Expectation Maximisation Algorithm for Brain Tissue Segmentation

Berta Calm Salvans[1] and Abdelrahman Habib[1]

[1]Universitat de Girona

October 25, 2023

1 Introduction

Automatic brain tissue segmentation into grey matter (GM), white matter (WM), and cerebrospinal fluid (CSF) from magnetic resonance images (MRI) has helped diagnose various types of neuro-disorders, such as multiple sclerosis, Alzheimer's, etc [1]. Different types of segmentation approaches were proposed in the literature, which include statistical segmentation techniques that have been widely used for segmenting brain MRI. In these methods, the voxels are used to characterize a Gaussian Mixture Model (GMM) as cited by [2]. These models are defined as a weighted sum of finite Gaussian components, where each of them is used to model the distribution of voxels from one tissue type. All voxels are weighted using the prior probability that corresponds to each tissue type [2]. As cited by [2], the GMM parameters are estimated using an Expectation-Maximization (EM) algorithm, where a Gaussian classifier is applied to obtain the class label of each voxel.

2 Objectives

The objectives of this laboratory session are to:

(a) Research information about EM algorithm implementation in lab groups.

(b) To understand, design, analyse and implement EM segmentation algorithm using Python.

(c) To evaluate the performance of the algorithm using the provided images: study the problems and possible improvements, and assess the results using the ground truth and the Dice similarity measure.

(d) To report the procedure and results obtained after the implementation.

3 Tools & Methods

3.1 Skull Stripping

As an initial step of the EM algorithm, skull stripping had to be performed to only segment the regions of the three brain tissues: CSF, GM, and GM. This step is performed using the provided ground truth (GT) volume, T1 sequence volume, and T2-FLAIR sequence volume. The GT labels volume was firstly binarized in order to create a binary mask of the brain tissues, and used in segmenting as it only contains the three brain tissues. This mask is then used to index and obtain the tissue intensity data from both sequences in the case of multi-modality or single sequence in the case of a single modality. The final tissue data array contains all of the tissue pixel values for the three regions, stored in columns. The shape of the tissue data is $N \times d$, where N is the number of samples, and D is the d-dimensions. This is all performed in a function called *perform_skull_stripping*.

3.2 Model Parameter's Initialisation

To start the algorithm two initialisation approaches were taken into consideration: random and K-Means initialisation. Both methods were used to obtain the mean and the covariance matrices, parameters used later to update the model performance. This initialization step was performed after the skull stripping step 3.1. For the K-Means initialisation, the data, either in multi-modality or single-modality dimensions was fitted to a K-Means function provided by sklearn. From there, both the K-Means labels and the cluster centroids were taken for initialization. The centroids represent the centre of the clusters (the mean), which has a shape of

$K \times d$, where K is the number of K clusters, and d is the dimension of the data fitted. For random initialization, random centroids were generated in the range between the minimum and maximum values of the data, and random labels were generated up to the maximum number of K clusters, for our case, K is equal to 3. The covariance matrices for each cluster were initialized for each cluster data separately, taking the shape of $K \times d \times d$. This process was performed in a function called *initialize_parameters*.

3.3 Expectation Step

Brain MR images are denoted as a set of D-dimensional vectors $X = \{x_i; i = 1, 2, ..., N\}$, where $x_i \in R^D$ characterizes voxel i [2]. According to the GMM, each of the x_i samples is assumed to be drawn independently from one of the K Gaussian distributions, each of which models the voxels from one brain tissue type [2]. The formula used to express the finite mixture model with K components is demonstrated in equation 1.

$$p(\underline{x}|\Theta) = \sum_{k=1}^{K} \alpha_k p_k(\underline{x}|z_k, \theta_k) \tag{1}$$

In equation 1, the $p_k(\underline{x}|z_k, \theta_k)$ are the mixture components, $1 \leq k \leq K$. Each is a density or distribution defined over $p_k(\underline{x}|z_k)$, with parameters θ_k that we estimate in our implementation. The model parameters are μ_k which refers to the mean of class k, and $\sum_k$ which refers to the covariance matrix of class k. In addition to those parameters, $\alpha_k = p(z_k)$, which are the mixture weights are computed to represent the probability that a randomly selected $\underline{x}$ was generated by component k, where $\sum_{k=1}^{K} \alpha_k = 1$. Those three parameters were also initiated inside the *initialize_parameters* function.

