détecteraient, quant à eux, les modulations spectrales et temporelles présentes dans les sons. Nous utilisons les enregistrements fMRI de sujets soumis à 49 modulations spectro-temporelles. Ceux-ci proviennent d'une étude effectuée par Schönwiesner (2009) [14]. Tout comme les données relatives aux filtres d'orientations, ces données sont difficiles à analyser avec les techniques standard.

Nos buts à travers ce projet sont multiples. D'une part, nous cherchons à mieux comprendre comment le cerveau humain encode les sons. C'est-à-dire, nous cherchons à localiser la/les régions d'encodage des modulations spectro-temporelles. D'autre part, un autre objectif important de ce projet, à plus long terme, est l'amélioration des techniques d'analyse des données fMRI. Nous observons scrupuleusement le comportement des différentes techniques et algorithmes utilisés et dégageons quelques pistes de recherche.

Ce mémoire est divisé en 5 chapitres. Le chapitre 1 est consacré à l'hypothèse d'encodage des sons postulée et aux données de Schönwiesner. Le chapitre 2 consiste en une introduction aux données fMRI. Nous résumons les caractéristiques de ces données et décrivons les différentes étapes du processus d'analyse standard. Au chapitre 3, nous nous attardons sur le potentiel des classifieurs linéaires pour l'analyse des données fMRI. Nous décrivons comment on peut les utiliser dans le cadre de cette application et discutons des principaux défis posés par les données fMRI. Au chapitre 4, nous présentons les différentes expériences réalisés dans le cadre de ce projet. Certaines ont surtout pour but d'étudier l'efficacité et la pertinence des techniques d'analyse à base de classifieurs linéaires. D'autres ont pour but de localiser la région d'encodage des modulations spectro-temporelles. Finalement, au chapitre 5, nous présentons et analysons les résultats de ces expériences. Nous discutons aussi de quelques pistes de recherche intéressantes pour le futur.

Chapitre 1

Encodage des sons complexes dans le cerveau humain

Un des buts du présent projet est de mieux comprendre comment le cerveau humain encode les **sons complexes**. Par l'expression son complexe, nous faisons ici référence aux sons auxquels nous sommes exposés dans la vie de tous les jours.

À la Figure 1.1, on peut apercevoir des représentations visuelles de quelques sons assez communs. Celles-ci indiquent, à l'aide d'un code de couleur, comment le contenu spectral du son varie dans le temps. Elles sont communément appelées **spectrogrammes**. L'axe horizontal représente le temps. L'axe vertical représente les fréquences et est aussi appelé axe spectral. Ces deux axes forment ce qu'on appelle l'espace spectro-temporel.

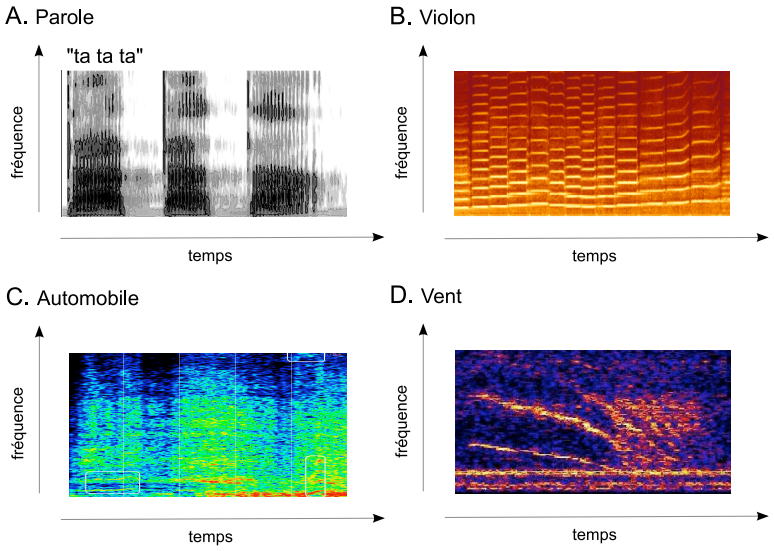


FIGURE 1.1: Spectrogrammes de quelques sons communs. Un spectrogramme est une représentation visuelle qui montre comment le contenu spectral d'un son varie dans le temps. L'axe horizontal représente le temps. L'axe vertical représente les fréquences. Les couleurs encodent, en chaque instant, les proportions des différentes fréquences dans le signal sonore. A. Parole ("ta ta ta"). B. Son d'un violon C. Bruit d'une automobile. D. Bruit du vent.

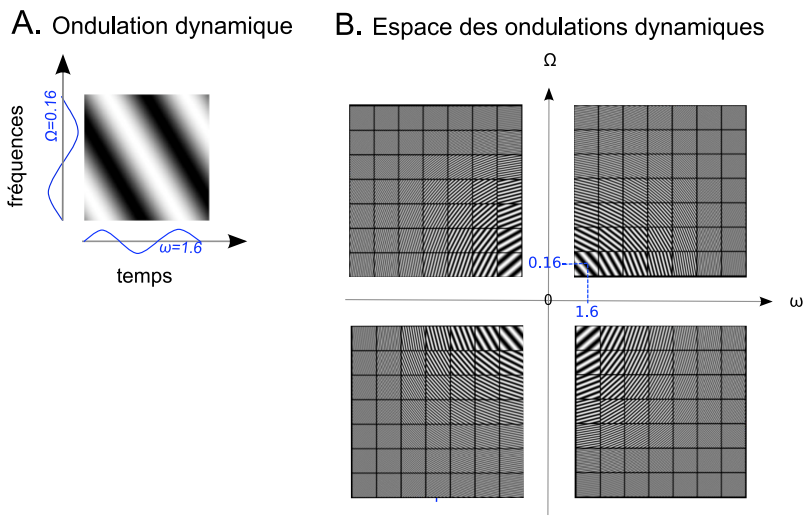


FIGURE 1.2: A. Spectrogramme d'une ondulation dynamique. L'axe horizontal représente le temps, l'axe vertical représente les fréquences à l'échelle logarithmique (octaves). Mathématiquement, dans l'espace spectro-temporel, les ondulations dynamiques sont des sinus bidimensionnels. Une modulation de fréquence $\Omega = 0.16$ est appliquée sur l'axe spectral. Une modulation de fréquence $\omega = 1.6$ est appliquée sur l'axe temporel. B. Spectrogrammes d'ondulations dynamiques positionnés selon leurs modulations spectrales Ω et temporelles ω. Ces deux paramètres définissent un espace mathématique, l'espace des ondulations dynamiques.

Ce qu'on appelle une **modulation spectro-temporelle** est un pattern sinusoïdal dans l'espace spectro-temporel. Les hachures dans les spectrogrammes de la Figure 1.1 en sont des exemples. On remarque qu'elles sont orientées selon différents angles et qu'elles sont de résolutions plus ou moins fines.

Selon une hypothèse d'encodage inspirée d'études sur les animaux et d'études sur la perception visuelle, le cerveau humain exploiterait ces modulations pour caractériser les sons présents dans l'environnement (Shamma [19]). Un tel encodage serait réalisé par une banque de neurones réglés à différentes modulations spectro-temporelles. Ceux-ci seraient localisés en bordure des régions d'encodage du spectrogramme auditif et seraient les homologues des **filtres d'orientation**, des neurones encodant les modulations spatiales présentes dans les images de la rétine.

Un excellent stimulus pour addresser la question de la sensibilité aux modulations spectro-temporelles est l' **ondulation dynamique** (Shamma [15]). Il s'agit en fait d'un cas particulier de modulation spectro-temporelle. Mathématiquement, le spectrogramme (Figure 1.2A) de ce son est un sinus bidimensionnel pur, défini par deux paramètres Ω et ω. Le paramètre Ω est la fréquence de la modulation appliquée sur l'axe spectral. Le second paramètre, ω, est la fréquence de la modulation appliquée sur l'axe temporel. L'ensemble des valeurs possibles pour les paramètres Ω et ω définit un espace mathématique, soit l'**espace des ondulations dynamiques**. À la Figure 1.2B, les spectrogrammes de différentes ondulations dynamiques sont positionnés dans cet espace.

Plusieurs études ont été réalisées chez les animaux à l'aide d'ondulations dynamiques. Les études de Shamma [15, 16, 17, 18, 19], effectuées sur des furets, sont particulièrement détaillées et ont eu beaucoup d'impact dans la communauté neuroscientifique. Des groupes de neurones réglés à des modulations

spectro-temporelles variées dans le cortex auditif primaire ont été repérés. Nous présentons ces études à la section 1.1.

Une étude a été réalisée récemment (2009) par Schönwiesner [14] chez les humains. Celle-ci est fortement inspirée des études de Shamma. Cette fois, cependant, on mesure les réponses de larges populations de neurones (fMRI) au lieu des réponses de neurones isolés. Selon Schönwiesner, il semble que les observations de Shamma soient aussi applicable à large échelle, i.e., on observe des populations de neurones réglées à des familles de modulations spectro-temporelles variées. Cela suggère l'existence de structures spécialisées pour des familles de sons catégorisées par leur contenu en modulations spectro-temporelles.

Dans le présent projet, nous analysons précisément les données de l'étude de Schönwiesner. En utilisant des techniques d'analyses basées sur des classifieurs linéaires, nous espérons repérer la position de la région d'encodage des modulations spectro-temporelles dans le cerveau humain. À la section 1.2, nous décrivons plus en détails l'étude de Schönwiesner. Nous introduisons aussi brièvement notre propre méthode d'analyse des données.

1.1 Études chez les furets

Plusieurs études faisant appel aux ondulations dynamiques ont été menées par Shamma [15, 16, 17, 18, 19], sur des furets. Les réponses de neurones isolés ont été enregistrées dans le cortex auditif primaire, à l'aide d'une microélectrode. Dans la région d'encodage, on s'attend à trouver un riche amalgame de neurones réglés à des modulations spectro-temporelles variées. Nous donnons ici davantage de précisions concernant la nature des données et résumons les principales conclusions des différentes études.

1.1.1 Acquisition des réponses des neurones

Shamma enregistre les réponses de neurones du cortex auditif primaire à différentes ondulations dynamiques. Des mesures de voltage sont prélevées à l'aide d'une microélectrode insérée à l'intérieur des neurones.

À la base, il existe plusieurs types de "réponses", i.e. plusieurs manières d'encoder de l'information dans le cerveau. Shamma s'intéresse surtout à la fréquence d'émission des **potentiels d'action**. Un potentiel d'action est une augmentation importante et brève du voltage qui provoque une transmission de courant par le neurone.

À partir des mesures de voltage, on détecte et dénombre les potentiels d'action à des intervalles de temps régulier, tel qu'illustré à la Figure 1.3.

1.1.2 Réponses synchrones et préférentielles aux ondulations dynamiques

À travers plusieurs études [15, 16, 17, 18], en analysant la réponse (fréquence des potentiels d'action) de neurones isolés dans le temps à différentes ondulations dynamiques, Shamma et son équipe ont reperé des neurones montrant un comportement synchrone et préférentiel par rapport aux ondulations dynamiques.

La réponse temporelle de ces neurones, aurait la forme d'un sinus de même fréquence que la modulation temporelle de l'ondulation dynamique utilisée. (Figure 1.4A). Un comportement préférentiel par rapport à certaines ondulations dynamiques se manifesterait dans l'amplitude du sinus. Celle-ci

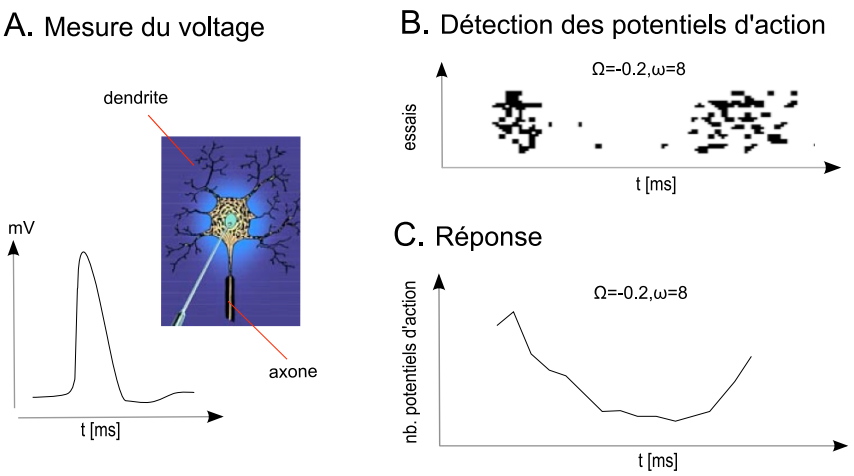


FIGURE 1.3: Acquisition des réponses des neurones. A. Une microélectrode permet de percer sans dommages la membrane d'un neurone et de mesurer le voltage (en mV) à l'intérieur. Si les dendrites (entrées) sont suffisamment excitées, un potentiel d'action, i.e. augmentation brusque du voltage, est généré et il y a transmission de courant dans l'axone (sortie). B. Pour un stimulus et un neurone donné, les potentiels d'action sont détectés pour 15 essais différents. L'axe horizontal indique le temps, en millisecondes. Chaque rangée correspond à un essai. Chaque point représente un potentiel d'action. (Image modifiée, Shamma (2001) [19]) C. La réponse du neurone est définie en fonction de la fréquence des potentiels d'action. Plus précisément, à intervalles de temps réguliers, on dénombre les potentiels d'action à travers les 15 essais. (C : Image modifiée, Shamma (2001) [19].)

correspondrait à une fonction $T(\Omega, \omega)$ qui dépend de la modulation spectrale Ω et de la modulation temporelle ω. L'amplitude ne serait raisonnablement élevée que pour des modulations spectro-temporelles précises, focalisées dans l'espace des ondulations dynamiques.

Les réponses des neurone seraient à peu près linéaires, i.e. la réponse à un ensemble d'ondulations dynamiques correspondrait environ à la somme des réponses individuelles. Quelques non-linéarités seraient présentes, mais auraient un impact minime sur les résultats de l'analyse. Celles-ci seraient principalement dûes à la saturation et la positivité des réponses, i.e. aux limites naturelles des neurones.

Afin d'obtenir un estimé complet de la fonction $T(\Omega, \omega)$, il faudrait en théorie considérer la réponse du neurone à toutes les combinaisons (Ω, ω), ce qui est peu réaliste. En effet, on ne peut maintenir les furets sous anesthésie que pour un temps relativement court. Une série d'études ont été consacrées à la démonstration d'une seconde propriété importante des réponses des neurones, soit la séparabilité spectro-temporelle. Cette propriété signifie que la fonction $T(\Omega, \omega)$ peut s'écrire comme un produit de deux fonctions indépendantes $F(\Omega)$ et $G(\omega)$:

$$T(\Omega, \omega) = F(\Omega)G(\omega)$$

Lorsqu'elle prévaut, le temps de calcul de $T(\Omega, \omega)$ est considérablement réduit, car les fonctions spectrale $F(\Omega)$ et temporelle $G(\omega)$ peuvent être estimées séparément. Selon les conclusions de Shamma, il semble que la séparabilité complète ne soit pas applicable, mais qu'il y ait bien séparabilité à l'intérieur d'un même quadrant dans l'espace des ondulations dynamiques. (Figure 1.4B).

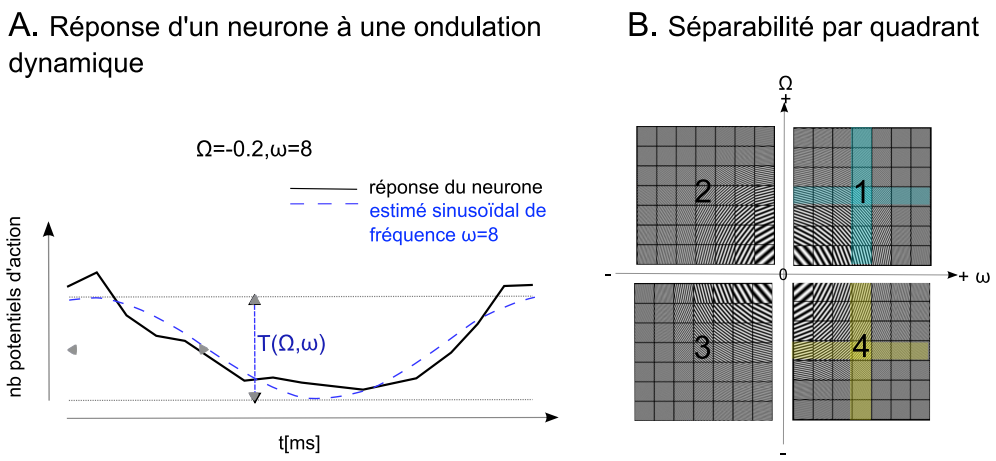


FIGURE 1.4: A. Le trait noir indique la réponse d'un neurone à une ondulation dynamique de paramètres $\Omega = -0.2$ et $\omega = 8$. Le trait pointillé bleu indique l'estimé sinusoïdal de la réponse du neurone. La fréquence de ce sinus correspondrait approximativement à ω, la fréquence de la modulation temporelle. Le comportement préférentiel du neurone serait encodé dans l'amplitude $T(\Omega, \omega)$ du sinus. (Image modifiée, Shamma (2001) [[19]]). B. Points de tests (Ω, ω) requis pour un estimé total de la fonction $T(\Omega, \omega)$, lorsque la séparabilité par quadrant est applicable. Les deux axes représentent les modulations spectrale Ω et temporelle ω définissant l'espace des ondulations dynamiques. Les ondulations dynamiques colorées en turquoise servent à estimer la fonction $T(\Omega, \omega)$ dans le quadrant 1 et aussi dans le quadrant 3 qui lui est symmétrique. Les ondulations dynamiques colorées en jaune servent à estimer la fonction $T(\Omega, \omega)$ dans les quadrant 2 et aussi dans le quadrant 4 qui lui est symmétrique.

1.1.3 Sensibilité des neurones aux ondulations dynamiques

Une fois les propriétés de linéarité et de séparabilité par quadrant validées, les fonctions $T(\Omega, \omega)$, définissant les sensibilités aux ondulations dynamiques, ont été estimées pour un bon nombre de neurones. Ces dernières se sont avérées très diversifiées et souvent focalisées autour de modulations (Ω, ω) caractéristiques. À la Figure 1.5A, on aperçoit la fonction $T(\Omega, \omega)$ estimée pour un neurone en particulier. Il semble y avoir une préférence pour les modulations spectrales très basses. En effet, il y a focalisation en $\Omega = 0$. Il semble aussi y avoir une préférence pour les modulations temporelles d'amplitude 8 Hz. En effet, on observe une double focalisation en $\omega = 8$ et $\omega = -8$.

À la Figure 1.5B, on aperçoit les spectrogrammes "préférés" de quelques neurones, i.e. les spectrogrammes correspondant à une combinaison linéaire des spectrogramme des ondulations dynamiques selon les sensibilités $T(\Omega, \omega)$:

$$S(\mathbf{x}, \mathbf{t}) = \Sigma_{\Omega,\omega}\, T(\Omega, \omega)\, S_{\Omega,\omega}(\mathbf{x}, \mathbf{t})$$

Ici, $S_{\Omega,\omega}(\mathbf{x}, \mathbf{t})$ désigne le spectrogramme de l'ondulation dynamique de modulation spectrale Ω et de modulation temporelle ω.
On remarque la présence des ondulations 2d caractéristiques aux ondulations dynamiques. Les résolutions et directions de ces ondulations varient d'un neurone à l'autre.

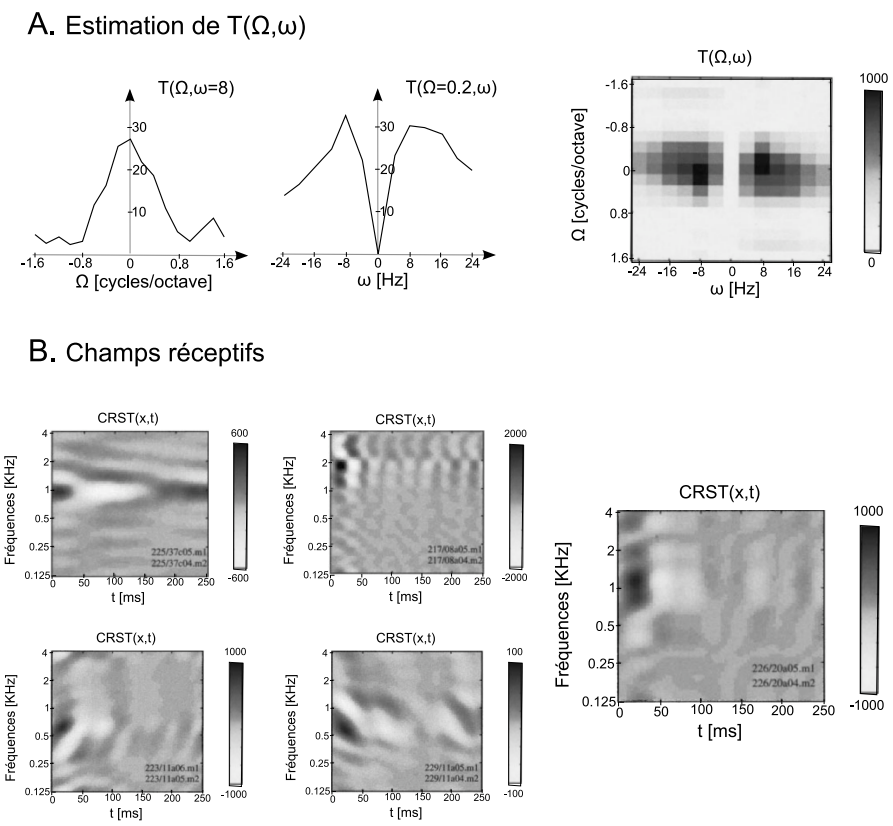


FIGURE 1.5: A. Exemple d'estimation de la fonction $T(\Omega, w)$. On suppose que la séparabilité par quadrant est applicable et qu'on peut décomposer la fonction de transfert $T(\Omega, \omega)$ en deux composantes indépendantes $F(\Omega)$ et $G(\omega)$. À gauche, l'estimé de $F(\Omega)$. Au centre, l'estimé de $G(\omega)$. À droite, la fonction $T(\Omega, \omega)$ complète obtenue en combinant les estimés de F et G. B. Spectrogrammes obtenus en combinant linéairement les ondulations dynamiques selon les préférences des neurones. Le spectrogramme correspondant à l'exemple en A est sur la droite. Sur la gauche, 4 spectrogrammes associés à 4 neurones différents. On remarque la présence d'ondulations 2d similaires aux ondulations dynamiques en certains endroits. (Images modifiées, Shamma (2001) [19].)

1.2 Étude chez les humains

Une nouvelle étude faisant appel à des ondulations dynamiques a été réalisée récemment (2009) par Schönwiesner [14]. Tout comme les études de Shamma (section 1.1), le but de cette étude est de valider l'hypothèse d'encodage présentée au début de ce chapitre. Cette fois, cependant, les sujets sont des humains.

Les données propres à cette étude sont très différentes des données recueillies par Shamma. En effet, pour des raisons éthiques évidentes, on ne peut procéder à une ouverture de la boîte crânienne et à l'exposition du cortex comme on le fait pour les animaux. Aussi, l'activité neuronale a été mesurée au moyen d'une technique dite non-invasive, soit l'imagerie par résonance magnétique fonctionnelle (fMRI). La résolution spatiale associée à cette technique est nettement plus grossière. Chaque unité de volume (voxel), en théorie, peut contenir plusieurs millions de neurones. Shamma enregistre quant à lui les potentiels d'action émis par des neurones isolés.

Le cadre d'analyse rappelle celui utilisé par Shamma ; pour chaque unité de volume (voxel), on mesure les sensibilités aux différentes ondulations dynamiques. Néanmoins, les mesures de sensibilité ne répondent plus à la même définition et se prêtent à une analyse à plus large échelle, i.e. à l'analyse

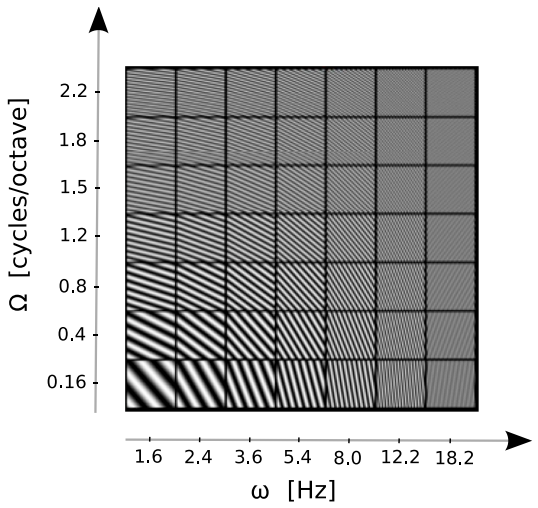


FIGURE 1.6: Spectrogrammes des ondulations dynamiques utilisées par Schönwiesner positionnés selon leurs modulations spectrales Ω et temporelles ω. En tout, 49 ondulations dynamiques différentes ont été utilisées. Leurs modulations spectrales varient entre 0.16 cycles/octave et 2.2 cycles/octave et leurs modulations temporelles entre 1.6 Hz et 18.2 Hz.

du comportement de larges populations de neurones.

Ici, nous présentons les données, méthodes d'analyse et principales conclusions de l'étude de Schönwiesner. Comme nous utilisons précisément les données de cette étude dans le présent projet, nous introduisons aussi brièvement notre propre méthode d'analyse.

1.2.1 Données fMRI

La technique d'**imagerie par résonnance magnétique fonctionnelle** (**fMRI**) est liée aux besoins en oxygène des neurones. Lorsqu'une population de neurones est très active, davantage d'oxygène est consommé. Dans les petits vaisseaux sanguins à proximité, cette consommation accrue perturbe les propriétés magnétiques de l'hémoglobine. Lors d'une scannographie fMRI, on cherche à capter la force de ces perturbations en plongeant la région du cerveau ciblée dans un champ magnétique de haute intensité. Les unités de volume pour lesquelles les mesures sont effectuées sont appelées **voxels**.

Les réponses fMRI de 7 sujets exposés à 49 ondulations dynamiques différentes ont été acquises (Figure 1.6). Les modulations spectrales Ω utilisées varient entre 0.16 cycles/octave et 2.2 cycles/octave. Les modulations temporelles ω utilisées varient quant à elles entre 1.6 Hz et 18.2 Hz.

Comme mentionné précédemment, les mesures fMRI ne peuvent être comparées à des enregistrements de potentiels d'action effectués sur des neurones isolés. D'abord, l'unité de volume (voxel), a une résolution de l'ordre du millimètre cube. Comme la grosseur d'un neurone est de l'ordre du micromètre cube, un seul voxel peut donc contenir, en théorie, des millions de neurones. De plus, la force des perturbations magnétiques causées par la consommation d'oxygène est une mesure bruitée, non-linéaire et délayée de l'activité électrique des populations de neurones.

Pour de plus amples explications quant à la nature des données fMRI, prière de se référer au chapitre 2. Nous donnons aussi davantage détails sur les données spécifiques à l'étude de Schönwiesner à la section 4.2.

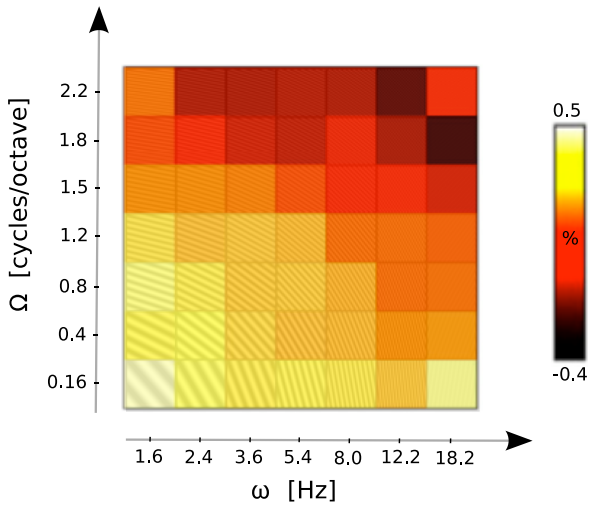


FIGURE 1.7: Répartition des préférences aux différentes ondulations dynamiques, tous voxels confondus. Pour chaque ondulation dynamique, i.e. chaque combinaison (Ω, ω), un code de couleur indique le nombre de préférences induites dans les voxels, ramené sur une échelle entre -0.4 et 0.5. Il semble que le nombre de préférences diminue au fur et à mesure que les paramètres Ω et ω augmentent. (Image modifiée, Schönwiesner (2009) [14].)

1.2.2 Analyse des données par Schönwiesner

Nous exposons ici les résultats qui appuient l'hypothèse d'encodage postulée. Comme mentionné précédemment, les données se prêtent à une analyse à large échelle, puisque les réponses des voxels réflètent le comportement de larges populations de neurones.

Similairement à Shamma, l'analyse de Schönwiesner est basée sur l'analyse de la **sensibilité** des unités de volumes, i.e. voxels, aux différentes ondulations dynamiques. En guise de mesures de sensibilité, on utilise, en un voxel donné, les réponse moyennes à chacune des ondulations dynamiques. On détermine les **préférences** des voxels : pour chaque voxel, on détermine l'ondulation dynamique pour laquelle la sensibilité (i.e. réponse moyenne) du voxel est la plus forte. Schönwiesner analyse la répartition de ces préférences dans l'espace des ondulations dynamiques et dans le cerveau.

La figure 1.7 montre par combien de voxels chaque ondulation dynamique est préférée. Il semble que le nombre de préférences induites dans les voxels diminue au fur et à mesure que les paramètres Ω et ω augmentent, i.e. au fur et à mesure que les fréquences des modulations spectrales et temporelles augmentent. Coïncidence intéressante, ce gradient réfléterait assez bien la distribution des modulations spectro-temporelles dans les sons présents dans notre environnemment.

La figure 1.8 illustre quant à elle la distribution des préférences à travers les lobes temporaux des sujets. Plus précisément, on aperçoit les lobes temporaux de 7 sujets différents. L'ondulation dynamique préférée en chaque voxel est encodée selon un code de couleur réflétant sa position dans l'espace des ondulations dynamiques.

Les préférences semblent regroupées en agrégats nets et uniformes. Il semble que, si deux voxels sont proches dans l'espace, alors leurs préférences ont aussi de bonnes chances d'être voisines dans l'espace des ondulations dynamiques. Ces corrélations spatiales sont intéressantes, néanmoins, il est difficile de savoir si elles sont significatives à l'aide d'une analyse traditionnelle, car les préférences, isolément, sont statistiquement très faibles.

Il y a beaucoup de variabilité d'un sujet à l'autre. Pour tous les sujets, cependant, peu de voxels montrent une préférence pour les ondulations dynamiques associées à des modulations spectrale et

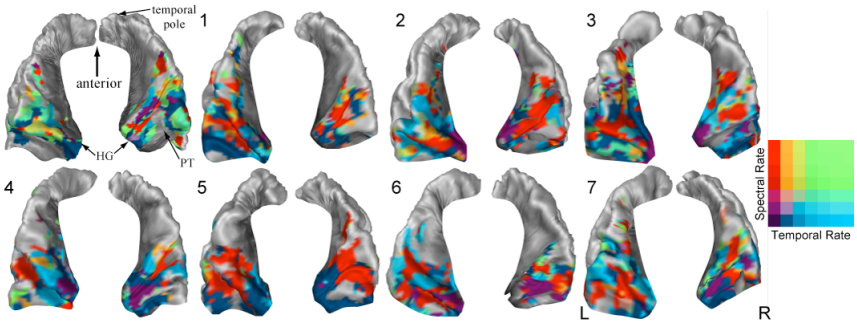


FIGURE 1.8: Lobes temporaux de 7 sujets différents. Les lobes en haut à gauche sont associés au groupe entier. L'ondulation dynamique préférée en chaque voxel est encodée selon un code de couleur réflétant sa position dans l'espace des ondulations dynamiques (grille de droite). Les préférences semblent organisées spatialement, i.e. forment des agrégats assez nets. Elles sont aussi très variables d'un sujet à l'autre. (Image tirée de Schönwiesner (2009) [14].)

temporelle toutes deux élevées (en vert).

Schönwiesner dresse aussi un inventaire du comportement des voxels face aux différentes ondulations dynamiques. Plus précisément, les voxels sont regroupés en catégories, selon leur type de sensibilité. On observe des patterns de focalisation similaires à ceux observés par Shamma (Figure 1.9A). On observe aussi d'autres patterns intéressants, notamment la focalisation sur un seul axe, Ω ou ω (Figure 1.9BC), la focalisation selon une direction arbitraire (Figure 1.9D) et finalement la focalisation multiple (Figure 1.9E).

De tels patterns de focalisation impliquent que les structures neuronales à l'intérieur des voxels associés sont fortement corrélées entre elles dans leurs réponses aux ondulations dynamiques. En effet, dans le cas contraire, les réponses moyennes estimées (sensibilités) seraient probablement disparates. Mentionnons que focalisation, ici, n'est pas nécéssairement synonyme d'appartenance à la région d'encodage. En fait, il n'est pas impossible que, dans la région d'encodage, les sensibilités des voxels aient justement une apparence plus disparate. Cependant, l'existence d'une organisation spatiale des populations de neurones selon des modulations spectro-temporelles caractéristiques n'en demeure pas moins un signe favorable à l'existence de la région d'encodage.

1.2.3 Notre analyse

Afin de valider statistiquement ses résultats, Schönwiesner s'en est remis à la stabilité des réponses moyennes des voxels. Plus précisément, on a partitionné les données en deux ensembles disjoints et vérifié si les estimés obtenus à partir des deux ensembles étaient similaires. On a répété cette procédure plusieurs fois sur différentes partitions. Selon Schönwiesner, les différences d'un estimé à l'autre seraient minimes.

Il s'agit d'une façon de faire non-standard sur données fMRI. En effet, une analyse sur données fMRI traditionnelle est plutôt basée sur les contrastes entre les réponses moyennes des voxels sous différentes conditions de stimulation. Plus le contraste est de forte amplitude, plus sa significativité est élevée, plus le voxel est considéré comme important.

