

Fouille interactive de séquences d'images 3D d'IRMf

Jerzy Korczak*, Christian Scheiber**

Jean Hommet*, Nicolas Lachiche*

*LSIIT, CNRS, Illkirch

<jjk,hommet,lachiche>@lsiit.u-strasbg.fr

**CHU Lyon

christian.scheiber@chu-lyon.fr

Résumé. Du point de vue de la fouille de données, le cerveau est un objet complexe par excellence. La discrimination des voxels d'image du cerveau qui présentent une réelle activité est en général très difficile à cause d'un faible rapport signal sur bruit et de la présence d'artefacts. Les premiers tests des algorithmes actuels de fouille dans ce domaine ont montré que leurs performances et leurs qualités de reconnaissance sont faibles. Dans cet article, nous présentons une nouvelle approche interactive de fouille des images IRMf, guidée par les données, permettant l'observation du fonctionnement cérébral. Plusieurs algorithmes de classification non supervisés sont testés sur des séquences d'images 3D d'IRMf. Les résultats des tests présentent notamment les performances des classifieurs en fonction du nombre de classes, du rapport signal sur bruit des données et de la taille des zones activées par rapport au volume exploré.

1 Introduction

Les techniques modernes d'imagerie cérébrale, comme l'Imagerie par Résonance Magnétique fonctionnelle (IRMf), offrent la possibilité d'enregistrer en même temps l'activité de l'ensemble du cerveau. C'est une force, mais cela génère une masse de données considérable (environ 300000 voxels, "pixels tridimensionnels", pour lesquels on recueille entre 100 et 1000 observations). Du point de vue de la fouille de données, le cerveau est un objet complexe par excellence. En général, la discrimination des voxels d'image du cerveau qui présentent une réelle activité est très difficile à cause d'un faible rapport signal à bruit et de la présence d'artefacts. Les premiers tests des algorithmes actuels de fouille dans ce domaine ont montré que leurs performances et leurs qualités de reconnaissance sont faibles [Sommer et Wichert, 2003]. En raison de la difficulté qu'il y a à manipuler de telles quantités d'informations, l'essentiel des études ne cherchent pas à les explorer, mais s'en servent pour tester un modèle par le biais de statistiques univariées effectuées en chacun des points. C'est le principe de logiciels de traitement tels que Statistical Parametric Mapping (SPM) [Friston *et al.*, 1995], AFNI [Cox, 1996] ou BrainVoyager [Goebel, 1997] qui consistent à mettre en évidence les voxels plus actifs dans une condition par rapport à une autre.

Sur le plan international, un grand nombre de recherches méthodologiques sont en cours pour mettre en évidence les variations qui ont du sens. On peut regrouper celles-ci

en deux grandes familles. La plus commune est l'approche par les statistiques multivariées comme les MANCOVA, PCA, PLS, analyse canonique ou plus récemment l'ICA [Arfanakis *et al.*, 2000][Beckmann et Smith, 2003]. La seconde famille d'approche regroupe toutes les méthodes de fouille de données, partant des techniques de clustering jusqu'aux algorithmes génétiques en passant par les réseaux neuromimétiques [Sommer et Wichert, 2003].

Dans cet article, nous présentons une nouvelle approche interactive de fouille des images IRMf, guidée par les données, permettant l'observation du fonctionnement cérébral à partir de données issues de l'IRMf. L'originalité de notre approche tient au fait que nous n'allons pas seulement mettre en oeuvre des techniques de fouille de données encore non appliquées au domaine de l'imagerie fonctionnelle, mais surtout que celles-ci vont être étendues par l'injection de connaissance à priori dans les mécanismes de fouille du système et l'interactivité, intégrant directement l'expert-médecin dans le processus de découverte et d'apprentissage de concepts pour mettre en évidence les zones fonctionnelles du cerveau et leur organisation.

L'article est structuré de la façon suivante. Dans le chapitre 2 nous présentons les caractéristiques des données IRMf. Dans le chapitre 3 nous décrivons notre approche de fouille interactive d'images IRMf, ainsi que les algorithmes de classification choisis. Le chapitre 4 illustre des cas d'études effectués sur des données synthétiques et réelles. Nous concluons dans le dernier chapitre en dégagant quelques perspectives.

2 IRMf comme objet complexe

Mesure indirecte de l'activité du cerveau : L'activité cérébrale qui se caractérise par des phénomènes électriques rapides, peut être mesurée directement avec les techniques d'électro-encéphalographie (EEG) et de magnéto-encéphalographie (MEG). Si ces deux techniques ont une très bonne résolution temporelle, leur résolution spatiale par contre est quasi-nulle. Pour pouvoir savoir où et quand a lieu l'activité cérébrale, on a recours à l'Imagerie par Résonance Magnétique fonctionnelle (IRMf). Cette technique est une mesure indirecte car elle n'est pas sensible à l'activité même des neurones, mais à la consommation d'oxygène qui en dépend. L'effet BOLD (Blood Oxygenation Level Dependent) est le phénomène qui permet la mesure de cette consommation, il se traduit par une évolution du signal RMN induite à la fois par des variations de concentration d'oxygène et des variations de débit sanguin. L'IRMf consiste donc en une IRM optimisée sur l'effet BOLD. La mesure de l'activité d'un cerveau revient à enregistrer des images IRM à la suite les unes des autres pendant que ce cerveau travaille. Un tel enregistrement est donc composé d'une série d'images en trois dimensions d'un même cerveau immobile, et dont les variations d'intensité d'une image sur l'autre sont liées avec l'activité de ce dernier. Les zones actives sont ainsi localisées aussi bien spatialement que temporellement avec des résolutions typiques de 3 mm et 3 s. A ces échelles, le signal IRMf d'un voxel représente l'activité d'un agrégat de millions de neurones moyenné sur un intervalle de temps très grand devant le signal isolé d'un seul neurone.

L'IRMf permet de localiser précisément chez un sujet des zones liées à un fonctionnement ou à un dysfonctionnement donné. Un exemple d'utilisation de ces localisations est la planification d'opérations chirurgicales du cerveau qui permettent d'une

part d'éviter d'endommager des zones stratégiques et d'autre part d'inhiber des régions entraînant des troubles comme la maladie de Parkinson. En recherche, l'IRMf est utilisée pour comprendre le fonctionnement du cerveau au delà des aspects sensoriels en mettant en évidence des effets cognitifs subtils.

Le cerveau étant toujours en activité et le rapport signal sur bruit d'IRMf étant très faible, cette technique a recours systématiquement à un paradigme pour faire ressortir un effet escompté. Le paradigme est une représentation du fonctionnement de l'ensemble de l'acquisition. Intégrant la séquence de stimuli ainsi que les paramètres d'acquisition, le paradigme doit être optimisé de manière à maximiser les contrastes entre les différentes conditions du test afin de relever au mieux leurs effets.

Les données IRMf suivent principalement cinq grands axes de complexité. Elles sont à la fois des données en trois dimensions spatiales plus une dimension temporelle, elles peuvent correspondre à plusieurs sujets ou bien à l'historique d'un même patient, enfin elles peuvent être combinées avec des connaissances anatomiques ou plus généralement médicales. Dans cet article, nous nous focalisons sur les deux premiers aspects que sont les dimensions spatiales et temporelles, et dont la complexité est issue de la masse des données, du bruit d'acquisition et de l'analyse elle-même.

Masse de données : Typiquement, une acquisition IRMf génère une série de 100 à 1000 images IRM. Chaque image d'acquisition est en général constituée de 32 ou 64 coupes de 64 pixels de côté. Les voxels de l'image volumique sont codés le plus souvent sur 16 bits ce qui produit des fichiers de 256 ou 512 Ko. Les images normalisées qui permettent la comparaison entre plusieurs sujets ou l'utilisation d'atlas anatomiques sont plus volumineuses, le format NMI utilisé par SPM est un volume de $79 \times 95 \times 68$ voxels qui représente des fichiers de 1 Mo. Au final les séries IRMf complètes forment des volumes de données compris entre 25 Mo et 1 Go.

Bruits : Les acquisitions IRMf sont effectuées par un appareil IRM réglé dans un mode de fonctionnement appelé T2. Le mode T2 n'étant pas sensible qu'à l'effet BOLD, le signal efficace lié à l'activité du cerveau n'est pas prédominant et les sources de bruit s'en retrouvent favorisées. Ces bruits sont nombreux et très variés. On trouve tout d'abord les bruits physiques instrumentaux classiques comme par exemple des dérives ou des parasites électroniques qui sont fréquents dans les conditions instrumentales extrêmes de l'IRM où il s'agit de capter des signaux RF très faibles au sein de champs magnétiques très intenses. Autre source de bruit, les mouvements du sujet qui génèrent des signaux parasites, ce sont non seulement les déplacements de la tête et des yeux, mais aussi des déplacements internes induits par les pulsations cardiaques à proximité des veines. Enfin les activités cérébrales sans relation avec le paradigme comme des souvenirs ou des envies sont également générateurs de bruit.

Complexité propre à la technique IRMf : La non-linéarité des signaux observés en IRMf, dont les sources dominantes sont des processus vasculaires et neuronaux, limite la détection et l'interprétation des données. La prise en compte de ces phénomènes spécifiques a été abordé dans divers travaux de pré-traitements des séries [Wager, 2001]. Les sources de bruit qui sont nombreuses peuvent interférer avec l'acquisition, ce qui complique grandement leur prise en compte. Voici quelques exemples d'artefacts liés à la mesure, puis au paradigme. Lors de la mesure, il peut survenir des battements entre une fréquence d'échantillonnage et la fréquence de la respiration du sujet ou de ses

battements cardiaques. Les acquisitions avec entrelacement des coupes font apparaître des bruits une coupe sur deux. Le codage spatial des mesures RMN associé à des mouvements peut affecter les signaux à de mauvaises zones. Quant aux artefacts liés au paradigme, ils sont forcément plus gênants, car ils peuvent facilement être intégrés aux résultats. L'exemple le plus courant est un bruit de mouvement généré par les clics de souris utilisés par le sujet pour répondre aux tests, ceci produit un bruit synchrone avec le paradigme. Encore d'autres artefacts plus complexes peuvent provenir de l'habitude ou de la lassitude du sujet par rapport aux tests.

