Machine Learning Lab : Homework-3 Report

Vidushee Jain : 2020KUCP1014

May 12, 2023

1 EM Algorithm Implementation

1.1 Introduction

Image segmentation is a crucial task in the field of image processing, computer vision, and machine learning. It is a process of dividing an image into multiple segments or regions based on some specific criteria. In this regard, the EMG algorithm (Expectation-Maximization algorithm for Gaussian Mixture Models) is a popular method for image segmentation. The EMG algorithm uses the Expectation-Maximization algorithm to fit a Gaussian Mixture Model to the pixel intensities of an image, which can then be used to segment the image.

1.2 Algorithm Steps

1. Read in the image and reshape it into a 2D data array with rows corresponding to pixels and columns corresponding to the RGB values of the pixels.
2. Perform KMeans clustering on the data with the specified number of clusters (k) and maximum iterations.
3. Calculate the mean and covariance matrix for each cluster.
4. Initialize an empty array h with dimensions n x k where n is the number of data points h is used to store the membership probabilities of each data point to each cluster.
5. Perform the Expectation step: Calculate the probability density function of a multivariate normal distribution with mean m[j] and covariance matrix sigma[j] for each data point using the multivariate_normal and calculate membership probabilities of each data point for all the clusters.
6. Normalize the membership matrix h by dividing it with the sum of probabilities of all clusters.
7. Calculate the loglikelihood values after the E-step and store them in the Q list.
8. Perform the Maximization step: Update the parameters of the model, which in this case are the mean and covariance of each cluster.
9. Repeat 5-8 steps until convergence or a maximum number of iterations is reached.

1.3 Convergence Condition

The difference between the log-likelihood values of the current iteration and the two previous iterations is less than a certain threshold (1e-14). If this is the case, it means that the change in the log-likelihood values is insignificant and the algorithm has converged, so the loop is broken and the algorithm stops iterating.

The condition npåbs(Q[-1][1] - Q[-3][1]) $<$ 1e-6 is used as a stopping criterion for the EM algorithm. The difference between the log-likelihood values of the current and previous iteration is calculated, and if the absolute difference is less than 1e-14, then the algorithm is terminated.

The value 1e-14 is chosen as a threshold because it is a small number and represents a very

small difference in log-likelihood values. The choice of this threshold is arbitrary and can be adjusted based on the specific application and the desired level of accuracy. In general, smaller values of the threshold lead to more accurate results but may increase the computational cost.

1.4 Explanation of Implementation

The provided code implements the EMG algorithm for image segmentation. It reads in an image and reshapes it into a 2D data array with rows corresponding to pixels and columns corresponding to the RGB values of the pixels. It then performs K-Means clustering on the data with the specified number of clusters (k) and maximum iterations. The mean and covariance matrix for each cluster are calculated, and an empty array h is initialized to store the membership probabilities of each data point to each cluster.

The Expectation step is performed by calculating the probability density function of a multivariate normal distribution with mean m[j] and covariance matrix sigma[j] for each data point and calculating membership probabilities of each data point for all the clusters. The membership matrix h is then normalized, and the log-likelihood values after the E-step are calculated and stored in the Q list.

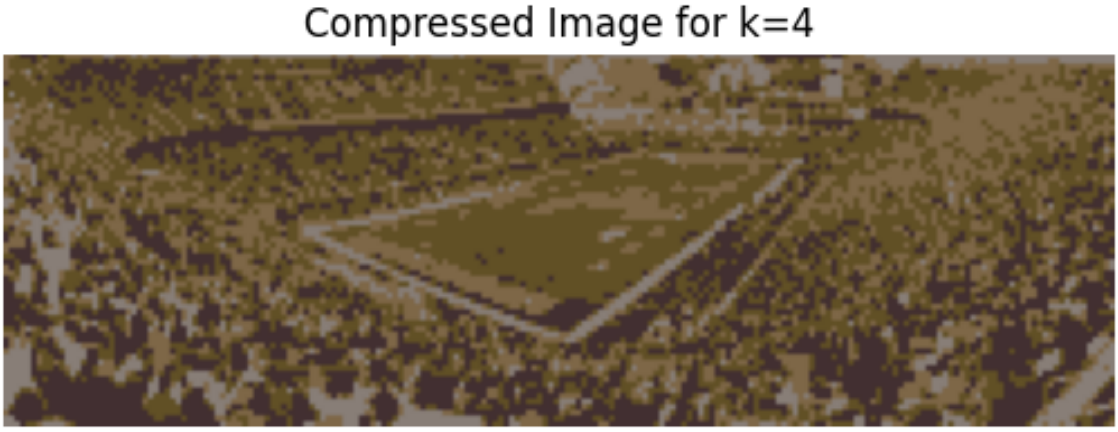
The Maximization step is performed by updating the mean and covariance matrix of each cluster. The entire process is repeated until convergence or a maximum number of iterations is reached.

