Assignment 2 Report

Jishnu Basavaraju, Akash Cherukuri, Silky Kumari

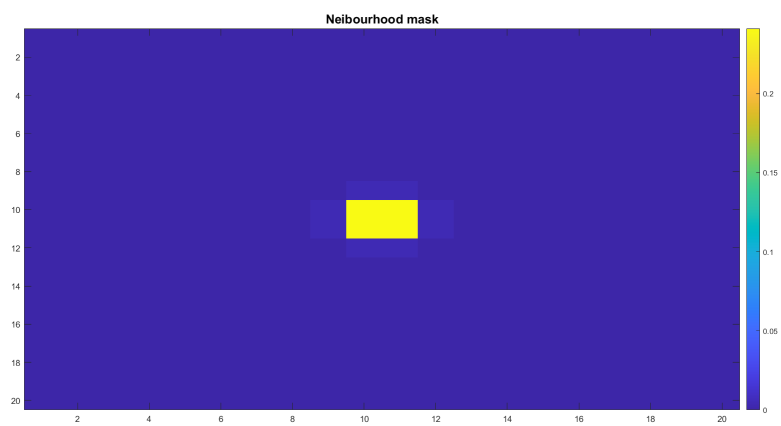
180050021, 190050009, 190070063

1 Segmenting a Brain Magnetic Resonance (MR) Image.

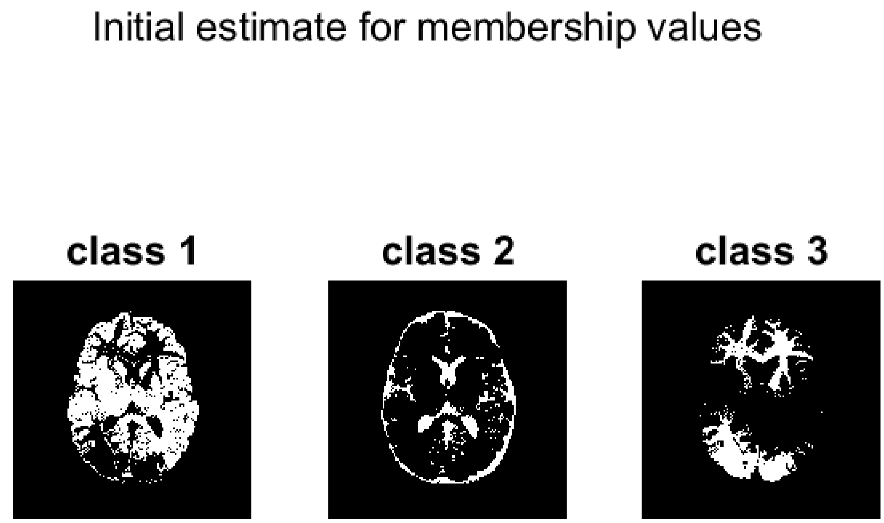
The code responsible for computing the optimal value of class means, memberships and bias field at every iteration has been commented appropriately.

(a) The chosen value of q is 1.6.

(b) The neighbourhood mask w_{ij} which has a window size of 20 can be seen as an image as follows:



(c) We performed kmeans image segmentation on the corrupted image and the mean estimates were used to calculated the membership function. Since kmeans is used for hard segmentation, the membership of the class whose mean was closest to the pixel intensity was set to 1 and the membership of the remaining classes were set to 0.
The image displaying the initial estimate is as follows:



(d) The initial estimates for the class means have been chosen by kmeans image segmentation as kmeans is one of the best algorithms for hard segmentation. It was implemented using MATLAB's pre-defined 'imsegkmeans' function.

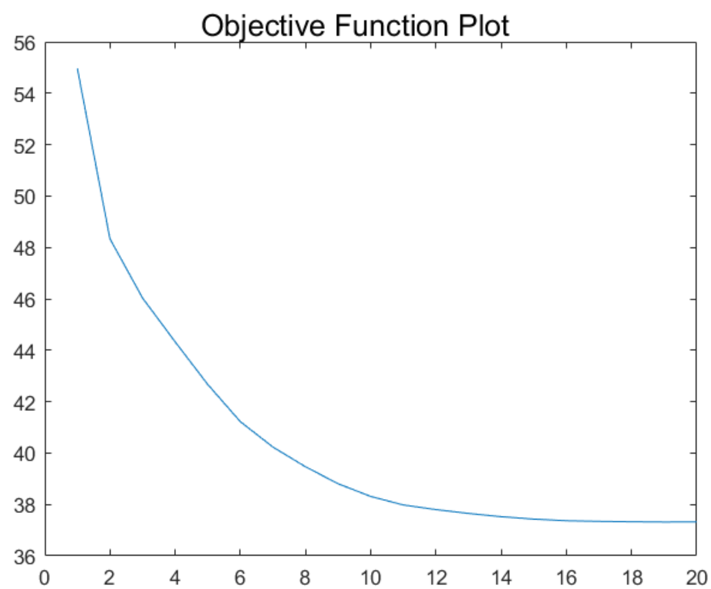
Class 1	Class2	Class 3
0.4571	0.2525	0.6361

Table 1: Initial Class Means

(e) A plot of the objective function with every iteration has been shown in the below graph.