Analyse IRMf : Le protocole expérimental d'IRMf classique consiste à définir un paradigme du fonctionnement du cerveau pour une expérience donnée, c'est-à-dire une façon de représenter son déroulement. Cela consiste à définir dans le même temps, le programme des tâches à soumettre au cerveau, aussi bien qu'un modèle de la réponse hémodynamique pour chacune de ces tâches. Chacune de ces réponses programmées est paramétrée par des coefficients de pondération. Ainsi, une acquisition IRMf consiste à faire dérouler le programme des tâches au sujet, les données IRMf récupérées sont ensuite comparées au modèle proposé. Pour chaque voxel pris séparément, les coefficients du modèle sont ajustés pour faire correspondre au mieux le modèle hémodynamique proposé à la réponse mesurée de ce voxel. Ensuite, des tests statistiques permettent de faire sortir les voxels qui répondent le mieux au schéma proposé. Lorsque le schéma correspond à l'ensemble du modèle, les voxels sélectionnés sont ceux qui ont répondu au moins à une tâche du programme. On peut ensuite forcer les coefficients de certaines tâches à zéro pour pouvoir différencier leurs effets. Implémentée notamment dans le programme SPM, cette méthode statistique basée sur un modèle paramétré est en pratique l'unique technique d'analyse IRMf. Lorsque le paradigme est soigneusement posé avec un modèle bien adapté, cette méthode statistique est très performante et permet de détecter des réponses extrêmement faibles.

La méthode statistique est performante mais ne peut conclure en dehors du modèle préfixé. Avec cette méthode les résultats doivent être forcément anticipés, ce qui n'est pas toujours possible. Le concept de fouille de donnée, peut se révéler utile en complément ou en remplacement de la méthode classique lorsqu'il est délicat de prévoir ce qui va se passer durant l'acquisition. La démarche proposée ici est d'utiliser des méthodes de classification neuronales non supervisées au sein d'un système interactif pour faire ressortir les zones actives sans avoir recours à un modèle.

3 Approche guidée par les données

3.1 Vue d'ensemble du processus

La fouille interactive d'images IRMf n'est pas basée sur un modèle mais au contraire est une approche guidée par les données qui de plus intègre directement un expert dans le processus de découverte. Cette approche est composée de cinq phases que sont l'acquisition et la sélection des données, leur pré-traitement, la classification, l'extraction de règles et de concepts, enfin la validation. Les données IRMf sont issues de l'acquisition de 100 à 1000 images IRM séquentielles d'un cerveau, l'enregistrement ayant lieu alors que ce dernier effectue une série de tâches programmées. Sur un sujet parfaitement

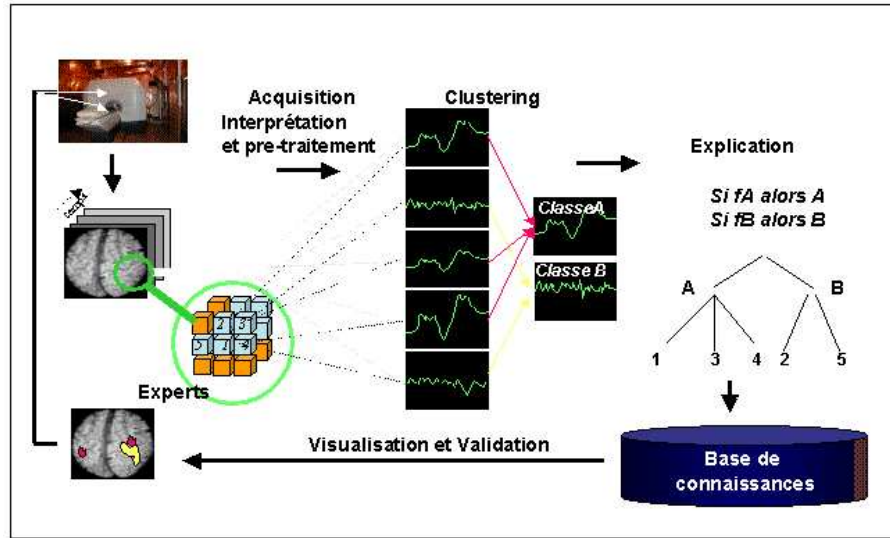


FIG. 1 – Schéma de fouille interactive d'IRMf

immobile, et en dehors de mouvements internes, les variations d'une image sur l'autre sont inhérentes à l'effet BOLD. L'activité cérébrale du cerveau en un point (voxel) se traduit par des variations d'intensité au cours du temps. En réalité le sujet n'est jamais complètement immobile, de plus, d'autres facteurs liés à l'acquisition viennent interférer avec les signaux d'intérêt. On a alors recours à des traitements spécifiques adaptés à chaque artefact en plus du pré-traitement des données en vue de la classification. La préparation des données consiste à choisir des attributs mettant en forme des vecteurs de manière à établir une relation de distance sur les données. Une acquisition de m images de taille n voxels, génère un ensemble de données de n individus. Un individu est un point dans un espace à m dimensions dont chaque coordonnée est l'écart de l'intensité du voxel considéré par rapport à sa moyenne sur la série. Au niveau de la phase de classification, on forme des classes composées d'un vecteur de référence dont la nature est la même que celle des individus qu'elle représente. En phase d'extraction, on génère des règles liées à l'acquisition, au sujet ou à un groupe de sujets.

La figure [Fig. 1] présente une vue simplifiée de fonctionnement du système de l'acquisition d'images IRMf jusqu'à l'interprétation et la validation par l'expert-médecin, ce dernier pouvant interagir à tous les niveaux. Les images seront récupérées après l'étape de pré-traitement. Le système devra permettre de faire une fouille de données dans des délais raisonnables pour pouvoir optimiser l'expérience et la renouveler aussitôt sur le sujet. La partie de droite montre un exemple d'interprétation de classes découvertes par un arbre de décision. En fonction des résultats obtenus, l'expert-médecin pourra modifier l'acquisition d'un point de vue purement technique par des paramètres géométriques ou temporels (résolution, zoom, etc). Cette réaction sera influencée par un retour d'information uniquement lié à l'état des signaux. Les algorithmes de classification et d'ex-

plication symbolique des classes aideront l'expert-médecin à comprendre les classes générées et à modifier l'expérience directement au niveau du paradigme. Ces modifications seront suscitées par les classifications de la fouille de données, mais seront choisies en définitive en fonction d'hypothèses sur le fonctionnement cérébral ou à partir de données connues comme des données anato-fonctionnelles préalablement saisies dans la base. Pour arriver à ce stade, le système intégré devra être très performant, en retournant des informations à un niveau d'interprétation suffisamment élevé. Le système sera capable d'apprendre et de stocker des connaissances acquises pendant les expérimentations et de s'en resservir comme connaissances préalables dans de futurs examens. Cette imagerie fonctionnelle interactive demande une analyse très rapide de résultats préliminaires pendant l'acquisition, cette analyse n'est envisageable que par une méthode spécifique qui devra s'insérer dans la démarche actuelle.

Le travail présenté dans cet article est centré essentiellement sur la phase de classification, dont la démarche se veut interactive et dynamique autour d'algorithmes de classification non supervisée. Une telle classification est un moyen d'analyser de grands ensembles de données. Elle consiste à décrire des données en les distribuant en un nombre limité de classes homogènes. Ces classes regroupent les éléments qui ont des caractéristiques et comportements similaires. Le terme classification est à distinguer de celui de classement (en anglais *classification*) avec lequel il y a parfois confusion. La classification consiste à partager les données en formant des classes alors que le classement se réduit à ranger les données dans des classes existantes. Non supervisée, la classification est dite "dirigée par les données" car le processus n'utilise pas de connaissance externe aux données et ne dispose que de leur description pour extraire des informations sur la structure de l'ensemble. Quelle que soit la formule adoptée, la classification se résume à regrouper n individus en k classes et se déroule suivant trois phases, initialisation des données et des paramètres de l'algorithme, l'algorithme lui-même et la validation des résultats. Chaque classe joue à la fois le rôle d'ensemble d'individus et de représentant de ces mêmes individus. Pour des données numériques, un individu peut être considéré comme un point dans l'espace multidimensionnel défini par ces attributs et être ainsi comparé à d'autres individus ou groupes d'individus à partir de distances de Manhattan ou de distances euclidiennes.

Les méthodes de classification non supervisée ont déjà été appliquées à l'IRMf à travers plusieurs méthodes dérivées de statistique, de logique floue ou encore de classification neuronale et hiérarchique. Les méthodes statistiques ayant fait l'objet de travaux sont l'algorithme des centres mobiles (K-means) [MacQueen, 1967] [Bock et Diday, 2000] [Dimitriadou *et al.*, 2004] [Goutte *et al.*, 1999] [Moller *et al.*, 2001], l'Analyse en Composante Principale (ACP) [Andersen *et al.*, 1999], [Lai et Fang, 1999] et l'Analyse en Composantes Indépendantes (ACI) [Esposito *et al.*, 2002]. L'algorithme des centres mobiles ou K-means est une référence dans le domaine de la classification de part ses performances respectables en regard de son extrême simplicité. L'ACP est une méthode statistique utilisée pour réduire l'information d'un ensemble de données vectorielles sur les axes orthogonaux principaux d'inertie ou composantes principales. Elle consiste à projeter les données dans un nouvel espace de dimension réduite. L'ACI recherche une transformation linéaire des vecteurs qui minimise la dépendance statistique entre ses composantes. L'ACI est une forme d'extension de l'ACP qui s'affranchit

de la contrainte liée à l'orthogonalité des composantes. L'intérêt de la méthode ACI réside dans le fait qu'elle permet de séparer les sources indépendantes contenues dans les signaux mesurés. En IRMf cette approche permet de sortir un signal fonctionnel masqué par divers signaux de bruits d'acquisition. ACP et ACI peuvent servir d'étape de pré-traitement.

La classification floue [Fadili *et al.*, 2000] [Barth *et al.*, 2001] est très voisine des centres mobiles, à la différence qu'un individu n'appartient pas à une classe unique, mais est partagé aux k classes en même temps avec une probabilité sur chaque classe. Un individu possède k valeurs comprises entre 0 et 1 qui caractérisent le degré d'appartenance à chaque classe. L'intérêt de la classification floue est qu'elle prend en compte les situations ambiguës situées aux frontières et intersections entre classes, ce qui limite les erreurs et augmente la stabilité du résultat.

Les classifications hiérarchiques [Baumgartner *et al.*, 2001] [Filzmoser *et al.*, 1999] produisent des classes qui sont organisées sous forme d'arbre et dont le nombre n'est pas forcément fixé à l'avance. La classification peut être effectuée soit par division soit par agrégation des ensembles d'individus. Les distances des individus n'interviennent qu'indirectement dans le partitionnement déterminé au niveau des classes à partir de critères comme l'inertie interclasse. L'avantage de telles méthodes consiste en un niveau supérieur d'interprétation, mais seulement au prix de temps de traitement ou d'espace mémoire très limitants.