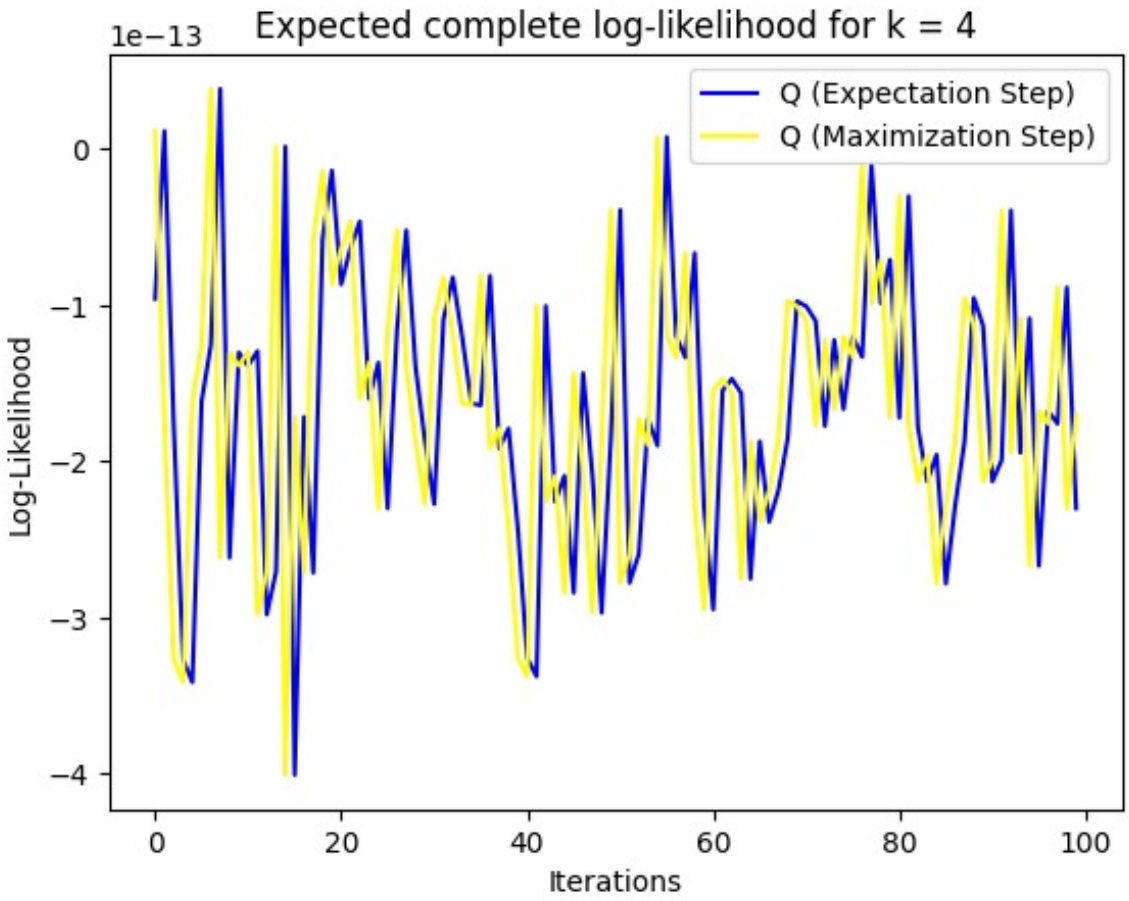
2 Expected Likelihood Plot

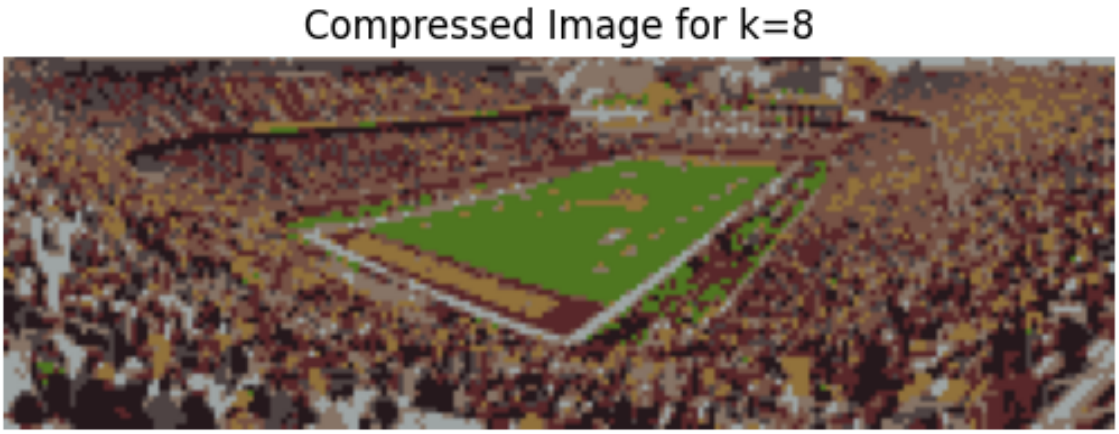
2.1 How to plot

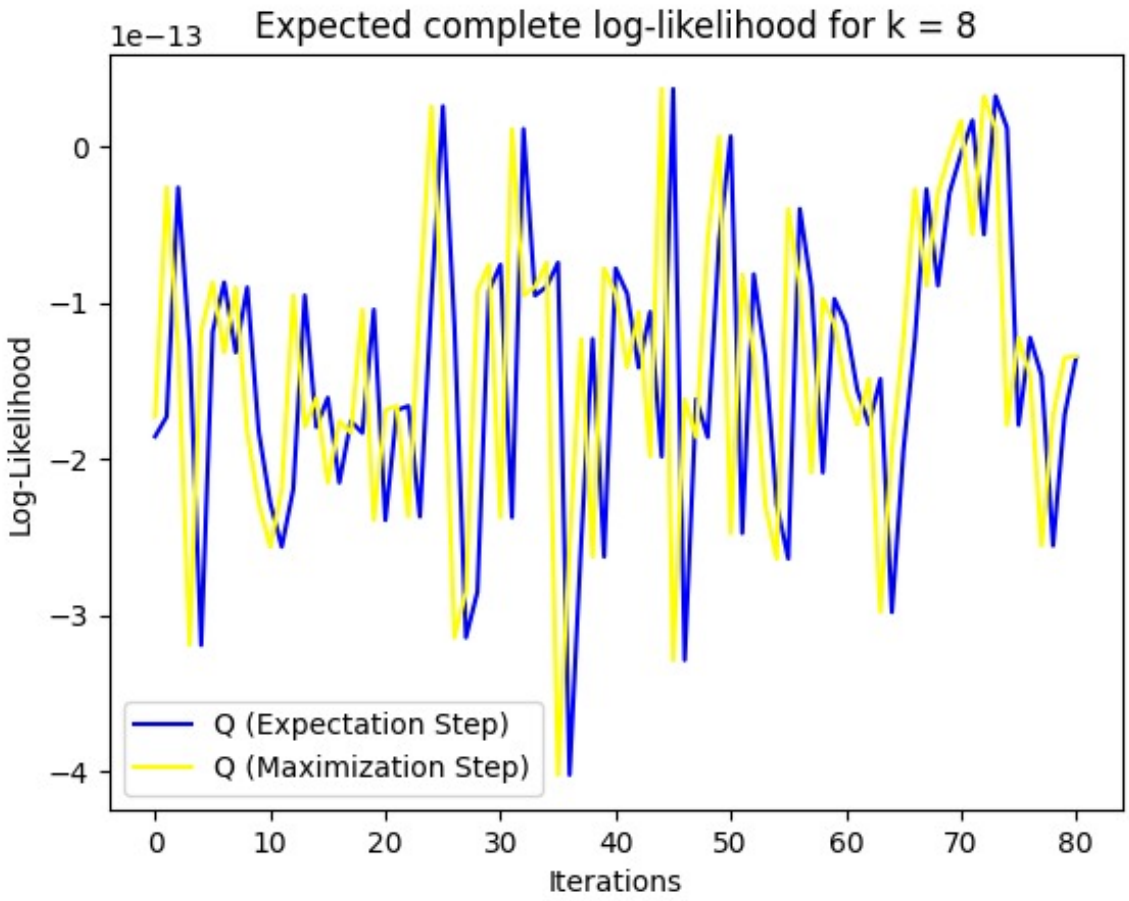
Initially I plotted compressed image using matplotlibś imshow function. The title of the plot indicates the value of k (number of clusters) used in the compression.

Plotting the expected complete log-likelihood (Q) values at each iteration of the EM algorithm. The x-axis represents the iteration number, and the y-axis represents the Q value. The zip(*Q) function is used to unzip the list of tuples into separate lists of x and y values. Then, two lines are plotted: one for the Q values after the E-step (odd iterations) and one for the Q values after the M-step (even iterations). The label parameter in the plot() method is used to create a legend for the two lines. The x and y labels and the title of the plot are also set using the xlabel(), ylabel(), and title() methods. Finally, the plot is displayed using the show() method.