To estimate the GMM for each K component, the multivariate Gaussian density equation 2 was implemented from scratch inside the *expectation* function to compute the posteriors.

$$p(\underline{x}|\theta_k) = \frac{1}{(2\pi)^{d/2} |\sum_k|^{1/2}} \\ e^{\frac{-1}{2}(\underline{x}-\underline{\mu_k})^t \sum_k |^{-1} (\underline{x}-\underline{\mu_k})} \tag{2}$$

The membership weights, which is the last step of the expectation step, are used to compute the probability of data point $\underline{x_i}$ to belong to each of the clusters k. It is denoted as $w_i k$ and has a shape of $N \times k$, where N is the number of data points, and cluster k. The formula for calculating the membership weights is shown in equation 3. The calculation of the membership weights is repeated for all k clusters, and for all data points. This calculation is also demonstrated in the *expectation* function.

$$w_{i_k} = \frac{p_k(\underline{x}|z_k, \theta_k).\alpha_k}{\sum_{m=1}^{K} p_m(\underline{x_i}|z_m, \theta_m).\alpha_m} \tag{3}$$

3.4 Maximisation Step

Once the membership weights are estimated during the expectation step, these weights together with the data are used to update the parameter values. The mixture weights, the means and the covariance matrices are recalculated again to maximize the likelihood. First, the total weight is obtained by summing all the weight from each data point from each cluster, see equation 4.

$$N_k = \sum_{i=1}^{N} w_i k \tag{4}$$

These values are used to calculate the new mixture weights (${\alpha_k}^{\text{new}}$), using equation 5, where N is the number of data points.

$${\alpha_k}^{\text{new}} = \frac{N_k}{N} \tag{5}$$

Then, the new mean for each cluster (${\mu_k}^{\text{new}}$) is obtained, using equation 6. Where for every cluster, each data point is multiplied by the corresponding weight, and the sum of all these values is divided by the sum of all weight of the specific cluster.

$${\mu_k}^{\text{new}} = \frac{1}{N_k} \sum_{i=1}^{N} w_{ik} \cdot x_i \tag{6}$$

Finally, the updated mean is used to obtain the covariance matrix, see equation 7.

$$\sum_{k}^{new} = \frac{1}{N_k} \sum_{i=1}^{N} w_{ik} \cdot (x_i - {\mu_k}^{\text{new}})(x_i - {\mu_k}^{\text{new}})^T \tag{7}$$

Once all the parameters have been updated the M step is finished. The next step would be to start again the E step and calculate new membership weights using the updated parameters. Following an iterative process, the algorithm will keep updating all the parameters to find the most suitable ones. This step is also demonstrated in the *maximization* function.

3.5 Log-likelihood

As mentioned, the EM algorithm iteratively repeats the E and M steps. For this, a log-