Les données de Schönwiesner se prêtent mal à ce genre d'analyse, car les différences entre les conditions de stimulation ne semblent pas se présenter sous forme de contrastes marqués entre les réponses moyennes. En effet, les préférences des voxels, isolément, sont très faibles et peu significatives. Il se peut, cependant, qu'elles le soient dans leur ensemble. En effet, les corrélations spatiales observées

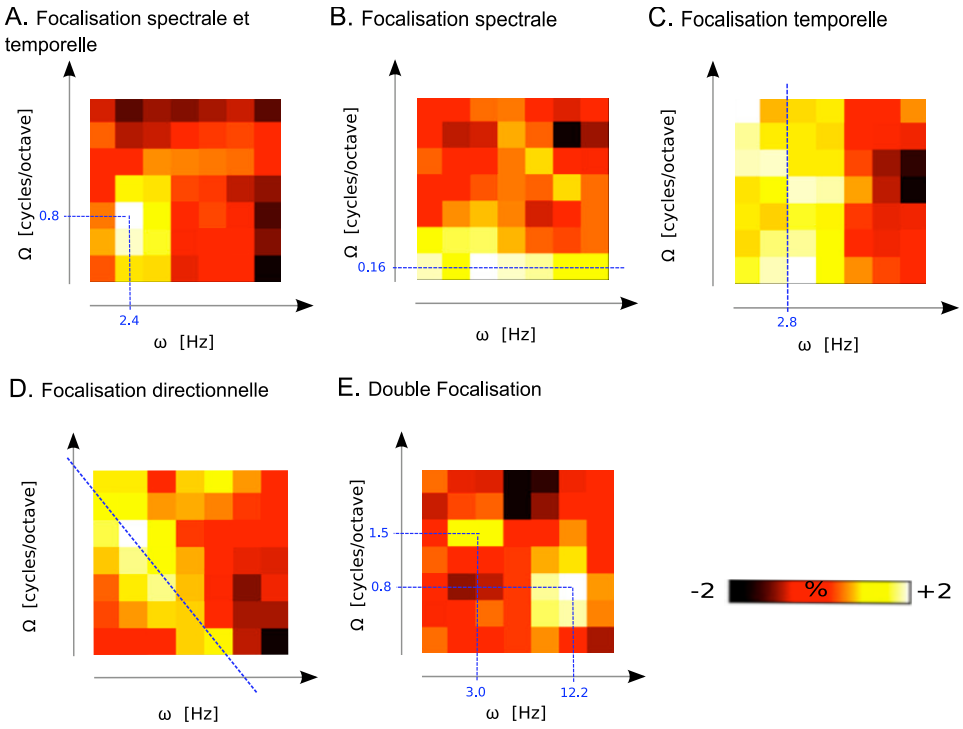


FIGURE 1.9: Exemples de focalisation des réponses des voxels dans l'espace des ondulations dynamiques. A. Focalisation de la réponse du voxel autour du point ($\Omega = 0.8, \omega = 2.4$) Les carreaux représentent les différentes ondulations dynamiques, positionnées selon leurs paramètres Ω et ω, i.e. selon leurs modulations spectrales et temporelles. Les couleurs des carreaux représentent les réponses fMRI moyennes (ré-échelonnées entre -2 et 2) du voxel aux ondulations dynamiques correspondantes. B. Focalisation de la réponse du voxel sur l'axe Ω seulement, en $\Omega = 0.16$. C. Focalisation de la réponse du voxel sur l'axe ω seulement, en $\omega = 2.8$. D. Focalisation le long d'une direction arbitraire. E. Double focalisation. (Images modifiées, Schönwiesner (2009) [14].)

entre les voxels sont prometteuses. En fait, même de faibles préférences spatialement désorganisées, ici, pourraient contenir de l'information. En effet, les structures soupçonnées d'encoder les modulations spectro-temporelles seraient très petites par rapport aux voxels et pourraient être distribuées aléatoirement dans les voxels.

Pour notre part, nous nous proposons d'utiliser des classifieurs linéaires pour analyser les données. Ces classifieurs combinent les réponses des voxels (linéairement) pour prédire la condition de stimulation. Une bonne performance de prédiction est un indicateur de la présence de patterns pertinents dans les réponses des voxels, i.e. des patterns stables et fiables.

Tout comme l'approche de Schönwiesner, il s'agit d'une approche non-traditionnelle, néanmoins nous avons bon espoir d'en tirer des résultats intéressants. En effet, récemment (2005), une telle approche a été utilisée par Kamitani et Tong [8] sur des données fMRI similaires, dans une étude portant sur la perception visuelle. Ils sont arrivés à localiser les filtres d'orientation, de petites structures neuronales détectant des modulations spatio-temporelles dans les images. Tout comme Kamitani et Tong, en repérant les ensembles de voxels associés aux meilleures performances de classification, nous aspirons à localiser la/les région(s) d'encodage.

Chapitre 2

Analyse traditionnelle des données fMRI

Nous effectuons ici un survol de certains aspects clé liés à la technique d'imagerie cérébrale fMRI. Nous résumons les contraintes régissant les résolutions spatiale et temporelle, de même que certains designs d'acquisition de base. Ensuite, nous présentons le processus d'analyse des données fMRI, des données brutes à la détection des **régions d'intérêt**, i.e. les régions du cerveau impliquées dans la tâche cognitive étudiée. Nous discutons ensuite brièvement des limites de l'analyse standard.

Pour davantage de détails concernant les processus d'analyse fMRI, voir Friston (1995) [6] ou Jezzard (2001) [7].

2.1 Résolutions spatiale et temporelle

Les mesures prises en chacune des unités de volume, i.e., les **voxels**, dépendent de changements locaux dans la réponse hémodynamique. Une activité neuronale importante induit généralement une augmentation du débit sanguin et de la consommation locale d'oxygène. Ces changements modifient localement les propriétés magnétiques de l'hémoglobine. Ce sont ces micros perturbations, précisément, qui sont captées durant la scannographie fMRI.

La résolution spatiale fMRI peut être aussi fine que 1 mm, cependant, les très fines résolutions ne sont pas toujours un choix qui est indiqué en pratique. Le cas échéant, on doit être prêt à tolérer des mesures plus faibles et plus bruitées. En effet, les structures neuronales mesurées étant plus petites, les perturbations magnétiques qu'elles créent sont plus faibles et plus difficiles à capter. De plus, les mesures deviennent aussi davantage sensibles aux mouvements du sujet pendant et entre les scans.

Généralement, on préfère effectuer les scans assez rapidement, afin de diminuer le risque qu'il y ait mouvement du sujet pendant l'acquisition. Cette contrainte de temps limite la résolution spatiale. En fait, l'image 3d finale est obtenue par l'acquisition d'une série d'images (tranches) en deux dimensions. En général, plus les images 2d sont de haute résolution, plus le scanner met du temps à les acquérir et plus il y a d'images, plus il faut du temps.

La résolution temporelle du signal dépend largement du design de l'expérience. Dans certains designs, on capte la réponse d'un seul stimulus à la fois, lorsque la réponse hémodynamique qu'il induit est à son maximum. Dans ce cas, il faut beaucoup de temps pour un seul stimulus puisque la réponse hémodynamique peut facilement s'étaler sur une dizaine de secondes. D'autres designs misent plutôt sur des scans très rapides et très fréquents (ms) pour capturer la réponse hémodynamique dans son ensemble. On suppose que les réponses hémodynamiques aux stimuli sont consistentes d'une stimulation à l'autre et s'additionnent linéairement. On présente alors les stimuli à vitesse élevée, dans un ordre aléatoire.

2.2 Étapes de prétraitements

2.2.1 Réalignement spatial

Typiquement, le sujet bouge légèrement d'un scan à l'autre. Avant de procéder à une analyse des données, on applique généralement une procédure de réalignement qui fait en sorte qu'on peut raisonnablement assumer que les coordonnées spatiales d'un voxel demeurent les mêmes d'un scan à l'autre.

Il existe plusieurs procédures, dont la plus répandue consiste à aligner chacun des volumes sur un volume de référence à l'aide d'une transformation linéaire. Les 6 paramètres de la transformation, soit 3 rotations et 3 translations, sont obtenus en minimisant la distance au carré avec le volume de référence. Ce dernier est généralement choisi parmis les premiers volumes, quoi que, dans certains cas, on lui préfère le volume moyen. Cette transformation est dite rigide parce qu'elle pré-suppose que la forme de la tête est constante d'un scan à l'autre.

Une fois la transformation estimée, on crée une nouvelle image pour laquelle les coordonnées des voxels sont identiques à ceux du volume de référence. Le signal en chacun des voxels du nouveau volume est estimé par interpolation avec les voxels voisins dans le volume original. Cette interpolation, introduisant des corrélations spatiales, consiste, en quelque sorte, en un premier lissage des données.

Malgré cette procédure de réalignement, les positions de différents voxels sont rarement exactes. D'abord, on ne tient pas compte des déformations non-linéaires, i.e. on suppose que la forme de la tête demeure constante d'un scan à l'autre. Or, on sait, entre autres, que la respiration induit de légères déformations. De plus, on assume généralement qu'il n'y a pas de mouvement durant l'acquisition d'un volume, ce qui est plutôt improbable. De tels mouvements existent et décalent légèrement les coordonnées des voxels à l'intérieur d'un même volume.

2.2.2 Lissage

Habituellement, les données sont lissées spatialement à l'aide d'un kernel sphérique gaussien.

Lorsque la largeur du kernel est proche de la variabilité spatiale naturelle de la réponse fMRI, cette opération est bénéfique, éliminant le bruit et uniformisant la réponse d'un scan à l'autre. Notamment, il peut être avantageux d'effectuer un léger lissage avant la procédure de réalignement spatial, afin de compenser pour l'imprécision spatiale naturelle dûe aux mouvements et déformations de la tête du sujet. Aussi, lorsqu'on cherche à comparer les données relatives à plusieurs sujets, on choisit généralement un kernel assez large, la variabilité spatiale étant amplifiée par les différences anatomiques d'un sujet à l'autre.

Le lissage peut cependant effacer de l'information importante lorsque le kernel est trop large. Notamment, on risque d'effacer les réponses induites par les petites structures neuronales, souvent plus faibles et spatialement désorganisées.

2.2.3 Detrending Temporel

Généralement, les réponses aux stimuli en un voxel donné sont superposées à un trend temporel lent et lisse. La présence de ce trend n'est pas dûe aux stimuli, mais plutôt à la réponse physiologique du sujet (i.e. respiration, fréquence cardiaque) et au réchauffement du scanner, aussi il est important de le supprimer.

Il n'existe pas vraiment de consensus quant à la manière standard d'estimer ce trend [21]. Il existe plusieurs modèles, notamment des splines cubiques ou quadratiques, des splines de Bézier, des filtres

passe-hauts, des approximations polynomiales, des ondelettes, etc. Pour notre part, nous avons opté au final pour un modèle quadratique local. Il s'agit d'un choix plutôt inhabituel, aussi nous donnons davantage de détails à ce sujet à la section 4.2.4.

2.2.4 Normalisation et masque

Avant de procéder à l'analyse, un premier filtrage des voxels est effectué. Plus précisément, les voxels en dehors des régions d'intérêt ou les voxels pour lesquels la réponse est trop faible par rapport à la condition de repos sont écartés de l'analyse. Les trajectoires temporelles des voxels sélectionnés sont ensuite normalisées. La normalisation peut se faire voxel par voxel ou encore à travers tous les voxels. Nous avons opté pour la normalisation voxel par voxel, celle-ci se prêtant mieux aux algorithmes d'apprentissage considérés dans la présente étude.

2.3 Analyse traditionnelle

Une fois les différentes étapes de prétraitement effectuées, on est prêt à procéder à la détection des régions d'intérêt. En chaque voxel, on calcule une statistique qui indique la probabilité que la réponse fMRI observée soit le fruit du hasard. Cette éventualité est appelée **hypothèse nulle** (H_0). Au final, on conserve seulement les voxels pour lesquels la probabilité de l'hypothèse nulle est très faible. Les différents agrégats de voxels ainsi sélectionnés sont ce qu'on appelle les **régions d'intérêt**. Nous décrivons ici plus en détail les étapes de cette analyse statistique.

D'abord, on cherche à mesurer la corrélation entre les réponses fMRI et les conditions auxquelles est soumis le sujet durant la scannographie (Figure 2.1). Soit $y_v(t)$ la réponse fMRI enregistrée au voxel v à différents moments t. Soit $x_v(t) \in \{0, 1\}$ une variable binaire encodant l'activation (1) ou la déactivation (0) d'une condition expérimentale (ex : stimulation vs repos) . On suppose qu'il existe b_{v1}, b_{v2} tels que :

$$y_v(t) = b_{v1}\, x_v(t) + b_{v2} + \epsilon_v(t)$$

où $\epsilon_v(t)$ est un bruit gaussien. Les paramètres b_{v1} et b_{v2}, sont déterminés par une régression linéaire standard, de manière à minimiser la somme des résidus au carré $\sum_t \epsilon_v(t)^2$. La solution analytique de ce problème d'optimisation est

$$\begin{aligned} b_{v1} &= \hat{\mu}_{v1} - \hat{\mu}_{v0} \\ b_{v2} &= \hat{\mu}_{v0} \end{aligned}$$

où $\hat{\mu}_{v1}$ et $\hat{\mu}_{v0}$ sont les moyennes empiriques sous activation et déactivation de la condition expérimentale, respectivement. Ainsi, "le degré de corrélation" entre la réponse fMRI $y_v(t)$ et la variable explicative $x_v(t)$, soit b_{v1}, est le contraste entre les moyennes empiriques $\hat{\mu}_{v,0}$ et $\hat{\mu}_{v,1}$. Plus il est élevé, plus le comportement du voxel est influencé par la variable $x_v(t)$ qui décrit les conditions expérimentales, moins l'hypothèse nulle est probable.

On remarque que la statistique $b_{v,1}$ est influencée par la variance du voxel, i.e. l'amplitude de la réponse fMRI captée. Afin de comparer les voxels de manière équitable, on normalise la statistique par l'écart type échantillonnal $\hat{s}_v$.

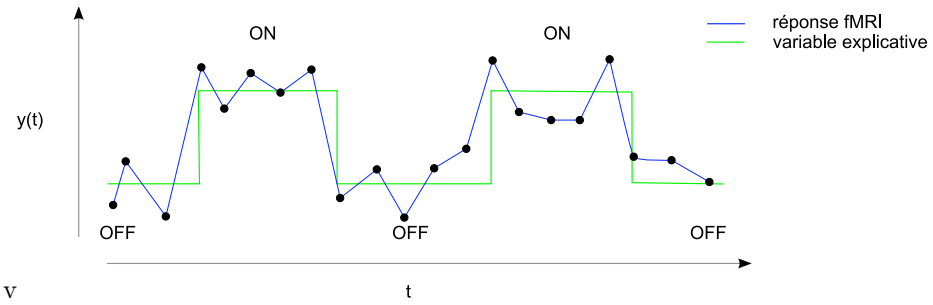


FIGURE 2.1: La trajectoire en bleu simule la réponse fMRI en un voxel à différents moments. En vert, on aperçoit l'évolution d'une variable explicative dans le temps. Celle-ci a deux états, soient ON et OFF. Il pourrait s'agir, par exemple, de l'activation ou de la déactivation d'un stimulus. La hauteur des plateaux ON et OFF indique les moyennes empiriques des réponses sous les conditions correspondantes . Ici, l'onde carrée résultante semble relativement bien expliquer la réponse fMRI. Il semble aussi qu'il y ait une bonne différence de hauteur entre les plateaux, ce qui suggère que les véritables moyennes ne sont identiques qu'avec une faible probabilité.

$$
\begin{aligned}
b'_{v1} &= \frac{b_{v1}}{\hat{s}_v} \\
\hat{s}_v^2 &= \frac{1}{n_1+n_0-2}\Big[\sum_{t\in x_v(t)=0}(y_v(t)-\hat{\mu}_{v1})^2+\sum_{t\in x_v(t)=0}(y_v(t)-\hat{\mu}_{v0})^2\Big] \\
n_1 &= |t\in x_v(t)=1| \\
n_0 &= |t\in x_v(t)=0|
\end{aligned}
$$

On peut généraliser l'analyse à un ensemble de J variables explicatives. Cette fois, la réponse du voxel est modélisée comme une combinaison linéaire des variables explicatives :

$$y_v(t) = b_{v,1}x_{v1}(t) + b_{v,2}x_{v2}(t) + \ldots + b_{v,J}x_{vJ}(t) + b_{v,J+1} + \epsilon_v(t)$$

Le degré de corrélation avec la j-ième variable est donnée par :

$$b_{v,j} = \hat{\mu}_{v,j} - \hat{\mu}_{v,repos}$$

où $\hat{\mu}_{repos}$ correspond à la moyenne empirique des scans associés à la condition expérimentale de "repos".

Il existe aussi d'autres variantes qui permettent d'intégrer des variables $x(t)$ non binaires. Par exemple, pour les designs de scannographie à haute résolution temporelle, i.e. avec des scans à fréquence très élevée, $x_v(t)$ modélise la réponse hémodynamique dans le temps. Parfois, on ajoute aussi des variables explicatives qui représente un trend. On incorpore ainsi le detrending dans l'analyse finale. Pour notre part, nous nous en tenons à des variables binaires.

Une fois la statistique de contraste b'_v obtenue, on mesure ensuite la probabilité de l'hypothèse nulle, i.e. la probabilité que le contraste observé soit dû au hasard. Pour se faire, on suppose que $y_v(t)$ est normalement distribuée conditionnellement à $x_v(t)$. Ainsi, à un facteur d'échelle près, la statistique de contraste b'_{v1} devient une statistique de Student. Plus précisément, pour la transformer en une statistique de Student t_v, il suffit de diviser par la racine du nombre de degrés de liberté :

$$t_v = \frac{b'_{v1}}{(n_1+n_0-2)^{1/2}} = \frac{\hat{\mu}_1-\hat{\mu}_0}{(n_1+n_0-2)^{1/2}\hat{s}_v}$$

On applique un test d'hypothèse standard sur la statistique t_v. Plus précisément, on considère que, sous l'hypothèse nulle, les véritables moyennes $\mu_{v,1}$ et $\mu_{v,0}$ sont identiques. On détermine la probabilité d'observer les estimés $\hat{\mu}_{v,1}$ et $\hat{\mu}_{v,0}$ dans ce cas. Plus $\hat{\mu}_{v,1}$ et $\hat{\mu}_{v,0}$ sont proches, plus l'hypothèse nulle est probable. Afin de déterminer si on rejette ou non l'hypothèse, on choisit ensuite un seuil T_α tel que, si cette hypothèse est vraie, alors la probabilité que $|t_v| \geq T_\alpha$ est inférieure à α. La probabilité $1 - \alpha$ représente la confiance avec laquelle on rejette l'hypothèse nulle.

Afin de déterminer si les contrastes observés pour un volume donné sont significatifs, il faut aussi prendre en considération le nombre de voxels. Par exemple, si la probabilité qu'une statistique de contraste en un voxel soit dûe au hasard est de 0.05 et qu'on dispose de 5000 voxels, alors, en supposant que ces voxels ne sont pas corrélés, on peut s'attendre à ce qu'il y ait environ $5000 * 0.05 = 250$ faux positifs, i.e. 250 contrastes déclarés à tort significatifs. Or, si on avait justement autour de 250 contrastes déclarés comme positifs, alors il se peut qu'il n'y ait pas vraiment de différences significatives entre les différentes conditions.

Afin de tenir compte du nombre de tests effectués, i.e. du nombre de voxels, on applique souvent une procédure appelée **correction Bonferroni** [7]. Cette procédure consiste à choisir un nouveau seuil $T_{\alpha'}$ tel que, cette fois, α corresponde à la probabilité qu'il y ait au moins un faux positif parmi un volume de D voxels. La relation entre α et α' est approximée par :

$$\alpha' = \frac{\alpha}{D}$$

Cette correction du seuil est souvent considérée comme trop sévère, du fait de la présence de corrélation spatiale parmis les voxels. Il existe d'autres techniques plus raffinées, basées sur la théorie des champs aléatoires, cependant nous n'en faisons pas usage dans le présent projet.

2.4 Limites de l'analyse traditionnelle

La méthode d'analyse traditionnelle implique que les différences d'une condition à l'autre se présentent sous forme d'un contraste marqué au niveau des réponses fMRI moyennes. De plus, elle est **univariée**, i.e. sonde les différents voxels isolément et pas dans leur ensemble. Ainsi, il est pratiquement impossible de détecter les patterns significatifs prenant la forme de faibles contrastes distribués. Ceux-ci sont confondus avec du bruit. La Figure 2.2 illustre un tel cas pour deux voxels et deux conditions.

Une autre limite importante de l'analyse traditionnelle est la question du lissage. Comme mentionné à la section 2.2.2, l'opération de lissage peut être bénéfique parce qu'elle efface le bruit. Cependant, elle peut aussi effacer de l'information en même temps. Dans [9], Kriegeskorte s'attarde à l'effet du lissage sur la détection des régions d'intérêt et remet en question la méthode d'analyse traditionnelle. Il applique une extension multivariée de la statistique de Student, basée sur la distance de Mahalanobis, à de petites sphères de voxels. Il met ainsi en évidence la significativité de patterns aux allures poivre et sel distribués dans ces sphères. Ces patterns sont effacés avec un lissage trop prononcé. Même sans lissage, ils sont indistinguables du bruit avec des statistiques de Student univariées, puisque les contrastes de moyenne en chacun des voxels sont faibles. Kriegeskorte montre aussi que, avec des tests de Student, beaucoup de régions d'intérêt ne peuvent être détectées sans lissage. Bref, selon cette étude, les formes d'information présentes dans les données peuvent être multiples.

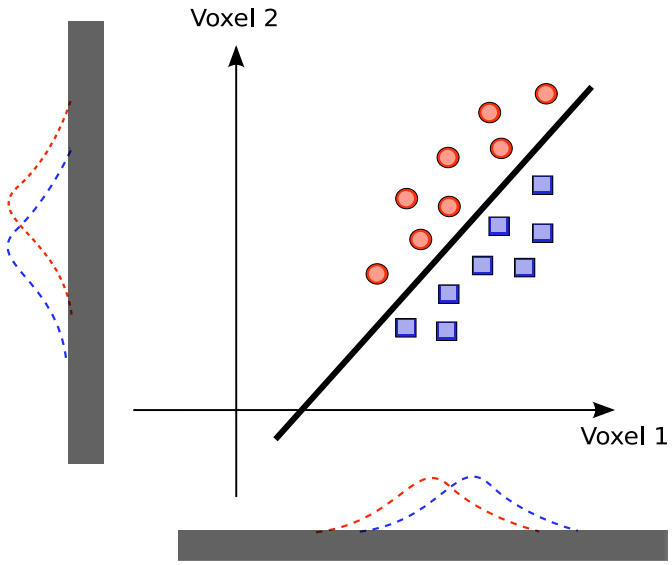


FIGURE 2.2: Les cercles rouges et les carrés bleus représentent les réponses fMRI, en deux voxels, associées à des conditions expérimentales différentes. Pour chaque voxel et chaque condition, la distribution des réponses fMRI est indiquée par une forme en cloche. Lorsqu'on observe la distribution des données sur un seul axe à la fois, il est difficile de voir des différences marquées entre les conditions. Cependant, lorsqu'on considère les deux voxels conjointement, on remarque qu'il existe une démarcation bien nette entre les exemples des deux conditions.

Il semble qu'il y ait avantage, dépendamment des données, à inclure dans l'analyse une statistique multivariée plutôt qu'univariée, et, de surcroit, une statistique un peu plus agnostique, i.e. qui implique moins de supposition quant à la nature de l'information dans les données. Au chapitre 3 qui suit, nous nous penchons sur l'analyse de données fMRI à l'aide de classifieurs linéaires. Ceux-ci combinent linéairement les réponses des voxels pour détecter et catégoriser différents cognitifs. Les classifieurs linéaires gagnent de plus en plus en popularité dans le milieu des neurosciences. Ils ont d'ailleurs été utilisés avec succès dans une étude similaire à la nôtre (Kamitani et Tong [8]).

Chapitre 3

Apprentissage automatique et applications aux données fMRI

L'application de classifieurs linéaires à l'analyse des données d'imagerie cérébrale est une tendance qui a grandit en popularité au cours des dernières années. Ces classifieurs combinent linéairement les réponses des voxels pour détecter et catégoriser différents cognitifs. Contrairement aux méthodes d'analyses traditionnelles, ceux-ci sont multivariés et, par conséquent, peuvent détecter une plus grande variété de patterns.

Dans ce chapitre, nous nous penchons sur le potentiel des classifieurs linéaires en tant qu'outils d'analyse fMRI. Nous débutons en situant la présente application dans le monde de l'apprentissage automatique. Ensuite, nous discutons des principaux défis associés à l'apprentissage sur données fMRI. Nous insistons beaucoup sur le phénomène du **sur-apprentissage**, particulièrement problématique pour notre application. Nous poursuivons avec une description du fonctionnement des classifieurs linéaires utilisés dans le présent projet, soient le **modèle bayésien gaussien naïf** (BGN) avec variances partagées, l'algorithme **régression logistique** (RL) et l'algorithme **machine à vecteurs de support** (MVS) linéaire. Nous décrivons différents outils d'interprétation basés sur ces derniers.

3.1 Survol de l'apprentissage automatique

L'apprentissage automatique est une famille d'algorithmes rappelant le processus d'apprentissage par exemple chez l'humain. Un algorithme d'apprentissage, essentiellement, construit un modèle de prédiction à partir d'une banque d'exemples. La phase de recherche d'information pertinente dans les données est communément appelée **entraînement**. Il existe bien sûr plusieurs degrés d'apprentissage. Un algorithme n'effectuant aucun apprentissage serait un algorithme qui se comporte de manière très stéréotypée, sans recherche active d'information.

Un autre élément important de l'apprentissage est la capacité de transfert à de nouveaux cas, soit la capacité de **généralisation**. Un algorithme consistant purement à stocker en mémoire tous les cas possibles n'est pas considéré comme un algorithme d'apprentissage.

Pour mesurer la capacité de généralisation, on s'en remet à la performance de prédiction du modèle sur une banque de nouveaux exemples. Une performance raisonnablement élevée indique que l'information exploitée par le modèle est pertinente à la tâche et représentative des données en général.

Les motivations à utiliser un algorithme d'apprentissage varient d'une application à l'autre (Figure 3.1). Le plus souvent, on veut éventuellement se servir du modèle construit pour effectuer des prédictions. La performance de prédiction est alors très importante ; même une petite amélioration peut

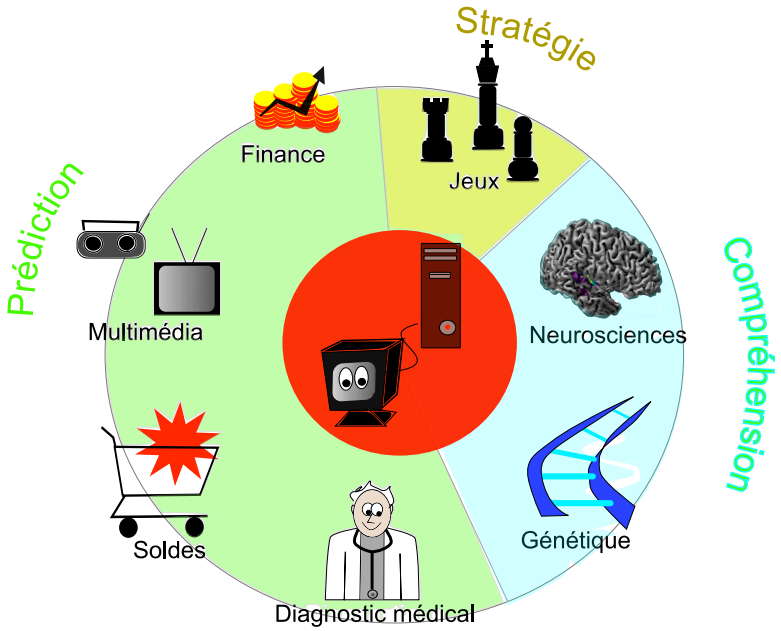


FIGURE 3.1: L'apprentissage automatique s'applique à toutes sortes de domaines. Les motivations varient d'une application à l'autre : effectuer des prédictions, construire une stratégie, mieux comprendre les données, etc. Les algorithmes d'apprentissage extraient automatiquement l'information pertinente de banque de données. (Image fortement inspirée de [30])

valoir son pesant d'or.

En ce qui concerne l'analyse des données fMRI, on cherche surtout à analyser le modèle construit en tant que tel. La performance de prédiction ne sert souvent qu'à valider la présence d'information pertinente dans les données. Quelques cas font cependant exception ; par exemple, les travaux visant la construction d'un détecteur de mensonge (Figure 3.2).

Pour notre part, nous utilisons des algorithmes d'apprentissage pour mieux comprendre les relations entre les réponses fMRI et diverses conditions de stimulation. Nous construisons et interprétons des modèles dont la tâche est de prédire le stimulus perçu à partir de la réponse fMRI. Nous identifions les patterns pertinents à la prédiction dans les réponses fMRI.

Le fait de réaliser l'entraînement avec des paires d'exemples de type (prédicteur,cible), i.e. ici (réponse fMRI, stimulus), est caractéristique des algorithmes d'apprentissage **supervisés**. Lorsque la cible prend des valeurs discrètes, on parle de tâche de **classification**, lorsqu'elle prend des valeurs continues ou appartient à un ensemble infini, on parle plutôt de tâche de **régression**.

Tous les algorithmes ne nécessitent pas la présentation de cibles explicites durant l'entraînement ; il existe aussi des algorithmes d'apprentissage **non-supervisés**. Notamment, une tâche non-supervisée très commune est l'**estimation de densité**, i.e. la construction d'un modèle de probabilité. Les approches non-supervisées peuvent s'avérer très efficaces pour comprendre les dynamiques inhérentes aux données et sont parfois utilisées en conjonction avec les approches supervisées.

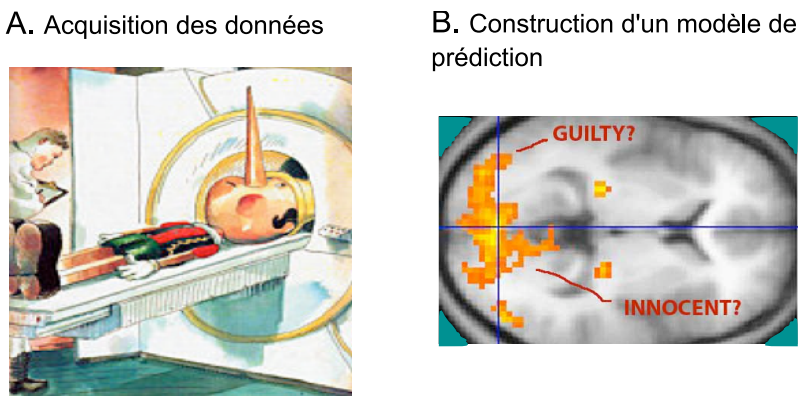


FIGURE 3.2: Une application de l'apprentissage machine dont le but est de construire un détecteur de mensonge. A. Une banque d'images fMRI enregistrées lorsque les sujets mentent ou disent la vérité est fournie à l'algorithme. (source : voir [31]) B. L'algorithme recherche les différences au niveau de la configuration des réponses des voxels entre les exemples où le sujet ment et les exemples où le sujet dit la vérité. (source : voir [29])

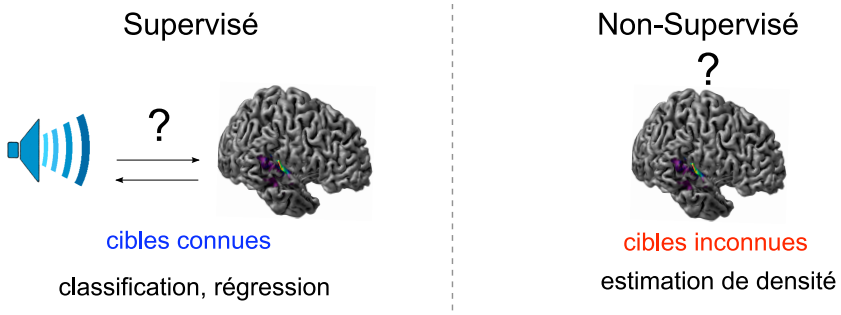


FIGURE 3.3: À gauche : caractéristiques de l'apprentissage supervisé. Des exemples de tâches sont de prédire le stimulus perçu par le sujet en utilisant la réponse fMRI comme prédicteur, ou vice-versa. À droite : caractéristiques de l'apprentissage non-supervisé. Les approches non-supervisées sont très utiles pour mieux comprendre la nature des données.