Les méthodes non-supervisées de fouille de données de grande taille ne sont pas encore très présentes dans le domaine d'analyse des IRM fonctionnelles. Dans ce article, nous nous sommes plus particulièrement focalisés sur les méthodes neuronales en comparaison avec des méthodes non hiérarchiques classiques comme Lloyd Généralisé (LBG) et K-means.

L'algorithme de Kohonen [Kohonen, 1982] [Fischer et Hennig, 1999] [Ngan et Hu, 1999] est l'algorithme le plus connu de la famille des algorithmes neuro-naux, dont les classes sont interconnectées sous forme de réseau et dénommées neurones. L'algorithme de Kohonen s'appuie sur un réseau ordonné de neurones dont les dimensions sont fixées. Cette topologie, lorsqu'il s'agit d'une grille 2D a l'intérêt de rendre les post-traitements et l'affichage plus pratique grâce à l'ordonnancement des classes.

La présente méthode s'appuie principalement sur l'algorithme des Gaz Neuronaux Croissants (GNC) [Fritzke, 1994] [Fritzke, 1995] un algorithme neuronal non encore appliqué à l'IRMf. Son originalité réside dans le fait que le nombre de classes n'est pas fixé à l'avance contrairement à la plupart des autres méthodes. Le nombre de neurones peut tout aussi bien augmenter que diminuer durant l'exécution, rendant l'algorithme particulièrement flexible. Ces particularités font que cet algorithme est prédisposé à notre approche. Nous rendons la procédure de classification interactive, de manière à engager l'expert dans le processus de découverte. L'utilisateur dispose de retours et de contrôles sur l'évolution de la classification, pouvant ainsi suivre et intervenir pour orienter le déroulement. Cette fouille visuelle en IRMf suppose des outils de vision 3D, des retours statistiques et un algorithme de classification suffisamment souple et dynamique pour supporter l'interaction [Fig. 2].

Dans ce travail, nous avons adapté l'algorithme GNC déjà très dynamique à l'IRMf et à l'interactivité. De plus, cet algorithme qui n'avait pas encore été appliqué à l'IRMf,

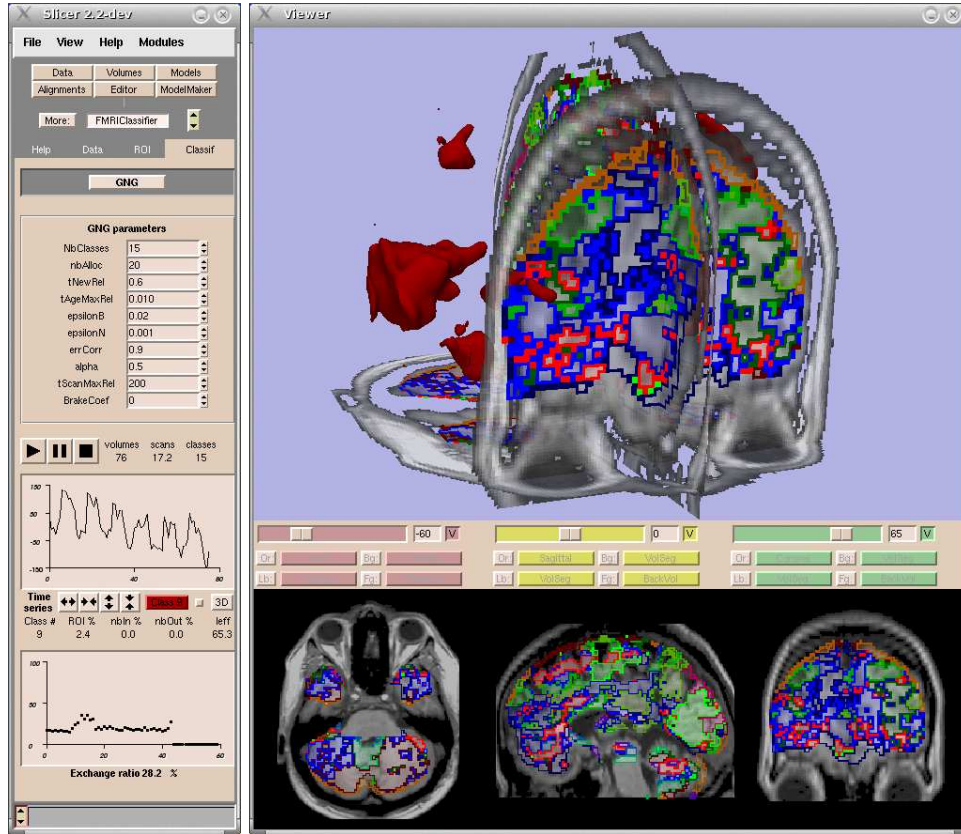


FIG. 2 – Module dédié à la classification IRMf dans le programme 3DSlicer

a été testé sur des données synthétiques et réelles en comparaison avec des algorithmes non-supervisés classiques.

A la suite des méthodes neuronales, nous envisageons de poursuivre nos travaux vers d'autres types d'algorithmes qui apportent des connaissances conceptuelles et structurales tels que COBWEB et vers des techniques plus récentes de fouille de données de grandes taille telles que CURE [Guha *et al.*, 1998] ou BIRCH [Zhang *et al.*, 1996]. Ces techniques sont plus efficaces pour appréhender des classes possédant des formes non sphériques à variance importante, et de plus permettent de réduire le volume des traitements à l'aide d'un échantillonnage des données par agrégation hiérarchique.

Les classes produites par les techniques d'apprentissage non-supervisé sont évaluées par l'expert qui ne retient que les classes pertinentes. Chaque classe correspond à un ensemble de voxels, régions du cerveau, ayant la même réponse hémodynamique au cours du temps. Cette réponse peut être caractérisée explicitement par la construction de règles. Ces règles combinent des motifs temporels observés avec des informations spatiales telles que l'activité des voxels ou des régions voisines, en plus des connaissances

du domaine, comme l’atlas des fonctions des régions du cerveau. En travaillant sur des données normalisées, c’est-à-dire quand les images sont recadrées pour correspondre à un cerveau type, les voxels de plusieurs séries d’images, d’une même personne ou de plusieurs personnes, de même position correspondent à la même région du cerveau. Des règles peuvent être apprises pour caractériser le fonctionnement du cerveau d’une personne en particulier, ou de plusieurs personnes, suivant l’objectif de l’expert. Les motifs temporels peuvent être synchronisés avec le paradigme, par exemple pour découvrir l’interaction entre les régions du cerveau utilisées pour la mémoire visuelle. D’autres motifs temporels peuvent être indépendants de tout paradigme, par exemple pour mettre en évidence la succession d’activations de régions du cerveau typiques de l’hallucination. Les techniques de fouilles de données, par exemple [Agrawal *et al.*, 1996][Zaki, 2001], doivent être adaptées aux données d’IRMf, en particulier pour utiliser les informations spatiales et les connaissances médicales.

3.2 Définitions et concepts de base des classifications

Dans ce chapitre nous détaillons particulièrement l’algorithme GNC, les autres algorithmes sont reportés en annexe.

Les algorithmes utilisés ont un certain nombre de propriétés en commun puisqu’ils doivent tous partager un ensemble de n individus en k classes. L’ensemble des classes évolue par l’adaptation d’une ou plusieurs classes à un individu tiré aléatoirement de l’ensemble des données et on peut ainsi assimiler le temps aux tirages successifs des individus.

Cas général :

L’ensemble D des données est constitué de n individus de dimension m

$$D = \{\xi_1, \xi_2, \dots, \xi_n\}, \quad \xi_i \in R^m \quad (1)$$

L’ensemble des classes C est constitué de k classes

$$C = \{c_1, c_2, \dots, c_k\} \quad (2)$$

Chaque classe c est associée à un *vecteur de référence* w_c de dimension m qui indique la position de son centre

$$w_c \in R^m \quad (3)$$

Pour un individu donné ξ , le *gagnant* $g(\xi)$ est la classe dont le vecteur de référence est le plus proche de ξ , $\|\xi - w_c\|$ est la distance de Manhattan entre les deux vecteurs

$$g(\xi) = \arg \min_{c \in C} \|\xi - w_c\| \quad (4)$$

R_c est l’ensemble des individus qui ont la classe c comme gagnante

$$g(\xi) = c \Rightarrow \xi \in R_c \quad (5)$$

Dans le cadre des approches de neuronales (classe = neurone), il existe un ensemble K de connexions symétriques

$$(i, j) \in K \iff (j, i) \in K, \quad K \in C \times C \quad (6)$$

N_c est l'ensemble des voisins topologiques directs de la classe (neurone) c

$$N_c = \{i \in C \mid (c, i) \in K\} \quad (7)$$

Cas partiulier de l'IRMf :

Soit I_{moy}^a , l'intensité moyenne du voxel a d'une série de m volumes

$$I_{moy}^a = \frac{1}{m} \cdot \sum_{i=1}^m I_i^a \quad (8)$$

ξ_a est le signal IRMf de ce voxel défini par ses variations d'intensité

$$\xi_a = \{\delta_1^a, \delta_2^a, \dots, \delta_m^a\} \text{ avec } \delta_i^a = I_{moy}^a - I_i^a \quad (9)$$

d_{Man} est la distance de Manhattan entre deux signaux ξ_a et ξ_b

$$d_{Man} = \sum_{i=1}^m |\delta_i^a - \delta_i^b| \quad (10)$$

$\Delta\delta_i^a$ est la variation d'une composante du signal IRMf ξ_a lors d'un déplacement relatif d'une valeur ϵ de ξ_a vers ξ_b

$$\Delta\delta_i^a = \epsilon \cdot (\delta_i^b - \delta_i^a) \quad (11)$$

Ces définitions communes sont utilisées dans les spécifications des algorithmes de classification présentées dans le paragraphe suivant ainsi que dans l'annexe.