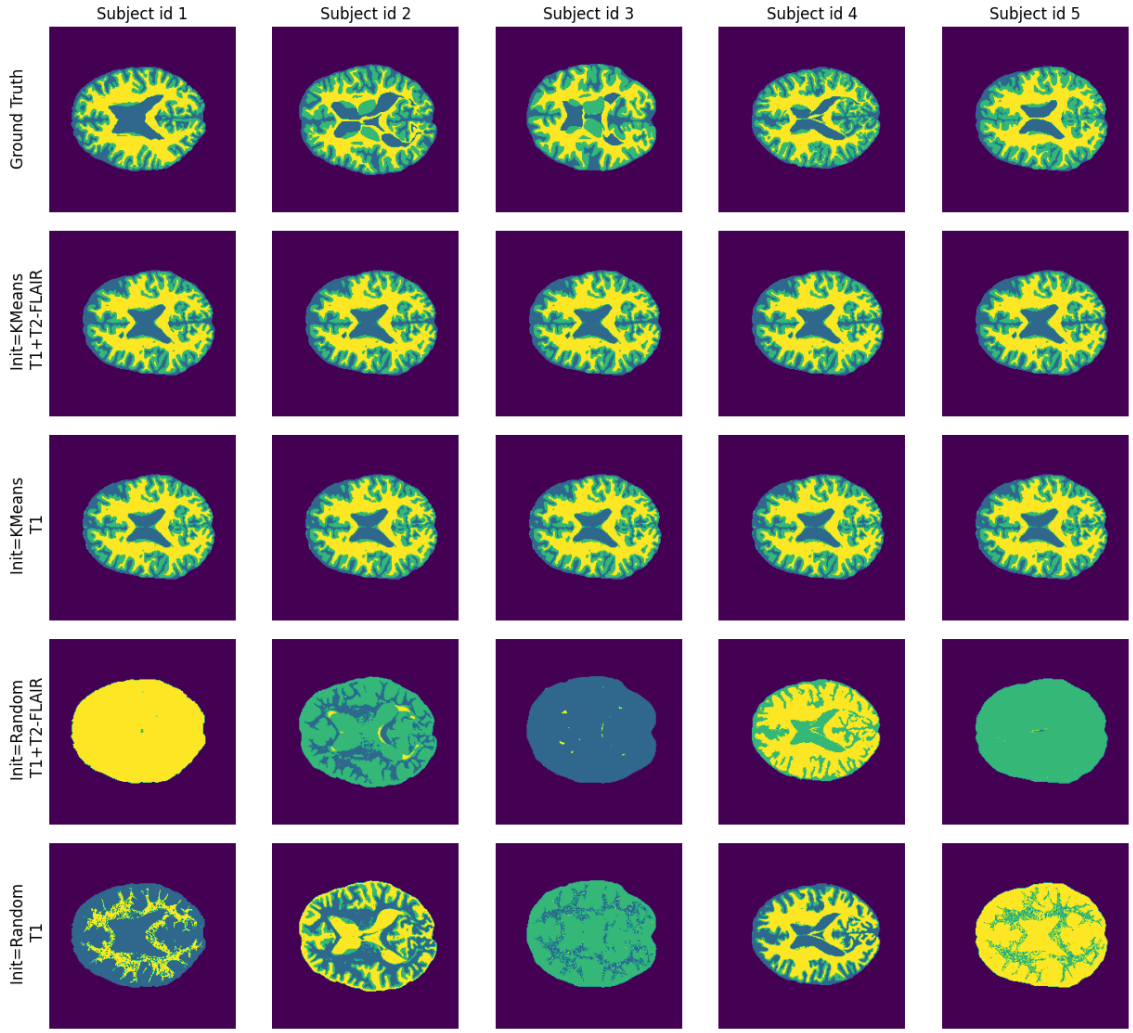

Figure 1: Final segmentation results for five subjects using different initialization techniques and different modalities.

likelihood function is introduced to stop the iterations when the model converges (equation 8).

$$\log l(\Theta) = \sum_{i=1}^{N} (\log \sum_{k=1}^{K} \alpha_k p_k(\underline{x}|z_k, \theta_k) \quad (8)$$

Convergence is detected by computing the value of log-likelihood after each E step and stopping the iterative process if the difference between the last value and the previous value is below a predetermined threshold. This step is also performed in the *log_likelihood* function.

3.6 Prediction Label's Correction

The final result of the EM algorithm is obtained from the final value of the membership weights when the algorithm converges. Those weights are used to reflect to which cluster the data point belongs, the point will always belong to the highest probability cluster. The prediction list is the same size as the N data points and is used to create a segmentation mask. This segmentation mask, for most of the cases observed, had labels different from the GT labels file for the three tissues. We created a simple function to map the wrong labels of the segmentation mask (predictions) to the GT labels, based on the different labels that we observed for the five subjects. This can be demonstrated in *correct_pred_labels* function.

4 Results and Discussion

We have evaluated the results by visual inspection and using the Dice score, by comparing the results to the provided GT file. All the segmentation results, using both K-Means and random initialization, for single and multi-modality can be seen in figure 1. The Dice score obtained for each case is shown in Table 1. The code used in the evaluation is implemented in $calc\_dice\_coefficient$ function, and the for-

Table 1: Dice score results for each patient for each brain tissue using both initialization approaches and different modalities configurations.