Un exemple bien connu d'une telle intégration est la classification par inférence bayésienne (Figure 3.4A). Donné un prédicteur $\mathbf{x} \in \mathbf{X}$ et une cible $c \in C$, l'apprentissage consiste à modéliser les probabilités conditionnelles $p(\mathbf{x}|c)$ et les probabilité marginales $p(c)$. Pour effectuer des prédictions, on exploite la règle de Bayes :

$$\begin{aligned} p(c|\mathbf{x}) &= \frac{p(\mathbf{x}|c)p(c)}{p(\mathbf{x})} \\ &\propto p(\mathbf{x}|c)p(c) \end{aligned}$$

La fonction de décision finale $y(\mathbf{x})$ infère la cible la plus probable en un point de test $\mathbf{x}$:

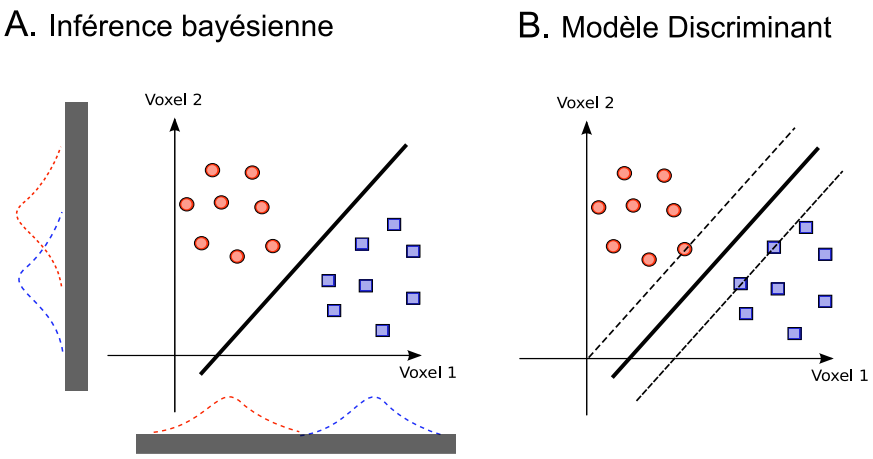


FIGURE 3.4: Problème de classification binaire en deux dimensions. Les carrés bleus représentent les exemples de la classe C=-1. Les cercles rouges représentent les exemples de la classe C=1. . A. Exemple de classifieur bayésien, soit le modèle bayésien gaussien naïf avec variances partagées. Les 2 dimensions (ici voxels) sont supposées indépendantes. Les densités $p(voxel_1|c)$ et $p(voxel_2|c)$ estimées sont indiquées sur chacun des axes. Elles sont modélisées comme des gaussiennes de moyennes distinctes et variance partagées. La frontière de décision obtenue par inférence bayésienne est tracée en noir. Dans le cas du modèle bayésien gaussien naïf, elle est linéaire. B. Exemple d'algorithme de classification discriminant, soir l'algorithme machine à vecteurs de support linéaire (SVM). On cherche la frontière linéaire qui permet de séparer les exemples des deux classes de manière optimale. La frontière permettant de bien classifier et de maximiser la distance des exemples à la frontière (marge) est choisie.

$$\begin{aligned} y(\mathbf{x}) &= \underset{c \in C}{argmax} \quad p(c|\mathbf{x}) \\ &= \underset{c \in C}{argmax} \quad p(\mathbf{x}|c)p(c) \end{aligned}$$

La classification par inférence bayésienne est considérée comme une approche supervisée, car il est nécéssaire de connaître les cibles des exemples. Cependant, la construction de chacun des estimateurs de densité $p(\mathbf{x}|c)$ relève de l'apprentissage non-supervisé. Une bonne performance de classification, avec ce type d'approche, est une conséquence d'une bonne "compréhension" des données. En fait, on la qualifie aussi de **générative**, car elle permet de modéliser le processus de génération des données à l'aide des distributions $p(\mathbf{x}|c)$ et $p(c)$ apprises.

L'approche de classification antagoniste à l'approche par inférence bayésienne est une approche purement supervisée, directement focusée sur la performance de classification, soit l'approche **discriminante** (Figure 3.4B). Celle-ci implique que la fonction de décision $y(\mathbf{x})$ a une certaine forme paramétrique $f(\mathbf{x},\boldsymbol{\theta})$, où $\boldsymbol{\theta}$ représente un ensemble de paramètres inconnus. Les paramètres $\boldsymbol{\theta}$ sont optimisés de manière à maximiser directement la performance de prédiction.

Comme nous l'expliquerons à la section 3.3.2, ces deux approches de classification sont complémentaires et il peut s'avérer intéressant de les comparer. Dans le présent projet, nous utilisons deux algorithmes de classification discriminants bien connus, soient les algorithmes machine à vecteurs de support avec noyau linéaire et régression logistique. À des fins de comparaison, nous incluons aussi un algorithme de classification par inférence bayésienne bien connu, soit le modèle bayésien gaussien naïf

avec variances partagées. Nous donnons une description plus détaillée de ces algorithmes à la section 3.5. Comme nous l'expliquerons à la section 3.5.1, le modèle bayésien gaussien naïf avec variances partagées est d'autant plus intéressant qu'il ressemble aux méthodes d'analyse standard des données fMRI et peut donc servir de point de référence.

Un point commun entre tous ces classifieurs est qu'ils sont **linéaires**. Soit $\mathbf{x} \in \mathbb{R}^d$ un prédicteur à d dimensions. Soit $c \in \{-1, 1\}$ une cible binaire. Un classifieur binaire $y(\mathbf{x})$ est dit linéaire s'il existe $\mathbf{w} \in \mathbb{R}^d$ et $b \in \mathbb{R}$ tels que

$$y(\mathbf{x}) = signe(\mathbf{wx} + b).$$

La frontière entre les données des deux conditions correspond à l'hyperplan défini par la contrainte

$$\mathbf{wx} + b = 0.$$

S'il existe $\mathbf{w}$ et b tels que les données sont parfaitement classifiées, on dit que les données sont **linéairement séparables**.

Comme nous l'expliquerons dans les sections qui suivent, les classifieurs linéaires sont de bons candidats pour l'analyse des données fMRI. Ils sont suffisamment robustes pour être utilisés dans des conditions où il y a beaucoup de bruit et relativement peu d'exemples, ce qui est typiquement le cas avec les données fMRI. De plus, ils sont relativement faciles à interpréter. Chacun des poids du vecteur $\mathbf{w} = w_1, w_2, ..., w_d$ est connecté à un des d voxels (Figure 3.5A), i.e. indique la contribution isolée du voxel à la décision du classifieur. Si $\mathbf{w}$ est près de la solution optimale, il peut être interprété comme la direction dans laquelle les données varient le plus d'une condition à l'autre (Figure 3.5B).

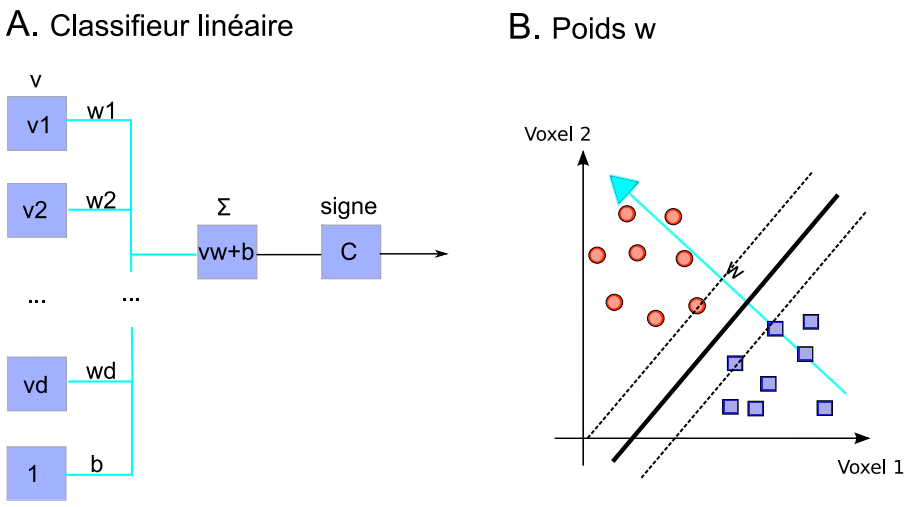


FIGURE 3.5: A. Structure d'un classifieur linéaire binaire. Ici le prédicteur $\mathbf{v} = (v_1, v_2, ..., v_d)$ correspond à la réponse fMRI dans d voxels. Chacun des poids du vecteur $\mathbf{w} = (w_1, w_2, ..., w_d)$ est connecté à un des d voxels. Le classifieur prédit de quel coté de la frontière se trouve $\mathbf{v}$, i.e. $signe(\mathbf{vw} + b)$. B. Classifieur linéaire binaire pour un prédicteur composé de seulement 2 voxels. Les carrés bleus sont les exemples pour lesquels la cible $C = -1$. Les cercles rouges sont les exemples pour lesquels la cible $C = 1$. La frontière de décision est indiqué par un trait noir épais. Le vecteur $\mathbf{w}$ est orthogonal à la frontière et peut être interprété comme la direction dans laquelle les données varient le plus d'une condition à l'autre.

Pour les problèmes de classification à plus de deux cibles, i.e. les problèmes **multiclasses**, il existe plusieurs extensions possibles de la définition d'un classifieur linéaire, dépendamment de l'algorithme et aussi plusieurs manières d'interpréter les paramètres. Nous donnons davantage de précisions à ce sujet à la section 3.5.

3.2 Prévenir le sur-apprentissage

Un problème fréquemment rencontré dans les applications d'apprentissage machine est ce qu'on appelle le **sur-apprentissage**. Ce phénomène se produit lorsque le degré de liberté de l'algorithme est trop élevé par rapport au nombre d'exemples d'entraînement disponibles. L'algorithme arrête alors son choix sur un modèle de prédiction qui colle excessivement aux exemples contenus dans l'ensemble d'entraînement et relève des patterns qui ne sont pas nécéssairement représentatifs des données en général (Figure 3.6). Typiquement, les capacités de généralisation à de nouveaux exemples en sont affaiblies.

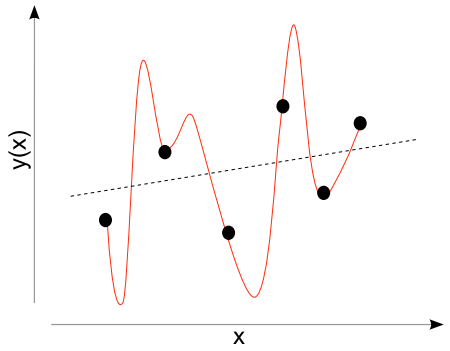


FIGURE 3.6: Exemple de sur-apprentissage pour un problème de régression 2d. Chacun des cercles représente un exemple d'entraînement, dans le plan cartésien. L'estimé idéal est tracé en noir, en pointillé ; les données sont distribuées de part et d'autre de cette droite, selon un bruit gaussien. Le modèle appris est tracé en rouge. L'algorithme d'apprentissage choisit un modèle qui arrive à faire des prédictions parfaites aux points d'entraînement. Ce modèle, cependant, épouse mal l'allure globale du trend.

En ce qui concerne l'analyse des données fMRI, le sur-apprentissage rend beaucoup plus hasardeuse l'interprétation des patterns appris par le modèle et est à éviter. Dans cette section, nous expliquons pourquoi les données fMRI sont un terrain particulièrement propice au sur-apprentissage. Nous présentons ensuite quelques techniques de prévention qui consistent à réduire le degré de liberté de l'algorithme. Le succès de celles-ci est lié au dilemme de biais variance, que nous présentons également.

3.2.1 Fléau de la dimensionalité et instabilité des données

Un algorithme d'apprentissage nécessite une quantité raisonnable d'exemples d'entraînement pour arriver à bien généraliser. Un ensemble d'entraînement trop petit risque de ne pas être représentatif des données en général et de mener à du sur-apprentissage. La quantité d'exemples nécéssaire dépend de l'instabilité dans les données et de la sensibilité de l'algorithme à cette instabilité.

En général, plus les données sont de dimensionalité élevée, plus l'espace des données est volumineux, et plus on a besoin d'exemples pour le représenter. Par exemple, supposons qu'on cherche à modéliser

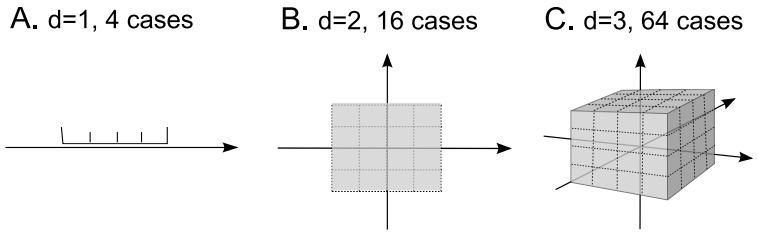


FIGURE 3.7: Nombre de cases dans un hypercube avec 4 divisions par dimension. A. Espace unidimensionnel : $4^1 = 4$ cases. B. Espace bidimensionnel : $4^2 = 16$ cases. C. Espace tridimensionnel : $4^3 = 64$ cases

des données uniformément distribuées dans un hypercube de dimensionalité d en construisant un histogramme, i.e. en divisant l'espace en cases et en comptant le nombre d'exemples dans chacune des cases. Le nombre d'exemples nécessaire est proportionnel au nombre de paramètres de l'histogramme, i.e. au nombre de cases. Ainsi, il faut donc de l'ordre de c^d exemples, où c correspond à la précision de l'histogramme voulue (Figure 3.7). Ainsi, dans ce cas particulier, le nombre d'exemples nécéssaire croît exponentiellement avec la dimensionalité.

Tous les algorithmes n'ont pas la même sensibilité que les histogrammes, néanmoins, ils sont tous influencés, à divers degrés, par la dimensionalité des données. Pour un même nombre d'exemples, une augmentation de la dimensionalité engendre typiquement davantage de sur-apprentissage : c'est ce qu'on appelle le **fléau de la dimensionalité**. L'effet est amplifié si un très grand nombre de dimensions ne sont pas pertinentes à la tâche qu'on veut réaliser ou si on introduit du bruit dans les données.

Les données fMRI, justement, sont de dimensionalité particulièrement élevée (parfois plus de 100000 voxels), comportent beaucoup de voxels qui ne sont pas pertinents dans la tâche cognitive considérée et sont de surcroît très bruitées. De plus, typiquement, on dispose de relativement peu d'exemples (ex : quelques centaines ou moins). L'apprentissage sur données fMRI, bref, est un défi pour le moins majeur et des mesures s'imposent pour prévenir le sur-apprentissage.

En fait, certains algorithmes sont plus robustes que d'autres au sur-apprentissage et sont donc des choix plus indiqués pour notre application. Comme nous l'expliquerons dans les sections 3.2.2 et 3.2.3 qui suivent, nous avons avantage à miser sur les algorithmes qui se limitent à un espace assez restreint de modèles, i.e., de faible **capacité**.

3.2.2 Contrôle de la capacité des algorithmes

Ce qu'on appelle la **capacité** d'un algorithme d'apprentissage est la richesse des formes que le modèle final peut adopter. Elle est souvent mesurée en terme de nombre de paramètres indépendants à estimer, ou **degrés de liberté**. En général, plus la capacité d'un algorithme est grande, plus ce dernier est sujet au sur-apprentissage. Les histogrammes, par exemple, ont une capacité illimitée ; si suffisamment d'exemples sont fournis, il peuvent modéliser n'importe quelle distribution. Cependant, comme nous l'avons expliqué à la section précédente, ils sont aussi très sensibles au bruit et à la dimensionalité des données.

Afin de mieux illustrer la relation entre la capacité et la tendance au sur-apprentissage, considérons le problème de régression polynomiale illustré à la Figure 3.8 . On cherche à approximer le modèle idéal $y(\mathbf{x})$ (trait discontinu) à l'aide de polynômes de différents degrés. Avec les polynômes de degrés 0

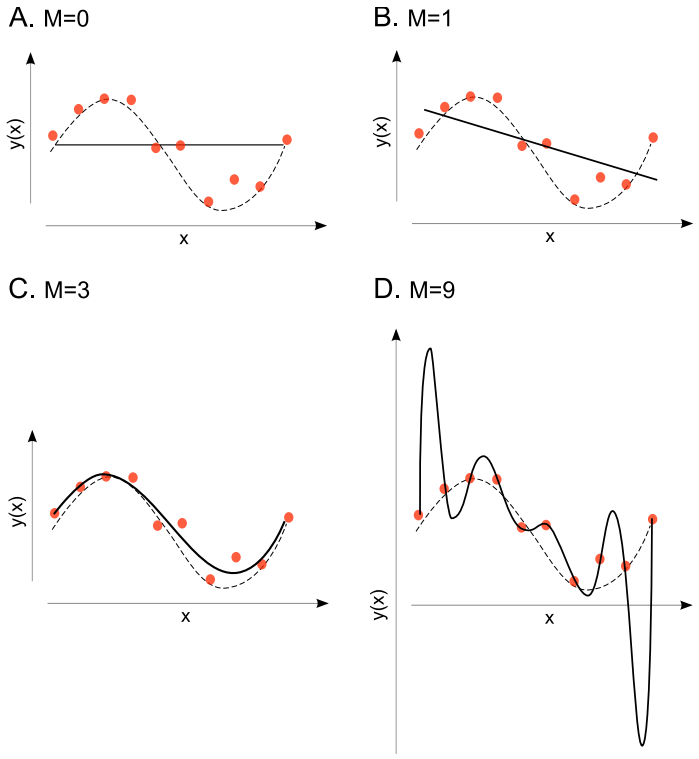


FIGURE 3.8: Problème de régression polynomiale 2d. Le modèle idéal correspond au trait discontinu. Les cercles rouges sont les exemples d'entraînement. Le modèle choisit correspond au trait continu. Le degré M de chacun des polynômes est indiqué sur chacun des graphiques. Plus le degré du polynôme augmente, plus la capacité de l'algorithme d'apprentissage augmente. Le meilleur modèle est obtenu en $M = 3$, soit le juste compromis de capacité. Cette image est une recontruction d'une image tirée de Legenstein (2009) [11]

et 1 (AB), la capacité du modèle est trop faible et on obtient un pauvre estimé, i.e. l'algorithme "sous-apprend". Avec un polynôme de degré 9 (D), le modèle a trop de capacité et "sur-apprend". En effet, le modèle obtenu passe par tous les points d'entraînement, mais n'épouse pas bien la forme globale de la courbe. Avec le polynôme de degré 3 (C), la capacité de l'algorithme semble raisonnable. En effet, l'estimé obtenu passe près des points d'entraînement et épouse bien la forme de la courbe.

La clé d'une bonne généralisation est un juste compromis de capacité. Une capacité trop élevée mène au sur-apprentissage et une capacité trop basse mène au sous-apprentissage. Bien sûr, plusieurs facteurs déterminent la capacité d'un algorithme.

La forme paramétrique du modèle est évidemment un facteur déterminant. Pour le présent projet, étant donné le risque élevé de sur-apprentissage, nous nous limitons à des modèles linéaires. Ceux-ci sont des candidats intéressants, car ils ont une capacité relativement faible. Cette fois, le nombre d'exemples nécessaires à une bonne généralisation croit linéairement avec le nombre de dimensions.

On peut aussi jouer sur la capacité des algorithmes d'apprentissage en introduisant un biais ou des indices durant la phase de recherche d'information. On appelle ce procédé **régularisation**.

Des techniques de régularisation bien connues sont les régularisations avec normes $L1$ et $L2$. Celles-ci font en sorte de privilégier les solutions pour lesquelles les paramètres du modèle sont de faible amplitude. Soient $\boldsymbol{\theta} = (\theta_1, \theta_2, ..., \theta_D)$ le vecteur de paramètres à apprendre par l'algorithme, D un

ensemble d'exemples d'entraînement, $f(\boldsymbol{\theta}, D)$ une fonction de coût à minimiser par l'algorithme. Pour une régularisation $L1$, on résoud :

$$argmin_{\boldsymbol{\theta}}\, f(\boldsymbol{\theta}, D) + \lambda_1 \sum_i |\theta_i|$$

Pour une régularisation $L2$, on résoud :

$$argmin_{\boldsymbol{\theta}}\, f(\boldsymbol{\theta}, D) + \lambda_2 \sum_i ||\theta_i||^2$$

Les régularisations $L1$ et $L2$ favorisent toutes deux les solutions avec des paramètres de faible amplitude, mais mènent à des solutions potentiellement très différentes. La norme $L2$ tend à favoriser les solutions distribuées, avec beaucoup de paramètres de faibles amplitude. La norme $L1$, quand à elle, favorise les solutions qui sont parcimonieuses, i.e. avec quelques paramètres d'amplitude élevée et la plupart des paramètres avec valeur 0. La norme L1 est généralement très efficace lorsque le prédicteur est composé d'un très grand nombre de dimensions non-pertinentes. En effet, en encourageant la parcimonie, seules les dimensions les plus pertinentes conservent une influence ; les dimensions du prédicteur associées à des paramètres de valeur 0 n'ont plus aucune influence.

La régularisation $L1$ est particulièrement indiquée pour notre application, puisque nous nous attendons à ce que peu de voxels soient pertinents. Un problème potentiel, cependant, est qu'elle risque d'altérer l'interprétabilité des paramètres en excluant les voxels pertinents, mais redondants. Nous présentons à la section 3.3.1 une alternative appelée **filet élastique**, basée sur une combinaison entre la norme $L1$ et la norme $L2$, qui mène à un meilleur compromis entre solutions distribuées et parcimonieuses.

Les régularisations $L1$ et $L2$, en quelque sorte, permettent de restreindre la forme paramétrique du modèle avec souplesse. La force de la régularisation est contrôlée en ajustant les constantes λ_1 et λ_2, avant l'entraînement. Ces paramètres sont des exemples d'**hyper-paramètres**. Nous expliquons comment les choisir à la section 3.4.

Il existe d'autres types de régularisation qui ne consistent pas à restreindre la forme paramétrique du modèle directement. Entre autres, une autre technique de régularisation beaucoup utilisée est la procédure d'arrêt prématuré. Celle-ci s'applique à des algorithmes basés sur des méthodes d'optimisation itératives, i.e. qui approche de la solution optimale progressivement en modifiant itérativement les paramètres. Elle consiste à limiter le nombre d'itérations. En stoppant l'optimisation plus tôt que tard, on peut prévenir le sur-apprentissage en évitant le perfectionnement excessif du modèle.

Il existe beaucoup d'autres techniques de contrôle de capacité. En fait, chacun des classifieurs linéaires considérés dans le présent projet fait l'objet d'une technique de contrôle particulière, comme nous l'expliquerons à la section 3.5.

3.2.3 Dilemme de biais-variance

Il existe de nombreuses techniques pour contrôler la capacité des algorithmes. Leur utilisation peut être motivée à travers ce qu'on appelle le **dilemme de biais-variance**, que nous introduisons ici.

Le modèle produit par un algorithme d'apprentissage est considéré comme un estimateur avec une certaine distribution qui dépend des conditions d'apprentissage, soit des données disponibles. Nous montrons que l'**erreur quadratique moyenne** (EQM) des prédictions de l'estimateur se réduit à deux composantes qui dépendent de l'algorithme d'apprentissage, soient le **biais** et la **variance**. Le biais représente la distance entre le modèle "moyen" et le modèle idéal. La variance représente l'instabilité du modèle appris dépendament des conditions d'apprentissage. En contrôlant la capacité d'un algorithme, on joue sur ces deux composantes.

Soit $\mathbf{x} \in \mathbf{X} \subseteq \mathbb{R}^d$ un prédicteur et $\mathbf{t} \in \mathbf{T}$ une cible. Soit $y(\mathbf{x}; D)$ un modèle de prédiction construit à partir d'un algorithme d'apprentissage avec un ensemble d'exemples d'entraînement fini D. On s'intéresse à l'erreur quadratique moyenne (EQM) de $y(\mathbf{x}; D)$:

$$EQM_D = \int_{\mathbf{X},\mathbf{T}} (y(\mathbf{x}; D) - \mathbf{t})^2 d\mathbf{x}\, d\mathbf{t}$$

Selon Bishop [1], l'erreur quadratique moyenne peut être décomposée de la manière suivante :

$$\begin{aligned} EQM_D &= \int_{\mathbf{X}} \{y(\mathbf{x}; D) - h(\mathbf{x})\}^2 p(\mathbf{x})\, d\mathbf{x} \quad + \quad \int_{\mathbf{X},\mathbf{T}} \{h(\mathbf{x}) - \mathbf{t}\}^2 p(\mathbf{x}, \mathbf{t})\, d\mathbf{x}\, d\mathbf{t} \\ h(\mathbf{x}) &= \mathbb{E}[\mathbf{t}|\mathbf{x}] = \int_{\mathbf{T}} \mathbf{t}\, p(\mathbf{t}|\mathbf{x})\, d\mathbf{t} \end{aligned}$$

Le modèle $h(\mathbf{x})$ introduit ici est le modèle en lequel l'erreur quadratique moyenne atteint son minimum, i.e. le modèle "idéal". Le deuxième terme de l'équation précédente représente donc l'erreur quadratique moyenne minimale et est indépendant du choix du modèle $y(\mathbf{x}; D)$.

Nous nous intéressons à la portion d'erreur attribuable à l'algorithme d'apprentissage, aussi nous considérons seulement le premier terme, soit celui qui dépend du modèle $y(\mathbf{x}; D)$ choisi. Plus précisément, nous nous intéressons à la performance moyenne de l'algorithme d'apprentissage indépendamment de l'ensemble d'entraînement D, aussi nous remplaçons l'intégrande $\{y(\mathbf{x}, D) - h(\mathbf{x})\}^2$ par son espérance sur D . On obtient :

$$\int_{\mathbf{X}} \mathbb{E}_D[\{y(\mathbf{x}; D) - h(\mathbf{x})\}^2]\, p(\mathbf{x})\, d\mathbf{x}$$

En fait, on peut montrer (Bishop [1]) que cette expression peut à son tour être décomposée de la manière suivante :

$$\begin{aligned} \int_{\mathbf{X}} \{\mathbb{E}[\{y(\mathbf{x}; D) - h(\mathbf{x})\}^2]\, p(\mathbf{x})\, d\mathbf{x} &= \quad biais^2 \quad + \quad variance \\ biais^2 &= \int_{\mathbf{X}} \{\mathbb{E}_D[y(\mathbf{x}; D)] - h(\mathbf{x})\}^2 p(\mathbf{x})\, d\mathbf{x} \\ variance &= \int_{\mathbf{X}} \mathbb{E}_D[\{y(\mathbf{x}; D) - \mathbb{E}_D[y(\mathbf{x}; D)]\}^2] p(\mathbf{x})\, d\mathbf{x} \end{aligned}$$

Le terme $biais^2$, ici, représente la moyenne, sur $x \in \mathbf{X}$, du carré de la distance entre les prédictions moyennes de l'algorithme, i.e. $\mathbb{E}_D[y(\mathbf{x}; D)]$, et les prédictions du modèle idéal $h(\mathbf{x})$. Plus le modèle choisi $y(\mathbf{x}; D)$ approche $h(\mathbf{x})$ en moyenne, plus $biais^2$ est petit. Le terme *variance* fait référence à la moyenne, sur $\mathbf{x} \in \mathbf{X}$, de la variance $Var_D[y(\mathbf{x}; D)]$ dans les prédictions de l'algorithme.

Les algorithmes avec plus de capacité ont typiquement moins de biais, i.e. l'ensemble des formes que le modèle peut épouser est davantage susceptible de contenir une forme qui approche $h(\mathbf{x})$. Cependant, plus de capacité va généralement de pair avec une augmentation de la tendance au sur-apprentissage, ce qui induit davantage de variance. En ajustant la capacité des algorithmes, on cherche le meilleur compromis entre le biais et la variance (Figure 3.9). Ce compromis est connu sous le nom de **dilemme de biais-variance**.

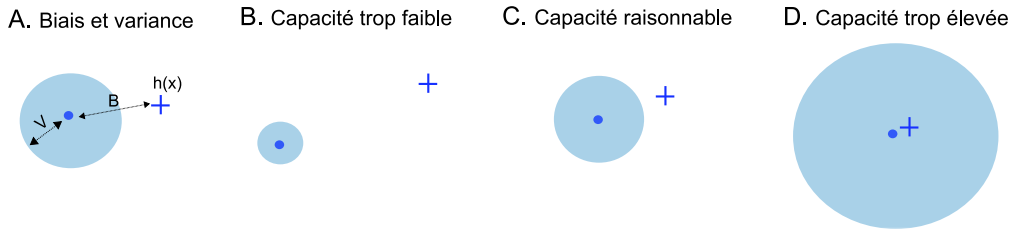


FIGURE 3.9: Dilemme de biais-variance. A. Diagramme de l'erreur dûe au biais (B) et de l'erreur dûe à la variance (V). La croix représente le modèle idéal $h(\mathbf{x}) = E[\mathbf{t}|\mathbf{x}]$. Le cercle bleu pâle représente la densité de probabilité dès différents modèles appris. Plus le cercle est grand, plus la variance est grande. Le point au centre du cercle représente le modèle appris "moyen". Plus celui-ci est loin du modèle $h(\mathbf{x})$, plus le biais est important. B. Exemple de capacité trop faible. La variance de l'algorithme d'apprentissage est faible, mais le biais est si grossier que le modèle final n'approche pas du tout la solution idéale. C. Exemple de capacité raisonnable. La variance et le biais sont relativement faibles. D. Exemple de capacité trop élevée. Le biais est très petit, cependant la variance de l'algorithme apprentissage est si grande qu'il y a peu de chance que le modèle appris approche $h(\mathbf{x})$.

3.3 Pertinence des techniques de contrôle de capacité à l'application

Puisque nous utilisons les algorithmes d'apprentissage surtout pour mieux comprendre les données, dans le cadre de notre application, l'interprétabilité des modèles appris est très importante. Dans le présent projet, nous prêtons attention à l'effet des différentes techniques de contrôle de capacité, en matière d'interprétabilité.

Un critère d'évaluation important, en matière d'interprétabilité, est la quantité de **faux positifs** et de **faux négatifs** . Par faux positif, ici, nous entendons un pattern dans les réponses fMRI relevé par l'algorithme d'apprentissage, mais pas représentatif des données en général. Inversement, par faux négatif, nous entendons un pattern pertinent et représentatif, mais ignoré par l'algorithme d'apprentissage.

De produire à la fois peu de faux positifs et de faux négatifs est un défi d'apprentissage important et intimement lié au défi posé par le sur-apprentissage. En fait, la réussite dépend aussi de la capacité de l'algorithme. Une capacité trop faible peut faire en sorte que le modèle n'a pas la complexité nécessaire pour capter certains patterns ; ainsi on augmente potentiellement la balance de faux négatifs. Une capacité trop élevée mène typiquement à du sur-apprentissage et peut résulter en une augmentation des faux positifs et des faux négatifs. En effet, les patterns pertinents sont alors souvent mis de côté au profit de patterns non-représentatifs.

Généralement, la capacité de l'algorithme est ajustée de manière à maximiser la performance de prédiction du modèle final sur de nouveaux exemples. Cette façon de faire permet une efficacité optimale du modèle final en terme de performance de prédiction. Cependant, elle ne permet pas nécéssairement une efficacité optimale en ce matière d'interprétabilité. En effet, la performance de prédiction ne réflète pas toujours bien la quantité de faux négatifs et faux positifs.

Un bon exemple de ce phénomène est la régularisation $L1$ introduite à la section 3.2.2. Cette dernière fait en sorte que seul un petit nombre de voxels, parmis les plus pertinents, conservent une influence. Ainsi, on réduit substantiellement les faux positifs, ce qui permet d'augmenter la performance de prédiction. Toutefois, il se peut que les voxels qui covarient avec un voxel déjà sélectionné soient écartés parce qu'ils représentent de l'information redondante, i.e. non essentielle pour le classifieur. La quantité de faux négatifs introduits, dans ce cas, n'est pas nécéssairement mesurable par la performance de prédiction du classifieur.

En fait, à la section 3.3.1, nous décrivons une technique de régularisation appelée **filet élastique**. Cette technique se veut une amélioration de la régularisation avec norme $L1$. Contrairement à cette dernière, en plus d'encourager la parcimonie, elle tient compte de la présence de corrélations entre les voxels. Nous comparons, dans l'une de nos expériences, cette technique de régularisation à la régularisation $L1$.

Puisque la performance de prédiction ne semble pas toujours bien réfléter l'interprétabilité d'un modèle, plutôt que de chercher directement à prédire le mieux possible, pourquoi ne pas chercher à comprendre le mieux possible ? En ce sens, les modèles génératifs, en particulier les classifieurs bayésiens, sont des candidats intéressants pour l'analyse fMRI. En effet, comme expliqué à la section 3.1, une bonne performance de prédiction, pour ce type de modèle, découle d'une modélisation du processus de génération des données.

À la section 3.3.2, nous nous penchons sur les effets bénéfiques d'un critère d'optimisation génératif classique, soit la log-vraisemblance des exemples, par rapport à un critère d'optimisation discriminant étroitement relié, soit la log-probabilité à posteriori des cibles. Nous expliquons comment le critère d'optimisation génératif peut être vu comme une version régularisée du critère d'optimisation discriminant et que l'effet dépend de la véracité des suppositions émises par rapport à la distribution des données.

D'investiguer pleinement le potentiel des algorithmes génératifs à l'analyse des données fMRI dépasse le cadre de ce projet. Cependant, en guise de point de départ, nous observons le comportement d'un classifieur bayésien en particulier, soit le modèle bayésien gaussien naïf avec variances partagées (section 3.5.1). Celui-ci est basé sur les mêmes suppositions quant à la nature des données que les méthodes d'analyse fMRI traditionnelles. Nous tâchons d'évaluer la pertinence de ces suppositions à travers nos expériences.

3.3.1 Régularisation avec le critère filet élastique (FE)

Comme mentionné précédemment (section 3.2.2), la régularisation $L1$ encourage la parcimonie au niveau des paramètres appris. Elle est généralement très bénéfique lorsqu'il y a un très grand nombre de dimensions qui ne sont pas pertinentes à la tâche de prédiction. Sur données fMRI, cependant, elle risque d'altérer l'interprétabilité des poids en excluant des voxels pertinents, mais redondants ou un peu plus bruités. Les réponses fMRI des voxels, typiquement, covarient. La présence de covariances peut être dûe à l'appartenance à une même région fonctionnelle. Elle peut aussi être introduite par le lissage spatial appliqué en guise de prétraitement et/ou par de légers mouvements du sujet.

Un modèle qui favorise la parcimonie tout en intègrant les corrélations spatiales entre les voxels pourrait éventuellement mener à une meilleure interprétabilité en combattant le sur-apprentissage et évitant les faux négatifs, i.e. les voxels qui sont déclarés à tort comme non-pertinents. Une technique de régularisation appelé **filet élastique** (FE), introduite par Zou et Hastie (2005) [25], est un candidat potentiellement intéressant.

Elle consiste en une combinaison convexe des normes $L1$ et $L2$ et favoriserait la sélection de features (dimensions) corrélés parmis un grand nombre de features non pertinents :

$$argmin_{\boldsymbol{\theta}}\ f(\boldsymbol{\theta}, D) + \lambda_1 \sum_d |\theta_d| + \lambda_2 \Sigma_d ||\theta_d||^2$$

La norme $L1$ déclencherait la sélection des voxels les plus discriminants, tandis que la norme $L2$, plus inclusive, aiderait à englober les voxels qui leur sont corrélés. Cette technique de régularisation semble d'autant plus prometteuse qu'elle a été appliquée avec succès sur données fMRI tout récemment (2009), par Carroll & Schapire [4].

3.3.2 Approches générative et discriminante

Comme expliqué à la section 3.1, dans le cadre d'une tâche de classification, on peut distinguer deux stratégies d'apprentissage complémentaires, l'une **générative**, soit l'approche de classification par inférence bayésienne, et l'autre **discriminante**. Soit $\mathbf{x} \in \mathbf{X} \subseteq \mathbb{R}^d$ un prédicteur et $c \in C := \{1, 2, \ldots, K\}$ une cible. L' approche dite générative consiste à modéliser la distribution $p(\mathbf{x},c)$. Cet estimé est ensuite utilisé pour inférer les probabilités $p(c|\mathbf{x})$ de chacune des cibles en un point de test. Plus la distribution apprise est fidèle à la véritable distribution, plus la performance de prédiction est élevée. L'approche discriminante, quant-à-elle, consiste à ajuster les paramètres de la fonction de décision de manière à optimiser directement la performance de prédiction.