3.3 Algorithme des Gaz Neuronaux Croissants

Dans cet article, nous nous concentrons sur l'étude et l'adaptation de l'algorithme des Gaz Neuronaux Croissants [Fritzke, 1994] [Fritzke, 1995] pour l'IRMf. Cet algorithme diffère des méthodes, comme K-means, SOM de Kohonen et LBG, décrites en annexe, par le fait que le nombre de classes change durant son déroulement. Il s'agit d'un algorithme dynamique de réseau de neurones inter-connectés dans lequel des classes (neurones) peuvent disparaître ou apparaître. Les connexions entre classes ont la propriété de vieillir et disparaissent lorsqu'elles atteignent un âge maximum prédéfini. Cette propriété est la cause de la disparition des classes qui sont éliminées lorsqu'elles ne sont plus connectées. L'apparition des classes se fait à intervalles de

temps régulier en insérant une nouvelle classe auprès de la classe qui présente la plus grande erreur. L'erreur d'une classe est une évaluation basée sur la somme des distances de cette classe aux individus qui l'ont déclaré gagnante. Ainsi le réseau se renforce auprès des classes qui parcourent les plus grandes distances ou qui réunissent le plus d'individus. Nos contributions portent sur deux aspects de classification, notamment sur la réduction de l'espace des données et une aide au paramétrage de l'algorithme assisté par des outils statistiques et de fouille de données visuelle.

L'algorithme des Gaz Neuraux Croissants adapté aux données IRMf est présenté ci-après.

Algorithme GNC

Définitions des variables :

- nD : taille de l'ensemble de données explorées
- k et k_{max} : nombre de neurones courant et nombre maximum de neurones autorisés
- a_{max} et ar_{max} : âge maximum d'une connexion et age maximum relatif à nD et k
- ϵ_g et ϵ_v : coefficients d'adaptation du gagnant et de ses voisins
- t_n et tr_n : délai de création d'une nouvelle classe et délai relatif à la nD
- t et t_{max} : temps d'exécution courant et durée maximale de l'algorithme exprimés en nombre de données tirées
- α : coefficient de répartition des erreurs
- β : coefficient de régulation des erreurs
- $a_{(c_1, c_2)}$: age d'une connexion entre deux classes c_1 et c_2
- e_a : erreur d'une classe a
- $stop$: booléen d'arrêt de l'algorithme

1. Choix des paramètres de l'algorithme :

- (a) nombre maximum de neurones k_{max}
- (b) age maximum relatif ar_{max}
- (c) délai relatif de création d'un nouveau neurone tr_n
- (d) coefficients d'adaptation ϵ_g et ϵ_v
- (e) coefficient de régulation des erreurs β
- (f) durée maximale d'exécution t_{max}

2. Initialisation des variables :

- (a) $stop = faux$
- (b) $t = 0$
- (c) $a_{max} = ar_{max} \cdot nD \cdot k$
- (d) $t_n = tr_n \cdot nD$
- (e) Tirage aléatoire de deux individus différents de l'ensemble des données pris comme vecteurs de référence des deux classes d'initialisation
- (f) $k = 2$

3. Tirage aléatoire d'un individu ξ de l'ensemble des données et incrémentation de t .

4. Recherche de $g_1(\xi)$ et $g_2(\xi)$, les classes gagnantes de ξ , c'est-à-dire les deux classes pour lesquelles le vecteur de référence est le vecteur le plus proche de ξ [Equ. 4].

5. Connexion des classes $g_1(\xi)$ et $g_2(\xi)$, remise à zéro de l'age de la connexion $a_{(g_1, g_2)} = 0$

6. Augmentation de l'erreur de la classe gagnante : $e_{g_1} = e_{g_1} + \|\xi - w_{g_1}\|^2$
7. Adaptation de la classe gagnante : $w_{g_1} = w_{g_1} + \epsilon_g(t) \cdot (\xi - w_{g_1})$
8. Adaptation des voisines de la classe gagnante : $w_c = w_c + \epsilon_v \cdot (\xi - w_c) \quad (\forall c \in N_{g_1})$
9. Incrémentation de l'âge des connexions de la gagnante : $a_{(g_1,c)} = a_{(g_1,c)} + 1 \quad (\forall c \in N_{g_1})$
10. Destruction de toute connexion dont l'âge est supérieur à a_{max}
11. Pour chaque classe n'ayant plus de connexion et Si $k > 2$:
 - (a) destruction de cette classe
 - (b) décrémentation de k
 - (c) $a_{max} = ar_{max} \cdot nD \cdot k$
12. Si (t est multiple t_n) et ($k < k_{max}$)
 - (a) détermination de la classe p_1 qui accumule la plus grande erreur
 - (b) détermination de p_2 , voisine de p_1 qui accumule la plus grande erreur
 - (c) création d'une nouvelle classe r entre p_1 et p_2 telle que $w_r = (w_{p_1} + w_{p_2})/2$
 - (d) destruction de la connexion entre p_1 et p_2
 - (e) connexion de r et p_1
 - (f) connexion de r et p_2
 - (g) initialisation de l'erreur de r : $e_r = \alpha \cdot (e_{p_1} + e_{p_2})$
 - (h) diminution de l'erreur de p_1 : $e_{p_1} = e_{p_1} - \alpha \cdot e_{p_1}$
 - (i) diminution de l'erreur de p_2 : $e_{p_2} = e_{p_2} - \alpha \cdot e_{p_2}$
 - (j) incrémentation de k
 - (k) $a_{max} = ar_{max} \cdot nD \cdot k$
13. Diminution des erreurs de toutes les classes par le facteur β
14. Intervention de l'utilisateur, cas d'interruptions :
 - (a) relevé de données pour la visualisation
 - (b) relevé de mesures statistiques
 - (c) modification de l'un des paramètres de l'algorithme
 - (d) augmentation ou réduction du volume de données
 - (e) suppression d'un neurone
 - (f) sauvegarde de l'état courant
 - (g) restauration d'un état précédent
 - (h) décision d'arrêt : $stop = vrai$
15. Si ($t < t_{max}$) ou (! $stop$) retour en 3
16. Fin de l'algorithme

Dans cet algorithme, des classes peuvent apparaître ou disparaître au cours du temps sous l'influence d'erreurs évaluées. Ceci donne à GNC une souplesse qui lui permet de s'adapter à une démarche interactive, dans laquelle l'évolution de la classification est également influencée par un expert.

Pour ce faire, nous utilisons un intervalle de pause au pas 14 de l'algorithme durant lequel GNC est dans un état cohérent. Ceci permet d'effectuer des mesures sur l'état

en question, de le modifier, de le sauver ou encore de le restaurer. Cette pause est ainsi invoquée systématiquement à intervalles de temps réguliers pour effectuer des mesures absolues sur la classification et alimenter les outils de visualisation. Elle est aussi utilisée à la demande lorsque l'expert intervient sur le cours de la classification. Ce dernier a donc accès et peut modifier tout point de la classification durant l'exécution. Tout peut être modifié sans avoir besoin d'arrêter et de relancer le processus. Les paramètres de l'algorithme ainsi que l'espace des données peuvent donc être optimisés et varier durant la classification.

Comme les effets des paramètres a_{max} et t_n sont fortement dépendants de la taille de l'ensemble des données et du nombre de neurones, nous avons également adapté ces deux paramètres originaux en leur adjoignant respectivement les paramètres ar_{max} et tr_n , avec

$$a_{max} = ar_{max} \cdot nD \cdot k \quad (12)$$

$$t_n = tr_n \cdot nD \quad (13)$$

A chaque modification de nD , a_{max} et t_n sont recalculés, a_{max} l'est également pour toute variation de k . Ainsi ar_{max} et tr_n font office d'interface pour le paramétrage de a_{max} et t_n , ce qui permet de se rendre indépendant des variations du nombre de neurones et de la taille des données. Les réglages restent ainsi relativement constants lorsque l'on intervient dans le déroulement de la classification mais aussi lorsque l'on change de jeu de données.

En ce qui concerne le nombre de neurones, nous avons encore ajouté un paramètre de limitation k_{max} , sans quoi dans la plupart des cas, le nombre de neurones ne cesserait d'augmenter.

3.4 Caractéristiques des algorithmes

Les principaux aspects qui caractérisent les algorithmes de classification se retrouvent au niveau du choix du nombre de classes, de l'adaptation des classes, de l'échelle d'itération de ces adaptations, de la prise en compte d'un voisinage entre classes et de ses possibilités d'interactions.

Le choix du nombre de classes est un facteur déterminant pour une classification. La plupart des algorithmes imposent de fixer ce nombre avant de débiter sans retour possible sur le choix, d'autres s'ils ne l'imposent pas sont souvent contraints de le limiter.

L'adaptation des classes aux données se fait en général par itération à l'une des échelles suivantes, l'échelle de l'ensemble des données ou l'échelle de la donnée. Dans le premier cas, toutes les classes sont recalculées en même temps après chaque passage de l'ensemble des données. Dans l'autre, une ou plusieurs classes sont modifiées après chaque tirage d'une nouvelle donnée. Le deuxième type d'algorithme dont la granularité est beaucoup plus fine, donne plus de souplesse dans son utilisation.

Pour les algorithmes neuronaux, le concept de connexion entre classes, permet de prendre en compte le voisinage des classes lors de leurs adaptations. La modification des classes se fait itérativement à l'échelle de la donnée, on adapte la classe la plus proche ainsi que son voisinage à la dernière donnée tirée, le voisinage étant formé d'une ou de plusieurs classes connectées à la classe en question. En modifiant de la sorte plusieurs

classes simultanément, on accélère le processus et l'on donne une cohésion à l'ensemble des classes. De plus, les adaptations sont réglables par des coefficients qui prennent en compte le degré de voisinage. La contre partie de cette organisation en réseau de classes est qu'elle peut générer des classes vides.

L'interaction avec l'algorithme de classification peut être réalisée au moyen d'un intervalle de pause comme décrit dans le paragraphe précédent sur GNC. Quatre niveaux d'interaction peuvent être discernés. Le niveau minimum correspond à un retour de mesures sur l'état courant et permet à l'utilisateur de pouvoir décider quand interrompre le processus. Le deuxième niveau est la modification du volume des données, ce niveau est rendu possible lorsque l'adaptation des classes est indépendante de l'historique de la classification. Le troisième niveau est accessible aux algorithmes qui disposent de paramètres non nécessairement fixés. Dans ces conditions l'utilisateur peut les faire varier pour les optimiser en cours d'exécution. Au delà de ces trois niveaux d'interaction, une gestion de sauvegarde et de restauration des états de la classification constitue un dernier aspect d'interaction avec la classification.