Configuration	Tissue	1	2	3	4	5	Average
K-Means (T1+T2)	CSF	0.90388	0.88434	0.87893	0.89787	0.86860	0.88670 ± 0.01272
	GM	0.80141	0.75029	0.79918	0.79549	0.83692	0.79666 ± 0.02757
	WM	0.86039	0.76112	0.87095	0.83896	0.89558	0.84540 ± 0.04591
K-Means (T1)	CSF	0.87772	0.86405	0.87103	0.87863	0.85959	0.87020 ± 0.00747
	GM	0.77720	0.74223	0.78328	0.77778	0.82961	0.78202 ± 0.02789
	WM	0.85852	0.78454	0.87520	0.83363	0.88558	0.84749 ± 0.03604
Random (T1+T2)	CSF	0.00024	0.01490	0.49235	0.86201	0.00628	0.27516 ± 0.34848
	GM	0.00031	0.64301	0.0	0.00575	0.61755	0.25332 ± 0.30789
	WM	0.42938	0.73676	0.02165	0.59892	0.00021	0.35738 ± 0.29924
Random (T1)	CSF	0.60115	0.82287	0.00092	0.87479	0.00089	0.45979 ± 0.38644
	GM	0.00346	0.67389	0.58539	0.63099	0.67784	0.51431 ± 0.25762
	WM	0.69172	0.78771	0.55535	0.78109	0.61536	0.68624 ± 0.09108

mula used for dice score is demonstrated in equation 9. To avoid the division by zero in any scenario a small value of epsilon $\epsilon = 1e - 8$ was added to the formula.

$$DSC = \frac{2 * |X \cap Y|}{|X| + |Y| + \epsilon} \tag{9}$$

Results show that K-Means initialisation had significantly better segmentation results than the random initialisation, given also that our approach for correcting the segmentation labels as in section 3.6 corrected the labels to match the GT labels for the three tissues. Both modality configurations, single- and multi-modality, had low to no segmentation accuracy. Random initialisation with a single T1 modality was quite close in some cases as in subject 4, while the random initialisation with the two modalities was significantly less accurate.

Overall, K-Means initialisation presented similar Dice results for single and multi-modal approaches. In the initialisation using both modalities, the model performed the best on CSF and GM tissues, resulting in 0.88670 ± 0.01272 and 0.79666 ± 0.02757 respectively. While the model outperformed WM tissue segmentation with an average dice score of 0.84749 ± 0.03604.

Finally, by implementing the log-likelihood function, segmentation results improved compared to when defining the number of iterations. This, ensured the use of optimal parameters for the cluster-based segmentation. When defining the number the iterations could lead to over-segmentation if the number was too big, or under-segmentation if the number was not enough to reach a good segmentation accuracy. For K-means initialization and multi-modality the log-likelihood parameters were plotted during 20 iterations, and when the function converged marked with a red dot, see figure 2. Showing that by manually defining the number of iterations, the accuracy of the result can be negatively affected. Also, in figure 2, in the right graphs, it is also shown how the dice coefficients increase until the log-likelihood function converges and the segmentation result is obtained. In each case, all dice scores gradually increased, maximizing the likelihood, except for subject 5, where dice scores for CSF and GM decreased slightly at the end.

Conclusions

This study shows that EM is suitable for medical image segmentation, that through an interactive process is able to estimate the parameters and compute the clustering based on prior probability even with incomplete data. However, the result depends on the parameter initialization and can affect directly the final results and the number of clusters needs to be known beforehand.

References

[1] Pulkit Kumar, Pravin Nagar, Chetan Arora, and Anubha Gupta. U-segnet: Fully convolutional neural network based automated brain tissue segmentation tool. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 3503–3507, 2018.

[2] Guangjian Tian, Yong Xia, Yanning Zhang, and Dagan Feng. Hybrid genetic and variational expectation-maximization algorithm for gaussian-mixture-model-based brain mr image segmentation. *IEEE Transactions on Information Technology in Biomedicine*, 15:373–380, 5 2011.