Afin de mieux comprendre les avantages et inconvénients de ces deux approches, nous présentons ici deux types d'algorithmes très apparentés, l'un génératif, l'autre discriminant, et nous analysons les différences au niveau de leurs critère d'optimisation.

Le premier, soit l'algorithme génératif, maximise la **log-vraisemblance** (LV) des exemples de l'ensemble d'entraînement. Plus précisément, il minimise la log-vraisemblance négative, ce qui est équivalent. Soit un ensemble d'exemples d'entraînement $D : \{\mathbf{x}_n, c_n\}_n \subset (\mathbf{X},C)$.

Soit $\mathcal{F}$ l'ensemble des distributions conditionnelles inférables :

$$\mathcal{F} :=\{f \quad |\exists\, p(\mathbf{x}, c) \quad t.q. \quad \forall(\mathbf{x}, c) \in (\mathbf{X}, C), \quad f(\mathbf{x}, c) = p(c|\mathbf{x})\}$$

On cherche $f^* \in \mathcal{F}$ telle que :

$$\begin{aligned} f^* &= \underset{f\in\mathcal{F}}{argmin} \quad \mathcal{L}_G(D, f) \\ \mathcal{L}_G(D, f) &= -LV(D|f) = -\Sigma_n\, log(p(\mathbf{x} = \mathbf{x}_n, c = c_n|f)) \end{aligned}$$

où $p(\mathbf{x}, c|f)$ est la distribution.

$$p(x, c|f) = \underset{p(\mathbf{x},c)\in P_{\mathbf{X},C}(f)}{argmin} \; -\Sigma_n\, log(p(\mathbf{x} = \mathbf{x}_n, c = c_n))$$

Ici, $P_{\mathbf{X},C}(f)$ dénote l'ensemble des distributions $p(\mathbf{x}, c)$ menant à une distribution conditionnelle f.

Nous considérons maintenant un deuxième algorithme, considéré comme l'**équivalent discriminant** du premier. Les fonctions f représentant les distributions conditionnelles sont de même forme paramétriques. Cette fois, on maximise directement la log-probabilités a posteriori des cibles au lieu de la vraisemblance des observations. Plus précisément, on cherche

$$\begin{aligned} f^* &= \underset{f\in\mathcal{F}}{argmin} \quad \mathcal{L}_{\mathcal{D}}(D, f) \\ \mathcal{L}_{\mathcal{D}}(D, f) &= -\Sigma_n log(f(\mathbf{x} = \mathbf{x}_n, c = c_n)) \end{aligned}$$

Examinons maintenant la différence entre le critère génératif $\mathcal{L}_{\mathcal{G}}$ et le critère discriminant $\mathcal{L}_{\mathcal{D}}$. On peut facilement montrer (Bouchard [3]) que

$$\mathcal{L}_G(D, f) - \mathcal{L}_{\mathcal{D}}(D, f) = -\Sigma_n log(p(\mathbf{x} = \mathbf{x}_n|f))$$

où $p(\mathbf{x}|f)$ est telle que

$$p(\mathbf{x}|f) = \underset{p(\mathbf{x})\in P_{\mathbf{X}}(f)}{argmin} \; -\Sigma_n log(p(\mathbf{x} = \mathbf{x}_n))$$

Ici, $P_{\mathbf{X}}(f)$ dénote l'ensemble des distributions $p(\mathbf{x})$ menant à une distribution conditionnelle f.

Cette équation signifie que l'algorithme génératif est plus contraint par rapport à son équivalent discriminant, car les solutions choisies maximisent davantage la probabilité des prédicteurs $\mathbf{x}_n$ de l'ensemble d'entraînement. En fait, lorsque les suppositions quant à la nature des données sont vraies, le biais induit par cette contrainte supplémentaire prévient le sur-apprentissage et tend vers 0 au fur et à mesure que le nombre d'exemples augmente ; les algorithmes génératif et discriminant convergent tous deux vers la même solution lorsque le nombre d'exemples augmente (Bouchard [3]). Dans le cas où les suppositions relatives à la nature des données sont peu réalistes, le biais du modèle génératif grossit et persiste en raison d'une pauvre modélisation de $p(\mathbf{x}, c)$. Il se peut alors fort bien que le modèle discriminant performe mieux peu importe la quantité de données.

Ainsi, l'utilisation d'une approche générative est une option intéressante lorsque les suppositions sous-jacentes intègrent les connaissances à priori relatives aux données. En revanche, lorsque les suppositions sont non-fondées, cette technique est peu indiquée.

Mentionnons que l'algorithme régression logistique est précisément l'équivalent discriminant du modèle bayésien gaussien naïf. Il sera intéressant, dans le présent projet, de comparer ces deux algorithmes en prenant en considération cette relation bien connue.

3.4 Évaluation et sélection de modèles

Parfois, le comportement de l'algorithme d'apprentissage dépend de paramètres dont la valeur est fixée avant l'entraînement. Par exemple, les algorithmes incorporant les critères de régularisation $L1$, $L2$ et FE introduits à la (section 3.2.2) dépendent des degrés de régularisation λ_1 et/ou λ_2 . De tels paramètres sont appelés **hyper-paramètres**. Pour les ajuster, le plus souvent, on effectue quelques essais avec différentes valeurs. Ensuite, on compare les modèles obtenus et la combinaison associée au "meilleur" modèle est sélectionnée. Il existe bien sûr plusieurs procédures pour évaluer les modèles.

Une des procédure les plus simples est la procédure de **validation**. Celle-ci consiste à évaluer les performances des différents modèles sur un ensemble de données indépendant appelé **ensemble de validation**. Les données disponibles pour l'entraînement sont partitionnées en deux ensembles disjoints. Les modèles sont construits à partir de l'un des ensembles et testés (validés) sur l'autre.

Cette procédure est avantageuse de par sa simplicité, mais peu recommendée lorsqu'on dispose de peu de données. En effet, dans ce cas, les performances de prédiction sur l'ensemble de validation risquent d'être instables et de mal représenter les capacités de généralisation des modèles. Il peut alors y avoir sur-apprentissage au niveau des hyper-paramètres, i.e. un hyper-paramètre peut, par pur hasard, être associé à de bons résultats sur l'ensemble de validation, mais pas sur les données en général.

Une autre alternative plus fiable, la procédure de validation croisée par blocs, est illustrée à la Figure 3.10. Cette procédure consiste à effectuer plusieurs procédures de validation successivement et à calculer la moyenne des performances de validation obtenues. D'abord les données disponibles pour l'entraînement sont partitionnées en K sous-ensembles (blocs) disjoints de taille égale. Tour à tour, on réserve l'un des K sous-ensembles pour la validation et on entraîne le modèle en utilisant les données des autres sous-ensembles. Au final, on obtient ainsi K performances de prédiction sur des ensembles indépendants pour chacun des modèles. En guise de critère de comparaison des modèles, on utilise la performance de prédiction moyenne. Le moyennage, ici, a pour effet de stabiliser la performance de prédiction et de prévenir le sur-apprentissage au niveau des hyper-paramètres.

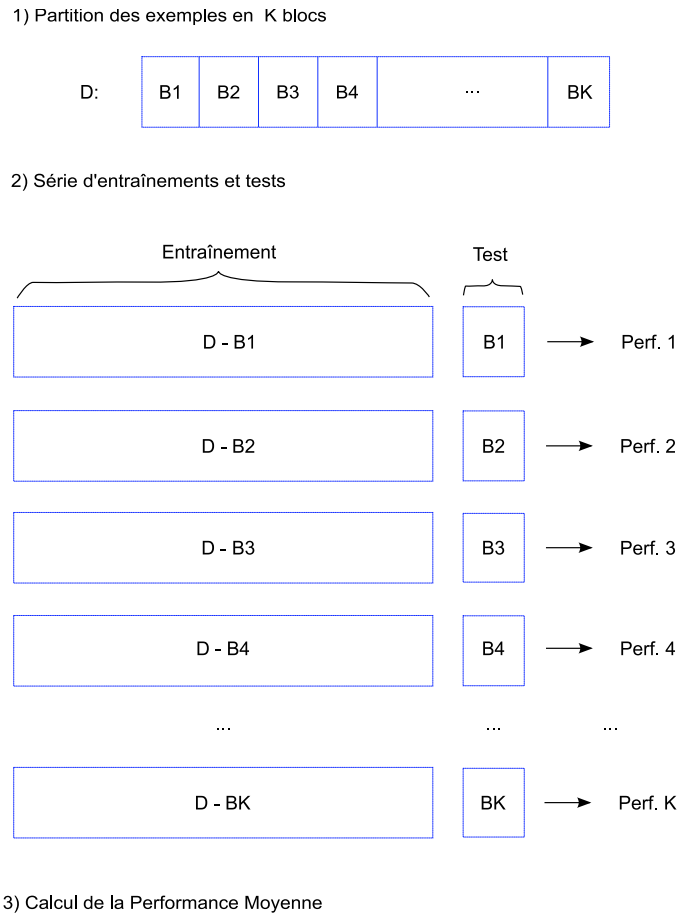


FIGURE 3.10: Procédure d'évaluation des modèles impliquée dans la validation croisée. 1) L'ensemble des exemples disponibles pour l'entraînement (D) est partitionné en K blocs disjoints, de taille identique. 2) Des paires d'ensembles d'entraînement et de test sont formées en réservant tour à tour l'un des blocs pour la phase de test. Pour chaque paire, on entraîne l'algorithme à l'aide de l'ensemble d'entraînement et on mesure la performance de prédiction sur l'ensemble de test. 3) La performance finale est la moyenne des performances en test obtenues pour chacune des paires.

Une fois les hyper-paramètres sélectionnés, l'apprentissage n'est pas tout à fait terminé. Un nouveau modèle est construit en utilisant cette fois toutes les données disponibles pour l'entraînement. La performance de prédiction du modèle final est calculée sur des données de test non utilisées lors du processus de sélection des hyper-paramètres. Cette étape finale permet d'utiliser au maximum les données d'entraînement. De plus, elle permet aussi d'obtenir un estimé non-biaisé de la capacité de généralisation du modèle final. En effet, puisque les hyper-paramètres sont optimisés spécifiquement pour les données de l'ensemble de validation, la performance de prédiction du meilleur modèle sur l'ensemble de validation est un estimé biaisé de la capacité de généralisation de celui-ci.

Évidemment, l'évaluation du modèle final est sujette aux mêmes problèmes que l'évaluation des modèles durant le processus de sélection des hyper-paramètres. En effet, lorsqu'il y a peu de données, la performance de prédiction sur les données de test est aussi sujette à beaucoup de variance.

Une solution assez commune est d'imbriquer deux procédures de validation croisées l'une dans l'autre, i.e. d'effectuer une **double validation croisée**. Le premier niveau d'imbrication sert à obtenir

une série de performances de prédiction sur des ensembles de test indépendants. Le deuxième niveau sert à effectuer la sélection des hyper-paramètres pour chacun des modèles du premier niveau.

Cette procédure permet d'obtenir un estimé plus stable de la capacité de généralisation, cependant cet estimé n'est plus attribuable à un seul modèle, mais bien à un ensemble de modèles. Cela pose problème pour notre application, puisque nous cherchons surtout à relever les patterns pertinents à la tâche de prédiction. Le fait de disposer de plusieurs modèles complique le processus d'analyse. Pour cette raison, nous nous contentons d'une procédure de validation croisée pour la sélection des hyper-paramètres, avec test sur un ensemble indépendant pour mesurer la capacité de généralisation du modèle final.

3.5 Classifieurs linéaires

Ici, nous décrivons birèvement les algorithmes d'apprentissage utilisés dans le présent projet, soit le **modèle bayésien gaussien naïf** (BGN) avec variances partagées, l'algorithme **régression logistique** (RL) et l'algorithme **machine à vecteurs de support** (MVS) linéaire. Pour davantage de détails, voir Bishop [1].

Il s'agit tous de classifieurs linéaires. Donné un prédicteur $\mathbf{x} \in \mathbb{R}^d$et une cible $c \in \{\pm 1\}$, on cherche $\mathbf{w} \in \mathbb{R}^d$ et $b \in \mathbb{R}$ tels que la frontière linéaire $\mathbf{w}\mathbf{x} + b = 0$ permet de bien séparer les données. S'il existe $\mathbf{w}$ et b tels que les données sont parfaitement classifiées, on dit que les données sont linéairement séparables.

Parmis toutes les frontières qui permettent de séparer à peu près correctement les données d'entraînement, toutes n'ont pas la même chance d'être sélectionnées par les différents algorithmes. En effet, ils incorporent tous une certaine forme de contrôle de capacité, de par des suppositions explicites, ou de par la stratégie d'optimisation employée. Le modèle bayésien gaussien naïf (BNG) est un classifieur à inférence bayésienne et suppose que les distributions $p(\mathbf{x}|c)$ sont gaussiennes. L'algorithme régression logistique (RL) suppose que les probabilités à posteriori $p(c|\mathbf{x})$ ont une forme sigmoïdale et maximise la probabilité des cibles des exemples d'entrainement. L'algorithme machine à vecteur de support (MVS), quant à lui, cherche $\mathbf{w}$ qui maximise la distance des exemples d'entraînement à la frontière de décision.

À la section 3.5.4, nous présentons différentes manières de visualiser les paramètres des classifieurs, dépendamment du classifieur utilisé et du nombre de cibles. Dans le cas binaire, lorsque la fonction de décision choisie approche la frontière de décision idéale, les paramètres représentent la direction dans laquelle les données varient le plus d'une cible à l'autre. Dans le cas multiclasse, on s'intéresse plutôt à la **sensibilité** du classifieur aux différents voxels, déterminée de par l'amplitude des différents paramètres reliés au voxel. Il existe plusieurs extensions multiclasses, dépendamment des algorithmes, et par conséquent plusieurs définitions de sensibilité possibles.

3.5.1 Modèle bayésien gaussien naïf (BGN) avec variances partagées

Le modèle bayésien gaussien naïf avec variances partagées fonctionne par inférence bayésienne. L'entraînement consiste à estimer les distributions conditionnelles $p(\mathbf{x}|c)$ et $p(c)$. En un point de test $\mathbf{x}$, en usant de la règle de Bayes, on prédit la cible pour laquelle la probabilité $p(c|\mathbf{x})$ est maximale.

Les distributions conditionnelles $p(\mathbf{x}|c)$ sont modélisées par des gaussiennes multivariées de moyennes $\boldsymbol{\mu}_c$ distinctes et de matrice de covariance diagonale$\sum$ partagée.

Le fait que de la matrice de covariance $\sum$ soit diagonale implique l'indépendance des prédicteurs, i.e, ici, des réponses fMRI dans les différents voxels. Cette hypothèse d'indépendance est communément appelée **hypothèse naïve**.

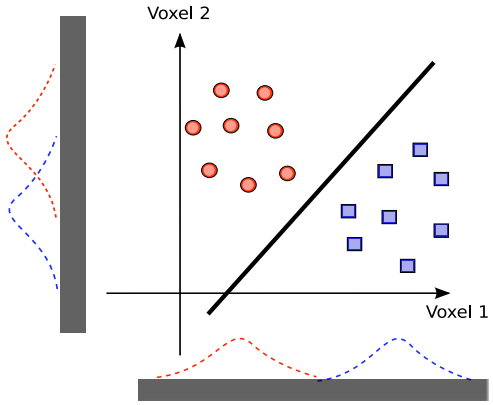


FIGURE 3.11: Modèle bayésien gaussien naïf avec variances partagées pour un prédicteur en 2 dimensions. Les carrés bleus représentent les exemples de la classe $c = 1$. Les cercles rouges représentent les exemples de la classe $c = 2$. Les 2 dimensions (ici voxels) sont supposées indépendantes. Les formes des densités $p(voxel_1|c)$ et $p(voxel_2|c)$ sont indiquées sur chacun des axes. Se sont des gaussiennes de moyennes distinctes et variances partagées. La frontière de décision, tracée en noir, est linéaire.

Lorsque les probabilités marginales p(c) ne sont pas connues au départ, elles sont généralement estimées à partir des proportions des exemples associés à chacune des cibles dans les données. Dans nos expériences, elles sont connues ; elles sont toutes égales, i.e. les cibles, à priori, sont équiprobables.

La Figure 3.11 illustre la frontière de décision pour un prédicteur en 2 dimensions, pour le cas binaire. Comme pour tous les classifieurs utilisés dans le présent projet, elle est linéaire. Pour le cas binaire, cela signifie qu'il existe, $\mathbf{w} \in \mathbb{R}^d$, $b \in \mathbb{R}$ tels que cette frontière correspond à l'ensemble $\mathbf{wx} + b = 0$.

$$\begin{aligned} \mathbf{w} &= (\hat{\boldsymbol{\mu}}_1 - \hat{\boldsymbol{\mu}}_2)^T \hat{\sum}^{-1} \\ b &= -\frac{1}{2}\left(\hat{\boldsymbol{\mu}}_1^T \hat{\sum}^{-1} \hat{\boldsymbol{\mu}}_1 - \hat{\boldsymbol{\mu}}_2 \hat{\sum}^{-1} \hat{\boldsymbol{\mu}}_2\right) \end{aligned}$$

Ici, $\hat{\boldsymbol{\mu}}_1, \hat{\boldsymbol{\mu}}_2 \in \mathbb{R}^d$ correspondent aux moyennes échantillonales des prédicteurs pour chacune des cibles et $\hat{\sum}$ est une matrice diagonale telle que $\hat{\sum}_{i,i}$correspond à la variance échantillonale du ième prédicteur toutes cibles confondues.

Remarquons que si les données ont été pré-normalisées, le vecteur de poids $\mathbf{w}$ correspond (à un facteur d'échelle près) aux statistiques de Student traditionnellement utilisées pour analyser les données fMRI.

Pour cette raison, le modèle bayésien gaussien naïf avec variances partagées est un point de référence intéressant aux méthodes d'analyse traditionnelles. D'abord, la performance de prédiction du modèle appris indique à quel point les statististiques de Student sont significatives dans leur ensemble. Les probabilités de l'hypothèse nulle calculées lors de l'analyse traditionnelle indiquent quant à elles à quel point les statistiques de Student sont significatives isolément. Il peut être intéressant de voir à quel point le fait d'analyser les statistiques de Student de manière multivariée plutôt qu'univariée donne un avantage. La comparaison de la performance de prédiction du modèle à celles de modèles

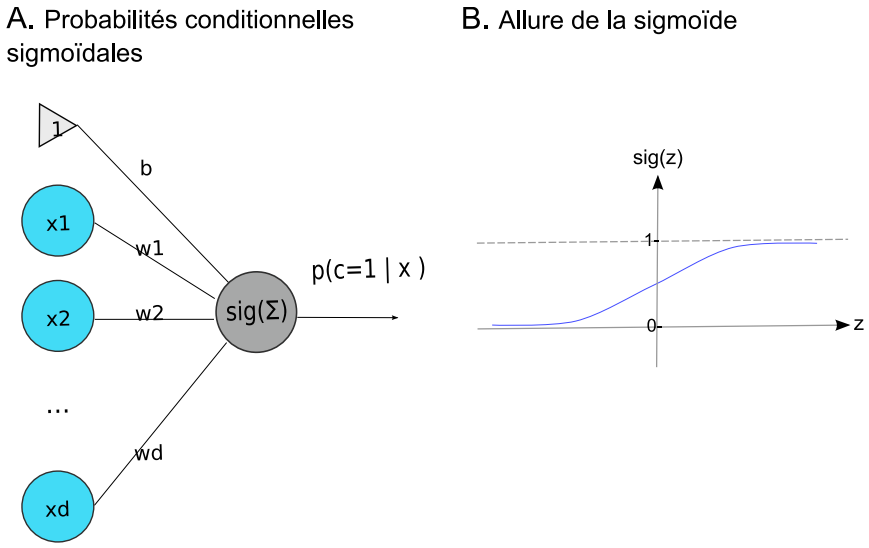


FIGURE 3.12: Mécanique de l'algorithme régression logistique pour un problème de classification binaire. A. Le classifieur effectue un produit scalaire entre le prédicteur $\mathbf{x}$ et un vecteur de poids $\mathbf{w}$. Ensuite, une fonction sigmoïde est appliquée. La sortie représente $p(c = 1|\mathbf{x})$. B. Allure de la fonction sigmoïde. Il s'agit d'un fonction strictement croissante entre 0 et 1.

obtenus via d'autres algorithmes permet d'évaluer la pertinence des statistiques de Student à la tâche de discrimination. Notamment, comme expliqué à la section 3.3.2, la comparaison à l'équivalent discriminant du modèle bayésien gaussien naïf, soit l'algorithme régression logistique, peut en dire long sur la véracité des hypothèses énoncées ci-haut, relatives à la nature des données.

3.5.2 Régression logistique (RL)

L'algorithme régression logistique est un algorithme de classification linéaire discriminant avec sorties à interprétation probabilistes. On suppose que les probabilités $p(c|\mathbf{x})$ ont une certaine forme paramétrique, i.e. que $p(c|\mathbf{x}) \approx p(c|\mathbf{x}; \mathbf{w})$. En un point de test $\mathbf{x}$, le classifieur prédit la cible pour laquelle la probabilité $p(c|\mathbf{x}; \mathbf{w})$ estimée est maximale.

Les paramètres $\mathbf{w}$ sont optimisés de manière à maximiser la log-vraisemblance des cibles des exemples :

$$\mathcal{L}_D(\mathbf{w}) = \sum_n log\left(p(c_n|\mathbf{x}_n, \mathbf{w})\right)$$

Ainsi, les paramètres $\mathbf{w}$ sont ajustés de manière à ce que les distributions $p(c|\mathbf{x_n}, \mathbf{w})$ soient le moins ambigües possibles, i.e. à ce qu'elles soient aiguillées en $c = c_n$.

La mécanique de ce classifieur est illustrée à la Figure 3.12, pour un problème de classification binaire. La probabilité $p(c = 1|\mathbf{x})$ est modélisée à l'aide d'une sigmoïde appliquée sur le produit scalaire entre le prédicteur $\mathbf{x}$ et un vecteur de poids $\mathbf{w}$. La probabilité $p(c = 2|\mathbf{x})$ est tout simplement fixée à $1 - p(c = 1|\mathbf{x})$.

Pour un problème de classification multiclasse, un vecteur de poids $\mathbf{w}_c$ est optimisé pour chaque cible. La fonction sigmoïde est remplacée par une fonction appelée **softmax** qui ramène les probabilités des différentes cibles entre 0 et 1 et fait en sortent qu'elles somment toujours à 1 :

$$p(c|\mathbf{x}) = \frac{exp(\mathbf{w}_c\mathbf{x})}{\sum_{c'} exp(\mathbf{w}_{c'}\mathbf{x})}$$

Pour $\mathbf{w}$ et c fixé, la courbe $p(c|\mathbf{x}, \mathbf{w})$ conserve une forme sigmoïdale.

Comme il n'existe pas de solution analytique au problème d'optimisation, la solution est obtenue par des méthodes d'optimisation numériques. Nous utilisons pour notre part une descente de gradient conjuguée. Voir Shewchuk[20] pour une description de cette méthode.

L'utilisation de formes sigmoïdales pour modéliser les probabilités des cibles à posteriori rend le critère d'optimisation $\mathcal{L}_D(\mathbf{w})$ problématique. En effet, s'il existe un vecteur $\mathbf{w}$ permettant de classifier parfaitement les données, alors le minimum de la fonction de coût est atteint lorsque la norme de $\mathbf{w}$ est infinie, lorsque les probabilités des cibles des exemples d'entraînement prennent toutes la valeur 1. Pour éviter une divergence de la procédure d'optimisation, on ajoute une régularisation $L1$ ou $L2$ (section 3.2.2) au critère $\mathcal{L}_D(\mathbf{w})$ pour contraindre la norme de $\mathbf{w}$. Une autre alternative est d'appliquer une procédure d'arrêt prématuré de l'optimisation (section 3.2.2).

Comme mentionné précédemment (section 3.3.2), l'algorithme régression logistique est précisément l'équivalent discriminant du modèle bayésien gaussien naïf avec variances partagées. En effet, si les distributions $p(\mathbf{x}|c)$ sont modélisées par des Gaussiennes multivariées de moyennes $\boldsymbol{\mu}_c$ et de même matrice de covariance $\sum$ et que la distribution $p(c)$ est multinomiale, alors, $p(c|\mathbf{x})$ a une forme sigmoïdale.

3.5.3 Machine à vecteurs de support (MVS)

L'algorithme machine à vecteurs de support (MVS) a été introduit pour la première fois en 1963 par Vapnik [22]. Tout comme régression logistique, cet algorithme est discriminant et cherche à éviter les décisions ambigües. Cette fois, cependant, on ne cherche pas à maximiser les probabilités à posteriori des cibles. En fait, le modèle final n'a pas d'interprétation probabiliste. La technique d'optimisation utilisée est plutôt de maximiser la **marge**, soit la plus petite distance d'un exemple à la frontière de décision.

La Figure 3.13 illustre cette stratégie pour un problème de classification binaire 2d. L'idée est que, parmis les droites qui permettent de séparer correctement les exemples, celle qui permet une généralisation optimale se trouve environ à égale distance des exemples les plus proches de la frontière, i.e. au centre sur la figure ci-dessous. Plus précisément, la frontière optimale est construite à partir de deux hyperplans parallèles qui passent par les points à proximité de la frontière. Ces points sont appelés les **vecteurs de support**.

L'algorithme MVS n'est pas typiquement linéaire. En fait, en 1992, Boser, Guyon et Vapnik [2] ont proposé une extension non-linéaire de l'algorithme, maintenant reconnue comme la version officielle. Cette dernière permet de projeter non linéairement les données dans un nouvel espace et à rechercher une frontière de décision linéaire dans ce nouvel espace. Bien que la forme de la frontière dans l'espace de projection soit linéaire, elle peut prendre des formes très complexes dans l'espace original, dépendamment de l'espace de projection choisi. Cette astuce est particulièrement avantageuse lorsque les données ne sont pas linéairement séparables dans l'espace original, mais qu'elles le sont lorsqu'on les projette dans un nouvel espace de dimensionalité plus élevée. En ce qui nous concerne, puisque les données sont déjà dans un espace de dimensionalité très élevée, nous nous en tenons à la version purement linéaire de l'algorithme.

L'algorithme, dans sa version originale, se limite seulement aux solutions qui permettent une classification parfaite des exemples d'entraînement. Cela peut entraîner du sur-apprentissage et poser pro-

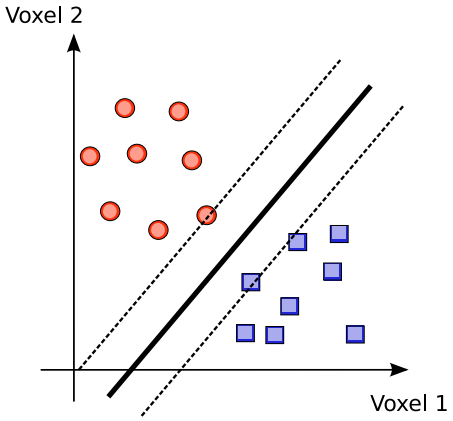


FIGURE 3.13: Problème de classification binaire en deux dimensions. Les carrés bleus représentent les exemples de la classe C=-1. Les cercles rouges représentent les exemples de la classe C=1. La frontière entre les deux classes est tracée en noir. On cherche la frontière linéaire qui maximise l'espacement (marge) entre les deux classes. Cette frontière est située entre deux hyperplans parallèles passant par des exemples situés près de la frontière, soit les vecteurs de support.

blème lorsque les données ne sont pas linéairement séparables. Il existe des variantes qui permettent de **relaxer** un peu le problème original, i.e., d'allouer un certain nombre d'exemples mal classifiés, moyennant une pénalité. Dans le présent projet, nous utilisons en fait les variantes **cMVS** et **nuMVS**. Bien que formulées différemment, ces dernières sont équivalentes, i.e. convergent vers la même solution.

La variante cSVM, introduite en 1995 par Cortes et Vapnik [5], est obtenue en insérant le coût de mauvaise classification suivant au problème d'optimisation :

$$C \sum_n \varepsilon_n,$$

Les contraintes de séparabilité originales $t_n y(x_n) \geq 1$ sont remplacées par $t_n y(x_n) \geq 1 - \varepsilon_n$. Ici, $\varepsilon_n = 0$ si $\mathbf{x}_n$ est correctement classifié et est à l'extérieur de la marge. Sinon, ε_n est correspond à la distance entre la prédiction $y(\mathbf{x}_n) = \mathbf{w}\mathbf{x}_n + b$ et la marge.

L'idée est de rendre la marge originale "molle" , i.e. de permettre, moyennant une pénalité, que certains points soient situés à l'intérieur des marges ou du mauvais côté de l'hyperplan. La variable C, ici, est un hyper-paramètre et contrôle la rigidité de la marge. Lorsqu'on augmente sa valeur jusqu'à l'infini, on retrouve le problème d'optimisation original, sans relaxation.

Une autre extension de MVS équivalente à cMVS, mais formulée différemment est la variante nuMVS. On introduit cette fois la pénalité :

$$-\nu p + \tfrac{1}{N} \sum_n \varepsilon_n$$

Les contraintes de séparabilité originales $t_n y(x_n) \geq 1$ sont remplacées par $t_n y(x_n) \geq p - \varepsilon_n$ et $p \geq 0$. Ici, on introduit une nouvelle variable p qui est optimisée en même temps que les autres paramètres. Les variables ε_n répondent à la même définition que pour la variante cMVS.

Ici, ν est un hyper-paramètre entre 0 et 1 qui vient remplacer l'hyper-paramètre C. On peut l'interpréter comme une borne inférieure sur la proportion de vecteurs de support et borne supérieure sur le taux de mauvaise classification des points de l'ensemble d'entraînement.

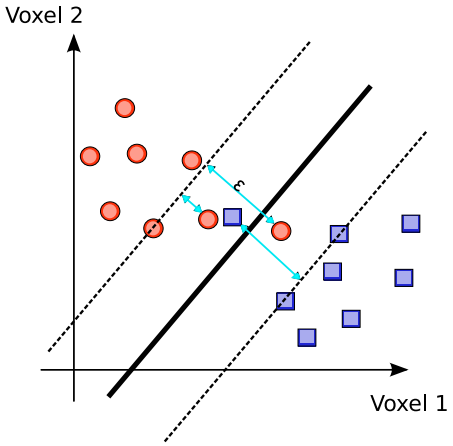


FIGURE 3.14: Variante cMVS pour un problème de classification binaire en deux dimensions. Les cercles rouges représentent les exemples associés à la cible 1. Les carrés bleus représentent les exemples associés à la cible 2. La frontière de décision finale est tracée en noir. Les marges sont parallèles à la frontière de décision et sont indiquées par des traits plus fins. L'idée est de permettre, moyennant une pénalité, que certains points soient situés à l'intérieur des marges ou du mauvais côté de l'hyperplan. Ici, 2 exemples sont mal classifiés et un exemple est bien classifié, mais situé à l'intérieur de la marge. Les distances à la marge (pénalités) de chacun de ces exemples sont représentées en turquoise.

Pour davantage de détails concernant l'algorithme MVS, de même que les variantes cMVS et nuMVS, voir Bishop [1].

Contrairement à l'algorithme régression logistique, l'extension multiclasse ($K \geq 3$ cibles) n'est pas aisée pour l'algorithme MVS. Une complication est que le concept de marge est difficile à définir pour plus de deux cibles. Il existe des extensions (ex : Weston & Watkins [23]) consistant à entraîner K classifieurs simultanément en maximisant la marge par rapport à chacune des frontières. Néanmoins, en pratique, elles sont peu utilisées, car elles sont computationnellement coûteuses. La plupart des implémentations MVS multiclasses disponibles misent plutôt sur une combinaison de plusieurs classifieurs binaires entraînés indépendamment les uns des autres.

L'approche **un-contre-tous** consiste à utiliser une combinaison de classifieurs linéaires binaires entraînés pour discriminer une cible en particulier. On obtient ainsi des fonctions $f_c(\mathbf{x}) = \mathbf{w}_c\mathbf{x} + b$ qui permettent d'associer un score à chaque cible. Pour un nouveau point de test, on prédit la cible qui obtient le score maximum. Le fait que chaque classifieur est entraîné indépendamment des autres peut être problématique, car rien ne garantit que les paramètres appris sont d'échelles équivalentes d'un classifieur à l'autre. Il se peut que le score d'une cible c soit toujours d'amplitude plus élevée à cause de la norme du vecteur de poids $\mathbf{w}_c$ correspondant.

Une autre alternative est d'entraîner un classifieur pour chaque paire de cibles, i.e. $\frac{K(K-1)}{2}$ classifieurs, et de combiner les votes de ceux-ci pour prendre une décision. Cette approche est appelée **un-contre-un** . Pour un ensemble de N exemples, le temps requis passe de $O(KN^2)$ à $O(K^2N^2)$. Malgré la perte d'efficacité, cette approche demeure avantageuse, car cette fois, on évite le problème d'échelle. C'est en fait l'approche que nous avons utilisée dans le présent projet.

3.5.4 Visualisation des poids des classifieurs

On veut mesurer comment la réponse fMRI en un voxel influence la fonction de décision finale du classifieur, soit la **sensibilité** du classifieur aux différents voxels. Nous décrivons ici les différentes

mesures de sensibilité utilisées dans le présent projet.