Les algorithmes utilisés dans ce travail illustrent tous les aspects que l'on vient de décrire. K-means tout d'abord est un algorithme itératif à l'échelle de la donnée. Il ne prend pas en compte le voisinage, une seule classe est adaptée à la fois. Le nombre de classes à fixer au départ étant son unique paramètre et l'adaptation de ses classes étant dépendante de l'historique du processus, les possibilités d'interagir avec cet algorithme sont minimales. L'intérêt de K-means repose sur sa simplicité et son principal inconvénient est sa convergence trop rapide, ce qui le rend très dépendant de son initialisation et donc peu reproductible. LBG est une sorte de K-means dont la convergence a été prolongée pour accroître sa reproductibilité. Son itération est réalisée à l'échelle de l'ensemble des données ce qui dans un sens limite sa souplesse mais d'un autre côté lui permet de s'affranchir de l'historique de la classification, LBG a ainsi la possibilité d'interagir au niveau des données. SOM est un algorithme neuronal, c'est-à-dire qu'à chaque tirage d'une donnée plusieurs classes sont modifiées. Le nombre de classes doit être fixé au départ pour former une grille 2D ordonnée. L'initialisation de la grille est prépondérante pour la suite de la classification et doit faire l'objet d'un traitement particulier. L'intérêt de cette grille réside dans des facilités d'affichage des classes ainsi qu'en leurs regroupements, par contre elle génère souvent des classes vides. Les coefficients d'adaptation sont réglables de manière interactive mais le format de la grille est une contrainte importante au niveau de la gestion des états de classification. GNC moins connu en imagerie médicale, est aussi un algorithme neuronal. Dans cette variante, le nombre de neurones ainsi que le nombre de connexions sont variables durant l'apprentissage. Cet algorithme est plus sophistiqué que SOM et utilise plus de paramètres. Ses inconvénients sont liés à sa complexité car mal réglé il génère des résultats inconsistants. En revanche, lorsque le paramétrage est correct, cet algorithme est très reproductible et peut isoler des classes de très faible cardinalité. Tous les paramètres peuvent être modifiés en cours d'exécution y compris le paramètre de nombre de classes. Cet aspect dynamique, déterminant pour la fouille interactive, en fait une technique de choix.

3.5 Fouille de données visuelle

Les quatre algorithmes ont tous des capacités d'interaction avec l'utilisateur, mais leurs possibilités sont très inégales. Le meilleur des cas est GNC où toutes les interactions sont permises. Dans ce paragraphe nous présentons quelles sont ces possibilités et comment l'expert-médecin peut orienter la classification des séquences IRMf.

L'utilisateur suit l'évolution de la classification au cours du temps à partir d'outils visuels alimentés à intervalles de temps réguliers par l'algorithme. Les informations fournies sont de trois ordres : statistiques, temporelles et spatiales. Les statistiques portent sur la classification en tant que telle, quant aux informations temporelles et spatiales, elles sont liées à la nature de l'IRMf.

Les informations statistiques décrivent l'évolution de la classification par des mesures sur les classes générées. L'utilisateur peut suivre à partir de graphiques, l'évolution des erreurs, des inerties intra-classe et inter-classes, du nombre de données réunies au sein de chaque classe et du nombre de données qui changent de classe par unité de temps. L'ensemble de ces informations renseigne notamment sur la dispersion des classes, la convergence et la stabilité de la classification.

Les informations temporelles et spatiales véhiculées dans chaque classe, représentent l'évolution des signaux IRMf au cours du temps et leurs localisations dans le cerveau. Les signaux sont affichés sous forme de courbes dans des graphiques où l'on peut également faire figurer le paradigme, soit par la séquence de stimuli, soit par un modèle de la réponse lorsqu'il existe. La visualisation spatiale est effectuée simultanément de deux manières, à partir de trois coupes perpendiculaires et directement en trois dimensions. La classification en cours alimente un volume où les couleurs des voxels sont associées aux classes. Cette vision 3D de la classification peut être superposée suivant les cas, à un des volumes de la série d'acquisition, à une IRM structurelle du sujet ou à un volume normalisé faisant office d'atlas anatomique.

Toutes ces informations sont rafraîchies régulièrement durant la classification permettant à l'utilisateur de suivre en direct l'évolution du processus. Il opère ainsi une visualisation active, en affichant les aspects qu'il juge opportuns et en intervenant sur le cours de la classification. Les leviers dont il dispose pour influencer le processus sont les paramètres de l'algorithme et le volume des données qu'il peut modifier quand il le désire.

L'utilisateur peut facilement optimiser GNC, par l'observation en direct des mesures statistiques et de leurs variations induites par des modifications de paramétrage. Ainsi, le réglage de l'algorithme peut être convenablement effectué sans avoir besoin d'arrêter et de recommencer sans cesse. D'autre part, les paramètres peuvent être adaptés en fin de classification pour stabiliser ou affiner un résultat. Sachant que le nombre de paramètres de GNC est assez élevé et qu'il n'est pas toujours facile de prédire leurs effets, cette facilité de réglage est particulièrement appréciable.

Dans la découverte de régions actives du cerveau, les moyens d'intervention de l'expert-médecin se situent au niveau du réglage du nombre de classes nécessaires et de la sélection des régions explorées.

Le nombre de classes k du réseau de neurones GNC a tendance à augmenter jusqu'au nombre de classes maximum paramétré k_{max} . Si la croissance du réseau est graduelle, en revanche, la réduction engendrée par une baisse de l'âge maximum des connexions

a_{max} est assez brutale. C'est pourquoi, le choix du nombre de classes est effectué par sa limite supérieure qu'il est toujours plus facile d'augmenter que de diminuer. Ainsi, l'augmentation du nombre de classe se fait naturellement, et sa réduction peut être réalisée directement par l'utilisateur qui a la possibilité d'éliminer des classes. Notons au passage que la suppression d'une classe doit être accompagnée par le système d'une gestion appropriée des connexions afin de maintenir la cohérence du réseau. En définitive pour adapter au mieux le nombre de classes aux données IRMf, l'expert-médecin fixe la valeur de k_{max} en observant l'apparition des nouvelles classes durant la croissance du réseau. Il peut par exemple décider de stopper cette croissance après l'apparition d'une classe escomptée ou bien revenir en arrière par l'élimination d'une classe de son choix.

Pour augmenter la vitesse d'exécution et diminuer la complexité des résultats, l'utilisateur a toujours intérêt à limiter au maximum le volume des données. L'espace d'exploration peut être ajusté au moyen de plusieurs techniques. Un outil inclus dans la visualisation 3D, permet de sélectionner des volumes d'intérêt ou de désintérêt. L'espace peut aussi être restreint à des structures anatomiques. Dans tous les cas un seuillage permet d'éliminer l'espace qui entoure le crâne. Et dans le cas d'images normalisées, il est possible de limiter les recherches à la matière grise. Les résultats intermédiaires de classification sont aussi utilisés pour la sélection des données. Ainsi, l'expert-médecin peut poursuivre une classification en focalisant son intérêt sur certaines classes après avoir éliminé l'ensemble des voxels des autres classes. L'augmentation de l'espace de recherche, également possible, est parfois désiré. Le cas se produit lorsque l'on veut étendre une classification à d'autres zones, par exemple après avoir optimisé certains paramètres de l'algorithme sur une région volontairement réduite pour gagner du temps.

4 Etude de cas

Les algorithmes ont été appliqués à l'IRMf sur des données synthétiques et réelles. Le jeu artificiel a été élaboré de manière à pouvoir évaluer et comparer les différents algorithmes en contrôlant précisément les paramètres des données. Le cas réel est une acquisition IRMf de tests auditifs.

4.1 Données synthétiques

Les données synthétiques sont un assemblage de deux parties, les activations simulées purement artificielles et le fond constitué de données réelles. La série d'IRM du cerveau utilisée comme fond a été composée à partir d'images issues du jeu de données réelles décrit par après. Les images sélectionnées coïncident avec une des conditions du test auditif qui en compte deux, "silence" et "parole". Toutes les images de la condition "silence" ont été rassemblées et enchaînées pour former le fond de notre jeu de test qui se rapproche ainsi au mieux de données réelles. Sur cette série de 40 images, viennent s'ajouter les activations synthétiques formées par des séries temporelles en créneau (signal carré) simulant un paradigme de type bloc. Le lieu des activations est un volume cubique de 5 voxels de côté. Le niveau de bruit moyen de ces 125 voxels du fond a été mesuré en prenant le double de la variance des intensités de ces voxels dans le temps.

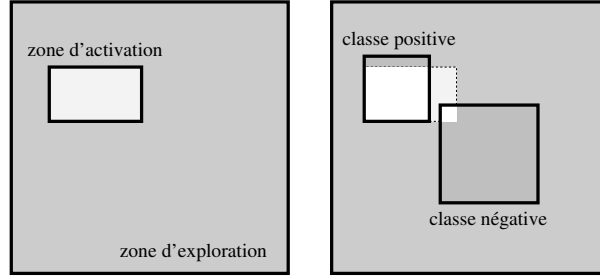


FIG. 3 – Schéma simplifié des classes d'activation positives et négatives

A partir de cette mesure, il est ensuite possible de contrôler le rapport signal sur bruit de nos activations, en ajoutant des créneaux d'intensités voulues aux voxels considérés. On dispose, avec ces données, d'un jeu de tests dont les paramètres sont les suivants :

- *algorithme* : GNC, SOM, LBG ou K-means,
- k : nombre de classes,
- s/b : rapport signal sur bruit, ajustable de manière continue,
- n_{ZA}/n_{ZE} : rapport de dilution de la zone d'activation dans la zone d'exploration.

Les données synthétiques permettent l'automatisation des tests qui sont ainsi répétés chacun 50 fois. La mesure des tests est basée sur une matrice de confusion indiquant notamment les ratios de vrai-positifs. Pour créer cette matrice, il faut déterminer la nature positive ou non des classes. La figure [Fig. 3] illustre de manière simplifiée comment sont déclarées les classes positives en fonction de leur recouvrement d'une zone d'activation. Ainsi, les classes ne possédant pas d'individus dans la zone d'activation sont toutes déclarées négatives, et l'on utilise le critère arbitraire indéfini par l'équation [Equ. 14] pour déterminer le caractère des autres.

$$C_a \text{ est positive} \iff \frac{inC_a}{outC_a} > 2 \cdot \frac{n_{ZA} \cdot (k-1)}{n_{ZE} - n_{ZA}} \quad (14)$$

avec :

C_a	classe dont le caractère est à déterminer,
ZE	zone d'exploration,
ZA	zone d'activation,
n_{ZE}	nombre de voxels de ZE ,
n_{ZA}	nombre de voxels de ZA ,
k	nombre total de classes,
$inC_a, outC_a$	nombre de voxels de C_a à l'intérieur et à l'extérieur de ZA .

