UNIVERSITAT DE GIRONA, MEDICAL IMAGE SEGMENTATION AND APPLICATIONS

Report: EM Algorithm | 2021-2022

OJEDA, Manuel; VASILE, Alexandru

October 26, 2021

1 EM-EXPECTATION MAXIMIZATION

1.1 Introduction and problem definition

The Expectation-Maximization (EM) algorithm is an iterative technique designed for probabilistic models. It is often used for finding the unknown parameters of a mixture model. Likelihood is the probability of a sample of belong to a class, given the parameters of that class. EM maximizes the log-likelihood of the sample as represented by the mixture model: maximizing the likelihood we can fit a mathematical model to the data.

This practical laboratory is to get the algorithm working with the brain images provided to get the 3 different classes: *white matter (WM), gray matter (GM), cerebrospinal fluid (CSF).*

1.2 Algorithm analysis

- **INITIALIZE the parameters of the model:** the initial values for mean, variances and proportions.
- **REPEAT:**
  - **EXPECTATION STEP:** Recompute labels for all the data set given the current cluster parameters.
  - **MAXIMIZATION STEP:** Use that classification to estimate the parameters again.
- **Until CONVERGES to a local minimum:** it is checked if the parameters do not have any significant changes, if they do not, it stops.

1.3 Proposed solution

Our solution is of a holistic nature as it uses information from both channels (T1 and T2 FLAIR), processes and segments the whole multidimensional array.

We worked with the concept of "Tracer Bullet Code", meaning that the first implementation of our project was very unpolished and didn't meet all the requirements, for example, it was only developed for one channel. With time, we added more layers of complexity, covering all the

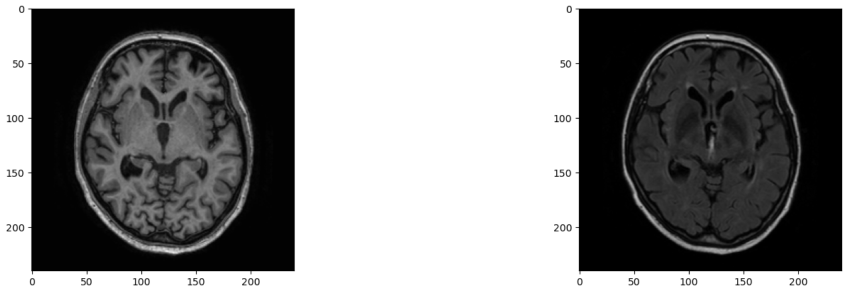
conditions of evaluation and even adding an enhancement to the Kmeans initialization that makes it more robust.
The framework was the one proposed in the paper and the programming was carried out in a sequential manner as the need for an object oriented approach was non-existent.
One other programming paradigm that was implemented was coding "comments first", meaning that the code does what the comments dictate. This is a very intuitive way of implementing complex equations like the ones that were associated with this project.
From an image processing perspective, the NIfTI files are loaded and transformed into 3D arrays, only to be fully vectorized later and be processed in this form for the whole of the EM section in which clustering was performed in a 2D space as information from both T1 and T2 FLAIR was used. After the segmentation by clustering was carried out, the image is then reconstructed from the vectors and is displayed along with metrics.

1.4 Experimental section and results

The experiments on the algorithm were realized in two different ways:

- The first one was with a completely random initialization of the variables.
- The second one was to apply the algorithm of *K-Means* and those values are the initial values of the variables.

All the results shown were obtained using the 20th from the folder "2". In the Figure 1.1 both original images can be seen, from the T1 and T2 FLAIR files.



(a) Original T1 image.

(b) Original FLAIR image.

Figure 1.1: Original images obtained from the data set.

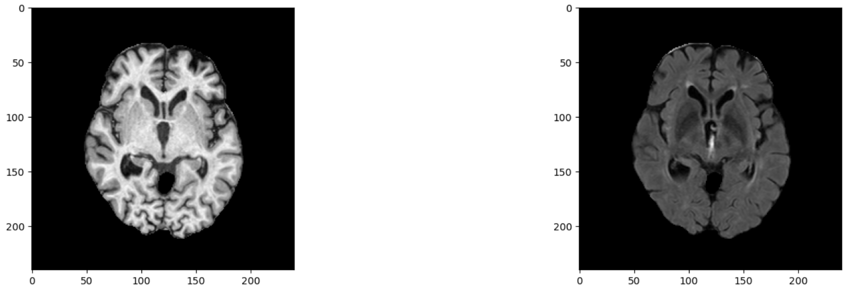
In order to be able to work with the proper regions, a pre-processing step is necessary. The figure 1.2 shows the results of this step, it can be visualized only the main brain tissue.

1.4.1 Random Initialization

In Figure 1.3, the results for the same slice, number 20, can be observed after running the algorithm. At this stage, the initialization was realized completely randomly.