Dans le cas d'un problème de classification binaire, nous utilisons directement les paramètres du classifieur. Un classifieur linéaire binaire fonctionne à partir d'un produit scalaire entre le prédicteur $\mathbf{x} = (x_1, x_2, ..., x_d) \in \mathbb{R}^d$ et un vecteur de poids $\mathbf{w} = (w_1, w_2, ..., w_d) \in \mathbb{R}^d$ (voir Figure 3.5). Si $\mathbf{w}$ approche la solution idéale, il représente la direction dans laquelle les données varient le plus d'une cible à l'autre. Chacun des poids w_v est associé à un voxel v. Ainsi, on définit

$$sens(v) = w_v$$

Pour un problème de classification à $K \geq 3$ cibles (multiclasse), une comparaison directe des poids du classifieur pourrait compliquer l'analyse, car les voxels sont reliés à plus d'un poids. Dans le cas du modèle bayésien gaussien naïf avec variances partagées et de l'algorithme régression logistique, le classifieur fonctionne à partir de K vecteurs de poids, i.e. un vecteur $\mathbf{w}_c$ par cible.

$$\begin{aligned} \mathbf{w} &= (\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_K) \\ \mathbf{w}_c &= (w_{c1}, w_{c2}, ..., w_{cd}) \end{aligned}$$

Ainsi, chaque voxel v est relié à K poids. Pour ce cas particulier, nous définissons la sensibilité du classifieur à un voxel v comme l'amplitude maximale rencontrée parmis les poids $\mathbf{w}$ du classifieur rattachés au voxel v.

$$sens(v) = max \quad \{|w_{1v}|, |w_{2v}|, ..., |w_{Kv}|\}$$

Nous varions légèrement cette définition pour l'algorithme machine à vecteurs de support. L'implémentation multiclasse que nous utilisons est l'approche un contre un. Comme expliqué à la section 3.5.3, un tel classfieur fonctionne en combinant les décisions d'une multitude de classifieurs binaires, plus précisément un classifieur binaire pour chaque paire de cibles. Ainsi, cette fois, les paramètres $\mathbf{w}$ du classifieurs consiste en $\frac{K(K-1)}{2}$ vecteurs de poids et, donc, chaque voxel est relié à $\frac{K(K-1)}{2}$ poids.

$$\mathbf{w} = \{\mathbf{w}_{c,c'} \in \mathbb{R}^d\}, 1 \leq c, c' \leq K, c \neq c',$$

Comme les différents classifieurs binaires ne sont pas nécéssairement d'échelles compatibles, de se fier simplement au poid d'amplitude maximale pourrait être trompeur. Afin de remédier à ce problème, nous nous ramenons à un ensemble de K vecteurs de poids en prenant la moyenne, pour chaque cible, des classifieurs impliquant celle-ci. Ainsi, on calcule

$$\hat{\mathbf{w}}_c = \frac{1}{K-1} \sum_{c' \neq c} \mathbf{w}_{c,c'}$$

Ensuite, la sensibilité au voxel v est définie comme l'amplitude maximale rencontrée parmis les poids moyens $\hat{\mathbf{w}}_c$rattachés au voxel v :

$$sens(v) = max \quad \{|\hat{\mathbf{w}}_{1v}|, |\hat{\mathbf{w}}_{2v}|, ..., |\hat{\mathbf{w}}_{Kv}|\}$$

Le fait de visualiser les paramètres du classifieur peut permettre de détecter des patterns à large échelle et donner des indices quant à la localisation des régions d'intérêt. Cependant, il importe de garder en mémoire que la performance de prédiction du classifieur est une statistique multivariée, i.e. attribuable à l'ensemble des poids du classifieur. Les différentes mesures de sensibilités définies ici ne permettent pas de localiser les régions d'intérêt avec certitude et rigueur. En effet, elles ne représentent

que l'influence des voxels isolément ; il se peut que certains voxels aient peu d'influence isolément, mais une influence importante lorsqu'on les combine ensemble. Il ne faut pas non plus négliger l'effet du sur-apprentissage.

À la section 3.6 qui suit, nous introduisons une stratégie appelée **recherche locale** qui permet mieux de dégager l'importance de groupes de voxels plus ou moins restreints.

3.6 Recherche locale et détection des régions d'intérêt

Plutôt que d'utiliser tous les voxels en même temps, il peut être avantageux de travailler avec de petits volumes de voxels à la fois. En effet, puisque la performance de prédiction d'un modèle est attribuable à l'ensemble des voxels, lorsqu'on utilise de gros ensemble de voxels, celle-ci ne nous renseigne plus sur les contributions plus isolées des voxels. Les poids appris par le classifieur peuvent fournir quelques indices à ce sujet, mais ne sont pas toujours très fiables. En effet, les faux positifs, i.e. les patterns dûs au sur-apprentissage, sont pratiquement indistinguables des patterns véritablement pertinents. Aussi, de se fier à l'influence isolée des voxels peut être trompeur. En fait, ce faisant, on retombe précisément dans les paradigmes de l'analyse standard.

Afin de mieux dégager la contribution d'ensembles de voxels précis, une stratégie possible est d'utiliser un algorithme de **recherche locale**. Ce dernier consiste à mesurer la quantité d'information en différents endroits du cerveau en considérant seulement un petit volume de voxels à la fois. Plus précisément, on produit une carte d'**information locale** qui, en chaque voxel, indique la quantité d'information contenue dans le voxel et son voisinage immédiat.

Il existe plusieurs types de recherche locale. Notamment, on peut utiliser différentes tailles et formes de voisinage : sphères, cubes, appartenance à une même circonvolution du cortex, etc. Quant à nous, nous utilisons des sphères de différents rayons, centrées sur les différents voxels. Bien sûr, on peut aussi utiliser diverses statistiques pour mesurer la quantité d'information. Dans [9], Kriegeskorte utilise une extension multivariée d'une statistique de Student basée sur la **distance de mahalanobis**. Pour notre part, nous nous en remettons à la performance d'un classifieur linéaire. Nous produisons une carte de **performance locale**, où chaque voxel se voit attribuer la performance d'un classifieur linéaire construit à partir des voxels situés à proximité.

Le but de la recherche locale est aussi, éventuellement, de détecter quelles sont les régions du cerveau impliquées dans la tâche cognitive étudiée, i.e. les régions d'intérêt. Afin de trancher cette question avec davantage de rigueur, on transforme la carte d'information en carte de **significativité**. C'est-à-dire, on remplace la statistique d'information considérée par la probabilité d'observer cette statistique en l'absence d'information. Ensuite, par un seuillage, on peut déclarer que certains groupes de voxels sont impliqués dans le processus cognitif avec une certaine probabilité. Ce procédé peut être vu comme une généralisation des tests de Student.

Lorsque la statistique d'information considérée est la performance du classifieur, il existe plusieurs manière de calculer la significativité. Nous présentons deux solutions possibles à la section 3.7 qui suit, soit la procédure de randomisation et l'approximation binomiale.

Le problème de détection des régions d'intérêt peut aussi être vu comme un problème d'apprentissage automatique classique qui est la sélection de "features" . En effet, on cherche à démêler les voxels qui sont pertinents à la tâche de classification de ceux qui ne le sont pas. La régularisation $L1$, introduite à la section 3.2.2, est en fait considérée comme une méthode de sélection de "feature" intégrée. En effet, cette dernière encourage la parcimonie, i.e. à ce que la plupart des paramètres aient pour valeur 0. Dans le cas d'un problème de classification binaire, si le poids relié à un voxel prend la valeur 0, alors

cela signifie que, pour le classifieur, la réponse fMRI en ce voxel est non-pertinente à la tâche. Seuls les voxels permettant d'améliorer substantiellement la performance du classifieur conservent une influence. Dans le présent projet, nous cherchons à savoir si la régularisation $L1$, ou encore la régularisation FE (combinaison $L1$ et $L2$) pourrait être une manière efficace de détecter les régions d'intérêt.

3.7 Tests de significativité

La performance d'un algorithme d'apprentissage est-elle statistiquement significative ? Plus précisément, quelle est la probabilité d'observer une certaine performance sous l'**hypothèse nulle**, i.e. lorsque les données ne contiennent pas d'information ? Trouver la réponse à cette question n'est pas trivial puisqu'on ne connaît pas, à priori, la distribution de la statistique de performance sous l'hypothèse nulle.

Pour un problème de classification à deux cibles où la statistique de performance considérée est le taux d'exemples bien classifiés, une option possible est de supposer une forme **binomiale** de paramètre $p = 0.5$. Plus précisément, on suppose que, sous l'hypothèse nulle, pour chaque exemple présenté, le classifieur tire une cible au hasard et a donc une probabilité de succès de $p = 0.5$.

La validité de cette approximation est discutable. Elle ne tient pas compte de la variance du classifieur dans ses performances, variance qui dépend de l'algorithme d'apprentissage utilisé, des données et de la composition des ensembles d'entraînement et de test. De plus, même lorsque l'hypothèse nulle prévaut, le classifieur final ne se comporte pas de manière complètement aléatoire. Notamment, il est consistant dans ses erreurs ; si on lui présente le même exemple deux fois, il prédira la même cible deux fois.

Une alternative, pour un problème supervisé, est de faire appel à un procédé appelé **randomisation**. Afin de modéliser le comportement du classifieur en l'absence d'information, on remplace les cibles originales des exemples d'entraînement par des cibles choisies aléatoirement. On entraîne ensuite le classifieur avec les cibles ainsi décorrélées et on mesure sa performance sur l'ensemble de test pour lequel les cibles originales ont été conservées. On répète le processus plusieurs fois avec de nouvelles cibles. À partir de l'échantillon de performances obtenues, on estime la distribution sous-jacente en construisant un histogramme.

La randomisation semble plus crédible que l'approximation binomiale, néanmoins cette technique est aussi beaucoup plus coûteuse computationnellement. Si la randomisation doit être appliquée à de nombreuses reprises, par exemple dans le cas d'une recherche locale, cela peut poser problème. Dans le présent projet, nous cherchons à savoir à quel point les bénéfices de la randomisation sont substantiels.

Chapitre 4

Organisation des expériences

Dans ce chapitre, nous présentons les expériences réalisées dans le cadre de ce projet.

Les données fMRI dont nous diposons ont été acquises par Schönwiesner (2009) [14], lors d'une étude récente portant sur l'encodage des sons complexes dans le cerveau humain. Nous rappelons brièvement les buts et résultats de cette étude antérieure et situons nos propres buts par rapport à celle-ci. Nous donnons aussi un aperçu du processus de scannographie et décrivons les étapes de prétraitement des données.

Les expériences consistent à analyser les relations entre les réponses fMRI et les différentes conditions de stimulation à l'aide d'algorithmes d'apprentissage. Nous appliquons des classifieurs linéaires standards, soient le modèle bayésien gaussien naïf (BGN) avec variances partagées, l'algorithme machine à vecteur de supports avec relaxation (cMVS) linéaire et l'algorithme régression logistique (RL). Nous appliquons aussi un algorithme de recherche locale. Certaines expériences ont surtout pour but d'évaluer la pertinence des algorithmes en tant qu'outils d'analyse. D'autres ont pour but de compléter l'analyse effectuée par Schönwiesner.

4.1 Contributions de la présente étude

Selon des études effectuées chez les animaux et des études portant sur la perception visuelle, les sons complexes seraient encodés par des populations de neurones réglés à différentes modulations spectro-temporelles. Afin de valider cette hypothèse chez l'humain, les réponses fMRI de 8 sujets exposés à 49 modulations spectro-temporelles différentes ont été acquises par Schönwiesner [14].

Les modulations spectro-temporelles en question sont des sons complexes appelés **ondulations dynamiques**. Mathématiquement, leurs spectrogrammes (Figure 4.1A) sont des sinus bidimensionnels purs, définis par deux paramètres Ω et ω. Le paramètre Ω est la fréquence de la modulation appliquée sur l'axe spectral. Le second paramètre, ω, est la fréquence de la modulation appliquée sur l'axe temporel. L'ensemble des valeurs possibles pour les paramètres Ω et ω définit un espace mathématique, soit l'**espace des ondulations dynamiques**. À la Figure 4.1B, les spectrogrammes des 49 ondulations dynamiques utilisées par Schönwiesner sont positionnés dans cet espace.

Schönwiesner dresse un inventaire des réactions des voxels face aux différentes ondulations dynamiques. Plus précisément, les voxels sont regroupés en catégories, selon la répartition de leurs réponses moyennes dans l'espace des ondulations dynamiques. Un comportement fréquent, selon cette analyse, est la focalisation des réponses les plus fortes, i.e. des "préférences", des voxels. La Figure 4.2A montre

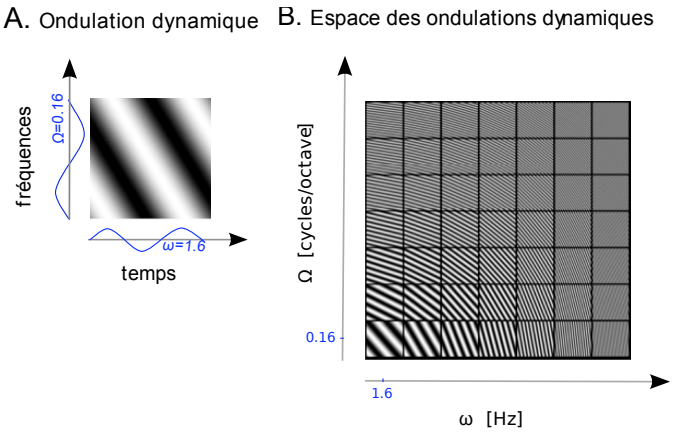


FIGURE 4.1: Ondulations dynamiques utilisées par Schönwiesner. A. Spectrogramme d'une ondulation dynamique. L'axe horizontal représente le temps, l'axe vertical représente les fréquences à l'échelle logarithmique (octaves). Mathématiquement, dans l'espace spectro-temporel, les ondulations dynamiques sont des sinus bidimensionnels. Une modulation de fréquence Ω =0 .16 est appliquée sur l'axe spectral. Une modulation de fréquence $\omega = 1.6$ est appliquée sur l'axe temporel. B. Spectrogrammes des ondulations dynamiques utilisées par Schönwiesner positionnés selon leurs modulations spectrales Ω et temporelles ω. Nous appelons l'espace mathématique défini par ces deux paramètres "espace des ondulations dynamiques".

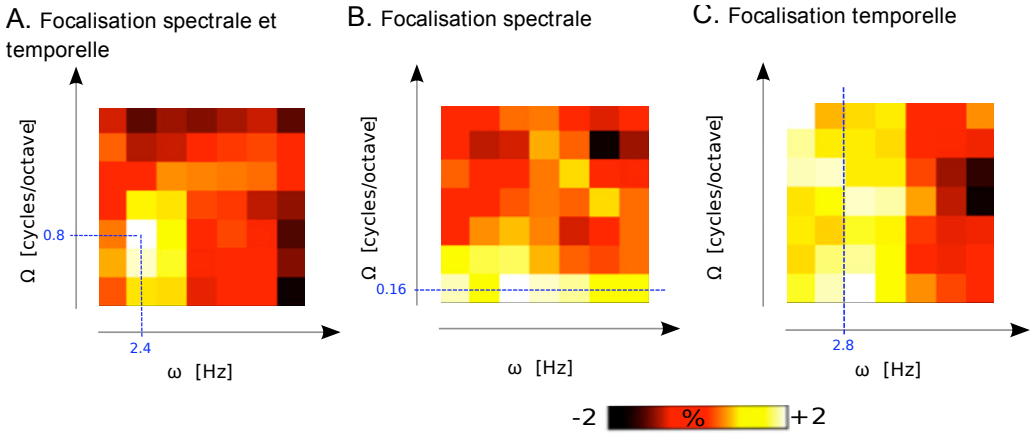


FIGURE 4.2: Exemples de focalisation des réponses de 3 voxels dans l'espace des ondulations dynamiques. A. Focalisation de la réponse du voxel autour du point ($\Omega = 0.8, \omega = 2.4$) Les carreaux représentent les différentes ondulations dynamiques, positionnées selon leurs paramètres Ω et ω, i.e. selon leurs modulations spectrales et temporelles. Les couleurs des carreaux représentent les réponses fMRI moyennes (ré-échelonnées entre -2 et 2) du voxel aux ondulations dynamiques correspondantes. B. Idem A, mais focalisation de la réponse du voxel sur l'axe Ω seulement, en Ω =0 .16. C. Idem A, mais focalisation de la réponse du voxel sur l'axe ω seulement, en $\omega = 2.8$. La variété et la focalisation observées dans les réponses moyennes des voxels appuient l'hypothèse d'encodage postulée.

un exemple de focalisation autour de l'ondulation dynamique ($\Omega = 0.8, \omega = 2.4$) pour un voxel en particulier. En B et C, on voit aussi des exemples de focalisation sur un seul axe de l'espace des ondulations dynamiques (Ω en B et ω en C).

La variété et la focalisation dans les préférences des voxels relevées par Schönwiesner appuient

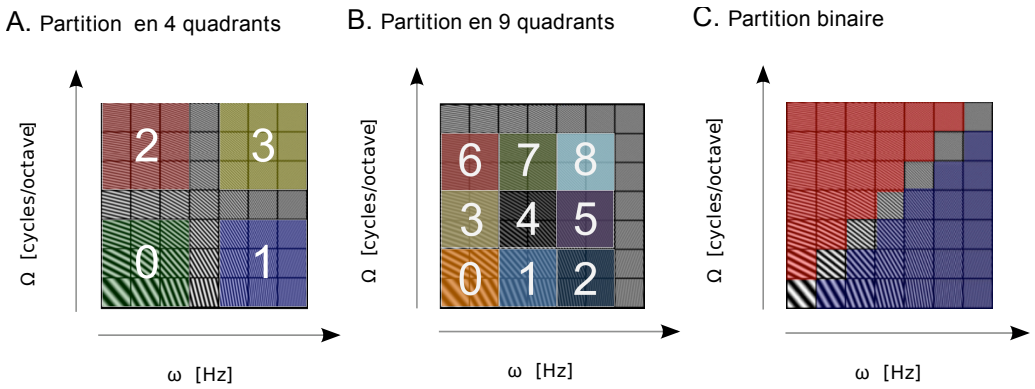


FIGURE 4.3: Partitions de l'espace des ondulations dynamiques. A. Partition en 4 quadrants. On obtient ainsi 4 cibles à distinguer B. Partition en 9 quadrants (9 cibles). C. Partition binaire

l'hypothèse d'encodage postulée. Quant à nous, nous tentons une analyse sous un angle différent. Nous appliquons différents classifieurs linéaires standards, soient le modèle bayésien gaussien naïf (BGN) avec variances partagées, l'algorithme machine à vecteurs de support avec relaxation (cMVS) et régression logistique (RL). Nous utilisons la réponse fMRI à travers les différents voxels pour prédire la condition de stimulation dans laquelle se trouvait le sujet. En repérant les ensembles de voxels associés aux meilleures performances de prédiction, nous aspirons à localiser la/les région(s) d'encodage dans le cerveau humain.

À des fins de pragmatisme, nous posons certaines limites. D'abord, il serait peu réaliste de chercher à prédire directement le stimulus perçu, puisque nous ne disposons que de 10 images fMRI par stimulus. Plutôt, nous considérons des partitions en 4 et 9 quadrants de l'espace des ondulations dynamiques (Figure 4.3AB). Ces dernières donnent lieu à des tâches de classification à 4 et 9 cibles, respectivement. Un autre problème potentiel est la variabilité d'un sujet à l'autre, au niveau des préférences des voxels et de la position de la présumée région d'encodage. Afin d'éviter les complications résultantes et de simplifier l'analyse des résultats, nous débutons avec les données relatives à un seul sujet.

Nous sommes conscients de la nouveauté de la méthodologie que nous empruntons. L'application d'algorithmes d'apprentissage aux données fMRI, bien que de plus en plus populaire et reconnue, n'est toujours pas un standard. C'est pourquoi nous prennons soin d'étudier le comportement des algorithmes sur nos données et, s'il y a lieu, de dégager des améliorations possibles. En fait, le présent projet est aussi pour nour une opportunité de dégager des avenues de recherche concernant le développement de nouveaux algorithmes spécialisés pour les données fMRI. Pour ces test, nous nous servons d'une autre partition des stimuli. Elle est illustrée à la Figure 4.3C ci-dessous. Contrairement aux précédentes, elle est très large et ne comporte que deux catégories. Comme nous l'expliquerons à la section 4.3.1, nous avons confiance en la présence de patterns discriminants clairs selon cette partition, ce qui en fait un excellent problème pour tester les algorithmes.

4.2 Acquisition et prétraitement des données

Nous décrivons ici les propriétés des données fMRI dont nous disposons. Nous donnons un aperçu de la séance de scannographie et résumons les principales caractéristiques des images fMRI brutes obtenues. Comme expliqué à la section 2.2, bon nombre d'étapes de prétraitement sont nécessaires

avant de procéder à une analyse. Nous spécifions les choix qui ont été faits à chacune de ces étapes, dans l'ordre.

4.2.1 Images brutes

Tel que discuté à la section 2.1, le déroulement d'une séance de scannographie doit être soigneusement étudié. Les résolutions et dimensions des images sont des choix très importants, de même que le timing entre les moments de stimulation, les scans, et les moments de repos.

En ce qui concerne le timing, lorsqu'on utilise des stimuli auditifs, des contraintes supplémentaires s'imposent du fait que, lors de l'acquisition des images, le scanner émet un bruit important. Si on ne prend pas garde, la réponse du sujet à ce bruit peut potentiellement interférer avec la réponse au stimulus. La solution préconisée, en général, est l'acquisition très rapide des images. La réponse hémodynamique au bruit met un certain délais avant de se manifester (Figure 4.4A). Si le temps d'acquisition des images est plus court que ce délais, on évite le chevauchement avec la réponse au stimulus. Naturellement, un temps d'acquisition limité impose à son tour de nouvelles contraintes quant à la résolution des images.

La Figure 4.4B illustre le timing de scannographie choisi par Schönwiesner. Des spectrogrammes (carrés bariolés) marquent les moments et durées (4s) des stimulations avec différentes ondulations dynamiques. Les bandes verticales grises numérotées marquent les moments et durées (1s) des scans. La courbe au trait discontinu (- - -) indique la réponse hémodynamique au bruit du scanner. On peut constater que le temps d'acquisition des images est suffisamment court pour que celle-ci demeure faible du début à la fin du scan. La courbe au trait continu représente quant à elle la réponse hémodynamique au stimulus. On constate que les scans ont lieu suite à chaque stimulation, alors que la réponse au stimulus est très forte, ou suite à une période de repos, lorsque les réponses au stimulus et au bruit émis lors du scan précédent sont très faibles.

Chaque image fMRI 3d est composée de 13 images 2d parallèles, superposées verticalement. Celles-ci sont de dimensions 128×128 et sont scannées séquentiellement. Le plan d'acquisition horizontal de chacune des images est légèrement incliné, de manière à épouser l'orientation naturelle des lobes temporaux, tel qu'illustré à la Figure 4.5. Les 13 images 2d englobent tout juste la partie supérieure des lobes temporaux. La résolution verticale, i.e. d'une image 2d à l'autre, est de 2.5 mm. La résolution horizontale est de 1.5 mm $\times$ 1.5 mm. Il s'agit de très fines résolutions, la résolution standard se situe plutôt aux alentours de 4 mm.

Les réponses aux 49 ondulations dynamiques ont été acquises à travers 4 séances de scannographie. En combinant les données des 4 séances, on obtient un total de 10 exemples pour chaque ondulation et, donc, un grand total de 490 exemples.

4.2.2 Réalignement et lissage

Nous avons procédé à un réalignement spatial des images pour compenser pour le mouvement du sujet d'un scan à l'autre. Nous avons ensuite appliqué un léger lissage spatial, à l'aide d'un noyau gaussien de 2 mm de rayon. La perte d'information avec un tel rayon devrait être minime, puisque le réalignement spatial introduit déjà des corrélations spatiales de même envergure. Par précaution, nous

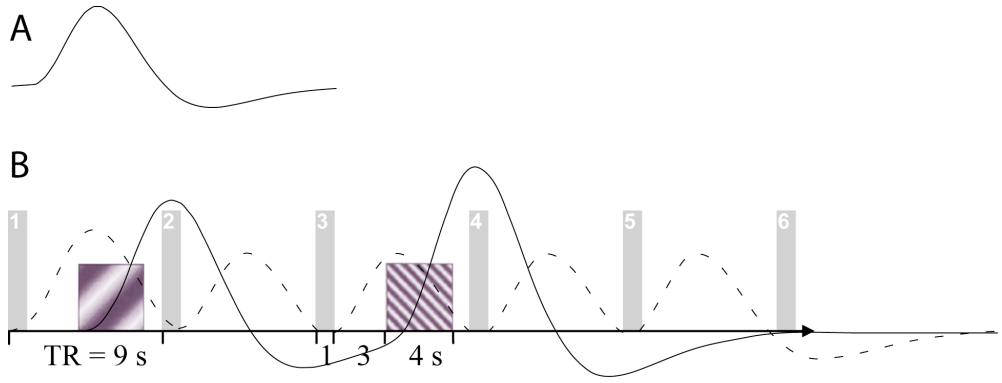
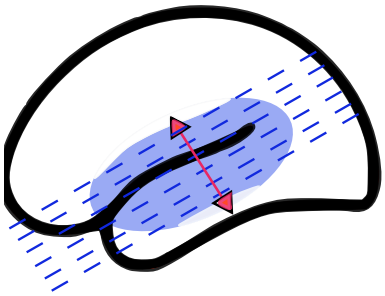


FIGURE 4.4: Déroulement de la séance de scannographie. A. Allure typique de la réponse hémodynamique suite à une courte stimulation. Le maximum d'amplitude est atteint quelques secondes après la stimulation. B. Les spectrogrammes marquent les moments où les stimuli sont présentés. Leur largeur représente le temps de stimulation, soit 4s. Les bandes verticales grises numérotées marquent quant à elles les moments où les scans sont effectués, soit à des intervalles de 9s. La largeur des bandes, toujours la même, indique le temps d'acquisition, soit 1s. La courbe au trait discontinu (- - -) représente la réponse hémodynamique au bruit du scanner. Le temps d'acquisition des images est suffisamment court pour que la réponse au bruit demeure très faible du début à la fin de l'acquisition. La courbe au trait continu représente la réponse hémodynamique au stimulus. Après une stimulation, le scan a lieu lorsque la réponse au stimulus est très forte. Après une période de repos, le scan a lieu lorsque les réponses au stimulus et au bruit émis lors du scan précédent sont très faibles.



s

FIGURE 4.5: Scan des lobes temporaux. La partie supérieure du lobe temporal gauche est colorée en bleu pâle. L'image 3d finale est composée de 13 images 128×128 superposées les unes au dessus des autres. Elles sont scannées séquentiellements de haut en bas, le long de la direction indiquée par la flèche bidirectionnelle rouge. Les traits bleus indiquent les plans d'acquisitions de quelques images 2d. Ceux-ci sont parallèles et légèrement inclinés, de manière à épouser l'orientation naturelle des lobes temporaux.

avons aussi effectué quelques tests avec un rayon de lissage plus petit, soit 1 mm. Néanmoins, à moins d'indications contraire, le lecteur peut assumer un rayon de lissage de 2 mm.

4.2.3 Masque

Nous avons sélectionné un total de 6342 voxels dans les lobes temporaux. Plus précisément, ceux-ci sont répartis à travers les tranches verticales 3 à 11 (de haut en bas). Ce masque a été construit en pré-sélectionnant les voxels les plus actifs et en appliquant des dilations et érosions successives. Pour évaluer le degré d'activitation des voxels, nous nous sommes basés sur des statistiques de Student calculées au préalable par Schönwiesner [14]. Celles-ci reflètent les contrastes, en moyenne, entre les

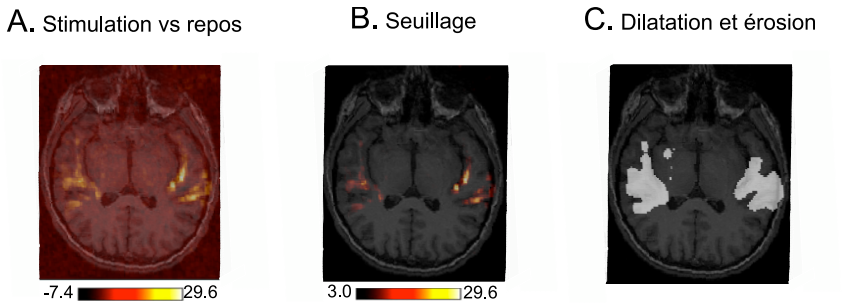


FIGURE 4.6: Construction du masque pour la tranche no 6. A. Statistiques de Student, condition de stimulation (tous stimuli confondus) vs condition de repos. Le code de couleur est indiqué au bas de l'image. Une statistique de Student élevée indique une forte activation du voxel correspondant par rapport à la condition de repos. B. Premier masque obtenu par un seuillage sur les statistiques de Student en A. Seuls les voxels des lobes temporaux sont considérés. Le seuil choisi pour la statistique de Student est 3.0. C. Masque final obtenu suite à des dilatations et érosions successives à partir du masque en B.

scans avec stimulation et les scans sans stimulation (Figure 4.6A). Nous avons appliqué un processus de seuillage sur ces statistiques de Student, afin de filtrer seulement les voxels les plus "actifs" par rapport à la condition de repos (Figure 4.6B). Nous avons ensuite appliqué des dilatations et érosions successives, afin de lisser et gonfler les régions pré-sélectionnées (Figure 4.6C). Selon les connaissances à priori disponibles quant à la localisation de la région d'encodage, il est raisonnable de supposer qu'elle soit située à proximité d'une région fortement activée par rapport à la condition de repos.

4.2.4 Detrending et normalisation

Les réponses fMRI brutes, en un voxel donné, sont généralement biaisées par la présence d'un trend lent et lisse qui n'est pas lié aux différents stimuli (Figure 4.7). Ces trends sont dûs en majeure partie au réchauffement du scanner et à la réponse physiologique du sujet. Il est impératif qu'ils soient soustrait des trajectoires temporelles des voxels avant l'analyse. En ce qui nous concerne, la qualité du detrending peut avoir beaucoup d'impact. En effet, si les préférences des voxels aux différentes ondulations dynamiques sont subtiles et bruitées, un detrending grossier risque de les masquer.

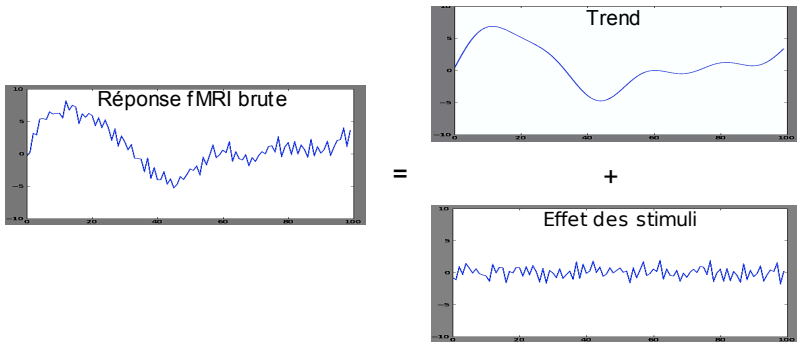


FIGURE 4.7: Allure typique de la réponse fMRI brute en un voxel. L'axe horizontal représente le temps, en nombre de scans. La réponse fMRI brute est biaisée par la présence d'un trend lent et lisse non lié à l'effet des stimuli. Lors du detrending, on cherche à estimer et supprimer ce trend.

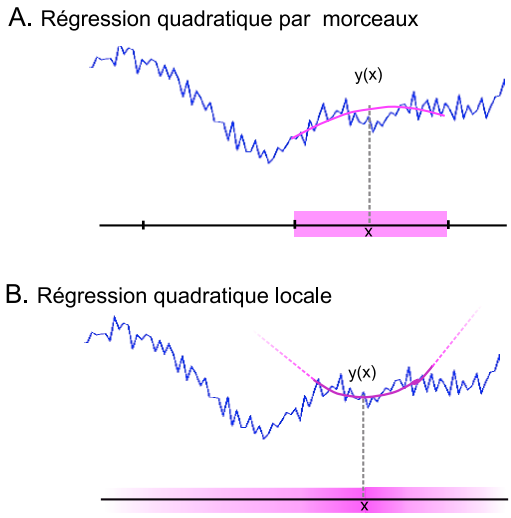


FIGURE 4.8: Différences entre les régressions quadratiques "par morceaux" et "locale". La point x correspond au point de test, le point $y(x)$ à la prédiction pour le point de test. Le degré d'influence des exemples pour la prédiction est indiqué sur l'axe du temporel. A. Régression quadratique par morceaux. L'axe temporel est découpé en segments de longueur fixe. Pour un point de test x donné, seulement les exemples appartenant au même segment dans lequel se trouve x influencent la prédiction. B. Régression quadratique locale. L'influence des exemples est déterminée selon leur promixité au point de test.

Nous avons opté, au final, pour une régression dite **quadratique locale** pour estimer la forme des trends en chacun des voxels. Avant d'arrêter notre choix sur cette technique, nous avons testé d'autres formes de régression plus communes, notamment des régressions quadratique et linéaire, des splines cubiques et des filtres passe-bas. La régression quadratique locale, à l'oeil nu, épousait mieux la forme des trends que les autres techniques.

La régression quadratique locale est analogue à une autre forme de régression quadratique plus simple dite **par morceaux**. Cette dernière (Figure 4.8A) consiste à diviser la trajectoire en segments de longueur fixe sur l'axe temporel et à effectuer une régression quadratique sur chacun de ces segments. Ainsi, pour un point de test donné, seulement les exemples appartenant au même segment que le point de test, sur l'axe temporel, influencent la prédiction. La régression quadratique locale, tout comme la régression quadratique par morceaux, accorde davantage d'importance aux exemples situés dans le voisinage du point de test. Cependant, plutôt que de s'en remettre à l'appartenance ou non au même segment, on pondère l'influence des exemples selon leur proximité au point de test (Figure 4.8B).

Ci-dessous la formulation mathématique de la régression quadratique locale :

Régression quadratique locale

Soient :
- *x un point de test sur l'axe temporel*
- *P^2 l'ensemble des fonctions quadratiques*
- *$(x_n, y_n)_{n=1}^N$ un ensemble d'exemples $\in \mathbb{R}^2$*

On cherche d'abord une fonction $y^* \in P^2$ telle que :

$$y^* = argmin_{\hat{y} \in P^2} \sum_n w_n (y_n - \hat{y}(x_n))^2$$

Donné $0 \leq \alpha \leq 1$, les poids w_n sont définis comme suit :

$$w_n = 0 \text{ si } x_n \notin N\alpha \text{ exemples les plus proches de } x$$
$$\propto \left(1 - \left(\frac{distance(x_n, x)}{maxdistance(x)}\right)^3\right)^3 \text{ sinon}$$

La valeur du trend en x est ensuite approximée par $y^*(x)$.

