

ESIR3-IN : I.M. TP 3

Segmentation "moyenne", Algorithme STAPLE, Courbe ROC

15 IRM (T1) de cerveaux non pathologiques ont été segmentées manuellement. On veut construire un atlas anatomique à partir de ces données. Une première étape consiste à recalcr spatialement les 15 IRM et à faire la moyenne de ces 15 volumes. Vous trouverez dans le fichier `TP3_donnees.mat` une image `IRM_T1` contenant une coupe (arbitrairement choisie) du résultat de cette première étape. `IRM_T1` correspond donc à la moyenne des 15 IRM différentes, toutes recalées les unes avec les autres.

1 Union, Intersection, Majorité

1. La matrice `Segm_binaire` contient les 15 segmentations de la matière blanche, recalées sur l'IRM moyenne `IRM_T1`. Ainsi `Segm_binaire(:, :, k)` contient la segmentation du k -ième sujet. Écrivez une fonction `displaySegOnMri(Seg, MRI)` qui affiche une segmentation binaire par dessus une IRM (en rouge par exemple). Affichez quelques-unes de ces segmentations par dessus l'image anatomique moyenne en utilisant cette fonction.
2. Construisez l'intersection et l'union de ces segmentations (une ligne pour chacune). L'un de ces résultats vous semble-t-il un candidat satisfaisant pour représenter la segmentation "moyenne" associée aux 15 segmentations initiales ?
3. Une stratégie plus évoluée consiste à appliquer la règle de la majorité ("majority voting") qui consiste à attribuer à chaque pixel l'étiquette qui revient la plus fréquemment dans les différentes segmentations initiales. Cette stratégie a également l'avantage d'être adaptable directement aux cas où L ne contient plus simplement des segmentations binaires mais des segmentations à n classes. En utilisant la fonction matlab `mode`, calculez la segmentation moyenne (binaire) de la matière blanche donnée par cette méthode, ainsi que la segmentation à 4 classes (matière grise, matière blanche, ventricule, noyaux gris centraux) calculée à partir de `Segm_4classes`.
4. Dans certains cas, une grande variabilité dans la qualité des différentes segmentations initiales peut créer des segmentations moyennes imparfaites. Un exemple très caricatural est donné par les matrices `I_Toy` et `Segm_Toy` (cf Fig. 1) : la segmentation 1 est une segmentation obtenue par seuillage (et donc imparfaite du fait de la présence de fort bruit), les segmentations 2 et 7 sont quasi-parfaites, les segmentations 3, 4, 5 et 6 ne semblent pas cohérentes

avec la donnée à segmenter (la 3 et la 6 ayant probablement été réalisées par des étudiants un vendredi matin). Testez et affichez les 3 stratégies précédentes (union, intersection et majorité) sur cet exemple.

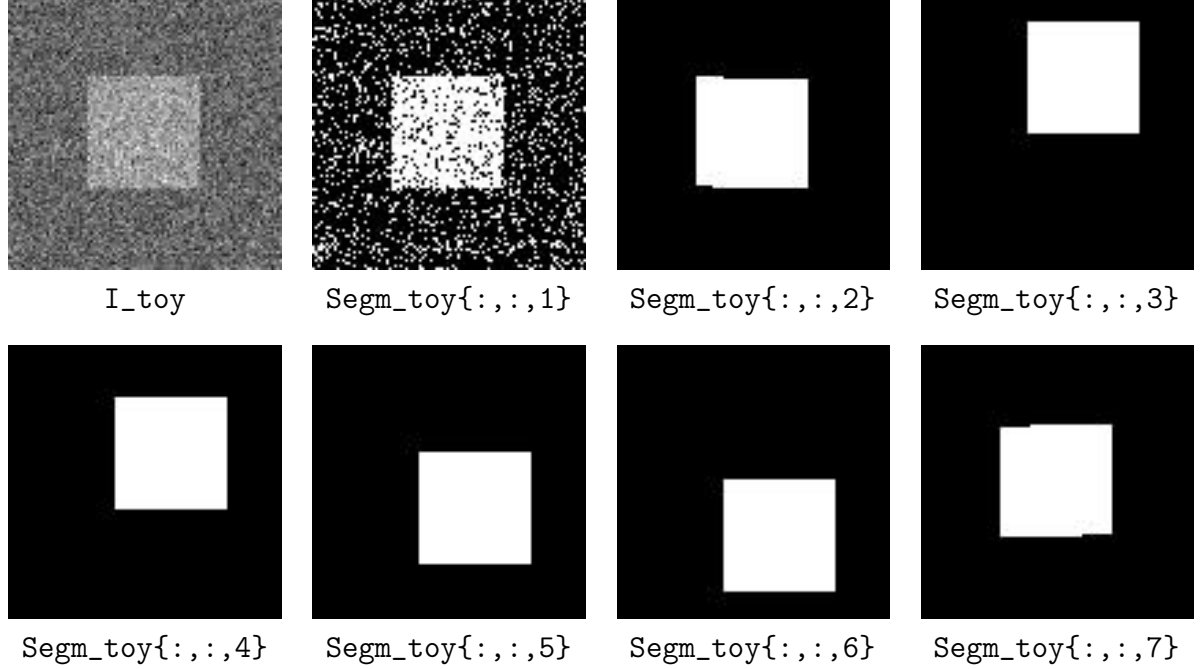


FIGURE 1 – Exemple "jouet"

2 Algorithmes STAPLE

Dans ces cas (grande variabilité de qualité des segmentations), une meilleure segmentation peut être obtenue en accordant plus de poids à certaines segmentations initiales par rapport à d'autres. L'algorithme STAPLE¹ introduit en cours permet simultanément de combiner n segmentations et d'estimer le niveau de performance de chacune des segmentations initiales.

