

Projet 2: Clustering de signaux spatio-temporels d'IRM de perfusion de patients atteints d'infarctus cérébral

December 9, 2017

Référents sujet:

- David Rousseau david.rousseau@univ-angers.fr

1 Mise en contexte

L'accident vasculaire cérébral (AVC) ischémique, ou infarctus cérébral, est causé par l'obstruction d'une artère au niveau du cerveau. L'imagerie par résonance magnétique (IRM) pondérée de perfusion est une des techniques les plus utilisées dans la prise en charge des patients atteints d'AVC ischémique en phase aiguë pour renseigner sur le débit de la micro-circulation sanguine dans les tissus. Elle consiste à injecter un agent de contraste qui va circuler dans le réseau vasculaire et qui traduit la perfusion tissulaire. La zone atteinte par l'AVC (lésion ischémique) est hypoperfusée car obstruée par un caillot sanguin (comparée à la perfusion d'un tissu sain).

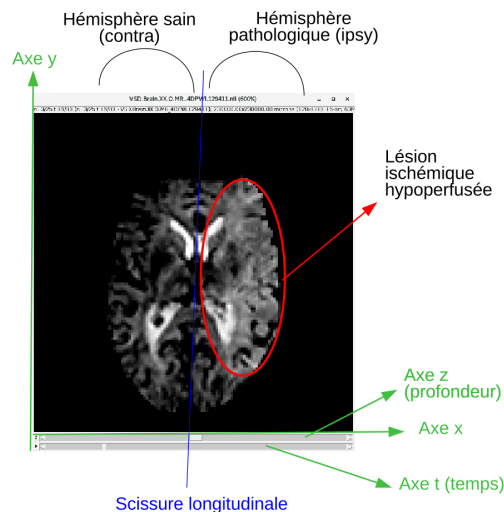


Figure 1: IRM de perfusion d'un patient atteint d'AVC ischémique (à ce temps donné, l'agent de contraste est visible en hypointensité et la lésion en hyperintensité)

L'IRM de perfusion est une image en intensité de gris en 4 dimensions: 3 dimensions spatiales (x,y,z) pour décrire le volume du cerveau, et une dimension temporelle (t) pour décrire la dynamique de passage de l'agent de contraste. Un moyen d'analyser ces images consiste à encoder l'environnement local de chaque voxel en signature spatio-temporelle comme décrit sur le schéma figure 2.

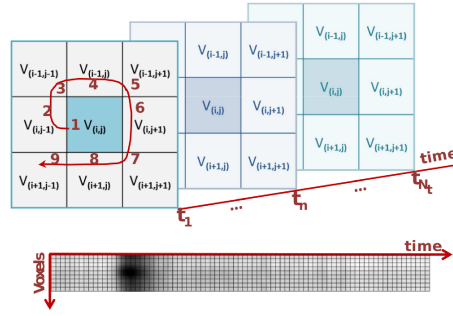


Figure 2: Exemple de signatures spatio-temporelles. Axe y: encodage du pixel et de son voisinage, Axe x: temps (*image: M. Giacalone*)

La signature spatio-temporelle est un graphique 2D traduisant une information locale 3D (x, y et t). On encode l'information contenue dans chaque voxel en déroulant les signaux temporels le long d'une dimension spatiale et en empilant les signaux des 9 voxels du voisinage (3×3) du voxel d'intérêt les uns à la suite des autres, créant ainsi une imagerie de taille $(9 \times t)$.

Empiriquement, on sait que les signatures spatiales des tissus lésés sont différentes des signatures des tissus sains en terme de texture (*travaux de thèse de M. Giacalone*).

2 Objectifs du projet

L'objectif de ce projet est de regrouper les pixels similaires entre eux par une (ou des) approche(s) d'apprentissage non supervisé(s). L'approche principale de ce projet est l'utilisation des K-means. En aval de l'apprentissage, il faudra vérifier la cohérence des regroupements réalisés.

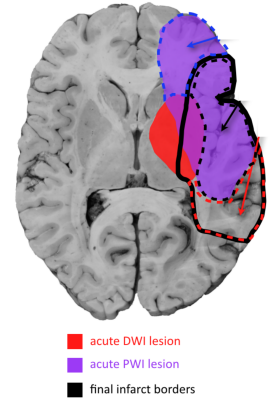


Figure 3: Mise en évidence des différents types de tissu du cerveau selon une coupe axiale

Sur la figure ci-dessus, vous pouvez distinguer plusieurs zones spatialement et physiologiquement distinctes. En rouge, il s'agit du coeur de lésion ischémique: les tissus de cette zone sont souvent déjà nécrosés et donc non récupérables. En bleu, il s'agit de la zone de pénombre: les tissus de cette zone sont touchés par l'infarctus mais il est difficile de prédire s'ils seront récupérables ou non. La zone entourée en noir correspond à la vraie lésion finale ischémique. Il s'agira donc de voir si les clusters de patches correspondent à ces zones plus ou moins définies.