Le paramètre α détermine la proportion d'exemples avec un poid non-nul. Plus α est petit, plus on se concentre sur les plus proches voisins pour effectuer la régression, plus l'estimé est "local", i.e. plus il épouse la forme du trend autour du point de test. La Figure 4.9 montre quelques trends estimés sur nos données en un voxel particulier, pour différentes valeurs du paramètre α. En $\alpha = 0.1$, le trend estimé semble absorber de petites oscillations potentiellement importantes. Il devient de plus en plus lisse au fur et à mesure qu'on augmente la valeur de α. Les trends estimés semblent robustes pour des valeurs de α entre 0.3 et 0.5.

Nous avons appliqué la régression quadratique locale sur chacun des voxels séparément, sur chacune des 4 trajectoires temporelles (200 scans chacune) correspondant aux 4 séances de stimulation. Nous avons utilisé l'implémentation R "loess". Le paramètre α choisi est le même pour tous les voxels, soit 0.4. Nous nous en sommes remis à un examen visuel sur les trajectoires de 100 voxels choisis au hasard pour juger de la qualité de l'estimé.

Suite à la procédure de detrending, en guise d'étape de prétraitement finale, nous avons normalisé les données voxel par voxel. Il est bien connu que les classifieurs linéaires cMVS et RL se comportent mieux sur données normalisées. De plus, afin de mieux réfléter l'importance relative des voxels, il est souhaitable que les poids appris par les classifieurs soient invariants à l'amplitude globale des réponses des voxels.

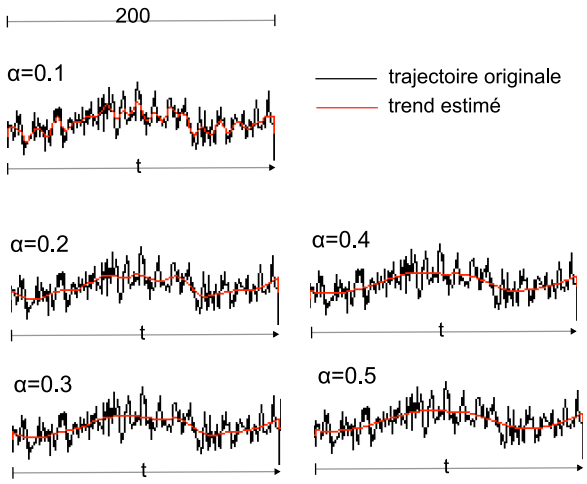


FIGURE 4.9: Régression quadratique locale pour une série de 200 scans. La réponse fMRI brute est tracée en noir. Le trend estimé par régression quadratique locale est tracé en rouge. Nous varions le paramètre de localité α. Plus ce dernier est petit, plus l'estimé est local. En $\alpha = 0.1$, le trend estimé semble absorber de petites oscillations potentiellement importantes. Il devient de plus en plus lisse au fur et à mesure qu'on augmente la valeur de α. Les trends estimés semblent robustes pour des valeurs de α entre 0.3 et 0.5.

4.3 Description des expériences

Ici, nous décrivons trois expériences réalisées dans le cadre de ce projet.

Les deux premières ont pour but d'évaluer la pertinence de différents algorithmes en tant qu'outils d'analyse. Nous considérons un problème de classification binaire. À l'expérience 1, nous appliquons des classifieurs linéaires de base. À l'expérience 2, nous nous penchons sur un algorithme de recherche locale.

La troisième expérience a quant à elle pour but de localiser la ou les régions d'encodage potentielles. Nous utilisons pour cette expérience des partitions plus fines de l'espace des ondulations dynamiques, soit des partitions en 4 et 9 quadrants.

4.3.1 Différentes partitions des stimuli

Dépendamment des buts des expériences, nous utilisons différentes partitions des stimuli, i.e. des ondulations dynamiques.

La partition "ST", illustrée à la Figure 4.10, découpe l'espace des ondulations dynamiques en deux ensembles disjoints S et T. L'ensemble S regroupe les modulations pour lesquelles la modulation spectrale Ω est plus élevée que la modulation temporelle ω, i.e. les sons plus complexes spectralement que temporellement. L'ensemble T regroupe au contraire les modulations pour lesquelles la modulation temporelle est plus élevée que la modulation spectrale, i.e. les sons plus complexes temporellement que spectralement.

Il s'agit d'un problème de classification approprié pour étudier et comparer les algorithmes. En fait, l'existence de structures neuronales spécialisées pour les sons temporellement complexes et spectrale-

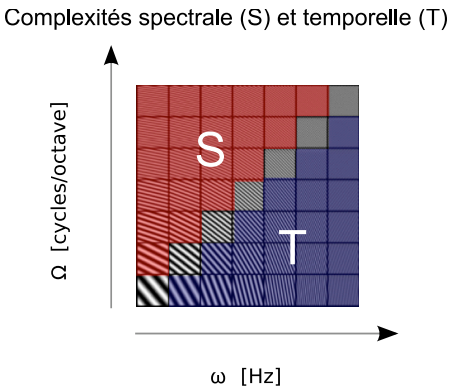


FIGURE 4.10: Partition des ondulations dynamiques selon leur complexité spectrale (S) ou temporelle (T). L'espace des ondulations dynamiques est défini par deux axes. L'axe horizontal représente la modulation spectrale Ω appliquée. L'axe vertical représente la modulation temporelle ω appliquée. L'ensemble S, coloré en rouge, regroupe les ondulations pour lesquelles la modulation spectrale est plus élevée que la modulation temporelle. L'ensemble T, coloré en bleu, regroupe les ondulations pour lesquelles la modulation temporelle est plus élevée que la modulation spectrale.

ment complexes est bien documentée (Schönwiesner (2005) [13], Zatorre-Belin (2001) [24]). Nous avons donc confiance en la présence de différences significatives entre ces deux conditions de stimulation.

Nous considérons aussi trois autres partitions similaires à la partition ST, soit les partitions ST_1, ST_2 et ST_3. Ces dernières sont construites à partir de la partition ST en excluant les ondulations dynamiques situées près de la frontière séparant les deux ensembles, i.e. près de la diagonale. Les différentes partitions sont illustrés à la Figure 4.11. Nous élargissant la marge entre les ensembles de 1 (ST_1), 2 (ST_2) et 3 (ST_3). Nous nous attendons ainsi à accentuer de plus en plus les différences entre les deux conditions S et T.

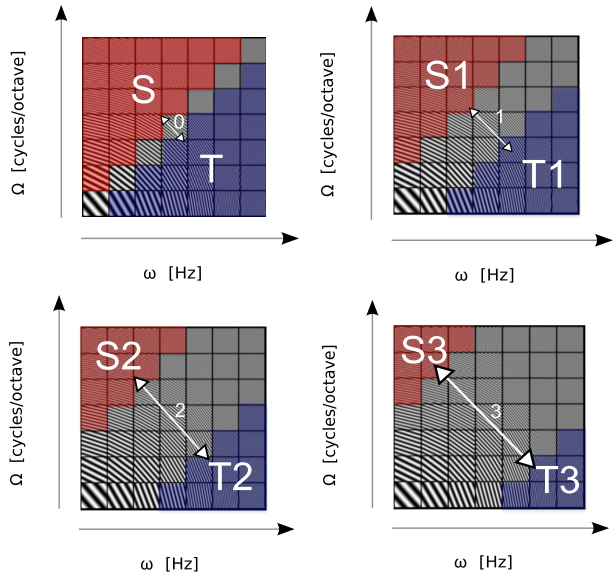


FIGURE 4.11: Exagération des différences entre les complexités spectrale et temporelle. Les ondulations dynamiques situées près de la frontière sont exclues, ce qui élargit la marge entre les deux ensembles originaux S et T. En élargissant la marge d'exclusion à 1, 2 et 3, on forme respectivement les partitions ST_1, ST_2 et ST_3.

Si les partitions ST, ST_1, ST_2 et ST_3 conviennent bien à l'analyse du comportement des différents algorithmes, elles sont cependant trop larges pour pouvoir identifier la région d'encodage avec précision. Afin de mieux repérer cette dernière, nous utilisons des partitions plus fines, en 4 et 9 quadrants. Ces dernières sont illustrées à la Figure 4.12.

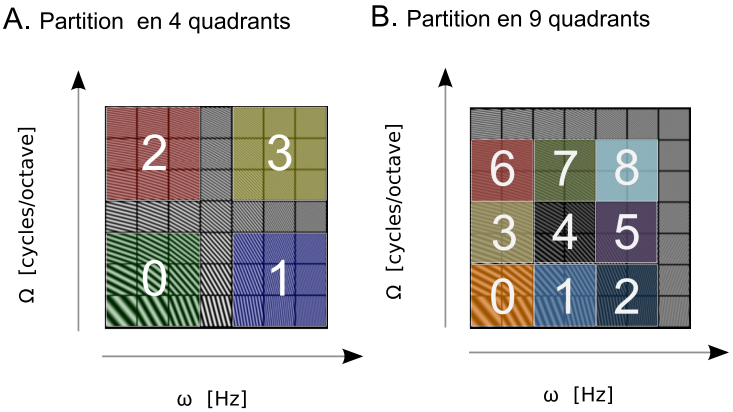


FIGURE 4.12: L'espace des ondulations dynamiques est découpé en sous-espaces plus raffinés pour les expériences ayant pour but d'identifier la région d'encodage. A. Partition en 4 quadrants B. Partition en 9 quadrants.

4.3.2 Expérience 1 : comparaison de classifieurs linéaires

L'objectif de cette expérience est d'entrevoir le potentiel de différents classifieurs linéaires en tant qu'outils d'analyse sur nos données. Nous considérons les problèmes de classification définis par les partitions binaires ST, ST_1, ST_2 et ST_3 (Figures 4.10 et 4.11). Nous comparons 3 classifieurs linéaires standards, soit le modèle BGN avec variances partagées, l'algorithme cMVS et l'algorithme RL. Nous tentons aussi quelques essais avec les régularisation $L1$, $L2$ et FE sur l'algorithme RL. Comme expliqué aux section 3.2.2 et 3.3.1, la régularisation FE consiste en un compromis entre les régularisation L1 et L2. Ce compromis mènerait à une meilleure interprétabilité des paramètres en favorisant à la fois la parcimonie et l'intégration des corrélations spatiales.

4.3.3 Expérience 2 : recherche locale

Dans cette expérience, nous cherchons à mieux cerner les avantages et inconvénients de la recherche locale. Par opposition à une recherche globale, une recherche locale consiste à analyser les voxels par petits groupes plutôt qu'en un seul bloc.

Nous appliquons l'algorithme de recherche locale défini à la section 3.6 sur le problème de classification défini par la partition ST (Figure 4.10). Plus précisément, nous centrons une sphère en chacun des 6342 voxels pré-sélectionnés et construisons des classifieurs linéaires (nuMVS) à l'aide des voxels contenus dans chacune des sphères. Nous produisons une carte de performance indiquant les performances de prédiction de chacun des classifieurs.

Nous tâchons d'abord de trouver un compromis raisonnable pour le rayon de la sphère de recherche et le rayon de lissage. Une sphère de recherche trop petite empêche l'algorithme d'apprentissage d'exploiter les patterns plus distribués. En revanche, une sphère de recherche trop grande fait perdre de la

précision et rend les classifieurs plus fragiles au sur-apprentissage de par l'augmentation de la dimensionalité du prédicteur. Le lissage, quant à lui, peut aider à éliminer le bruit, mais peut aussi effacer de l'information et introduire des corrélations spatiales qui n'étaient pas présentes initialement.

Une fois ces paramètres choisis, nous construisons une carte de performance et procédons à des tests de significativité pour détecter les régions d'intérêt. Nous en profitons pour comparer deux techniques pour modéliser l'hypothèse nulle, soit l'approximation binomiale et une procédure de randomisation. Comme expliqué à la section 3.7, l'approximation binomiale revient à supposer que les classifieurs, sous l'hypothèse nulle, prédisent chaque cible aléatoirement, indépendamment du prédicteur. La procédure de randomisation, quant à elle, consiste à estimer la forme de la distribution des performances sous l'hypothèse nulle en collectant les performances de classifieurs entraînés à l'aide de cibles aléatoires.

Pour terminer, nous comparons qualitativement les régions d'intérêts finales obtenues avec les poids du classifieur linéaire régression logistique régularisé avec la norme $L1$. Comme expliqué à la section 3.6, la régularisation $L1$ est une alternative intéressante à la recherche locale pour détecter les régions d'intérêt.

4.3.4 Expérience 3 : localiser la région d'encodage

Nous considérons cette fois des partitions plus raffinées, soient les partitions en 4 et 9 quadrants décrites à la Figure 4.12. Nous utilisons une fois de plus trois classifieurs linéaires standards, soient le modèle BGN avec variances partagées, l'algorithme cMVS et l'algorithme RL. Notre principal but est cette fois de repérer la(s) région(s) d'encodage potentielle(s). Nous nous attendons à une influence de plus en plus prépondérant de celle(s)-ci au fur et à mesure qu'on raffine la partition de l'espace des ondulations dynamiques.

Nous visualisons les poids appris d'une partition à l'autre à l'aide des mesures de sensibilité décrites à la section 3.5.4.

Chapitre 5

Résultats et analyse

Nous présentons ici les résultats des différentes expériences.

Nous effectuons quelques essais à l'aide de différents classifieurs linéaires à l'expérience 1. Nous tâchons de mieux cerner leur potentiel pour l'analyse fMRI et, s'il y a lieu, de dégager des améliorations possibles en ce qui concerne les techniques de régularisation.

À l'expérience 2, nous appliquons un algorithme de recherche locale. Nous prenons soin de comparer la carte de performance obtenue avec les poids appris par les classifieurs régularisés avec norme L1 de l'expérience 1.

Ces deux expériences sont réalisées sur un problème de classification binaire, défini par la partition ST introduite précédemment. Comme mentionné à la section 4.3.1, nous avons confiance en la présence d'information selon cette partition.

Finalement, à l'expérience 3, nous tâchons de repérer des candidats potentiels pour la région d'encodage des ondulations dynamiques. Nous utilisons cette fois des partitions en 4 et 9 quadrants de l'espace des ondulations dynamiques.

5.1 Expérience 1 : classifieurs linéaires

L'objectif de cette expérience est d'entrevoir le potentiel de différents classifieurs linéaires en tant qu'outils d'analyse de nos données. Nous considérons les problèmes de classification binaires définis par les partitions ST, ST_1, ST_2 et ST_3. Nous comparons trois principaux algorithmes, soient le modèle bayésien gaussien naïf (BGN) avec variances partagées, l'algorithme régression logistique (RL) et l'algorithme machine à vecteurs de support avec relaxation (cMVS). Nous tentons aussi quelques essais avec les régularisations $L1$, $L2$ et FE pour l'algorithme régression logistique (RL).

5.1.1 Détails du protocole expérimental

Les deux tiers des données sont réservées pour l'entraînement des algorithmes et le dernier tiers pour tester les classifieurs construits. Nous utilisons une validation croisée en 5 blocs pour sélectionner les hyper-paramètres, s'il y a lieu.

Nous utilisons l'implémentation libsvm [26] de l'algorithme cMVS. Les différentes variantes de l'algorithme RL ($L1$, $L2$, FE) sont basées sur la classe NNet de la librairie PLearn [27]. L'optimisation est réalisée par descente de gradient conjuguée. Voir Shewchuk (1994) [20] pour des explications concernant cette procédure d'optimisation.

Nous manipulons ces implémentations via la librairie python pyMVPA [28]. Celle-ci supporte, notamment, la procédure de validation croisée nécessaire à la sélection des hyper-paramètre. Nous avons,

au préalable, annexé l'implémentation PLearn de l'algorithme RL à cette librairie. L'implémentation cMVS libsvm est déjà annexée par défaut.

5.1.2 Premiers essais sur *ST*

Dans un premier temps, nous nous assurons de la faisabilité de l'analyse et en dégageons les bénéfices par rapport à l'analyse traditionnelle. Nous comparons la performance de prédiction et les poids appris par les trois principaux algorithmes sur le problème de classification ST. Nous insistons particulièrement sur le modèle BGN avec variances partagées qui est un point de référence important à la méthode d'analyse traditionnelle.

Les performances de prédiction des classifieurs construits par les différents algorithmes sont indiquées à la Table 5.1. Celles-ci confirment la présence de différences significatives d'une condition à l'autre. En effet, la performance de prédiction du modèle BGN avec variances partagées, soit la plus faible, est de 70%, ce qui est nettement plus élevé que le hasard, soit 50%.

Algorithmes	Performance	Hyper-paramètres
BGN	70%	-
cMVS	81%	$c = 0.01$
RL (Arrêt prématuré)	79%	-

TABLE 5.1: Performances de prédiction des algorithmes sur la partition ST et valeurs des hyper-paramètres choisis, s'il y a lieu. Dans le cas de l'algorithme RL, une procédure d'arrêt prématurée a été appliquée afin d'éviter le phénomène de divergence (voir section 3.5.2). La performance de prédiction rapportée est le pourcentage d'exemples de test bien classifiés. Dans tous les cas, la performance est beaucoup plus élevée que celle d'un classifieur aléatoire (50%).

La performance relativement élevée du modèle BGN avec variances partagées suggère que les statistiques de Student traditionnellement utilisées ne sont pas complètement non-pertinentes lorsqu'on les considère dans leur ensemble. En effet, on se rappelle que les poids appris par le modèle BGN avec variances partagées sont très similaires aux statistiques de Student (voir section 3.5.1).

Nous n'arrivons pas du tout aux mêmes conclusions avec une analyse univariée traditionnelle. Pour tous les voxels, en usant de tous les exemples, nous avons calculé les statistiques de Student, de même que les probabilités de l'hypothèse nulle correspondantes. Les histogrammes présentés à la Figure 5.1 permettent de visualiser la distribution des valeurs obtenues. La plus faible probabilité de l'hypothèse nulle observée est de 0.9971, ce qui signifie qu'on peut rejeter l'hypothèse nulle avec une confiance maximale de moins de 1%. Ainsi, chacune des statistiques de Student, isolément, est très peu significative ; il est essentiel de les analyser de manière multivariée pour en tirer de l'information pertinente.

Les poids appris par le modéle BGN avec variances partagées semblent relativement efficaces, mais ne semblent pas exploiter les différences entre les conditions S et T aussi bien que les poids appris par les algorithmes cMVS et RL. En effet, il y a un écart de 10% entre la performance du modèle BGN avec variances partagées et les performances des algorithmes cMVS et RL. Plus versatiles, ces derniers ne tiennent pas uniquement compte du contraste entre les moyennes empiriques des réponses pour

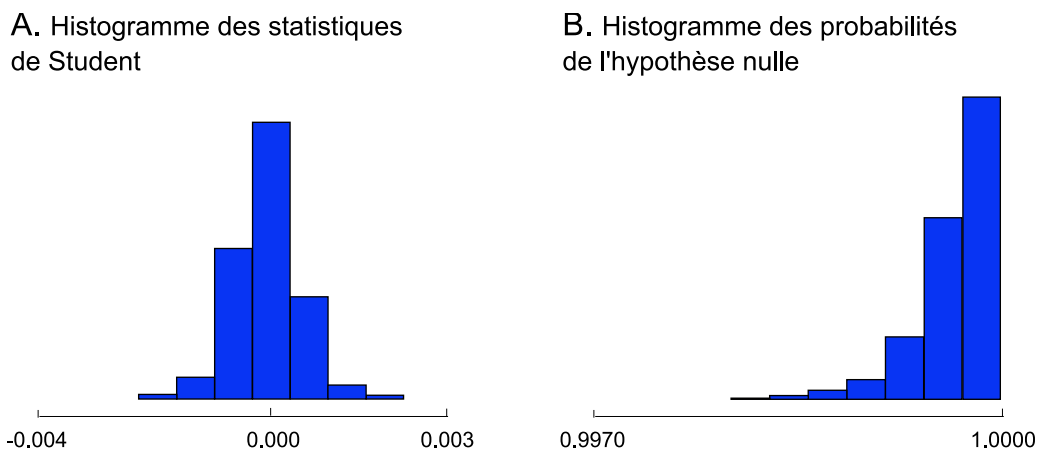


FIGURE 5.1: A. Histogramme des statistiques de Student sur la partition ST, calculées en utilisant la totalité des données. B. Histogramme des probabilités de l'hypothèse nulle correspondantes

déterminer l'influence des voxels. Par exemple, ils peuvent aussi prendre en considération la stabilité des corrélations entre les réponses d'un groupe de voxels. Ici, il semble que leur agnosticité quant à la forme de l'information dans les données leur confère un avantage assez net.

Malgré les différences de performance assez marquées, les poids appris par les trois algorithmes (Figure 5.2) sont cohérents entre eux. La correspondance entre les poids des algorithmes cMVS et RL semble particulièrement forte. La correspondance au modèle BGN avec variances partagées ne semble pas aussi exacte, cependant on note de fortes ressemblances. En fait, il semble que les patterns exploités par les algorithmes cMVS et RL englobent ceux exploités par le modèle BGN avec variances partagées. En effet, hormis l'amplification de quelques contrastes ça et là, leurs poids appris sont très similaires.

Afin de décrire plus rigoureusement le degré de similarité entre les poids appris par les différents algorithmes, nous avons calculé les coefficients de corrélation, i.e. les produits scalaires entre les vecteurs de poids normalisés (Figure 5.3). Les poids appris par RL et cMVS sont corrélés à un degré de 0.90. Cette forte corrélation n'est pas étonnante étant donné la similarité de leurs stratégies d'optimisation ; tous deux sont discriminants et cherchent à éviter les décisions ambigües. Les corrélations entre ces deux algorithmes et le modèle BGN avec variances partagées sont aussi assez fortes (0.81 et 0.64), particulièrement en ce qui concerne l'algorithme RL. La plus forte corrélation à l'algorithme RL est intéressante, considérant que l'algorithme RL est précisément l'équivalent discriminant du modèle BGN avec variances partagées.

5.1.3 Essais sur les partitions ST_1, ST_2 et ST_3

Nous avons discuté, à la section 3.3.2, des bénéfices potentiels d'un classifieur bayésien par rapport à son équivalent discriminant. Lorsqu'on dispose de peu de données et que les hypothèses émises par le modèle bayésien sont réalistes, alors ce dernier est généralement moins sujet au sur-apprentissage et performe mieux. En ce qui nous concerne, cependant, c'est le contraire qui se produit. Sur la partition ST, on observe un net avantage de performance en faveur de l'algorithme RL, soit l'équivalent discriminant. Ce résultat suggère que les hypothèses sur lesquelles se base le modèle BGN avec variances

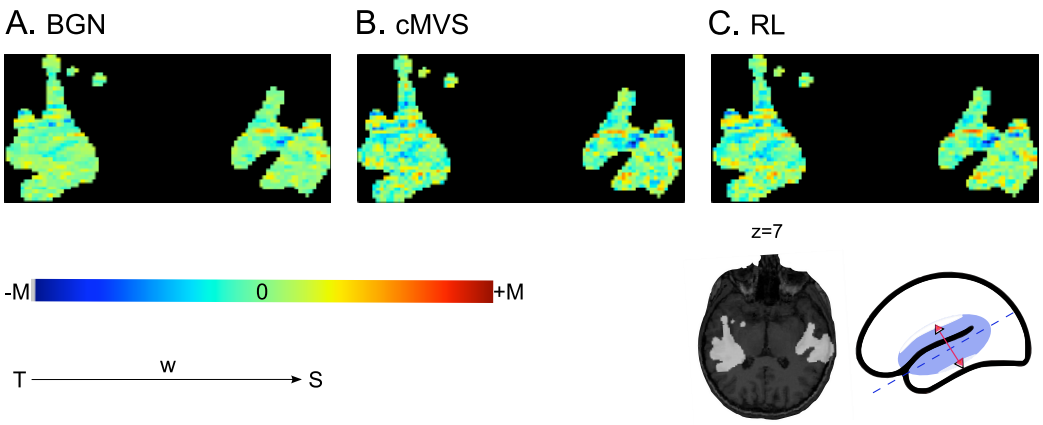


FIGURE 5.2: Poids appris par le modèle BGN avec variances partagées (en A), l'algorithme cMVS (en B) et l'algorithme RL (en C), sur la partition ST. Nous montrons seulement les poids relatifs à une tranche de voxel ($z = 7$). En bas, à droite, on aperçoit l'orientation de cette tranche de voxels 2d dans l'espace, de même que les voxels pré-sélectionnés pour l'analyse (colorés en blanc) à l'intérieur. L'échelle de couleur utilisée est indiquée en bas à gauche. Les poids des différents algorithmes ont été ramenés sur une échelle entre -1 et 1. Plus précisément, soit M l'amplitude maximale des poids d'un algorithme, le bleu représente la valeur $-M$ et le rouge la valeur $+M$. Le vert est associé à la valeur 0. Les poids indiquent la direction des changements de la condition T vers la condition S. Ainsi, un poids négatif/positif indique une inhibition/excitation des exemples de la condition S par rapport aux exemples de la conditions T au voxel correspondant. La cohérence d'un algorithme à l'autre est rassurante quant à la pertinence des poids appris par les algorithmes.

Partitions	Entraînement	Test
ST	280	140
ST_1	207	93
ST_2	133	57
ST_3	80	40

TABLE 5.2: Description des ensembles de données relatifs aux partitions ST, ST_1, ST_2 et ST_3. L'ensemble d'entraînement est composé des 2/3 des exemples et l'ensemble de test du 1/3. D'une partition à l'autre, la taille des ensembles diminue.

partagées sont simplistes ou erronnées. Afin d'éclaircir cette question davantage, nous comparons aussi les algorithmes sur les partitions ST_1, ST_2 et ST_3.

Celles-ci correspondent à des conditions d'apprentissage différentes. La Figure 5.4 montre la distribution des exemples mal classifiés dans l'espace des ondulations dynamique, pour le problème de classification ST. Les trois classifieurs semblent commettre beaucoup d'erreurs pour les ondulations dynamiques situées près de la frontière (en noir) entre les conditions S et T . En créant les partitions ST_1, ST_2 et ST_3, on retire progressivement ces exemples plus difficiles à classifier. Évidemment, puisqu'on exclut de plus en plus d'exemples, le nombre d'exemples diminue d'une partition à l'autre. La Table 5.2 indique les tailles des ensembles d'entraînement et de test relatives à chacune des partitions.

La Table 5.3 indique les performances de prédiction des algorithmes pour les différentes partitions. Pour les algorithmes cMVS et RL, on observe une amélioration progressive. En revanche, le modèle

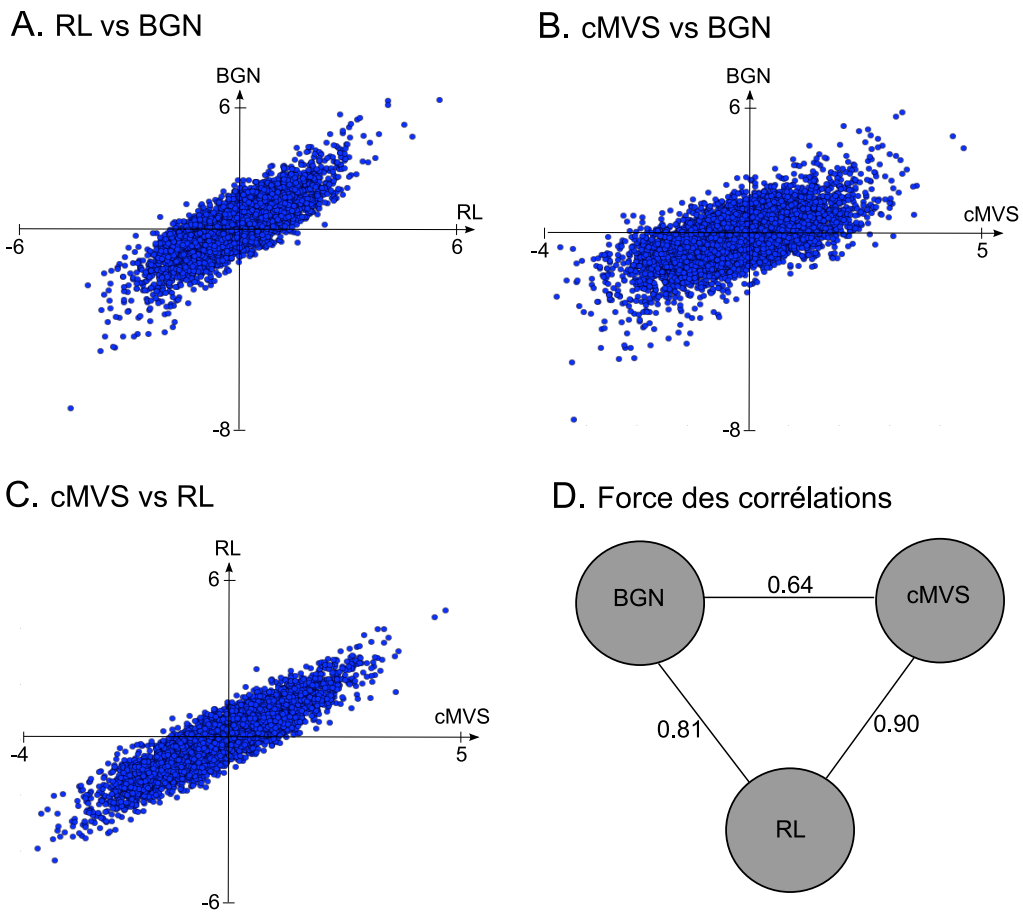


FIGURE 5.3: Cohérence des classifieurs. A.B.C Chaque point est associé à un voxel. Chaque axe indique la valeur du poids (normalisée) associé au voxel pour un classifieur en particulier. Les corrélations semblent particulièrement fortes, puisque, dans tous les cas, le nuage de points obtenu ressemble beaucoup à une ligne droite. D. Degrés de corrélation entre les poids des classifieurs. Les poids des modèles appris RL et cMVS sont particulièrement corrélés, ce qui réflète la similarité de ces algorithmes. Le modèle RL est beaucoup plus corrélé au modèle BGN avec variances partagées que le modèle cMVS, ce qui réflète le fait que l'algorithme RL est l'équivalent discriminant du modèle BGN avec variances partagées.

BGN avec variances partagées semble très instable. Dans tous les cas, cependant, les performances de prédiction sont suffisamment élevées pour indiquer la présence de différences significatives d'une condition à l'autre.

Le fait que, dans tous les cas, la performance du modèle BGN avec variances partagées soit nettement plus faible que celles des algorithmes cMVS et RL suggère d'autant plus fortement que les suppositions sur lesquelles se base le modèle BGN avec variances partagées sont incorrectes.

La Figure 5.5 montre les poids appris par les différents classifieurs, pour chacune des partitions. Les poids des modèles RL et cMVS semblent assez similaires d'une partition à l'autre, hormis une légère amplification des contrastes au niveau de la partition ST_3. En ce qui concerne le modèle BGN avec variances partagées, il semble, au contraire, que le contraste entre les poids diminue de façon marquée pour les partitions ST_2 et ST_3. Les signes des poids, cependant, demeurent cohérents d'une partition à l'autre, et cohérents avec ceux des poids des autres algorithmes.

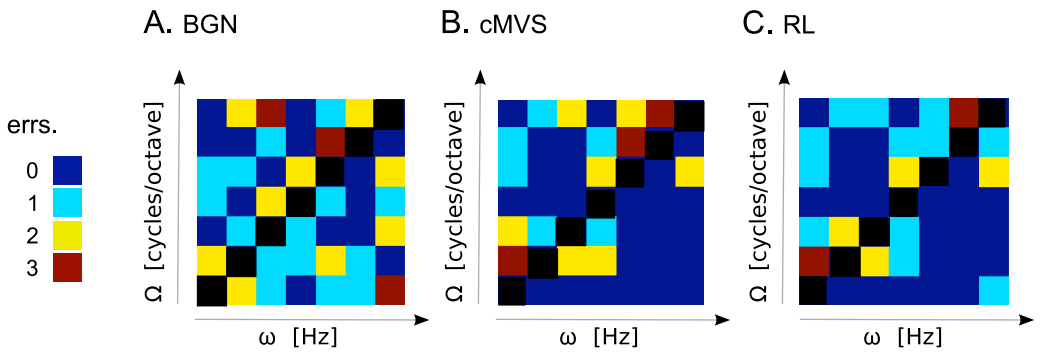


FIGURE 5.4: Distribution des exemples mal classifiés dans l'espace des ondulations dynamiques. A. cSVM B. RL, C. Modèle BGN avec variances partagées. Les ondulations dynamiques situées sur la frontière entre les conditions S et T sont colorées en noir. Les autres couleurs représentent le nombre d'erreurs, tel que défini par le code indiqué sur la gauche. Beaucoup d'erreurs sont commises près de la frontière.

Algorithmes	ST	ST_1	ST_2	ST_3
cMVS	81%	88%	90%	97%
RL (arrêt prématuré)	79%	86%	88%	92%
BGN	70%	78%	70%	80%

TABLE 5.3: Performances de prédiction des algorithmes sur les partitions ST, ST_1,ST_2 et ST_3. La performance de prédiction rapportée est le pourcentage d'exemples de test bien classifiés. La performance des algorithmes RL et cMVS s'améliore, alors que celle du modèle BGN avec variances partagées est instable. Dans tous les cas, le modèle BGN avec variances partagées performe moins bien que les algorithmes RL et cMVS.