Nous allons l'implémenter dans le cas d'une segmentation binaire mais la généralisation au cas n -aire est simple et rapide (cf article par exemple). Dans le cas binaire, la qualité de la segmentation j (ou de "l'expert" associé) sera mesurée via sa sensibilité p_j et sa spécificité q_j :

$$p_j = P(D_{ij} = 1 \mid T_i = 1) \quad , \quad q_j = P(D_{ij} = 0 \mid T_i = 0)$$

où, pour un pixel i :

- T_i est la segmentation "idéale" (inconnue mais que l'on va estimer)
- D_{ij} est la segmentation, l'étiquette (0 ou 1 ici) donnée par l'expert j . Dans notre cas, si $i = (x, y)$ on a $D_{ij} = \text{Segm_toy}(x, y, j)$.

1. Simultaneous truth and performance level estimation (STAPLE) : an algorithm for the validation of image segmentation, Warfield SK, Zou KH, Wells WM, IEEE Trans Med Imaging. 2004 Jul

On souhaite donc estimer T_i et (p_j, q_j) **simultanément**. L'algorithme STAPLE fait partie de la classe des algorithmes "espérance-maximisation" (en anglais "Expectation-maximisation algorithm", souvent abrégé EM)². Ces algorithmes itèrent sur 2 étapes, l'étape E et l'étape M. L'article (disponible sur l'ENT) détaille chacune de ces 2 étapes dans le cas qui nous intéresse.

1. On initialise $p_j^{(0)}$ et $q_j^{(0)}$
2. étape E : à l'étape k , on suppose que l'on connaît la sensibilité $p_j^{(k)}$ et la spécificité $q_j^{(k)}$ de chaque expert et on calcule, pour chaque pixel i , la probabilité qu'il fasse partie de la classe 1, que l'on note $W_i^{(k)}$.

Indication : Cette étape est décrite à la page 8 de l'article (équations 14, 15 et 16). $f(T_i = 1)$ et $f(T_i = 0)$ sont les probabilités a priori d'avoir l'une ou l'autre des classes et peuvent être estimées en calculant la proportion de 1 et de 0 dans les segmentations initiales (P1 et P0 dans le squelette fourni).

3. étape M : on suppose que l'on connaît $W_i^{(k)}$, et on met à jour la sensibilité $p_j^{(k+1)}$ et la spécificité $q_j^{(k+1)}$ de chaque expert.

Indication : Cette étape est décrite à la page 9 de l'article (équations 18 et 19).

4. On itère jusqu'à convergence. La segmentation finale est alors obtenue en affectant 1 aux pixels pour lesquels $W_i^{(k)} > 0.5$.

Vous trouverez dans `staple.m` un squelette de fonction à compléter ... complétez le! et testez le sur `Segm_toy` et `Segm_binaire`.

3 Courbe ROC

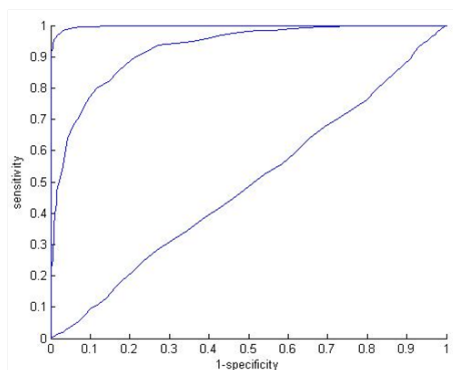


FIGURE 2 – Exemples de courbes ROC

2. https://en.wikipedia.org/wiki/Expectation-maximization_algorithm

Comme on l'a vu, la sensibilité et la spécificité associée à une segmentation sont deux mesures permettant de mesurer la qualité de celle-ci vis-à-vis d'une "vérité terrain". Une méthode de segmentation dépend très souvent d'un ou plusieurs paramètres, et une courbe ROC permet de représenter les différentes mesures de sensibilité/spécificité correspondantes aux segmentations associées à différents paramètres.

1. On souhaite tout d'abord segmenter l'image `I_toy` par seuillage. Tracez la courbe ROC correspondante, le paramètre qui varie étant donc le seuil utilisé. En déduire le seuil qui fournit la segmentation correspondante au meilleur compromis sensibilité/spécificité.

Indication : pour une segmentation donnée, pour pouvoir calculer sensibilité/spécificité on a besoin de la "vérité terrain". Pour l'image `I_toy`, celle-ci est un carré blanc sur un fond noir et peut être construite ainsi :

```
T = zeros(size(I_toy)); T(30:70,30:70) = 1;
```

2. La fonction `segmentation`, fournie dans `segmentation_LS.zip`, implémente la méthode "contours actifs / Level Set" (cf MATI). Cette fonction prend en argument l'image I à segmenter et un `sigma` indiquant l'écart-type de la gaussienne appliquée à l'image avant de calculer son gradient (pour obtenir l'image g_I qui servira de fonction d'arrêt dans l'évolution). Tracez la courbe ROC en faisant varier `sigma` entre 1 et 20. Identifiez le `sigma` optimal et comparez les deux méthodes (seuillage et contours actifs).