

Université Toulouse 3 Paul Sabatier - Master Informatique

**Rapport Projet CSAA : Segmentation d'images TEP
par classification spectrale.**

Présenté par :

DAOULATLI

Nour

DAHMANI

Meriem

Sommaire

Introduction

Préparation des données.....1

Méthode 1

Méthode 2

Évaluation des résultats

Conclusion

Introduction

L'objectif de ce projet est de réaliser classification non supervisée. Nous avons à notre disposition des courbes temps d'activité (TACs) simulées et des courbes temps d'activité théoriques.

Ce mini projet est divisé en cinq parties et dans chacune d'elle nous allons appliquer notre classification sur deux deux coupes : transverse et sagittale. Nous allons ainsi implémenter deux méthodes différentes : dans un premier temps, une classification par partitionnement puis, dans un second temps nous allons tester notre classification spectrale sur nos courbes TACs, et enfin, nous allons implémenter la seconde méthode de classification : K Means, en premier lieu, puis par réductions ACP+Kmeans. Par la suite, comparerons nos résultats avec vérité terrain : pour cela nous calculerons la précision (l'exactitude des prédictions positives) et FMI (l'indice de Fowlkes-Mallows qui permet de mesurer la similarité entre deux partitionnements). Pour finir, nous allons faire notre analyse et conclure en testant différentes valeurs de sigma et de classes.

Préparation des données

En termes de données, nous avons deux coupes : Transverse et Sagittale stockées respectivement dans data_T et data_S. De plus, nous avons vérité terrain pour chaque coupe stockés dans dataROI_S et dataROI_T. D'autre part, on dispose d'un autre exemple d'application : Toy Example qu'on a stocké dans ExpleJouet. En effet, toutes ses données ont été récupérées à partir des fichiers ".mat" qu'on a retrouvé sur moodle.

Méthode 1 : Classification spectrale

Pour cette partie, nous avons implémenter d'abord la fonction classification_spectrale qui prend en paramètre les données (Data), k qui représente le nombre de clusters et sigma. Tout d'abord, on a commencé par construire la matrice d'affinité, puis on a initialisé et rempli la matrice diagonale D de A afin de calculer la matrice normalisée L qui a pour formule : $L = D^{-1} \times A$. Ensuite, on calcule les vecteurs propres afin de créer la matrice X qui contient le K+ grand vecteurs de L. Après son obtention, on va extraire les extraire dans Y en normalisant X. Pour finir, on les classe en k clusters en utilisant la méthode kmeans.

Méthode 2 : Kmeans et réduction de dimension par ACP + Kmeans

Dans un premier temps, nous allons implémenter le Kmeans sans réduction ACP : en utilisant la méthode Kmeans, avec en paramètres k (le nombre de clusters), puis on applique le kmeans sur notre data avec fit, pour ensuite récupérer les clusters avec .labels_. Par suite, nous allons appliquer la réduction de dimension avec ACP: au départ, nous devons décider du nombre de composantes principales que nous souhaitons conserver en fonction du graphique

de variance cumulée, ce pourquoi nous allons essayer avec le nombre de dimension de notre data puis calculer la somme cumulée des variances et pour finir récupérer la valeur qui va conserver 95% de l'information. Ainsi, on applique le kmeans sur la data réduite.

Évaluation des résultats

Afin d'évaluer nos algorithmes, on a implémenté des fonctions pour le calcul des quatre métriques : rappel ,précision , taux de fp et spécificité. A partir des résultats obtenus, nous pouvons déduire les remarques suivantes : pour commencer, la classification spectrale donne de meilleurs résultats que la méthode des Kmeans, ceci est visible sur les images en comparant avec la vérité terrain ainsi qu'avec le FMI : pour la coupe transverse nous avons un FMI de 0.61 pour classification spectrale (CS) contre 0.44 pour Kmeans et pour la coupe sagittale nous avons un FMI de 0.6 pour CS contre 0.5 pour Kmeans. En effet, le recall et la précision confirment ceci :

- coupe sagittale : recall = 0.20 et précision = 0.17 contre recall = 0.16 et précision = 0.02 pour Kmeans
- coupe transverse : recall = 0.02 et précision = 0.17 contre recall = 0.08 et précision = 0.18 pour Kmeans

De plus, nous avons remarqué que la réduction par ACP + Kmeans donne des résultats plus ou moins similaires au Kmeans ou bien avec amélioration peu importante : pour la coupe sagittale nous avons un FMI de 0.503 sans CPA contre 0.508 avec CPA et pour la coupe transverse l'écart est encore moins important nous avons 0.4428 sans CPA contre 0.4423 avec CPA.

Pour avoir une optimisation maximale, nous avons cherché à connaître la valeur sigma qui donnera le meilleur résultat possible pour la coupe sagittale : celle-ci est de 0,9. Cette dernière a été trouvée en comparant les FMI pour chaque valeur de sigma comprise entre 0,1 et 1.

Conclusion

Dans ce projet, nous avons eu l'occasion d'implémenter et tester différents algorithmes pour réaliser une classification non supervisée, nous avons essayé de bien choisir nos algorithmes qui sont adaptés au problème et nos paramètres afin d'en tirer le meilleur résultat et de pouvoir à la fin comparer les différents algorithmes entraînés.