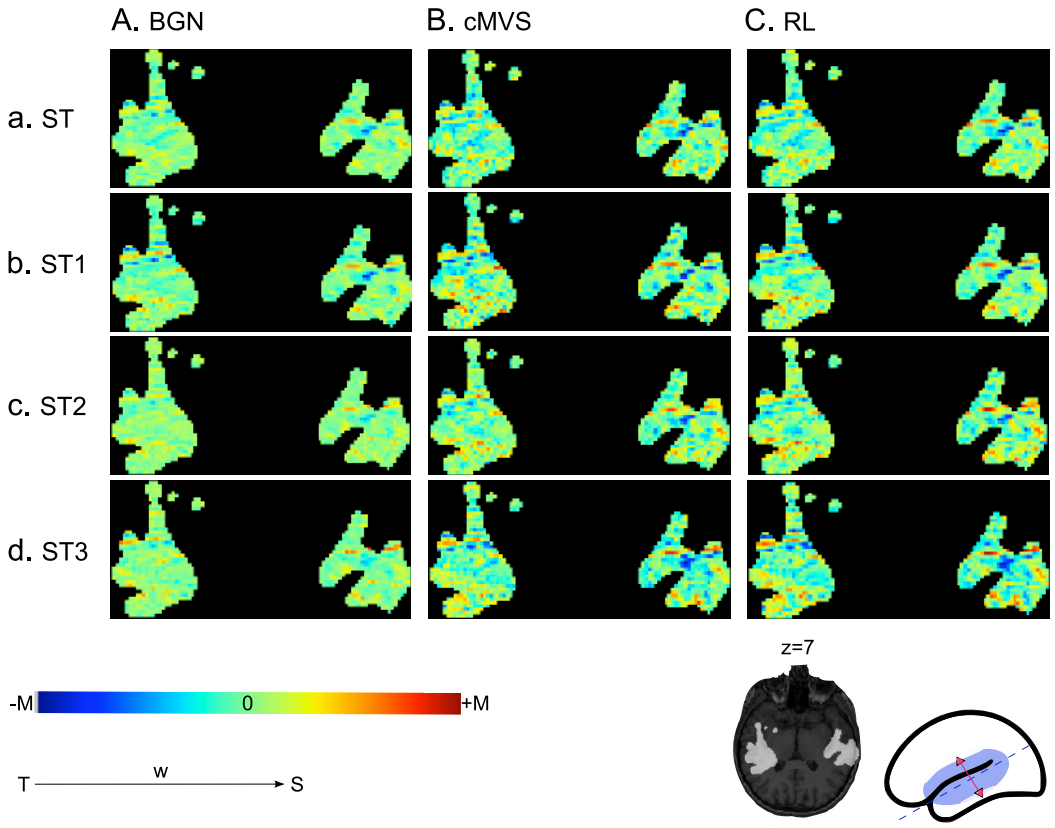


FIGURE 5.5: Poids appris par le modèle BGN avec variances partagées (colonne A), l'algorithme cMVS (colonne B) et l'algorithme RL (colonne C), sur les partitions ST (rangée a), ST_1 (rangée b), ST_2 (rangée c) et ST_3 (rangée d) . Nous montrons seulement les poids des voxels situés sur la tranche $z = 7$. Nous utilisons la même échelle de couleur qu'à la Figure 5.2. Les poids appris par les algorithmes RL et cMVS sont similaires et relativement stable d'une partition à l'autre. Les contrastes dans les poids

Notons que l'hypothèse naïve assumée par le modèle BGN avec variance partagées, voulant que les voxels soient indépendants les uns des autres, ne semble pas s'appliquer ici. En effet, les poids des voxels voisins semblent corrélés spatialement pour tous les classifieurs et toutes les partitions. En fait, il se peut fort bien que les algorithmes RL et cMVS misent précisément sur la stabilité de certaines corrélations pour arriver à bien discriminer.

En résumé, il semble que, de par leur agnosticité, les algorithmes cMVS et RL soient de meilleurs candidats que le modèle BGN avec variances partagées sur nos données. Malgré son instabilité, le modèle BGN avec variances partagées demeure toutefois cohérent avec les algorithmes cMVS et RL au niveau des signes des poids.

5.1.4 Pertinence de la régularisation

Nous revenons maintenant à la partition ST initiale. Nous effectuons quelques essais avec différentes techniques de régularisation sur l'algorithme RL. En plus de la procédure d'arrêt prématurée déjà appliquée précédemment, nous testons les techniques de régularisation $L1$, $L2$ et FE.

Comme expliqué aux sections 3.2.2 et 3.3.1, la technique de régularisation FE (filet élastique) consiste en une combinaison des régularisations $L1$ et $L2$. Nous avons bon espoir qu'elle mène à un meilleur compromis de parcimonie qu'une simple régularisation $L1$. Cette dernière peut parfois se montrer trop sévère en excluant des voxels qui sont pertinents, mais redondants.

La Table 5.4 indique les performances des différentes variantes de l'algorithme RL obtenues. En général, les variantes avec régularisation $L1$ et/ou $L2$ semblent mieux performer. Les différences de performance entre les variantes avec régularisation $L1$ et/ou $L2$ sont quant à elles minimes.

Algorithmes	Performance	Hyper-paramètres
RL (AP)	79%	-
RL + $L2$	83%	$\lambda_2 = 1.0$, $\varepsilon = 0.001$
RL + $L2$ (AP)	83%	$\lambda_2 = 1.0$
RL + $L1$	84%	$\lambda_1 = 0.0175$, $\varepsilon = 0.001$
RL + $L1$ (AP)	86%	$\lambda_1 = 0.0145$
LR + FE	84%	$\lambda_1 = 0.02$, $\lambda_2 = 0.25$, $\varepsilon = 0.001$
LR + FE (AP)	86%	$\lambda_1 = 0.008$, $\lambda_2 = 0.04$

TABLE 5.4: Effets des techniques de régularisation sur la performance de prédiction de l'algorithme RL. L'hyper-paramètre ε, ici, détermine le moment d'arrêt de l'optimisation lorsque la procédure d'arrêt prématuré (AP) n'est pas appliquée. Plus précisément, l'optimisation est stoppée lorsque le changement maximum au niveau des amplitudes des poids est inférieur à ε. Les constantes λ_1 et λ_2 contrôlent la force des régularisation $L1$ et $L2$, respectivement.

Bien que les différences entre les performances soient légères, on observe tout de même des différences assez importantes au niveau des poids appris. La Figure 5.6 montre l'effet des régularisations $L1$, $L2$ et FE. Les poids appris par l'algorithme RL avec régularisation $L2$ (Figure 5.6B) ressemblent en fait beaucoup aux poids appris par l'algorithme RL original (Figure 5.6A). Les poids appris par l'algorithme RL avec régularisation $L1$ (Figure 5.6D) sont quant à eux beaucoup plus parcimonieux, i.e. valent presque tous 0. Le critère de régularisation FE mène à une solution légèrement moins parcimonieuse que celle associée à la régularisation $L1$, mais tout de même très parcimonieuse. Malgré

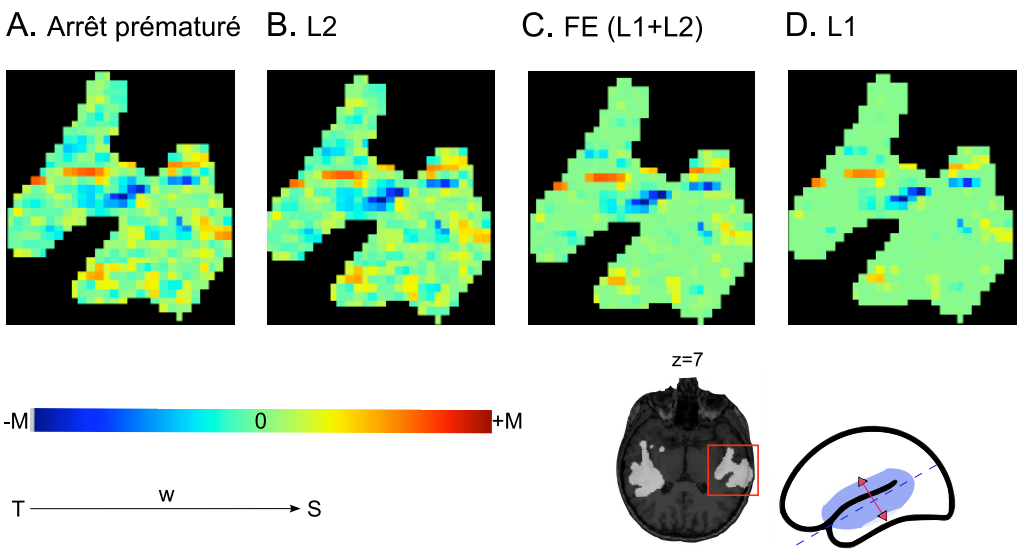


FIGURE 5.6: Poids appris par l'algorithme régression logistique avec arrêt prématuré (A), régularisation $L2$ (B), régularisation FE (C) et finalement régularisation $L1$(D). Nous montrons seulement les poids des voxels situé sur la tranche $z = 7$, dans le lobe temporal droit. Nous avons utilisé la même échelle de couleur qu'à la figure 5.2. Il y a de bonnes différences d'un algorithme à l'autre, au niveau de la parcimonie. La régularisation FE semble mener à une solution légèrement moins parcimonieuse que la régularisation $L1$.

les différences de parcimonie, les modèles demeurent tout de même cohérents entre eux. Les voxels qui résistent à la régularisation L1, i.e. qui conservent une influence, sont pour la plupart associés à des poids assez élevés pour tous les modèles.

La Figure 5.7 montre l'effet de la procédure d'arrêt prématurée sur les poids appris. Elle semble agir comme un frein sur la régularisation $L1$. À la rangée du haut, on aperçoit les poids appris lorsque l'optimisation est poursuivie jusqu'à la convergence de l'algorithme et à la rangée du bas, les poids appris lorsque la procédure d'arrêt prématurée est appliquée. Il ne semble pas y avoir d'effet notable sur la variante RL avec régularisation $L2$ (colonne A). En ce qui concerne les variantes avec régularisation FE et $L1$ (colonnes B et C, respectivement), les solutions semblent nettement moins parcimonieuses lorsque la procédure d'arrêt prématuré est appliquée.

La régularisation FE semble donner des résultats encourageants, cependant il y a place à l'amélioration. Cette technique semble bel et bien inclure plus de voxels que la régularisation $L1$, en particulier lorsqu'elle est utilisée en combinaison avec la procédure d'arrêt prématurée. Cependant, le compromis entre normes $L1$ et $L2$, ici, est faible et instable. En effet, les poids appris ressemblent beaucoup à la solution $L1$ dans le cas sans arrêt prématuré, et beaucoup à la solution $L2$ dans le cas avec arrêt prématuré. Ce comportement bipolaire a aussi été observé par Zou et Hastie (2005) [25] sur données simulées. Un autre problème est que les voxels supplémentaires annexés, par rapport à la régularisation $L1$, sont un peu disparates. Règle générale, les régions d'intérêt ne consistent pas en des voxels isolés, dispersés par ci par là. On s'attend plutôt à ce qu'elles prennent la forme d'ensembles de voxels contigüs dans l'espace, i.e. d'agrégats de voxels. Cet a-priori n'est pas vraiment intégré dans le critère de régularisation FE.

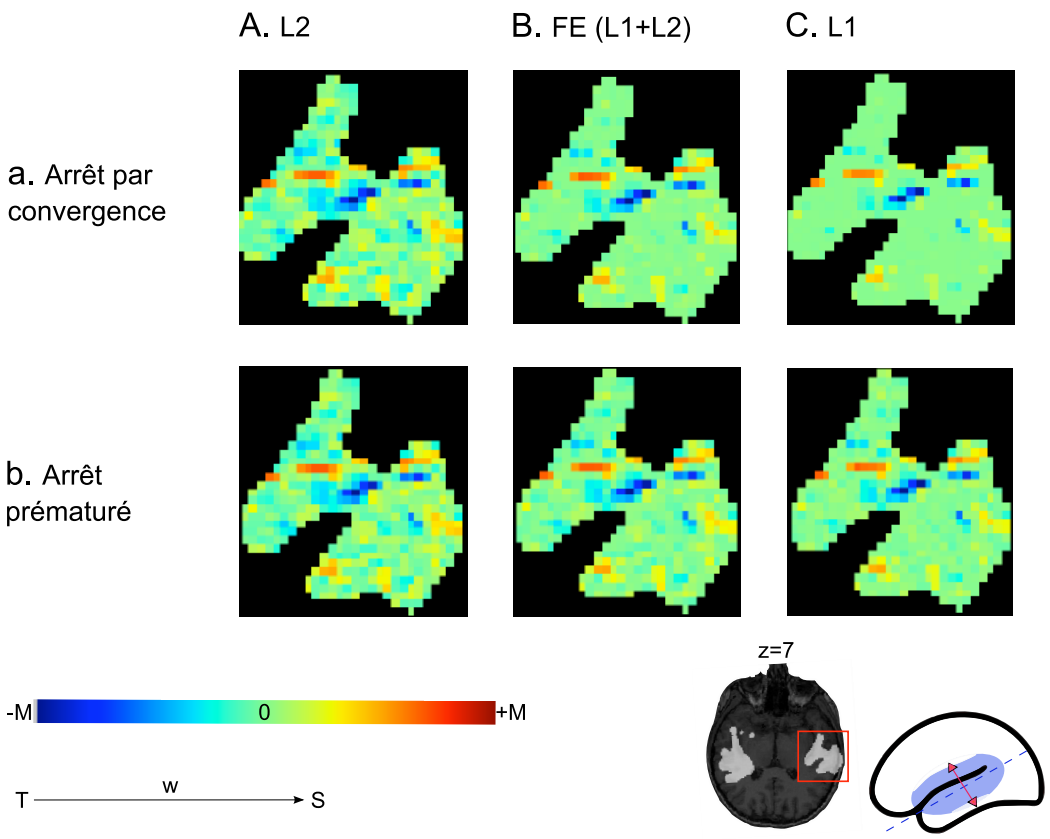


FIGURE 5.7: Poids appris par l'algorithme régression logistique avec régularisation $L2$ (A), régularisation FE (B) et régularisation $L1$ (C), sans arrêt prématuré (rangée a) ou avec arrêt prématuré (rangée b). Nous montrons seulement les poids des voxels situé sur la tranche $z = 7$, dans le lobe temporal droit. Nous avons utilisé la même échelle de couleur qu'à la Figure 5.2. La procédure d'arrêt prématuré semble agir comme un frein sur la régularisation $L1$.

5.2 Expérience 2 : recherche locale

Dans cette expérience, nous cherchons à mieux cerner les avantages et inconvénients de la recherche locale. Nous appliquons l'algorithme de recherche locale défini à la section 3.6 sur le problème de classification défini par la partition ST. Celui-ci se base sur la performance de prédiction d'un classifieur linéaire (nuMVS) pour représenter la quantité d'information dans des sphères de voxels.

Nous effectuons d'abord quelques tests afin d'ajuster le rayon de la sphère et le rayon de lissage. Comme mentionné précédemment (section 4.2.2), nous avons bon espoir qu'un rayon de lde 2 mm soit approprié, cependant, par précaution, nous effectuons aussi quelques essais avec un rayon de lissage de 1 mm.

Une fois les rayons de recherche et de lissage choisis, nous appliquons des tests de significativé sur les performances de prédiction obtenues. Nous comparons deux techniques pour modéliser le comportement de l'algorithme sous l'hypothèse nulle, soit l'approximation binomiale et la randomisation.

Pour terminer, nous comparons les régions d'intérêt obtenues par recherche locale avec les régions d'intérêt obtenues par régularisation $L1$.

5.2.1 Détails du protocole expérimental

Nous appliquons l'algorithme de recherche locale défini à la section 3.6 sur le problème de classification défini par la partition ST. Plus précisément, nous centrons une sphère en chacun des 6342 voxels pré-sélectionnés et, à l'aide de l'algorithme nuMVS, nous construisons des classifieurs linéaires à partir des voxels contenus dans chacune des sphères. Nous produisons une carte de performance indiquant les performances de prédiction de chacun des classifieurs.

Tout comme à l'expérience 1, nous réalisons la plupart des manipulations de l'expérience via la librairie python pyMVPA [28]. En plus des procédures d'entraînement, validation et validation croisées, celle-ci supporte la randomisation et comporte des procédures pour faciliter la recherche locale. Nous nous servons de l'implémentation nuMVS libsvm [26] que nous manipulons également à travers la librairie pyMVPA .

Tout comme à l'expérience 1, nous avons utilisé les deux tiers des donnés pour l'entraînement (incluant la sélection des hyper-paramètres) et le dernier tiers pour tester les classifieurs construits. Nous nous limitons à un seul hyper-paramètre nu pour tous les voxels. Celui-ci est sélectionné au moyen d'une validation croisée en 5 blocs appliquée sur chacun des classifieurs. Nous choisissons l'hyper-paramètre associé à de bons résultats dans l'ensemble.

5.2.2 Sensibilité aux rayons de recherche et de lissage

Nous effectuons ici quelques essais avec différents rayons de recherche et de lissage, dans le but d'évaluer la sensibilité de l'algorithme à ces paramètres sur nos données. Plus précisément, nous testons deux rayons de recherche, soit $r_R = 2.5$ mm et $r_R = 5$ mm, et deux rayons de lissage, soit $r_L = 1$ mm et $r_L = 2$ mm. Rappelons que la résolution horizontale des données est de 1.5 mm $\times$ 1.5 mm et que la résolution verticale est de 2.5 mm. Un rayon de recherche de 2.5 mm englobe donc tout juste les voisins immédiats du voxel situé au centre de la sphère.

À la Figure 5.8, on peut visualiser les cartes de performances associées aux différentes combinaisons. Ainsi, il semble que la combinaison ($r_R = 2.5$, $r_L = 1$) soit trop locale. En effet, pour celle-ci, très peu de voxels sont associés à de fortes performances. Pour toutes les autres combinaisons, i.e. lorsqu'on augmente le rayon de recherche ou le rayon de lissage, la performance de prédiction augmente de manière spectaculaire dans l'ensemble des voxels.

Il semble que certaines régions ne soient activées que lorsque le rayon de lissage est augmenté à 2 mm. À la Figure 5.8E, on peut voir la différence entre les cartes de performances associées aux combinaisons ($r_R = 2.5$, $r_L = 2$) et ($r_R = 5$, $r_L = 1$). Des contrastes positifs très prononcés apparaissent en certains endroits, indiquant une augmentation de performance notable avec davantage de lissage. En revanche, il ne semble pas y avoir de contrastes négatifs très prononcés, ce qui suggère que le lissage supplémentaire n'efface pas trop d'information et exploite déjà suffisamment bien l'information contenue dans le voisinage. Bref, il semble qu'un rayon de lissage de 2 mm soit bel et bien un bon choix sur nos données.

Il semble que lorsqu'on étend le rayon de lissage de à 2 mm, il n'y ait pas grand avantage à étendre aussi le rayon de recherche. En effet, les cartes de performances pour les combinaisons ($r_R = 5$, $r_L = 2$) et ($r_R = 2.5$, $r_L = 2$) sont très similaires. À la Figure 5.8F, on peut voir la différence entre ces deux cartes. Les contrastes sont en général assez faibles, ce qui indique que les cartes diffèrent peu. Les contrastes sont aussi pour la plupart négatifs, ce qui indique une baisse de performance généralisée

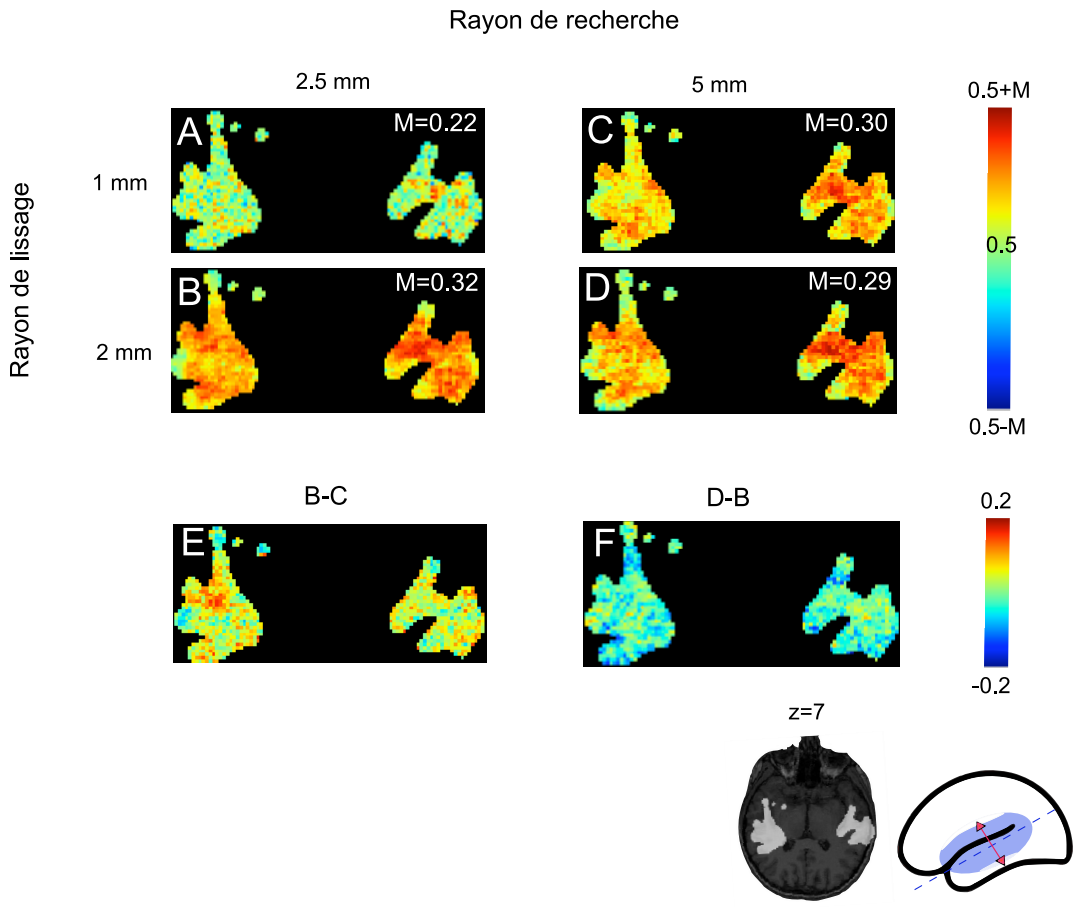


FIGURE 5.8: ABCD. Cartes de performance pour différentes combinaisons de rayons de recherche et de lissage. Nous montrons seulement les performances pour les voxels situé sur la tranche $z = 7$. Les performances sont beaucoup plus faibles en A, lorsque le rayon de lissage et le rayon de recherche sont tous deux petits. E. Différence entre les cartes B et C. Il y a des contrastes assez nets en certains endroits, montrant la présence de différences importantes entre les deux cartes. F. Différence entre les cartes D et B. B. Les contrastes sont pour la plupart faiblement négatifs, indiquant une baisse généralisée de la performance.

pour le rayon de recherche de 5mm. Cette baisse de performances pourrait être dûe à un mauvais choix d'hyper-paramètre, ou encore à un sur-apprentissage plus prononcé en raison de la dimensionalité plus élevée du prédicteur. Le dernier facteur semble légèrement plus probable, puisqu'on observe la même baisse généralisée de performance pour $r_R = 5$ lorsqu'on considère les différences entre les combinaisons ($r_R = 2.5$, $r_L = 2$) et ($r_R = 5$, $r_L = 1$).

Notons que plus on considère de larges rayons, plus l'algorithme est imprécis. En effet, il est assez difficile de faire la différence entre une bonne performance dûe à la présence d'un pattern distribué dans l'ensemble de la sphère et une bonne performance dûe à la présence d'un pattern restreint à un groupe de voxels en périphérie de la sphère. En d'autres mots, lorsque qu'on augmente le rayon de recherche ou de lissage, on augmente aussi le rayonnement des voxels pertinents.

Il semble donc que nous ayons peu à gagner à augmenter le rayon de recherche ou de lissage encore davantage, puisque, en ce qui nous concerne, les régions activées sont déjà relativement larges, i.e. couvrent beaucoup de voxels. Mentionnons, toutefois, qu'afin de détecter davantage de patterns et améliorer la précision de l'algorithme, nous aurions sans doute intérêt à ne pas nous limiter à une

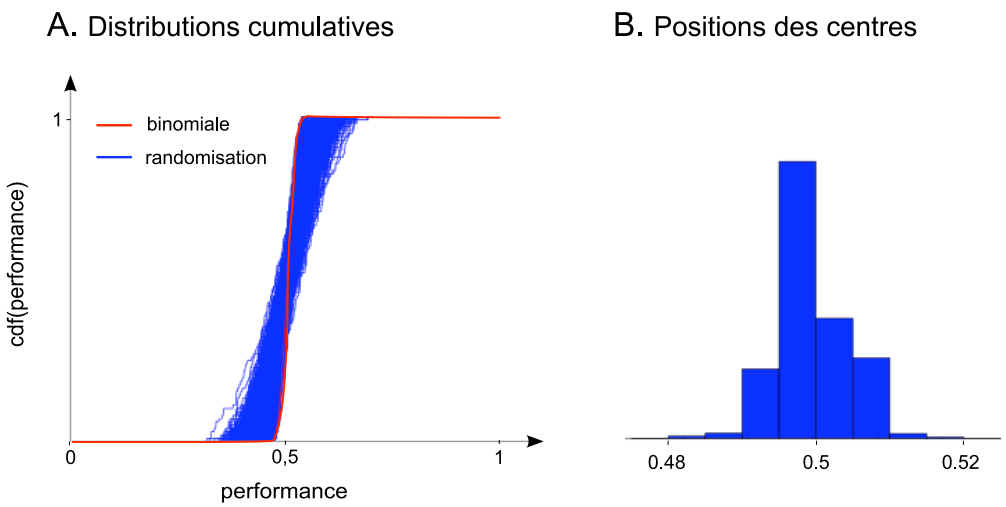


FIGURE 5.9: A. Distributions cumulatives des performances sous l'hypothèse nulle. Les estimés obtenus par randomisation sont tracés en bleu. La distribution cumulative de la loi binomiale est tracée en rouge. Les distributions cumulatives estimées par randomisation ont une forme sigmoïdale centrée en 0.5, tout comme la loi binomiale. Cependant, elles convergent plus lentement que la loi binomiale. B. Histogramme des positions des centres des distributions cumulatives estimées par randomisation. Les centres sont tous très prêts de 0.5.

forme de voisinage sphérique. Par exemple, il se peut fort bien que les patterns pertinents s'étendent plutôt le long des circonvolutions du cortex.

5.2.3 Tests de significativité

Nous appliquons des tests de significativité sur la carte de performance obtenue au moyen d'un rayon de recherche de 2.5 mm et un rayon de lissage de 2 mm. Nous en profitons pour comparer deux manières de modéliser la distribution des performances sous l'hypothèse nulle (H_0), soient l'approximation par une loi binomiale et l'appromixation par une procédure de randomisation.

À la Figure 5.9A, les distributions cumulatives obtenues par randomisation sont tracées en bleu. Elles sont de forme sigmoïdale et sont centrées environ en 0.5, tout comme la distribution cumulative associée à la loi binomiale (en rouge). L'histogramme présenté à la Figure 5.9B permet de voir plus clairement que les positions des centres sont en fait très proches de 0.5. Ainsi, les densités correspondant aux distributions cumulatives estimées par randomisation, tout comme la densité binomiale, ont l'allure d'une cloche centrée en 0.5. Leurs variances, cependant, ne sont pas les mêmes et sont en général plus élevées que celle de la densité binomiale. En effet, les distributions cumulatives obtenues par randomisation semblent croître moins abruptement que la distribution cumulative de la loi binomiale. Les vitesses de convergence diffèrent plus ou moins selon les distributions.

À la Figure 5.10, on aperçoit les régions d'intérêt finales obtenues lorsque les tests de significativité sont basés sur la distribution binomiale et lorsqu'ils sont basés sur les distributions estimées par randomisation. Il y a des différences très prononcées entre les résultats des deux tests. Il semble que l'approximation binomiale ne soit pas assez sévère. En effet, les régions retenues couvrent les lobes temporaux pratiquement en entier après le seuillage. Il semble que la variance de la binomiale sous-estime exagéremment les véritables variances. En revanche, les régions d'intérêt finales obtenues avec la

procédure de randomisation semblent quant à elles beaucoup plus réalistes. Elles semblent cependant un peu trop restreintes, les régions d'intérêt obtenues après le premier seuillage sont considérablement réduites et morcellées par rapport aux performances originales. La correction Bonferroni, ici, semble un peu trop sévère ; mentionnons qu'il s'agit d'un autre exemple de conséquence d'une intégration inadéquate des corrélations spatiales dans le processus d'analyse des données.

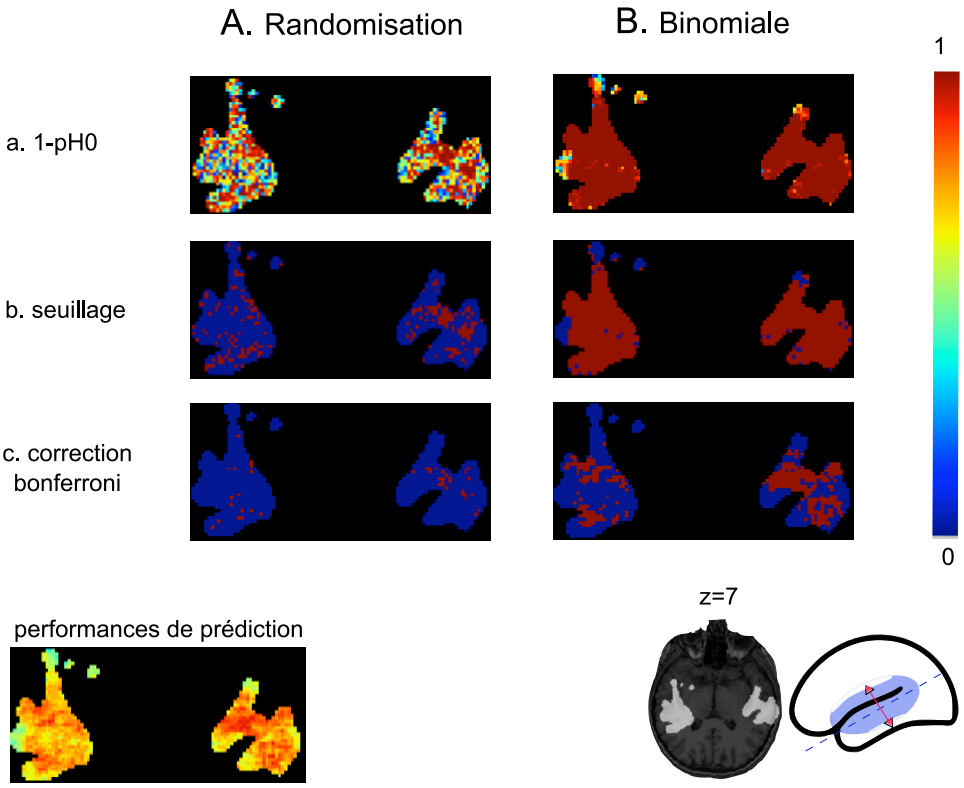


FIGURE 5.10: Tests de significativité appliqués sur les voxels de la tranche $z = 7$ à l'aide des distributions estimées par randomisation (colonne A) et à l'aide de la distribution binomiale (colonne B). Les performances de prédiction originales, pour un rayon de lissage de 2 mm et un rayon de recherche de 2.5 mm, sont montrées dans le coin inférieur gauche. (Voir la Figure 5.8B pour l'échelle de couleur). D'abord, on calcule la probabilité de l'hypothèse nulle pour chaque performance (rangée a). Ensuite, on retient seulement les voxels pour lesquels les probabilités de l'hypothèse nulle inférieures à un certain seuil de 0.05 (rangée b). Afin de tenir compte du nombre de voxels, on effectue une correction Bonferroni (rangée c). Cette opération consiste à diviser le seuil original par le nombre de voxels. Il y a des différences très prononcées entre les régions d'intérêt obtenues au moyen de la distribution binomiale et les régions d'intérêt obtenues au moyen des distributions estimées par randomisation. Il semble que l'approximation binomiale ne soit pas assez sévère. En effet, les régions retenues couvrent pratiquement tous les voxels après le premier seuillage.

Nous avons finalement choisi de nous en remettre à la procédure de randomisation pour estimer les distributions de l'hypothèse nulle. Bien qu'elle soit plus coûteuse computationnellement, elle semble beaucoup plus fiable que l'approximation par la loi binomiale sur nos données. Il pourrait être inté-

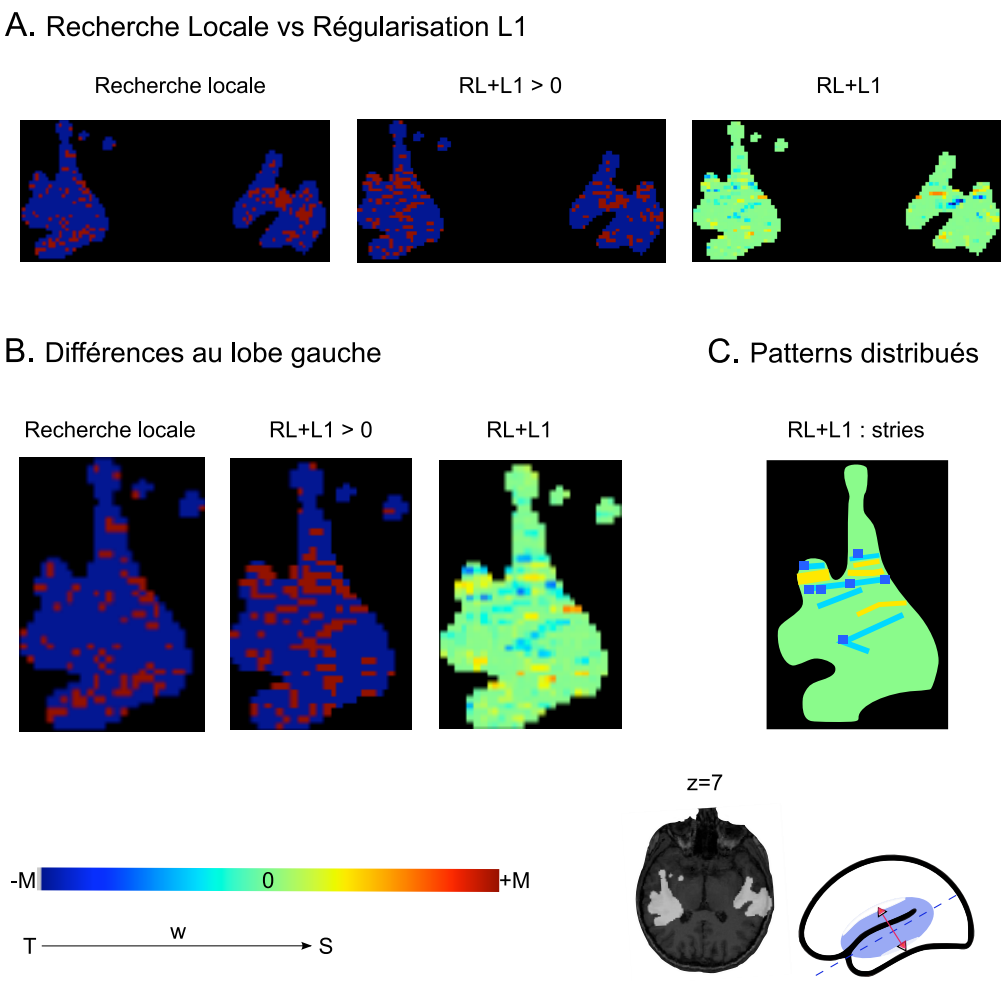


FIGURE 5.11: Régions d'intérêt obtenues par recherche locale et par régularisation $L1$ pour la tranche de voxels $z = 7$. A. À gauche : régions d'intérêt (en rouge) obtenues par recherche locale. Nous avons utilisé un rayon de recherche de 2.5 mm, un rayon de lissage de 2 mm, une procédure de randomisation pour la modélisation de l'hypothèse nulle (H_0) et un seuillage simple ($p(H_0) \leq 0.05$). Au centre : régions d'intérêt (en rouge) obtenues par régularisation $L1$. Plus précisément, les voxels correspondant aux poids non nuls appris par l'algorithme RL avec régularisation $L1$ sont sélectionnés. À droite : poids appris par l'algorithme RL avec régularisation $L1$. Dans la partie supérieure du lobe gauche, il y a des différences importante entre les régions d'intérêt sélectionnées par l'algorithme de recherche locale et les régions d'intérêt sélectionnés par régularisation $L1$. B. Idem A, avec zoom sur le lobe gauche. C. Schéma de quelques patterns intéressant dans les poids du classifieur RL, au lobe gauche. Les poids dans la partie supérieure semblent organisés en stries. De telles stries peuvent être difficile à repérer par recherche locale puisque celle-ci n'est appliquée que sur de petites sphères de voxels.

ressant, éventuellement, de considérer d'autres approximations que la loi binomiale. Il se peut qu'on obtienne des résultats plus réalistes avec une distribution de variance un peu plus élevée.