Avec ce critère, le taux de vrai-positifs qui résulte des classifications bascule rapidement vers les extrêmes que sont aucune zone d'activation détectée et toute la zone détectée. La moyenne des 50 essais constitue le résultat définitif d'un test et renseigne sur la fréquence de détection de la zone activée.

s/b	GNC	SOM	LBG	K-means
1.0	0	0	0	6
1.1	0	0	5	1
1.2	55	0	0	15
1.3	95	0	15	35
1.4	100	0	45	15
1.5	100	40	30	40
1.6	100	70	55	44
1.7	100	85	40	70
1.8	100	100	65	50
1.9	100	100	75	55
2.0	100	100	69	70
3.0	100	100	100	82
4.0	100	100	100	90

TAB. 1 – Fréquence de détection (%) en fonction du rapport signal sur bruit s/b .

n_{ZA}/n_{ZE}	GNC	SOM	LBG	K-means
1 :343	0	0	10	10
1 :216	85	15	34	35
1 :125	100	40	30	40
1 :64	100	100	95	45
1 :27	100	100	100	95
1 :8	100	100	100	100

TAB. 2 – Fréquence de détection (%) en fonction du rapport n_{ZA}/n_{ZE} .

Les tableaux [Tab. 1], [Tab. 2] et [Tab. 3] présentent des résultats des classifications sur les données de tests en faisant varier respectivement les paramètres s/b , n_{ZA}/n_{ZE} et k en fonction de *algorithme*. Les trois séries de données de test sont construites à partir d'un signal d'activation carré dont la période est de quatre volumes et dont les paramètres lorsqu'ils sont fixes, sont $s/b = 1.5$, $n_{ZA}/n_{ZE} = 1/125$ et $k = 9$. Les résultats sont exprimés en pourcentage de détection de la zone d'activation. Il apparaît tout de suite que les quatre algorithmes sont classés par les trois expériences, dans le même ordre. Quel que soit le rapport signal sur bruit, le nombre de classes imposé et le rapport de dilution de la zone à détecter dans la zone explorée, la meilleure détection est assurée par GNC suivi par SOM, LBG et enfin K-means. Que K-means soit dernier n'est pas surprenant, puisque cette version ne remet pas complètement en cause la position de ses classes contrairement à LBG. Ainsi, K-means et LBG dans une moindre mesure sont fortement dépendants de leurs initialisations. Mais ces deux algorithmes ne sont pas dénués d'intérêt dans la mesure où ils n'ont aucun paramètre à déterminer en dehors du nombre de classes. Cette simplicité leur permet d'obtenir souvent de meilleurs résultats que GNC et SOM lorsque le paramétrage de ces derniers

k	GNC	SOM	LBG	K-means
4	10	4	2	9
6	0	24	26	16
8	95	56	36	36
9	100	40	30	40
10	100	52	36	44
12	100	70	54	40
14	100	100	64	60
15	100	98	72	58
16	100	100	88	64
18	100	100	96	84
20	100	100	100	68
21	100	100	96	70
24	100	100	100	84

TAB. 3 – Fréquence de détection (%) en fonction du nombre de classes imposé

n'est pas optimum.

Les performances de SOM et GNC dépendent fortement de leurs paramétrages respectifs. Les critères décisifs de paramétrage de ces algorithmes sont pour SOM, l'initialisation de la grille 2D et pour GNC la bonne gestion de l'âge maximum des connexions. L'algorithme de Kohonen est basé sur un réseau de neurones organisé sous forme de grille, où les plus proches voisins d'un neurone au sens de la grille correspondent avec les plus proches voisins selon le critère de distance. Cette propriété doit être vérifiée dès le départ de l'algorithme sans quoi les performances s'en ressentent très nettement. Pour assurer un bon départ à l'algorithme, nous avons défini des opérateurs d'addition et de produit scalaire sur nos vecteurs de données, ainsi qu'un vecteur nul. Munis de ces propriétés, la grille peut être initialisée convenablement à partir de trois signaux suffisamment séparés tirés d'une portion de l'ensemble des données. Les trois signaux et le vecteur nul définissent les coins de la grille, les autres neurones s'en déduisent et sont positionnés avec l'aide des opérateurs.

Pour GNC, le paramètre critique est l'âge maximum des connexions. Pour que l'algorithme fonctionne convenablement de manière autonome, ce qui est le cas dans les tests automatiques de comparaisons avec les autres algorithmes, la mort de neurones doit subvenir un minimum de fois. Ce nettoyage automatique de neurone garantit la reproductibilité de l'algorithme et évite qu'il converge vers des situations d'exception. L'âge maximum k_{max} doit être ajusté juste en dessous du seuil pour lequel il n'y a plus de morts. Ce paramétrage est fonction de la taille de l'ensemble des données et du nombre de classes. Nous avons rendu le paramétrage de l'âge indépendant de ces deux paramètres en définissant un autre paramètre ar_{max} suivant l'équation [Equ. 12] Le nouveau paramètre devient ar_{max} et a_{max} reste une variable interne à l'algorithme qui est recalculée à chaque fois qu'un neurone apparaît ou disparaît. De cette manière la série de tests dépendants du nombre de classes a été réalisée avec le même paramètre. En l'absence de la modification, il faut redéfinir ce paramètre à chaque situation, sans

quoi les performances de GNC passent en dessous de LBG lorsque l'on s'éloigne un peu du nombre de classes pour lequel l'algorithme a été paramétré.

4.2 Données réelles de type bloc

Le jeu de données réelles est une série IRMf de tests auditifs. Ces données proviennent du site de l'institut de recherche londonien "The Wellcome Department of Imaging Neuroscience" ¹, elles font partie du jeu de tests du programme SPM99 à disposition à des fins d'éducation et d'évaluations. La série est constituée de 96 acquisitions d'IRM du cerveau enregistrées avec un temps de répétition de 7 s. Le paradigme de type bloc alterne les deux situations suivantes, une condition sans stimulus et une condition de stimuli auditifs par répétitions de mots de deux syllabes. Chaque bloc dure 42 secondes (6 images), les deux conditions sont ainsi répétées 8 fois. Les données figurent sous forme de deux versions, la série d'acquisition brute et la série prétraitée en vue d'une analyse par SPM. Les images d'origine ont été acquises avec une résolution spatiale de 3 mm, formant chacune un volume de données de 64x64x64 voxels. La seconde série est plus volumineuse car les traitements font passer les images à 79x95x68 voxels. Les traitements effectués sont au nombre de trois, recalage, normalisation et lissage spatial. Tout d'abord, les faibles mouvements de tête du sujet sont corrigés en recalant tous les volumes de la série sur le premier. Ensuite, les images de cerveau sont déformées pour obtenir un cerveau normalisé, ceci permet la comparaison entre plusieurs sujets et l'utilisation d'un atlas structurel. Enfin les images sont lissées spatialement par convolution avec une gaussienne de largeur 6 mm pour forcer la cohérence spatiale entre les données. Les résultats présentés dans cette partie portent sur des classifications effectuées sur les deux séries. Avec la série pré-traitée, nous présentons des classifications réalisées avec les quatre algorithmes placés dans les mêmes conditions, puis les conséquences du nombre de classes sur les résultats IRMf. Avec la série brute nous obtenons une classification en utilisant les données du paradigme.

Les quatre algorithmes ont été testés sur la série prétraitée avec un nombre de classes limité à 9. Les images étant normalisées, on a pu restreindre l'exploration au volume du cerveau, de plus les coupes supérieures et inférieures ont été ôtées, réduisant la masse des données à 26% du volume initial. La figure [Fig. 4] illustre le résultat de cette classification pour les 4 algorithmes répétés chacun 4 fois. Cette figure montre le résultat sur une même coupe du cerveau pour les seize essais, une colonne représente les quatre essais d'un même algorithme. On affecte les couleurs aux neuf classes en fonction de leur taille, ce qui permet à deux classifications identiques d'avoir le même aspect. Sur cette figure les classes d'intérêt ont été fixées à la couleur blanche. Les classes blanches issues de GNC et SOM correspondent sur l'ensemble du volume aux régions d'activation pointées par le résultat du test SPM. Les deux algorithmes sont reproductibles non seulement sur ces régions mais aussi sur les huit autres classes. Les résultats issus de LBG et K-means ne sont pas aussi constants. Sur cet exemple, LBG aggrège un autre ensemble de voxels avec la classe attendue et K-means sous-estime la classe en question. Dans les deux cas, le reste du cerveau est classifié de manière très variable.

¹<http://www.fil.ion.ucl.ac.uk/spm/>

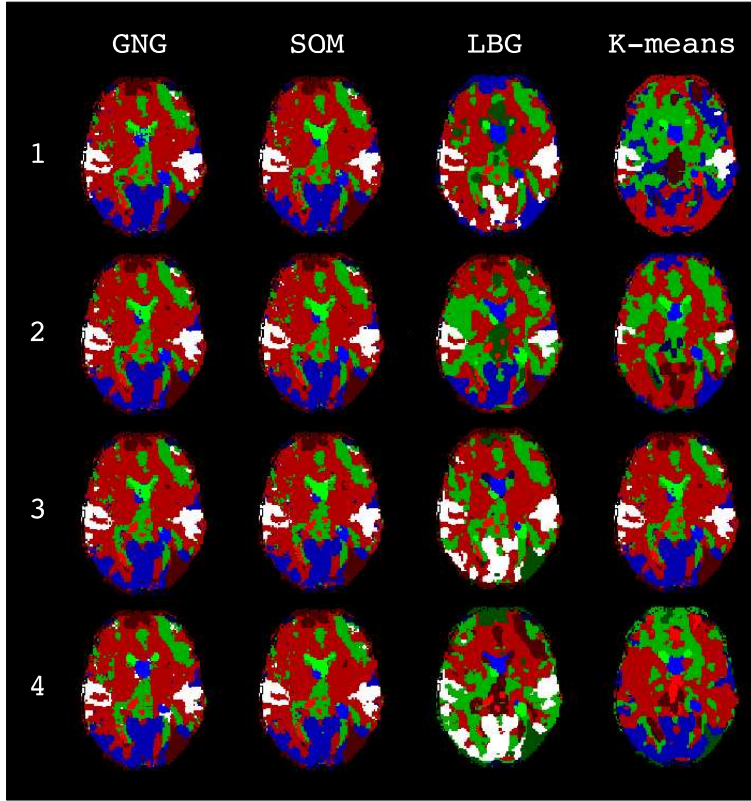


FIG. 4 – Reproductibilité des algorithmes GNC, SOM, LBG et K-means sur le test auditif.

