Segmentation Workshop

MPHYGB06: Expectation-Maximisation (Python)

Implement an Expectation-Maximisation based segmentation algorithm in order to classify and segment the brain's different tissues. The datasets to segment are in http://dl.dropbox.com/u/502383/EM_Lecture.zip

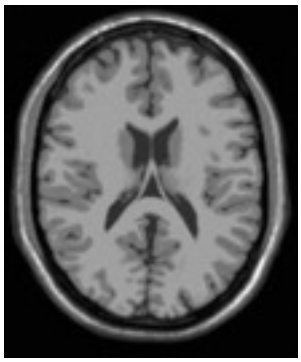
Basic Tasks:

Implement a segmentation algorithm that segments the brain into 4 tissue types: white matter, grey matter, CSF and others (skull, dura, skin, fat). This segmentation algorithm assumes that every pixel is independent. To speed up the implementation, a 1D vector can be created with all the pixels in the image, and the EM solved in a matricial/ vectorial fashion.

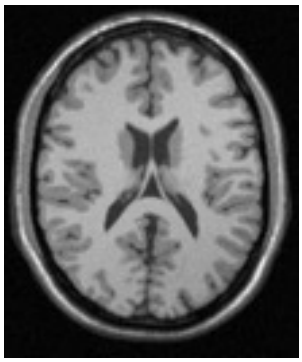
Write a function that takes the image and an initial set of parameters (number of classes, average, std and convergence ratio) and outputs the EM segmentation.

[WM,GM,CSF,Other] = EM_segment(image, k_classes, [averages], [stds], epsilon);

Use your implementation to segment the datasets ***brain.jpg***, ***brain_bias.jpg***, ***brain_noise.jpg***. Is it possible to segment and distinguish each tissue type without having spatial information?



brain.jpg



brain_bias.jpg

brain_noise.jpg

For this section you will only need these equations:

$$p_{ik}^{(m+1)} = \frac{f\left(y_i \mid z_i = e_k, \Phi_y^{(m)}\right) f\left(z_i = e_k\right)}{\sum_{j=1}^{K} f\left(y_i \mid z_i = e_j, \Phi_y^{(m)}\right) f\left(z_i = e_j\right)}$$

$$\mu_k^{(m+1)} = \frac{\sum_{i=1}^{n} p_{ik}^{(m+1)} y_i}{\sum_{i=1}^{n} p_{ik}^{(m+1)}}$$

$$\left(\sigma_k^{(m+1)}\right)^2 = \frac{\sum_{i=1}^{n} p_{ik}^{(m+1)} \left(y_i - \mu_k^{(m+1)}\right)^2}{\sum_{i=1}^{n} p_{ik}^{(m+1)}}$$

with,

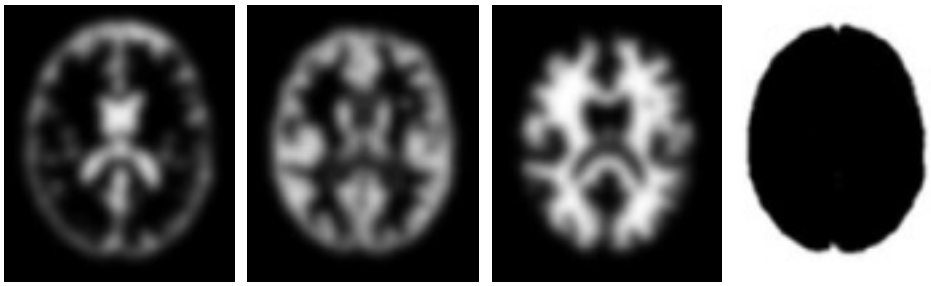
$$f(y_i \mid z_i = e_k, \Phi_y) = G_{\sigma_k}\left(y_i - \mu_k\right) \quad \text{and} \quad G_{\sigma_k}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x)^2}{2\sigma^2}}$$

Here, y_i is the intensity at pixel i, $F(z_i=e_k)=1$, and ϕ_y contains the model parameters μ_k (average) and σ_k (std). The value of p_{ik} contains the current estimate of the probability that voxel i belongs to class k and m is the current iteration number.

Advanced Tasks:

To improve the performance of the segmentation algorithm, several incremental changes can be done:

1. Create a new segmentation function that takes the image and the priors, and uses them to segment the image into its components. Also use these priors to initialize the averages and std of each class. Does it improve the segmentation. Don't forget that here, you should set $F(z_i=e_k)=\pi_{ik}$ were π_{ik} is the *a priori* probability for class k in pixel i. The priors for each class are shown bellow.



2. To add spatial consistency, create an MRF function with the matrix G as the energy functional. Here, you should replace $F(z_i=e_k)$ with:

$$f(z_i = e_k \mid p_{\mathcal{N}_i}^{(m)} \Phi_z^{(m)}) = \frac{\pi_{ik}\, e^{-\beta U_{\mathrm{MRF}}(e_k \mid p_{\mathcal{N}_i}^{(m)}, \Phi_z^{(m)})}}{\sum_{j=1}^{K} \pi_{ij}\, e^{-\beta U_{\mathrm{MRF}}(e_j \mid p_{\mathcal{N}_i}^{(m)}, \Phi_z^{(m)})}}$$

where,

$$U_{MRF}(e_k|p_{Ni}, \phi_z) = \sum_{j=1}^{K} \Big(\sum_{i=N_i^G} p_{ij} G_{kj} \Big)$$

, N_i is the neighbourhood for pixel i and,

G=

0	1	1	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Here, one needs to know the spatial relationship between all the voxels. Thus, using the 1D vectors is not ideal. Instead, use the reshape function to convert 1D arrays into the original 2D matrices.

A basic code structure can be found here: http://dl.dropbox.com/u/502383/EM.py

Literature:

• Van Leemput et al. ***Automated model-based tissue classification of MR images of the brain.*** IEEE Transactions on Medical Imaging (1999) vol. 18 (10) pp. 897-908

• Wells III et al. ***Adaptive segmentation of MRI data***. IEEE Transactions on Medical Imaging (1996) vol. 15 (4) pp. 429-442

• Zhang et al. ***Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm***. IEEE Transactions on Medical Imaging (2001) vol. 20 (1) pp. 45-57