1.4.2 K-Means Initialization

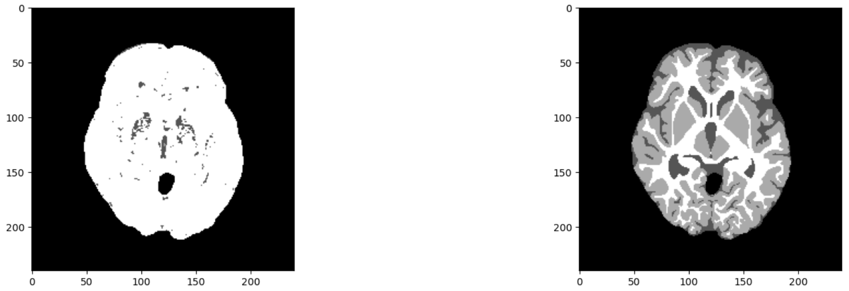
In the Figure 1.4, it can be observed the results for the same slice, number 20, after running the same algorithm again, the difference is the initialization. For this stage, the function K-Means was applied at the beginning to have the initial values for the clusters.



(a) T1 image without skull.

(b) FLAIR image without skull.

Figure 1.2: Images obtained after pre-processing.



(a) Result with random initialization.

(b) Original image with its labels.

Figure 1.3: Comparison between the random initialization result and the ground truth.

In the table 1.1, it can be observed the different values for the Dice coefficient depending on the region, also it can be observed the amount of time it took to run the algorithm.

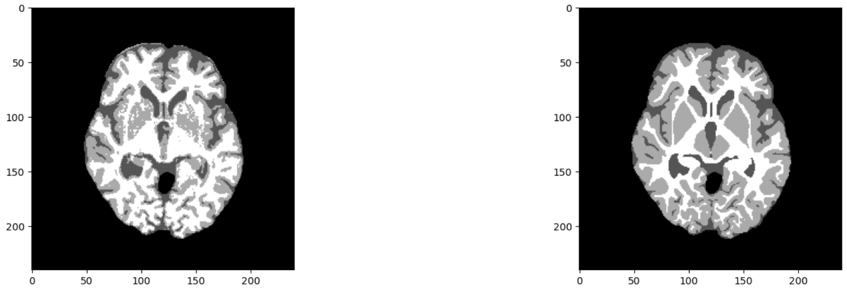
Table 1.1: Comparison of the results between the random and the k-means initialization.

Dice coefficient and computational time (seconds)			
	WM Dice/time	GM Dice/time	CSF Dice/time
Random	0.4/456.10	0/456.10	0.0648/456.10
K-Means	0.711/422.73	0.677/422.73	0.86/422.73

1.5 Project Management details

The plan for the first week of the task was for both team members, to look for information related to the algorithm. To fully understand how the algorithm works from the beginning, which approaches are the best ones for the task. A balance between performance and time consuming to implement. After looking for more information related to the algorithm, the coding part began with analyzing the data provided and get to know how to process it, if necessary.

The plan for the second week of the task was to code the algorithm stage by stage in MatLab



(a) Result with K-Means initialization.

(b) Original image with its labels.

Figure 1.4: Comparison between the K-Means initialization result and the ground truth.

with the visual help of images, plots of histograms and the "Command window", this last one to see the values and vectors and to analyze what happens with the data. During this week, the report was done and the experiments ran, in order to have the results and have the quantitative and qualitative analysis.

At the end, the real time dedicated was not the fully two weeks because of the different tasks from different subjects. Due to deadlines with different activities, the time dedicated to achieve the algorithm to be working was less as planned. Also the pipeline was developed with Python, because several problems were encountered in MatLab, the most relevant is: the functions sometimes did not work as expected so the functions were implemented line by line. Of course we could do the "from scratch" implementation with MatLab as well, but since this project will tie into the ATLAS implementation, it was all the more incentive to transfer the project to a more generalist language that allows for more flexibility of implementation.

1.6 Conclusions

From a qualitative point of view, the algorithm performs quite well when paired with Kmeans as an initialization method. The result coming from the kmeans implementation are definitely visually closer to the ground truth, although edges can seem jagged and some blocking can be observed, this is typical of clustering methods. The initialization with random distributions seems to generate an image in which white matter covers almost all the cranial space, this may be because of a local maximum that was encountered during the optimization.
From a qualitative point of view, Kmeans initialization is once again, the superior method, with the Dice score indicating better segmentation than in the case of random parameter initialization. Not only this, but also the time elapsed until convergence was consistently lower throughout our experiments. Unfortunately, due to format constraints, examples from all cases could not be presented in this report but by simply running the code with different paths, one can test this otherwise intuitive conclusion.
The EM algorithm is a robust segmentation tool that can find its place in various pipelines as it has an intuitive theoretical basis that has its roots in fuzzy logic.