La figure [Fig. 5] donne un exemple de l'influence du choix du nombre de classes. Cette figure montre une même coupe issue d'une image IRM structurée avec en surimpression les classes d'intérêt issues de classifications GNC dont le nombre de classes a été limité à cinq, neuf et dix-huit. En dessous figurent les séries temporelles de ces mêmes classes d'intérêt. Avec cinq classes, la classe d'intérêt intègre des voxels en dehors des zones d'activation, la série temporelle s'en retrouve perturbée en comparaison au cas 9 où l'on perçoit les huit périodes alternant les conditions avec et sans stimulus. Dans ce cas, la classe d'intérêt correspond bien à la région d'activation. Avec dix-huit classes, la zone d'activation engendre deux classes d'intérêt séparées par leurs intensités respectives. La zone la plus intense est entourée par la seconde zone et les signaux correspondants mettent bien en évidence la nature de l'expérience de type bloc.

Les tests de classification sur les données brutes n'ont pas abouti à des classes révélant les zones détectées précédemment. La classification résultante met en évidence les bruits les plus significatifs, comme des mouvements de tête du sujet qui génèrent des classes le long des contrastes les plus intenses de l'image. En regroupant les données en

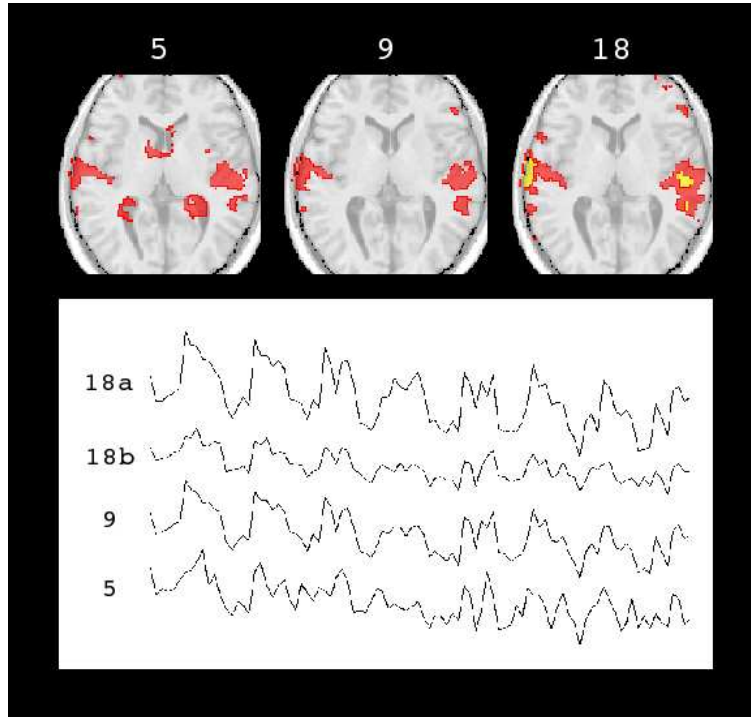


FIG. 5 – Classifications des données pré-traitées pour différents nombres de classes

accord avec le paradigme, nous avons pu révéler les lieux d'activation du test auditif. Le paradigme de ce test est du type bloc, où deux conditions se succèdent et se répètent, formant un motif périodique. Ici, il s'agit de deux conditions durant chacune 6 images et se répétant 8 fois. Nous avons donc moyenné les 8 périodes en une seule période constituée ainsi de 12 images. La classification sur cette petite série fait apparaître préférentiellement les zones attendues sous forme de blocs de voxels compacts, au sein d'une classe qui comporte toutefois quelques voxels répartis un peu partout [Fig. 6]. L'ensemble des données à traiter est réduit d'un facteur 16 par rapport à la série pré-traitée. Ce résultat est intéressant dans la mesure où il n'y a pas de pré-traitement et que le volume des données est extrêmement réduit, augmentant considérablement la vitesse d'exécution.

5 Conclusion

Dans cet article, nous avons considéré les séries d'images 3D IRMf comme objets complexes en prenant en compte le volume, la dimension temporelle, les relations spatiales et le bruit. Nous avons détaillé la partie centrale de notre système de fouille de séquences d'images 3D IRMf : la classification. Dans notre approche, nous avons

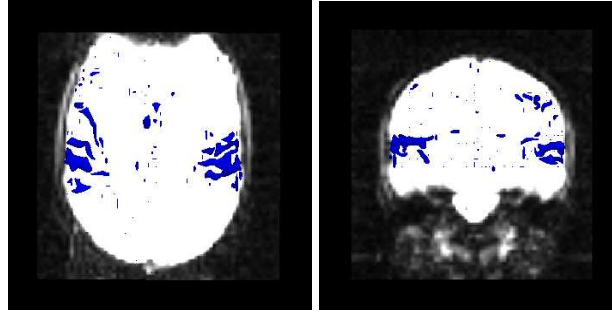


FIG. 6 – Classification des données brutes compactées, zones d'intérêt en projection sur des coupes axiales et coronales

imposé la nécessité de l'engagement d'un expert-médecin dans le processus de classification d'une part, et d'autre part, l'intégration d'un algorithme de classification capable de s'adapter aux spécificités de l'IRMf avec de faibles connaissances préalables. Plusieurs algorithmes de classifications ont été étudiés. Les expérimentations ont montré que les méthodes de classification classiques ne sont pas toujours performantes. Nous avons choisi et étendu le modèle des Gaz Neuraux Croissants qui s'est avéré plus robuste et reproductible dans les mêmes conditions de fonctionnement. Tous ses paramètres peuvent être modifiés en cours d'exécution y compris le paramètre du nombre de classes. Cet aspect dynamique est déterminant pour la fouille interactive. L'outil de fouille de données visuelle qui en découle permet notamment d'optimiser les paramètres de l'algorithme, de réduire l'espace de recherche, de présenter le processus de découverte sous plusieurs angles et participe ainsi à la diminution de la complexité des objets concernés pour fournir à l'expert-médecin des classifications de régions cervicales plus compréhensibles. Les premiers résultats sur des données de type bloc sont encourageants et nous permettent d'envisager la poursuite de ces travaux vers des expériences de type événementiel. Sur ce type de données, le rapport signal sur bruit est particulièrement faible et les premiers tests effectués ont abouti à la classification des bruits les plus conséquents qui masquent les signaux escomptés. Pour révéler les signaux d'intérêt aussi ténus parmi la somme des bruits, il faut préalablement à la classification, opérer un prétraitement spécifique. Les perspectives tendent vers la méthode ICA, combinée à des connaissances préalables pour dégager les principales sources de bruit.

Remerciements. Les auteurs remercient Karl Friston et Geraint Rees pour les données de test SPM, ainsi que les étudiants de l'Université Louis Pasteur de Strasbourg H. Hager, P. Hahn, V. Meyer, J. Schaeffer et O. Zitvogel, pour leur participation dans la phase initiale de réalisation du projet.

Références

- [Agrawal *et al.*, 1996] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, et A. Verkamo. Fast discovery of association rules. In U. Fayyad *et al.*, editor, *Avances in knowledge discovery and data mining*, pages 307–328. AAAI Press, 1996.
- [Andersen *et al.*, 1999] A. H. Andersen, D. M. Gash, et M. J. Avison. Principal component analysis of the dynamic response measured by fmri : a generalized linear systems framework. *Magnetic Resonance Imaging*, 17 :795–815, 1999.
- [Arfanakis *et al.*, 2000] K. Arfanakis, D. Cordes, V. M. Haughton, C. H. Moritz, M. A. Quigley, et M. E. Meyerand. Combining independent component analysis and correlation analysis to probe interregional connectivity in fmri task activation datasets. *Magnetic Resonance Imaging*, 18 :921–930, 2000.
- [Barth *et al.*, 2001] M. Barth, C. Windischberger, M. Klarhofer, et E. Moser. Characterisation of bold activation in multi-echo fmri data using fuzzy cluster analysis and a comparison with quantitative modeling. *NMR Biomed.*, 14 :484–489, 2001.
- [Baumgartner *et al.*, 2001] R. Baumgartner, R. Somorjai, R. Summers, et W. Richter. Ranking fmri time courses by minimum spanning trees : Assessing coactivation in fmri. *NeuroImage*, 13 :734–742, 2001.
- [Beckmann et Smith, 2003] C. Beckmann et S. M. Smith. Probabilistic independent component analysis for functional magnetic resonance imaging. *IEEE Trans. on Medical Imaging*, 2003.
- [Bock et Diday, 2000] H. Bock et E. Diday. *Analysis of Symbolic Data. Exploratory methods for extracting statistical information from complex data*. Springer Verlag, 2000.
- [Bozo et Gray, 1980] Y. Linde A. Bozo et R. M. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, COM-28 :84–95, 1980.
- [Cox, 1996] RW. Cox. Afni : Software for analysis and visualization of functional magnetic resonance neuroimages. *Computers and Biomedical Research*, 29 :162–173, 1996.
- [Dimitriadou *et al.*, 2004] E. Dimitriadou, M. Barth, C. Windischberger, K. Hornick, et E. Moser. A quantitative comparison of functional mri cluster analysis. *Artificial Intelligence in Medicine*, 31 :57–71, 2004.
- [Esposito *et al.*, 2002] F. Esposito, E. Formisano, E. Seifritz, R. Goebel, R. Morrone, G. Tedeschi, et F. DiSalle. Spatial independent component analysis of functional mri time-series : To what extent do results depend on the algorithm used ? *Human Brain Mapping*, 16 :146–157, 2002.
- [Fadili *et al.*, 2000] M. J. Fadili, S. Ruan, D. Bloyet, et B. Mazoyer. A multistep unsupervised fuzzy clustering analysis of fmri time series. *Human Brain Mapping*, 10 :160–178, 2000.
- [Filzmoser *et al.*, 1999] P. Filzmoser, R. Baumgartner, et E. Moser. A hierarchical clustering method for analysing functional mr images. *Magnetic Resonance Imaging*, 17 :817–826, 1999.