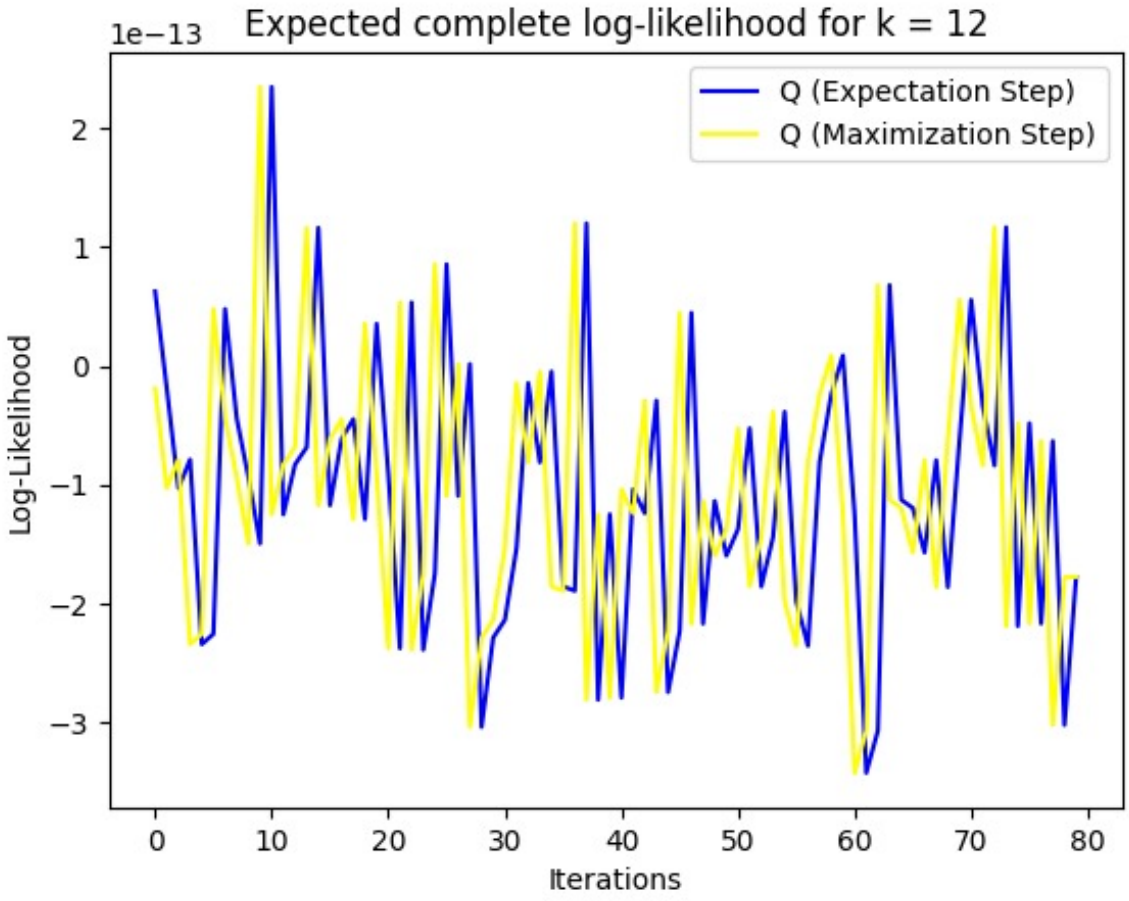



Expected complete log-likelihood for k = 4
1e−13
Q (Expectation Step)
Q (Maximization Step)
Log-Likelihood
Iterations
0
−1
−2
−3
−4
0
20
40
60
80
100


Compressed Image for k=8



Compressed Image for k=12



2.2 Plot Result

The plot shows the convergence of the expected complete log-likelihood (Q) over the iterations of the EM algorithm. The x-axis represents the number of iterations, and the y-axis represents the Q value.The darkyellow line shows the Q values after the Estep, and the other shows the Q values after the M-step. The convergence of Q indicates the convergence of the algorithm towards the optimal solution.If the Q values converge to a certain threshold or stop changing significantly, it suggests that the algorithm has found the optimal solution. The plot helps us to determine whether to stop the algorithm or continue running it for more iterations to obtain a better clustering result.Overall, this plot is a useful tool to monitor the progress of the EM algorithm and assess its convergence.