3 Données à disposition

Vous avez à votre disposition pour 4 patients atteints d'AVC:

1. 26 526 patches générés à partir de tous les pixels du volume d'IRM de perfusion (tissu sain, ischémique, autre) ¹,
2. l'IRM de diffusion, un autre examen qui permet de voir à l'admission l'étendue de la lésion.

Tout l'enjeu est de réaliser des groupes cohérents de patches à partir des descripteurs associés et sans aucun a priori sur leur appartenance à une classe.

4 Étapes du projet

Pour la réalisation de ce projet, il vous est proposé de coder en python. Beaucoup de tâches qui vous sont demandées sont bien documentées sur internet et peuvent être réalisées à travers de fonctions déjà implémentées. Consultez attentivement la page suivante :

- <http://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>

4.1 Caractérisation des patches en vecteur d'intensité de gris

Dans un premier temps, vous devez créer des vecteurs d'intensité moyenne de volume en fonction du temps.

1. Traduisez chaque patch en matrice d'intensité de gris avec le code suivant :


```
>>> from scipy import misc
>>> gray_matrix = misc.imread('patch.png')
```
2. A partir de chaque matrice (9*t), moyennerez l'intensité de gris à chaque pas de temps pour retrouver un vecteur de taille (1*t)
3. Sauvegardez dans un fichier les 11 000 vecteurs (1*t) caractéristiques des 11 000 patches dans un fichier.

4.2 Apprentissage du modèle

Pour ce projet, on choisit de faire des groupes de patches (clustering) avec la méthode des K-means. Pour chacun des 2 fichiers de descripteur, entraînez votre modèle K-means avec le code suivant:

```
>>> from sklearn.cluster import KMeans
>>> import numpy as np
>>> data = np.loadtxt('descripteur1_haralickTime.txt')
>>> data.shape
(26 526,14)
>>> kmeans = KMeans(n_clusters=N,init='random').fit(data)
>>> print kmeans.labels_
```

Faire varier le nombre de classes N de 2 à 10. Comparer les temps de calculs de chacun des apprentissages. Vos différentes classes correspondent-elles à des composantes connexes? Font-elles sens au regard de la clinique (cf Figure 3) ?

4.3 Sensibilité à l'initialisation

1. Tester la stabilité de vos clusters en réalisant plusieurs itérations de chaque apprentissage.
2. Remplacer dans le ligne de code de votre modèle *init='random'* par *init='k-means++'*. Refaites tourner vos apprentissages. A quoi sert cette option ? Quel est son impact ?

¹encodage du nom des patch: *patient_n_sim_01_typeTissu_coords_x_y_z.png*, avec *n* l'indice du patient, *typeTissu* la classe tissulaire (lésion, matière blanche ou matière grise), et *x,y,z* les coordonnées du pixel central du patch

4.4 Validation du clustering

1. Afin d'évaluer l'efficacité de votre clustering K-means, calculez le coefficient silhouette de chacun des modèles avec le code suivant :

```
>>> from sklearn.metrics import silhouette_samples, silhouette_score
>>> silhouette_score(data, kmeans.labels_, metric='euclidean')
```

Le coefficient silhouette est une mesure de similitude entre un objet et son propre cluster (cohésion) par rapport aux autres clusters (séparation). La valeur du coefficient va de -1 à +1, où une valeur élevée indique que l'objet est bien adapté à son cluster et mal adapté aux clusters voisins. Si la plupart des objets ont une valeur élevée, la configuration de cluster est appropriée. Si de nombreux points ont une valeur faible ou négative, la configuration de clustering peut comporter trop ou trop peu de clusters.

2. Représentez chaque patch dans l'espace des coordonnées (x,y) par un point en couleur, de façon à ce qu'une couleur corresponde à un cluster. Notez bien qu'il y aura une carte par coupe axiale (coupe selon l'axe z). Est-ce que les clusters réalisés correspondent à des clusters spatiaux ?

4.5 Comparaison avec d'autres méthodes

1. Chercher d'autres méthodes d'apprentissage non supervisées ?
2. Essayer d'en implémenter une au choix pour comparer vos résultats à ceux obtenus avec la méthode K-means.