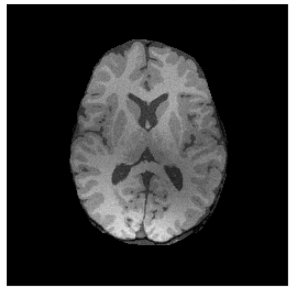
Objective function value each iteration
54.9641
48.3349
46.0312
44.3264
42.6679
41.2189
40.2319
39.4644
38.8053
38.3111
37.9793
37.7992
37.6501
37.5233
37.4290
37.3655
37.3411
37.3264
37.3167
37.3198

Table 2: Objective function values



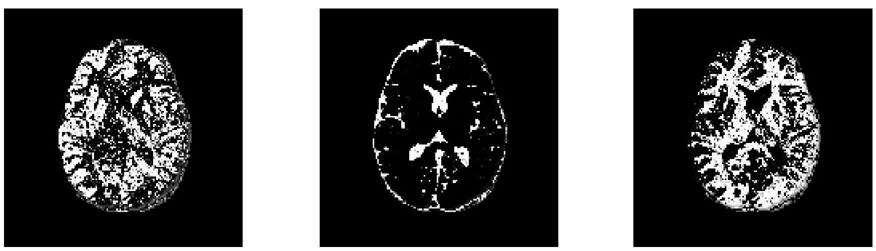
(f) The images are as follows:

Corrupted Image

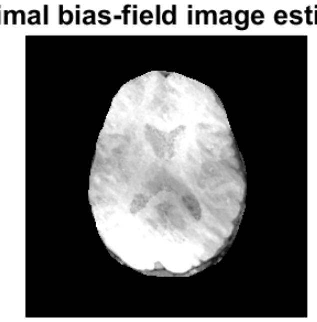


Optimal class-membership image estimates

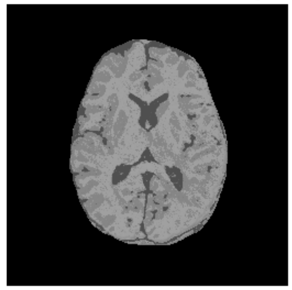
class 1 class 2 class 3



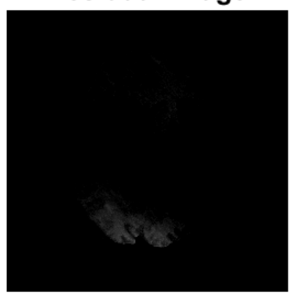
Optimal bias-field image estimate



Bias-removed image



Residual image



(g) The optimal estimates for the class means are:

Class 1	Class2	Class 3
0.5244	0.3092	0.6280

Table 3: Optimal estimates of Class Means

Uniqueness of Solution

The implementation discussed in class does not guarantee a unique solution. In our formulation, the only known parameter is y_i in the equation:

$$y_i = x_i * b_i + n_i$$

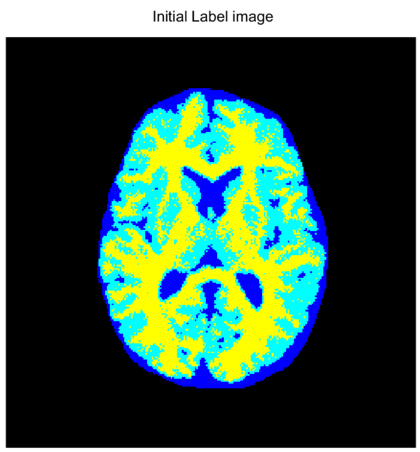
Thus there does not exist a unique solution to the estimated values of x_i and b_i.

2 Segmenting a Brain Magnetic Resonance (MR) Image.

The submitted code uses the EM algorithm for segmenting a Magnetic Resonance image. Again, the code has been commented to emphasize the parts which compute the optimal values of class means, memberships and the bias field at every iteration.

(a) The chosen value of β is 50.

