Assignment 7: Segmentation

CS 473/673: Medical Image Processing

Due 4:00pm Monday 4 April 2016

[8 marks] **Segmentation using GMM Clustering**

In this question, you will complete the supplied Matlab function `GMMSegmentation` that implements a segmentation algorithm by performing clustering in the joint intensity space using a Gaussian Mixture Model (GMM). As the accompanying skeleton code indicates, the function's calling prototype is,

```
[memb phi] = GMMSegmentation(stack, phi0)
```

See the function's help for a description of the input and output variables. Your solution should use the Estimation/Maximization (EM) method, similar to that discussed in Lesson 36. The EM algorithm using a GMM is outlined below. In the algorithm, n indexes the pixels, k indexes the Gaussian components, I_n represents the column vector of intensity values for pixel n, τ_{nk} is the membership of pixel n to component k, and μ_k, Σ_k and π_k are the mean, covariance and mass of Gaussian component k.

EM Algorithm using a GMM

1. Start with D registered images and an initial GMM with K components, where each component is characterized by a mean vector ($\mu_k \in \mathbb{R}^D$) and covariance matrix ($\Sigma_k \in \mathbb{R}^{D\times D}$).

$$\phi_k = (\mu_k, \Sigma_k) \quad , \quad k = 1, \ldots, K$$

2. Update the pixel memberships, τ_{nk}, using

$$\begin{aligned} \tau_{nk} &= \frac{\pi_k N(I_n|\phi_k)}{\sum_k \pi_k N(I_n|\phi_k)} \ , \\ \text{where} \quad N(I|\phi_k) &= \frac{1}{\sqrt{(2\pi)^D|\Sigma_k|}} \exp\left(-\frac{1}{2}(I-\mu_k)^{\mathrm{T}}\Sigma_k^{-1}(I-\mu_k)\right) \end{aligned}$$

   and D is the number of images, and hence the dimension of the joint space.

3. Update the means of the Gaussian components using

$$\mu_k' = \frac{\sum_n \tau_{nk} I_n}{\sum_n \tau_{nk}} \ .$$

   Note that this formula uses the updated membership values just calculated above.

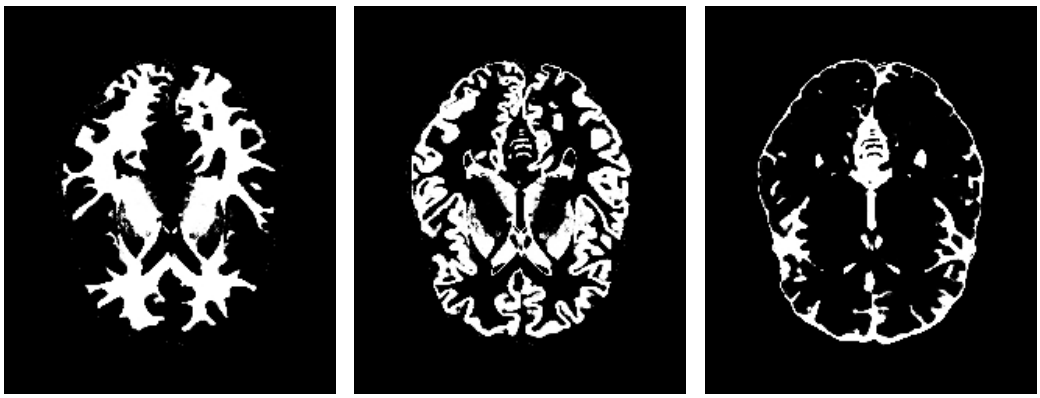
4. Update the covariance matrices using

$$\Sigma_k = \frac{\sum_n \tau_{nk}(I_n - \mu_k')(I_n - \mu_k')^{\mathrm{T}}}{\sum_n \tau_{nk}} \ .$$

   Notice that the formula uses the updated means.

5. Update the component masses using

$$\pi_k = \frac{\sum_n \tau_{nk}}{\sum_k \sum_n \tau_{nk}} \ .$$



(a) White Matter (b) Gray Matter (c) CSF

Figure 1: Sample membership maps using the larger BrainWeb dataset (the other two components are not shown).

I am supplying you with a Matlab script called `segmentation_test.m`. It reads in three images (so $D = 3$), sets up the initial GMM with four components (so $K = 4$), and calls `GMMSegmentation`. It also displays the resulting memberships. The script can use two different datasets, one big and one small. I suggest you use the small one for initial development of your code, but make sure it works on the full-resolution dataset. The script selects D pixels as initial μ's for the Gaussian components. You can edit the script so that the output looks similar to the images in Figure 1.

Notes

- It is possible for the memberships of a pixel to be essentially zero. For example, if one pixel's intensity vector, I, is far away from all the Gaussian components, then $N(I|\phi_k)$ will be negligible for all k. This causes divide-by-zero problems when normalizing the memberships. So, if the sum of the memberships for a given pixel is smaller than 10^{-10}, then simply set the memberships for the pixel to be $1/K$, where K is the number of Gaussian components.
- The supplied starter code for `GMMSegmentation` displays the membership maps after each iteration. It also calls a function called `ScatterPlot` that displays the joint intensity scatter plot with a set of ellipses that represent the dominant axes of each of the K Gaussian components. The small crosses show the component means. You might find it useful to display this scatter plot as your EM iteration progresses. See the help file for more information on how to use the function.
- You might find that the method converges better if you prevent the Gaussian components from updating their means (μ_k) for the first 3 iterations.
- Continue iterating until you either reach 30 iterations, or the average change in membership values (from one iteration to the next) falls below 0.002. That is, iterate for 30 iterations, or until

$$\frac{1}{NK}\sum_k\sum_n |\tau_{nk} - \tau'_{nk}| < 0.002 \ ,$$

  where N is the number of pixels, τ'_{nk} are the updated memberships and τ_{nk} are the previous memberships.

What do I submit?
You have to submit two things: your code for `GMMSegmentation.m`, and your edited version of `segmentation_test.m`.