5.2.4 Recherche locale vs recherche globale

Ici, nous comparons les régions d'intérêt détectées au moyen d'une recherche locale et les régions d'intérêt détectées au moyen d'une régularisation $L1$.

À la Figure 5.11A, on aperçoit les régions d'intérêt obtenues à l'aide des deux méthodes. Les régions d'intérêts obtenues par recherche locale correspondent, plus précisément, aux voxels sélectionnés au moyen des tests de significativité appliqués sur la carte de performance locale. Cette dernière est associée à un rayon de recherche de 2.5 mm et un rayon de lissage de 2 mm. Nous n'avons appliqué qu'un seuillage simple (Figure 5.10Ab) sur les p-valeurs et avons utilisé la procédure de randomisation pour estimer les distributions des performances sous l'hypothèse nulle. Les voxels sélectionnés par l'algorithme RL avec régularisation $L1$ sont les voxels dont le poids ne vaut pas 0, i.e. les voxels qui conservent une influence sur le classifieur.

Il semble y avoir des différences plus marquées entre les deux techniques au niveau de la partie supérieure du lobe gauche. Or, il se trouve que les poids du classifieur, dans cette région, sont organisés en stries (Figure 5.11BC). Il ne serait pas surprenant que de tels patterns ne soient pas détectés par l'algorithme de recherche locale que nous avons appliqué. En effet, ce dernier ne considère que de petites sphères de voxels et non des bandes de voxels. Notons qu'il n'est pas garanti que ces stries soient pertinentes. En effet, elles pourraient être dûes à du sur-apprentissage.

Cet exemple illustre bien la complémentarité des deux techniques de détection des régions d'intérêt. La recherche locale permet de dégager de manière rigoureuse la force des contributions des différents groupes de voxels sur lesquels elle est appliquée, en particulier lorsqu'elle est couplée à des tests de significativité. Cependant, la recherche locale implique aussi davantage de suppositions quant à la configuration géométrique des patterns pertinents. La régularisation $L1$, quant à elle, sélectionne les voxels de manière globale et est moins limitée de ce côté. Cependant, il est peu recommendé de s'y fier aveuglément, en raison de la présence potentielle de sur-apprentissage et en raison de sa tendance à exclure parfois un peu trop rapidement les voxels plus redondants.

Afin de tirer parti des avantages de ces deux techniques, une alternative intéssante serait d'effectuer d'abord une pré-sélection des régions d'intérêts à l'aide d'une régularisation $L1$ et, ensuite, d'utiliser une recherche locale pour mieux chiffer l'information contenue dans ces régions. Il pourrait aussi être intéressant de se tourner vers d'autres méthodes de sélection de "features" que la régularisation $L1$, incorporant mieux les connaissances à priori relatives à la forme des régions d'intérêt.

5.3 Expérience 3 : localisation des régions d'encodage

Le but de cette expérience est de localiser les régions d'encodage potentielles des ondulations dynamiques. Nous utilisons des partitions plus raffinées de l'espace des ondulations dynamiques, soit les partitions en 4 et 9 quadrants. Nous appliquons des classifieurs linéaires et visualisons leurs poids. Nous nous attendons à ce que la région d'encodage prenne de plus en plus d'importance au fur et à mesure qu'on raffine la partition de l'espace des ondulations dynamiques.

5.3.1 Détails du protocole expérimental

Les procédures d'entraînement et sélection de modèles sont similaires aux expériences 1 et 2.

L'extension multiclasse utilisée pour les algorithmes cMVS et nuMVS est l'approche un-contre-un (voir section 3.5.3). Les mesures de sensibilités utilisées pour visualiser les paramètres des classifieurs sont définies à la section 3.5.4.

Encore une fois, nous faisons appel à la libraire python pyMVPA [28] pour la majeure partie des manipulations. Les implémentations des algorithmes utilisées sont les mêmes qu'aux expériences 1 et 2. C'est-à-dire, on utilise libsvm [26] pour cMVS et nuMVS et la librairie PLearn [27] pour RL.

5.3.2 Classifieurs linéaires

Les performances de prédiction des algorithmes pour les différentes partitions sont réunies à la Table 5.5. La dernière rangée indique la performance d'un classifieur "aléatoire", i.e. qui prédit la cible au hasard peu importe le prédicteur. Tous les algorithmes performent sensiblement mieux que le classifieur aléatoire, ce qui suggère fortement la présence de différences significatives entre les différentes conditions. Encore une fois, les algorithmes cMVS et RL performent nettement mieux que le modèle BGN avec variances partagées. La régularisation $L1$ semble considérablement améliorer la performance de l'algorithme RL sur la partition en 4 Quadrants.

Modèles\Découpages	4 Quadrants	9 Quadrants
BGN	51%	28%
cMVS	73%	49%
RL (Arrêt prématuré)	66%	44%
RL+L1 (Arrêt prématuré)	74%	46%
Aléatoire	25%	11%

TABLE 5.5: Performances des algorithmes pour les partitions en 4 et 9 quadrants. La dernière rangée indique la performance d'un classifieur "aléatoire", i.e. qui prédit la cible au hasard peu importe le prédicteur. Tous les algorithmes performent beaucoup mieux que le classifieur aléatoire. Encore une fois, les algorithme cMVS et RL performent nettement mieux que le modèle BGN avec variances partagées.

À la Figure 5.12, on peut voir les **matrices de confusion** associées aux deux meilleurs classifieurs, soient cMVS et RL avec régularisation $L1$ et arrêt prématuré. L'élément (i, j) d'une matrice de confusion indique la proportion de cibles i classifiées comme des cibles j. Ici, dans tous les cas, la diagonale ressort assez bien, ce qui signifie que les cibles sont en majeure partie bien classifiées. Pour les deux algorithmes, il semble y avoir un peu plus de confusion pour la partition en 9 quadrants. En effet, la dominance de la diagonale devient assez faible par endroits, notamment aux cibles 4 et 8. En ce qui

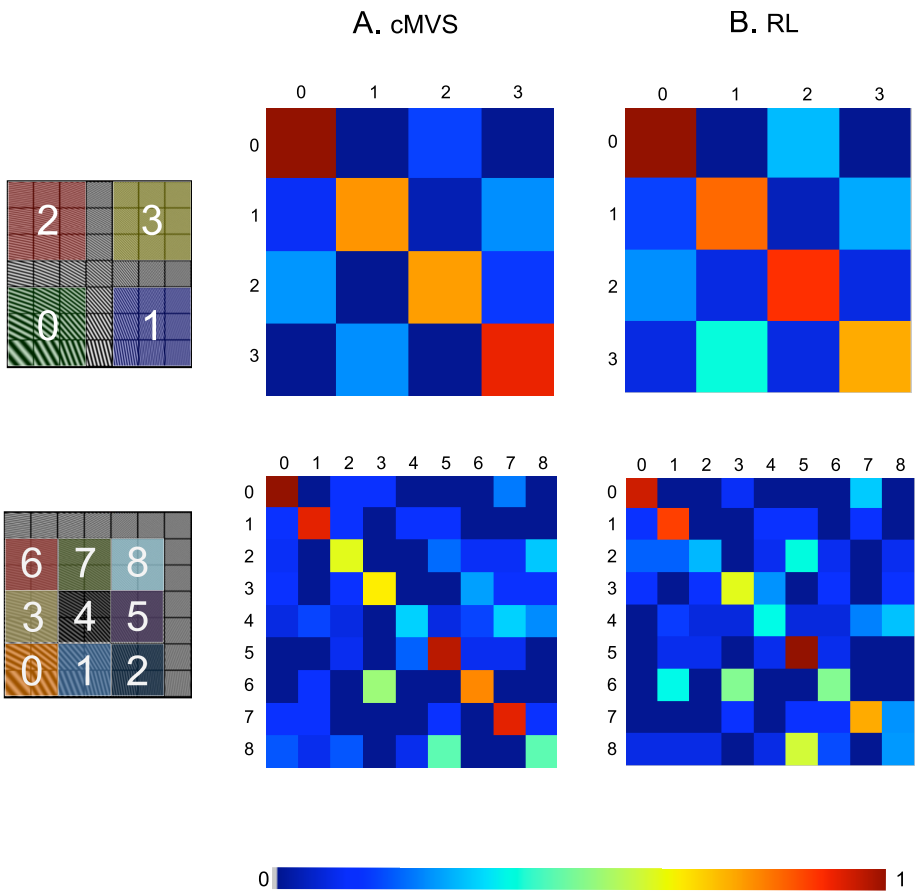


FIGURE 5.12: Matrices de confusion pour les partitions en 4 et 9 quadrants pour les algorithmes cMVS (colonne A) et RL (colonne B). La couleur d'une case (i, j) indique la proportion de cibles i classifiées comme des cibles j. Dans tous les cas, il semble que la diagonale soit dominante, ce qui signifie que les cibles sont bien classifiées. Il semble y avoir légèrement plus de confusion pour la partition en 9 quadrants, en particulier pour l'algorithme RL. Les cibles les plus confondues sont souvents proches dans l'espace des ondulations dynamiques.

concerne l'algorithme RL, la dominance est brisée pour les cibles 2 et 8. Malgré tout, bien que certaines cibles semblent plus faciles à discriminer que d'autres, dans l'ensemble, elles sont bien classifiées en assez forte proportion.

Le fait que les patterns appris par les algorithmes mènent à une bonne discrimination des cibles dans leur ensemble et pas, par exemple, à une cible en particulier au détriment des autres est rassurant quant à l'influence de la région d'encodage dans ces patterns. Une autre détail intéressant dans les matrices de confusion est que les régions correspondant aux paires de cibles les plus confondues sont souvent voisines dans l'espace des ondulations dynamiques. Pour le découpage en 4 quadrants, c'est vrai pour les paires (0,2) et (1,3) communes aux deux algorithmes. En ce qui concerne le découpage en 9 quadrants, c'est vrai pour les paires (3,6), (4,7), (4,8) communes aux deux algorithmes. Pour l'algorithme RL, où il y a un peu plus de confusion, c'est aussi vrai pour les paires (2,5),(3,4) et (7,8).

Nous nous penchons maintenant sur les poids appris par les deux algorithmes les plus performants, soient les algorithmes cMVS et RL avec régularisation $L1$ et arrêt prématuré. Plus tâchons de voir comment la sensibilité des classifieurs au différents voxels varie au fur et à mesure qu'on raffine les partitions. Nous utilisons les mesures de sensibilité définies à la section 3.5.4 sur les partitions en 4

et 9 quadrants. En ce qui concerne la partition ST, nous utililisons les valeurs absolues des poids appris. Il y a plusieurs patterns communs aux deux algorithmes et qui semblent être de bons candidats pour la région d'encodage. Nous en présentons ici deux en particulier, situés sur la tranche de voxel $z = 8$, soit au environ au centre des lobes temporaux. Pour la partition en 9 quadrants, les sensibilités correspondant à l'algorithme RL avec régularisation $L1$ sont plus difficiles à utiliser pour le repérage des régions d'encodage, car les contrastes entre les voxels sont peu prononcés. Cependant, en général, elles semblent cohérentes avec les sensibilités de l'algorithme cMVS.

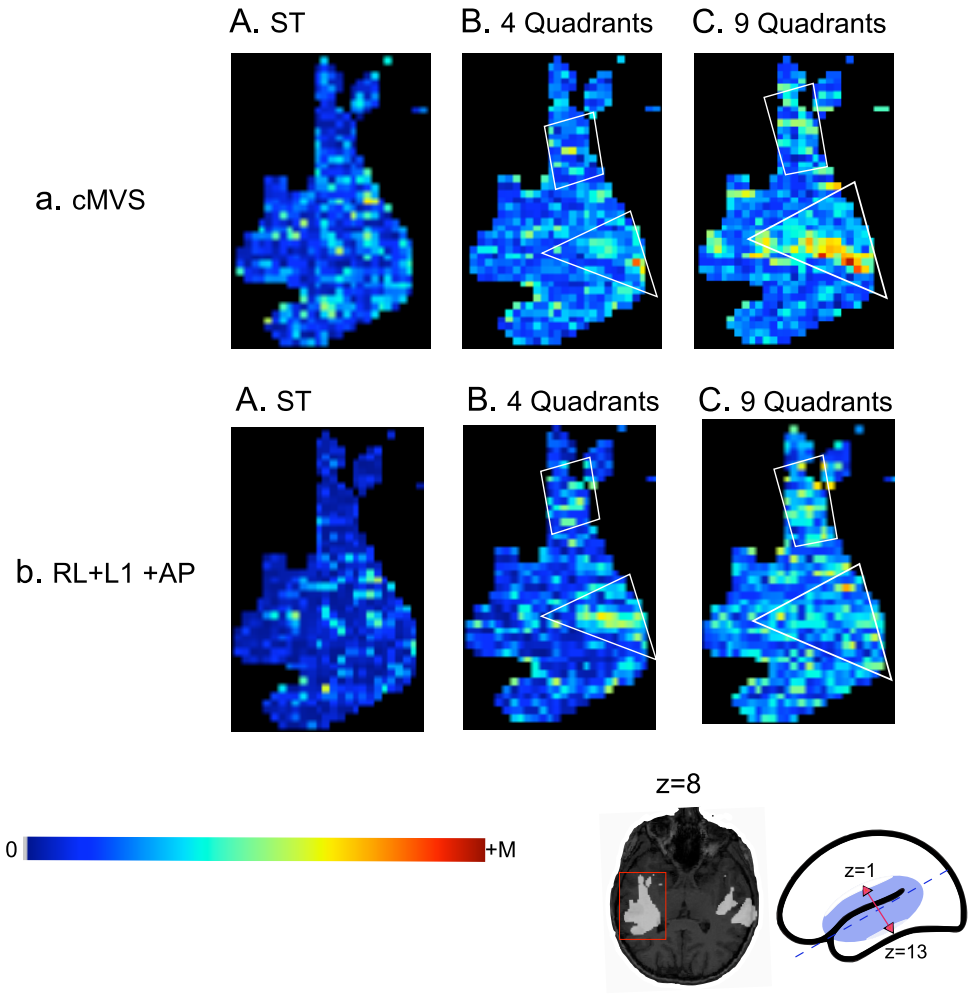


FIGURE 5.13: Mesures de sensibilités pour les partitions ST (colonne A) , 4 quadrants (colonne B) et 9 quadrants (colonne C) dérivées des poids des algorithmes cMVS (rangée a), et RL avec régularisation L1 et arrêt prématuré (AP) (rangée b), dans le lobe temporal gauche, à la tranche $z = 8$. Dans le cas de cMVS, au centre du love, un agrégat de voxels ayant la forme d'un faisceau devient nettement plus influent pour la partition en 9 quadrants. L'influence commence à poindre légèrement sur la droite pour la partition en 4 quadrants. Dans le haut du love, il semble y avoir une légère activation qui augmente et s'étend d'une partition à l'autre. Des patterns similaires sont présents dans les mesures de sensibilités de l'algorithme RL avec régularisation L1 et arrêt prématuré. Les mesures de sensibilité pour la partition en 9 quadrants, cependant ne sont pas aussi contrastées que celles de l'alorithme cMVS.

À la Figure 5.13, on aperçoit les mesures de sensibilités paur le lobe gauche de la tranche $z = 8$. Dans le cas de l'algorithme cMVS, au centre droit du lobe, un agrégat de voxels ayant la forme d'un faisceau devient nettement plus influent pour la partition en 9 quadrants. L'influence commence à poindre légèrement sur la droite pour la partition en 4 quadrants. De plus, dans le haut du lobe, il semble y avoir une légère activation qui augmente et s'étend d'une partition à l'autre. Des patterns similaires sont présents dans les mesures de sensibilités de l'algorithme RL avec régularisation $L1$ et arrêt prématuré. Les mesures de sensibilité pour la partition en 9 quadrants, cependant ne sont pas aussi contrastées que celles de l'alorithme cMVS.

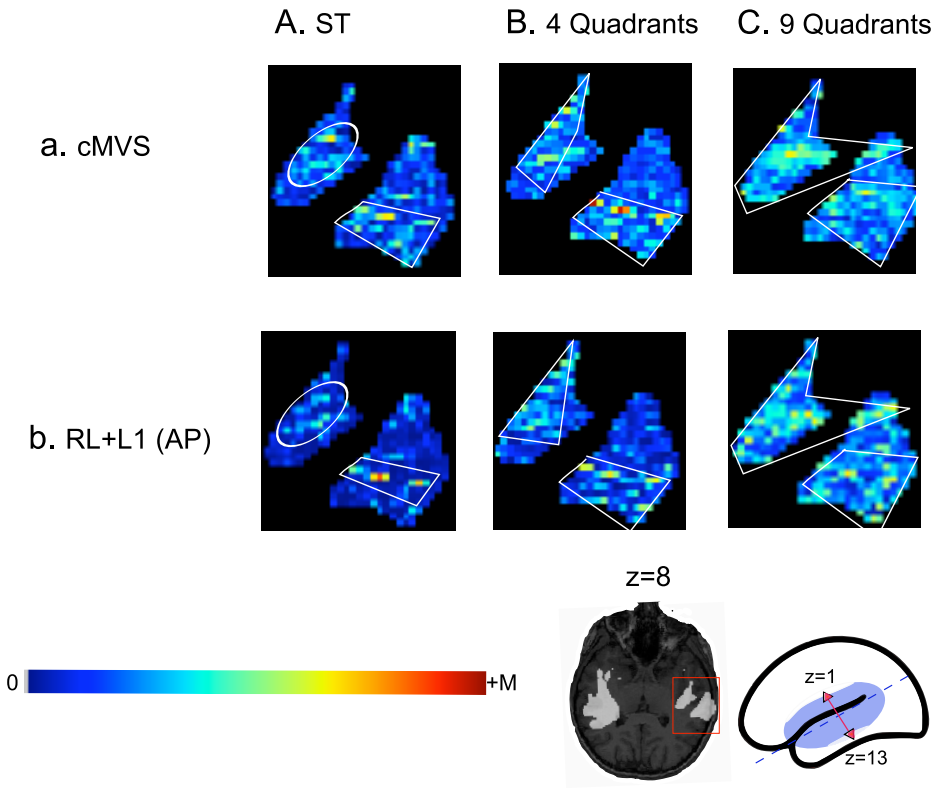


FIGURE 5.14: Mesures de sensibilités pour les partitions *ST* (colonne A) , 4 quadrants (colonne B) et 9 quadrants (colonne C) dérivées des poids des algorithmes cMVS (rangée a), et RL avec régularisation L1 et arrêt prématuré (AP) (rangée b), dans le lobe temporal gauche, à la tranche $z = 8$. Il semble y avoir 3 faisceaux de voxels qui prennent de l'influence et/ou s'élargissent d'une partition à l'autre.

À la Figure 5.14, on aperçoit les mesures de sensibilités pour le lobe droit de la tranche $z = 8$. Il semble y avoir 3 faisceaux de voxels qui prennent de l'influence et/ou s'élargissent d'une partition à l'autre. Le faisceau situé dans la partie inférieure droite est un candidat intéressant, mais un peu plus ambigü. En fait, on observe bien un élargissement du faisceau, mais aussi une diminution de son influence.

5.4 Travail futur

Un objectif à plus long terme de ce projet est aussi le développement d'algorithmes d'apprentissage plus spécialisé pour les données fMRI. Dans cette optique, nous avons porté une attention particulière, à l'expérience 1, à la pertinence de certaines techniques de régularisation à l'application. À la section 5.4.1, nous effectuons un retour sur l'efficacité de la régularisation FE à l'intégration des corrélations spatiales. Celle-ci donne des résultats intéressants, mais comporte certaines imperfections. Nous proposons une alternative qui pourrait éventuellement donner de meilleurs résultats. À la section 5.4.2, nous nous penchons cette fois sur le potentiel des algorithmes génératifs. Comme expliqué à la section X, l'utilisation d'un critère génératif peut être vu comme une forme de régula- risation. Le modèle BGN avec variances partagées utilisé dans nos expériences est un cas particulier d'algorithme génératif, plus précisément un classifieur de Bayes. Nous faisons le point sur ce dernier et dégageons quelques avenues de recherche intéressantes.

5.4.1 Intégration des corrélations spatiales

Il semble qu'une direction de recherche intéressante, pour améliorer le comportement des algorithmes d'apprentissage, soit de favoriser à la fois la parcimonie et l'intégration des corrélations spatiales au niveau de la régularisation.

La technique de régularisation FE (Filet Élastique) va en ce sens et semble donner des résultats encourageants sur nos données, du moins par rapport à la norme $L1$. Cependant, comme nous en avons déjà discuté à la section 5.1.4, le compromis entre normes $L1$ et $L2$ choisi semble faible et instable. En effet, les poids appris ressemblent soit beaucoup à la solution $L1$, sait beaucoup à la solution $L2$. Ce comportement bipolaire a aussi été observé par Zou et Hastie (2005) [25] sur données simulées. Un autre problème avec le critère FE est que l'identité des voxels qui sont corrélés n'est peut-être pas assez explicitement définies. Les voxels supplémentaires annexés, par rapport à la régularisation $L1$, sont un peu disparates. Règle générale, les régions d'intérêt ne consistent pas en des voxels isolés, dispersés par ci par là. On s'attend plutôt à ce qu'elles prennent la forme d'ensembles de voxels contigüs dans l'espace, i.e. d'agrégats de voxels.

Dans le futur, nous nous proposons d'expérimenter un nouveau critère de régularisation sur les paramètres $\boldsymbol{\theta}$ des algorithmes qui est le suivant :

$$\sum_{i<j} \lambda_{i,j}(|\theta_i| - |\,\theta_j|)^2$$

Avec

$$\lambda_{i,j} = k(voxel_i, voxel_j)$$

Ici, le noyau k indique le degré de similarité entre deux voxels. Une possibilité est de baser k sur la distance euclidienne. Il pourrait aussi être pertinent d'intégrer des connaissances à priori quant aux régions fonctionnelles, par exemple l'appartenance ou non à une même circonvolution. Ainsi, par le choix de la fonction k, on peut prioriser l'annexion de certains voxels avant les autres.

Une autre propriété intéressante de ce critère est qu'il agit à la fois comme une érosion et une dilatation. C'est-à-dire, si un voxel est important, alors il y a des chances que les voxels voisins soient importants. En revanche, si aucun voxel dans le voisinage n'est important, alors il y a des chances que le voxel ne soit pas important non plus.

Ce critère n'encourage pas nécessairement la parcimonie, du moins cela reste à vérifier. Une combinaison avec la norme $L1$ pourrait être intéressante pour pallier à cette faiblesse, le cas échéant.

5.4.2 Algorithmes génératifs vs discriminants

Le fait de chercher à mieux expliquer les données plutôt que de bien classifier à tout prix pourrait potentiellement aider à améliorer l'interprétabilité des modèles.

Les classifieurs de Bayes, basés sur des estimateurs de densités, vont en ce sens. Ici, le défi est de déterminer un degré de complexité adéquat pour les estimateurs de densités. Des suppositions trop rigides ou erronées mènent à une modélisation grossière des densités qui ne permet pas de bien capter les différences entre les distributions d'une condition à l'autre. En revanche, trop de souplesse peut mener à du sur-apprentissage dans la modélisation des densités et, au final, à de pauvres performances de classification.

En ce qui concerne le modèle bayésien gaussien naïf, il semble que les suppositions émises quant à la nature des données soient clairement erronées. En effet, son équivalent discriminant, soit l'algorithme régression logistique, performe systématiquement mieux sur toutes les partitions considérées. Le fait de considérer que les voxels sont indépendants les uns des autres semble particulièrement douteux, étant donné la présence de corrélations spatiales évidentes au niveau des poids appris par tous les algorithmes sur toutes les partitions.

Il se pourrait qu'on obtienne de meilleurs résultats avec des estimateurs de densités un peu plus complexes que le modèle BGN avec variances partagées. En fait, du travail a déjà été fait en ce sens sur données fMRI avec des RBMs par Schmah et Hinton (2008) [12].

Notons que, plutôt que de se restreindre à des classifieurs de Bayes, il pourrait aussi être intéressant de considérer des algorithmes hybrides qui optimisent à la fois un critère génératif et un critère discriminant. De tels algorithmes, justement basés sur des RBMs, ont été investigués assez récemment par Larochelle et Bengio (2008) [10]. Il pourrait être intéressant d'appliquer ces variantes sur données fMRI.

Conclusion

Dans le cadre de ce projet, nous nous sommes penchés sur une hypothèse d'encodage concernant les premières étapes de la perception auditive. Selon celle-ci, les modulations spectrales et temporelles présentes dans les sons seraient détectées et encodées dans le cerveau humain. Un tel encodage serait réalisé par une banque de filtres sensibles à des modulations spectro-temporelles variées.

Nous avons analysé les enregistrements fMRI de sujets soumis à 49 modulations spectro-temporelles différentes à l'aide de classifieurs linéaires. Ces données ont été recueillies lors d'une étude préalable effectuée par Schönwiesner [14] (voir sections 1.2, 4.1 et 4.2).

Comme nous l'avons expliqué aux sections 1.2 et 2.4, les données de Schönwiesner se prêtent très mal à une analyse statistique traditionnelle. Cette dernière ne permet de détecter que les différences présentes sous forme de forts contrastes au niveau de la réponse fMRI, en des voxels isolés. Les structures neuronales impliquées dans l'étude de Schönwiesner seraient assez petites par rapport à la résolution fMRI et induiraient plutôt des préférences faibles et distribuées à travers les voxels.

Nos buts à travers les différentes expériences réalisées étaient multiples. D'abord, nous cherchions à mieux cerner le potentiel des classifieurs linéaires sur nos données et sur les données fMRI en général (expériences 1 et 2). Nous cherchions aussi à repérer la région d'encodage des modulations spectro-temporelles (expérience 3).

Selon les résultats de l'ensemble des expériences, il semble que les classifieurs linéaires soient des outils d'analyse intéressants et prometteurs pour nos données. Le comportement des différents classifieurs est rassurant. Bien qu'ils ne performent pas toujours aussi bien les uns que les autres, ils sont cohérents entre eux au niveau de leurs paramètres et, dans tous les cas, performent mieux qu'un classifieur aléatoire. Malgré quelques imperfections, notamment une mauvaise intégration des corrélations spatiales, ils nous permettent déjà de dépasser les limites de l'analyse traditionnelle. En effet, des différences significatives impossibles à voir avec des tests de Student univariés sont détectées. Nous avons même réussi à repérer quelques candidats intéressants pour la région d'encodage des modulations spectro-temporelles à l'expérience 3 (voir Figures 5.13 et 5.14).

Comme nous l'avons illustré à l'expérience 2 (section 5.2.4), la recherche locale est un excellent complément à une analyse plus globale par les poids des classifieurs. Elle ne permet pas de détecter les patterns pertinents avec autant d'exhaustivité et de précision que la régularisation $L1$, cependant elle permet de dégager avec rigueur (significativité statistique) la contribution de régions isolées. Une amélioration intéressante à l'algorithme de recherche locale serait d'effectuer la recherche le long des circonvolutions du cortex. Évidemment, comme nous en avons déjà discuté à la section 5.2.4, une autre alternative intéressante serait de d'abord sélectionner les trajectoires de recherche à l'aide des régions d'intérêts sélectionnées par régularisation $L1$ ou une autre méthode de sélection de features.

En plus des améliorations à la recherche locale, ce projet nous a aussi permis de dégager quelques pistes de recherche intéressante pour le futur. Comme nous en avons discuté à la section 5.4, il se pourrait qu'on ait intérêt à miser sur l'intégration des corrélations spatiales et à investiguer davantage

le potentiel des algorithmes plus génératifs.

Bibliographie

[1] C. M. Bishop, ***Pattern recognition and machine learning***

[2] B. E. Boser, I. M. Guyon, V. Vapnik, 1992, ***A training algorithm for optimal margin classifiers***, Fifth Annual Workshop on Computational Learning Theory, ACM, 144-152

[3] G. Bouchard, B. Triggs, ***The trade-off between generative and discriminative classifiers***, CompStat 2004 Symposium

[4] M. K. Carroll, G.A. Cecchi, I. Rish, R. Garg, A. Rao, 2009, ***Prediction and interpretation of distributed neural activity with sparse models***, NeuroImage, vol. 44, no 1, 112-122.

[5] C. Cortes, V. Vapnik, 1995, , Machine Learning Journal, ***Support Vector Networks***, vol. 20, 273-297

[6] K. Friston, A. Holmes, K. Worsley, J.-B. Poline, C. Frith, R. Frackowiack, 1995, ***Statistical parametric maps in functional imaging : a general linear approach***, Human Brain Mapping, vol. 2, 189–210.

[7] P. Jezzard, P. Matthews. S. Smith, 2001, ***Functional MRI : an introduction to methods***, Oxford : OUP

[8] Y. Kamitani, F. Tong, 2005, ***Decoding the visual and subjective contents of the human brain***, Nature Neuroscience, vol. 8, no. 5, 679-685

[9] N. Kriegeskorte, R. Goebel, P. Bandettini, ***Information-based functional brain mapping***, PNAS, vol. 103, no. 10, 3863-3868

[10] H. Larochelle, Y. Bengio, ***Classification using Discriminative Restricted Boltzmann Machines***, , ICML

[11] R. Legenstein, octobre 2009, ***1.2 Introductory Example : Polynomial Curve Fitting***, Institute for Theorical Computer Science

[12] T. Schmah, G. Hinton, R. Zemel, 2008, ***Competing RBM density models for classification of fMRI images***, NIPS

[13] M. Schönwiesner, R. Rübsamen, D. Y. von Cramon, 2005, ***Hemispheric asymmetry for spectral and temporal processing in the human antero-lateral auditory belt cortex***, Eur J Neurosci 22 :1521–1528

[14] M. Schönwiesner, R. J. Zatorre, 2009, ***Spectro-temporal modulation transfer function of single voxels in the human auditory cortex measured with high-resolution fMRI***, PNAS, vol. 106 no. 34 14611-14616

[15] S. Shamma, H. Versnel and N. Kowalski, 1995a, ***Ripple Analysis in the Ferret Auditory Cortex : I. Response Characteristics of Single Units to Sinusoidally Rippled Spectra***, J. Auditory Neuroscience (1) 233-254

[16] S. Shamma and H. Versnel, 1995b, ***Ripple Analysis in the Ferret Primary Auditory Cortex. II :. Prediction of Unit Responses to Arbitrary Spectral Profiles***, J. Auditory Neuroscience (1) 255-270

[17] N. Kowalski, D. Depireux and S. Shamma, 1996a***, Analysis of dynamic spectra in ferret AI : Characteristics of single unit responses to moving ripple spectra***, J. Neurophysiology 76 (5) 3503-3523

[18] N. Kowalski, D. Depireux and S. Shamma, 1996b, ***Analysis of dynamic spectra in ferret AI : Prediction of single unit responses to arbitrary dynamic spectra***, J. Neurophysiology 76 (5) 3524-3534

[19] D. A. Depireux, J. Z. Simon, D. J. Klein, S. Shamma, ***Spectro-Temporal Response Field Characterization With Dynamic Ripples in Ferret Primary Auditory Cortex,*** J. Neurophysiology 85 1220-1234

[20] J. R. Shewchuk, août 1994, ***An Introduction to the Conjugate Gradient Method Without the Agonizing Pain***, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213

[21] J. Tanabe, D. Miller, J. Tregellas, R. Freedman, F. G. Meyer, 2001, ***Comparison of Detrending Methods for Optimal fMRI Preprocessing***, NeuroImage 15, 902–907

[22] V. Vapnik, A. Lerner, ***Pattern Recognition using Generalized Portrait Method***, Automation And Remote Control, vol. 24, 774-780

[23] J. Weston, C. Watkins, 1999, ***Support vector machines for multiclass pattern recognition***, Proceedings of the Seventh European Symposium On Artificial Neural Networks

[24] R. J. Zatorre, P. Belin, 2001, ***Spectral and temporal processing in human auditory cortex***, Cereb. Cortex, 11, 946–953

[25] H. Zou, T. Hastie, 2005, ***Regularization and variable selection via the elastic net***, J. R. Statist. Soc. B 67, Part 2, 301–320

[26] www.csie.ntu.edu.tw/~cjlin/libsvm/

[27] http ://plearn.berlios.de/

[28] http ://www.pymvpa.org/

[29] scienceline.org/_s/files/2008/11/fmri-copy.jpg

[30] mmp.kaist.ac.kr/ra_images/ML_img10.jpg

[31] dericbownds.net/uploaded_images/duped.jpg