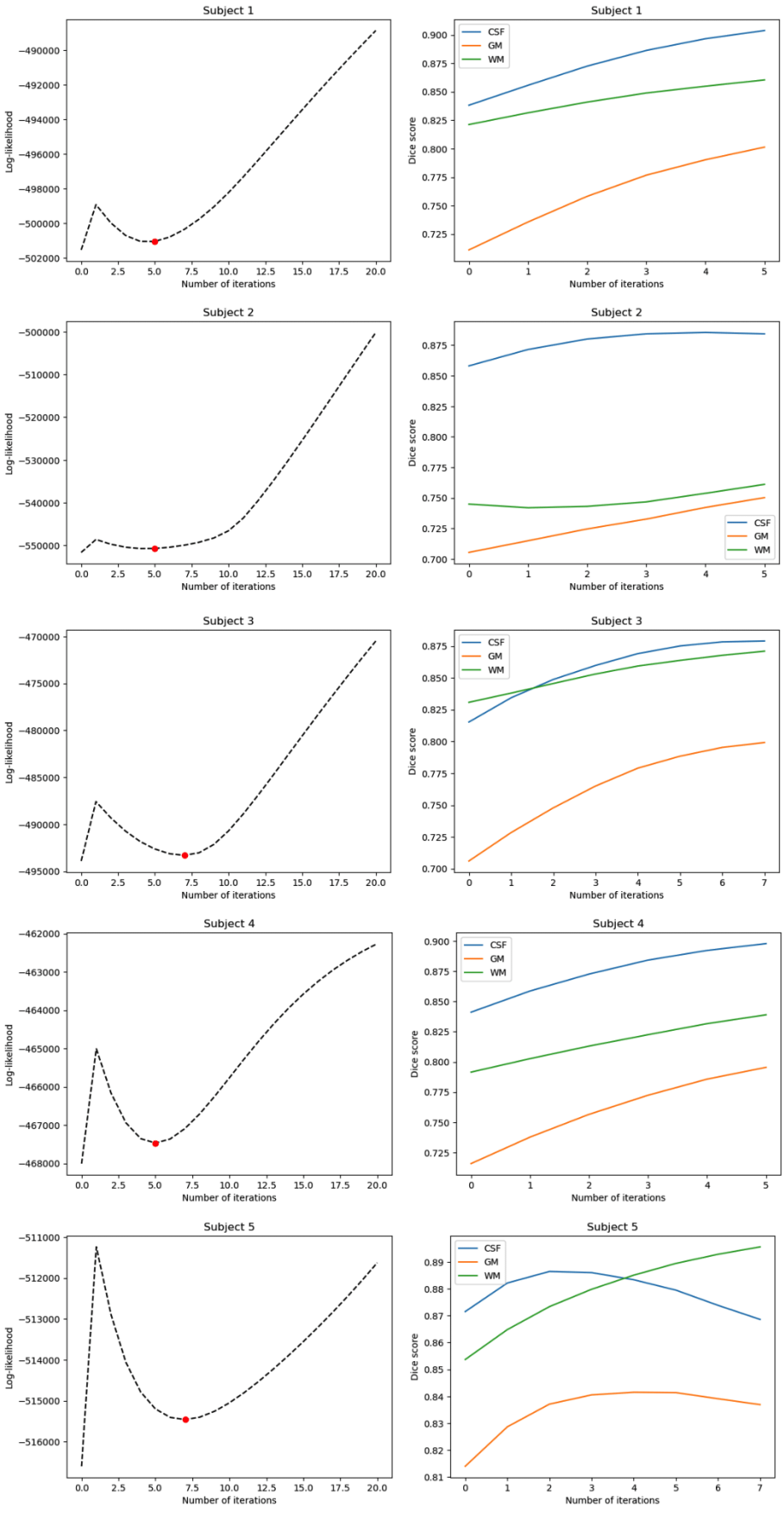

Figure 2: Log-likelihood function and dice scores plotted for each iteration for K-Means initialization and multi-modality approach.