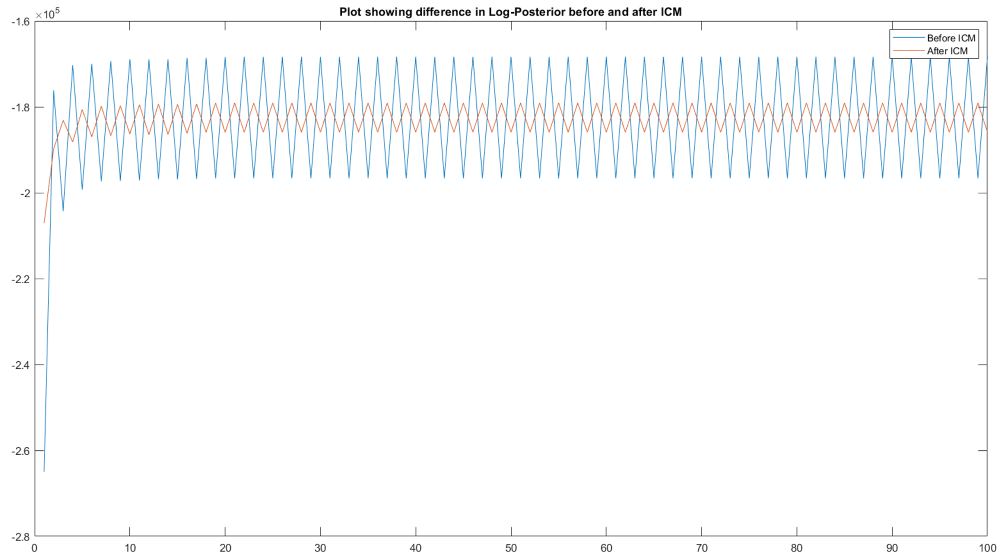
(b) Initial estimate for the label image has been obtained by using the KMeans clustering algorithm on the image. It was observed by playing around with the given image that the KMeans clustering algorithm produced "pretty good" clusters, and it was felt that these clusters would act as a great starting point. This is the motivation behind choosing the KMeans algorithm for label image initialization. The initial label image obtained is:



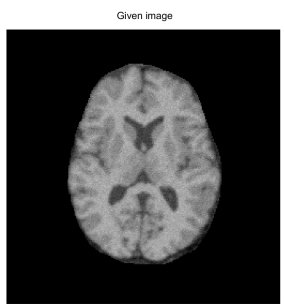
(c) The initial estimates for the Gaussian parameters $\theta = (\mu_i, \sigma_i)$ for a particular cluster are obtained by calculating the empirical mean and variance for points that were classified under that cluster by the KMeans initialization. The motivation behind this is the same as mentioned previously.

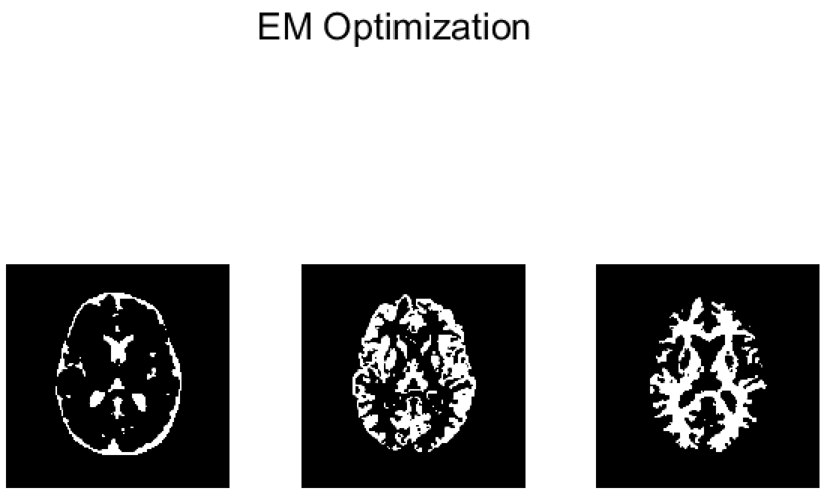
Parameter	**Initial Value**
μ_1	0.2931
σ_1	0.0655
μ_2	0.5102
σ_2	0.0413
μ_3	0.6305
σ_3	0.0358

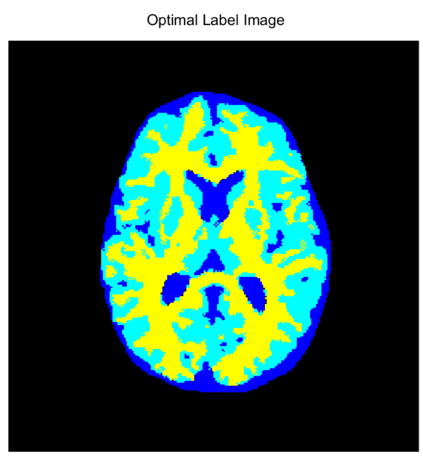
(d) The values of the log probability $\log P(x|y, \theta, \beta)$ before and after the ICM update with every iteration has been plotted below:



(e) The images requested are given below:

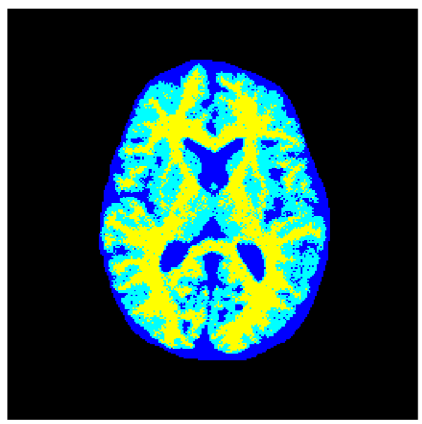



EM Optimization


Optimal Label Image

Optimal Class Image Estimates with β = 0

Optimal Label Image with β=0



(f) The optimal estimates for the class means for the chosen value of β are:

Class1	**Class2**	**Class3**
0.2949	0.5162	0.6316

3 Extending EM framework with prior information

Since the prior on the parameters θ are known, trying to maximize the posterior is a valid way to approach this problem.

Let θ denote the parameters, Y denote the observed variables and X denote the hidden parameters introduced by the EM optimization method. By applying Baye's rule, we get;

$$P(\theta|y) = \frac{P(y|\theta)P(\theta)}{P(y)}$$
$$\log P(\theta|y) = \log P(y|\theta) + \log P(\theta) - \log P(y)$$

We now introduce the hidden parameters and marginalize over them using the EM algorithm.

E Step

Let the function used in the EM Algorithm be denoted by F'. Replacing the log-likelihood with its definition, we get;

$$\log P(\theta|y) = F' + KL(q||P(x|y,\theta)) + \log P(\theta) - \log P(y)$$

$$F' = \log P(\theta|y) - KL(P(x|y,\theta^t)||P(x|y,\theta)) - \log P(\theta) + \log P(y)$$

This is the formulation for the F' which was discussed in class. From this, we now define the F relevant to this question as follows:

$$\begin{aligned} F &= E_q\left[P(\theta) - KL(q||P(x,y|\theta))\right] \\ &= \sum_x q(x)\log(q(x)/P(x,y|\theta)) + \log P(\theta) \end{aligned}$$

It can be seen that the last two terms are not functions of q. Therefore, computing the value of q turns out to be exactly the same as the computations performed in class. Therefore, using the results used in class we get the following value of q at the t'th iteration:

$$q(.) = P(x|y,\theta^t)$$

Therefore, the function F the needs to be maximized at every iteration would be given by;

M Step

A value of θ to maximize the value of F would be computed by the following equation:

$$\theta^* = \arg\max_\theta \left[\sum_x q(x)\log(q(x)/P(x,y|\theta)) + \log P(\theta)\right]$$

Application to GMM+EM

The priors for the weights, mean vectors, and covariance matrices has been mentioned below.

- **Means**
  A multivariate Gaussian is used to model the prior of μ_k. Here, d is the number of dimensions, I_d is the identity matrix of size $d \times d$ and 0 is the zero matrix of the same size.

  $$\mu_k \sim \mathcal{N}(0, I_d)$$

- **Weights**
  We know nothing about the weights, so we chose the most general (and the easiest to handle) distribution, the uniform distribution.

  $$W_k \sim \mathcal{U}(0,1)$$

- **Covariances**
  Using the hint provided, an Inverse-Wishart distribution is used to model the prior of the inverse covariance matrix.

$$P(C_k; \Phi, \nu) = \frac{|\Phi|^{\nu/2}}{2^{\nu p/2}\Gamma_p(\nu/2)}|C_k|^{-(\nu+p+1)/2} \exp\left(-\frac{trace(\Phi C_k^{-1})}{2}\right)$$

Modified EM Algorithm

We have already seen that the **E Step** is unaffected by the introduction of prior knowledge. Therefore, the computation of the memberships γ_{nk} would be unaffected.

- **Updating** W_k We had chosen the prior for the weights to be uniform. Therefore, the update of these parameters would be exactly the same as the ones done in class.

$$W_k = \frac{\sum_n \gamma_{nk}}{N}$$

- **Updating** μ_k

  We partially differentiate wrt μ_k and equate it to 0 to get the new value.

$$\frac{\partial}{\partial \mu_k} F = 0$$
$$\implies \sum_n \gamma_{nk} C^{-1}(y_n - \mu_k) = \mu_k$$

  Because the matrix is PSD, **Cholesky Decomposition** can be used to solve for the value of the updated class means μ_k. That is, we first decompose the matrix as LL^T where L is a lower triangular matrix with real and positive values.

- **Updating** C_k

  We partially differentiate wrt C_k and equate it to 0 to get the new value.

$$\frac{\partial}{\partial C_k} F = 0$$