- [Fischer et Hennig, 1999] H. Fischer et J. Hennig. Neural network-based analysis of mr time series. *Magnetic Resonance in Medicine*, 41 :124–131, 1999.
- [Forgy, 1965] E. Forgy. Cluster analysis of multivariate data : efficiency vs. interpretability of classifications. *Biometrics*, 21 :768–780, 1965.
- [Friston *et al.*, 1995] K. J. Friston, A. P. Holmes, K. J. Worsley, J. P. Poline, C. D. Frith, et R. S. J. Frackowiak. Statistical parametric maps in functional imaging : A general linear approach. *Human Brain Mapping*, 2 :189–210, 1995.
- [Fritzke, 1994] B. Fritzke. Fast learning with incremental rbf networks. *Neural Processing Letters*, 1 :2–5, 1994.
- [Fritzke, 1995] B. Fritzke. A growing neural gas network learns topologies. In G. Tesauro, D. S. Touretzky, et T. K. Leen, editors, *Advances in Neural Information Processing Systems 7*, pages 625–632. MIT Press, Cambridge MA, 1995.
- [Goebel, 1997] R. Goebel. *BrainVoyager : Ein Programm zur Analyse und Visualisierung von Magnetresonanztomographiedaten*. T. Plesser and P. Wittenburg, Forschung und wissenschaftliches Rechnen, 1997.
- [Goutte *et al.*, 1999] C. Goutte, P. Toft, E. Rostrup, F. A. Nielsen, et A. K. Hansen. On clustering fmri time series. *NeuroImage*, 9 :298–310, 1999.
- [Guha *et al.*, 1998] S. Guha, R. Rastogi, et K. Shim. Cure : An efficient clustering algorithm for large databases. In *Proceedings of the ACM SIGMOD*, pages 73–84, 1998.
- [Kohonen, 1982] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43 :59–69, 1982.
- [Lai et Fang, 1999] S. H. Lai et M. Fang. A novel local pca-based method for detecting activation signals in fmri. *Magnetic Resonance Imaging*, 17 :827–836, 1999.
- [Lloyd, 1982] S. P. Lloyd. Least squares quantization in pcm. technical note, bell laboratories, 1957. *IEEE Transactions on Information Theory*, 1982.
- [MacQueen, 1967] J. MacQueen. Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical statistics and probability*, 1 :281–297, 1967.
- [Moller *et al.*, 2001] U. Moller, M. Ligges, C. Grunling, P. Georgiewa, W. A. Kaiser, H. Witte, et B. Blanz. Pitfalls in the clustering of neuroimage data and improvements by global optimization strategies. *NeuroImage*, 14 :206–218, 2001.
- [Ngan et Hu, 1999] S. C. Ngan et X. Hu. Analysis of functional magnetic resonance imaging data using self-organizing mapping with spatial connectivity. *Magnetic Resonance in Medicine*, 41 :939–946, 1999.
- [Sommer et Wichert, 2003] F.T. Sommer et A. Wichert. *Exploratory Analysis and Data Modeling in Functional Neuroimaging*. The MIT Press, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2003.
- [Wager, 2001] T. Wager. Characterizing nonlinear bold responses in human visual cortex. *NeuroImage*, 13(6) :278, 2001.
- [Zaki, 2001] M. J. Zaki. Spade : An efficient algorithm for mining frequent sequences. *Machine Learning*, 42 :31–60, 2001.

[Zhang *et al.*, 1996] T. Zhang, R. Ramakrishnan, et M. Livny. Birch : an efficient data clustering method for very large databases. In *Proceedings of the ACM SIGMOD*, pages 103–114, 1996.

Annexe

Algorithme des centres mobiles

Cet algorithme appelé usuellement K-means, pour K moyennes [MacQueen, 1967] débute avec l'ensemble $C = \{c_1, c_2, \dots, c_k\}$ de k classes dont les vecteurs de référence w_c sont initialisés avec les valeurs de k individus $\{\xi_1^1, \xi_1^2, \dots, \xi_1^k\}$ différents choisis aléatoirement dans l'ensemble des données D . Le déroulement de l'algorithme est rythmé par le tirage aléatoire d'un individu de l'ensemble de données. Chaque nouvel individu déplace le vecteur de référence de la classe la plus proche. A chaque classe $c \in C$ est associé un coefficient d'apprentissage $\epsilon(t) = 1/t$. Le paramètre temps t de la classe c correspond au nombre d'individus tirés pour lesquels c a été déclarée gagnante ($t = 1$ à l'initialisation). L'algorithme porte le nom de K-means car pour toute classe, le vecteur référence est à la moyenne arithmétique des individus appartenant ou ayant appartenu à la classe.

Algorithme K-means :

1. Initialisation des k classes à partir de k individus différents tirés aléatoirement dans D : pour chaque classe c , $w_c = \xi_1^c$, $t_c = 1$ et $\epsilon_c(t) = 1$.
2. Tirage aléatoire d'un individu ξ de l'ensemble des données D .
3. Recherche de $c = g(\xi)$ la classe gagnante de ξ , c'est-à-dire la classe pour laquelle le vecteur de référence w_c est le vecteur le plus proche de ξ [Equ. 4].
4. Incrémentation du paramètre temps t_c de c , $t_c = t_c + 1$.
5. Ajustement du vecteur de référence w_c de la classe c

$$w_c(t) = w_c(t-1) + \epsilon(t) * (\xi_t^c - w_c(t-1))$$

6. Retour en 2 ou Arrêt suivant le critère choisi par le programmeur.

Au cours du temps, le vecteur référence w_c de la classe c prendra les valeurs suivantes :

$$w_c(1) = \xi_1^c$$

$$w_c(2) = w_c(1) + \epsilon(2) * (\xi_2^c - w_c(1)) = w_c(1) + 1/2 * (\xi_2^c - w_c(1)) = (\xi_1^c + \xi_2^c)/2$$

$$w_c(3) = w_c(2) + \epsilon(3) * (\xi_3^c - w_c(2)) = (\xi_1^c + \xi_2^c + \xi_3^c)/3$$

$$w_c(t) = w_c(t-1) + \epsilon(t) * (\xi_t^c - w_c(t-1)) = (\xi_1^c + \xi_2^c + \dots + \xi_t^c)/t$$

Le seul paramètre de cet algorithme est le nombre de classes c à définir au début. Il est important de noter que des individus qui ont participé à l'adaptation d'une classe en début de processus, peuvent ne plus faire partie de cette classe à la fin de l'algorithme bien que leur vecteur de référence soit intégré au calcul de la moyenne de cette classe.

Algorithme Lloyd Généralisé

Cet algorithme est aussi appelé la plupart du temps K-means [Bozo et Gray, 1980] [Forgy, 1965] [Lloyd, 1982]. Le calcul du vecteur de référence de chaque classe est aussi la moyenne arithmétique des vecteurs de chacun de ses individus. Contrairement à l'algorithme précédent, ce calcul n'est pas effectué à chaque tirage, mais lorsque l'ensemble des individus a été passé en revue. Il débute avec l'ensemble $C = \{c_1, c_2, \dots, c_k\}$ de k classes dont les vecteurs de référence w_c sont initialisés avec les valeurs de k individus $\{\xi_1^1, \xi_1^2, \dots, \xi_1^k\}$ différents choisis aléatoirement dans l'ensemble des données D . Cette fois, le déroulement de l'algorithme est rythmé par les balayages entiers de l'ensemble des données. A chaque itération, tout individu est associé à la classe la plus proche et les classes sont ensuite recentrées par le calcul des moyennes.

Algorithme LBG :

1. Initialisation des k classes à partir de k individus différents tirés aléatoirement dans D : pour chaque classe c , $w_c = \xi_1^c$, $t_c = 1$ et $\epsilon_c(t) = 1$.
2. Association de chaque individu ξ à sa classe la plus proche $c = g(\xi) : \xi \in R_c$.
3. Ajustement des vecteurs de référence w_c des classes c comme suit : $w_c = \frac{1}{|R_c|} \sum_{\xi \in R_c} \xi$
4. Retour en 2 ou Arrêt suivant la condition définie par le programmeur ou l'utilisateur.

Algorithme SOM de Kohonen

Les classes de l'algorithme de Kohonen [Kohonen, 1982] sont organisées sous forme de grille à deux dimensions (c_{ij}). Chaque classe possède ainsi plusieurs voisines topologiques qu'elle garde tout au long du déroulement de l'algorithme. Les classes situées en bordure de grille ont deux voisines directes, les autres en ont quatre. Comme dans K-means, l'algorithme se déroule par une succession de tirages aléatoires d'individus tirés de l'ensemble des données. Les individus tirés influencent non seulement la classe gagnante mais aussi ses voisines topologiques. D'autre part, l'adaptation des classes est atténuée par deux critères que sont la distance vis à vis de la classe gagnante et le temps écoulé.

Algorithme SOM :

Définitions des variables :

- t_{max} : durée de l'algorithme.
- $d_{rs} = |i - k| + |j - m|$: distance de Manhattan d_{rs} entre deux classes $r = c_{ij}$ et $s = c_{km}$
- $\sigma(t) = \sigma_i \cdot (\sigma_i / \sigma_f)^{t/t_{max}}$: coefficients de voisinage
- $\epsilon(t) = \epsilon_i \cdot (\epsilon_i / \epsilon_f)^{t/t_{max}}$: coefficients d'adaptation
- $h_{rs} = \exp(\frac{-d_{rs}^2}{2 \cdot \sigma(t)^2})$: fonction de voisinage h_{rs} entre les classes r et s

1. Choix des paramètres de l'algorithme :

- (a) largeur l et hauteur h de la grille

- (b) la durée de l'algorithme t_{max}
 - (c) les coefficients de voisinage σ_i et σ_f
 - (d) les coefficients d'adaptation ϵ_i et ϵ_f
2. Initialisation de la grille en prenant soin que les voisins topologiques directes de chaque classe soient aussi les plus proches voisines au regard du critère de distance euclidienne définie sur les données [Equ. 4].
 3. Tirage aléatoire d'un individu ξ de l'ensemble des données D.
 4. Recherche de $c = g(\xi)$ la classe gagnante de ξ , c'est-à-dire la classe pour laquelle le vecteur de référence w_c est le vecteur le plus proche de ξ [Equ. 4].
 5. Adaptation de chaque classe $c : w_c = w_c + \epsilon(t).h_{c\xi}(\xi - w_c)$
 6. Incrémentation du temps t
 7. Si $t < t_{max}$ retour en 3 sinon arrêt de l'algorithme.

Summary

The human brain provides typical complex data for data mining. Extracting active voxels from brain images is often very difficult due to a very high level of various noises. First experiments of current data mining algorithms in this domain showed their low performances and recognition abilities. In this article, we present a new interactive, data-driven, approach to mine fMRI images in order to show how the brain works from fMRI data. Several unsupervised classification algorithms are compared on 3D image sequences from fMRI, reporting results with respect to the number of classes, the noise level, and the ratio of the activated/observed areas.