Image Segmentation using EM algorithm

Mcmillon Walton Student
2017chb1047@iitrpr.ac.in

The Indian Institute
of Technology,Punjab
Ropar, India

Abstract

The **EM algorithm is a multi stage algorithm** used for image segmentation, **based on colors** present in the picture. It is widely applicable in the field of Computer Vision and serves as the basis for many projects. This report discusses on its implementation and application.

1 Introduction

The **EM algorithm** also known as the Expectation-Maximisation algorithm is an **unsupervised 2-stage machine learning algorithm** which alternates between the E and M steps/stages each iteration.It finds the **maximum likelihood parameters of the statistical models**[3](like Gaussian model etc.) which best describe the distribution of data. The first E step assumes **the latent variables**[1] of the statistical model[1], and then the M step assigns the data to the distribution which is more likely to produce that point. Then based on the data points assigned to the distribution, it re-evaluates the latent variables of the model- then the cycle starts again. Ideally this is done till convergence is achieved [2]

2 Methodology and Algorithm

The EM algorithm mainly involves 2 steps each iteration:

E-step

Based on the statistical model/distribution parameters[3], we assign the data points [4] to each of the distribution/cluster which are more likely to produce that point.*i.e.* Have the maximum a-posterioiri estimate of that pixel in the given mixture of Distributions.We get the "weights" of each pixel in each of the segements in the process. For more details on how the "weight" for each pixel in each segment is computed. Refer to Algorithm 1 E-step.

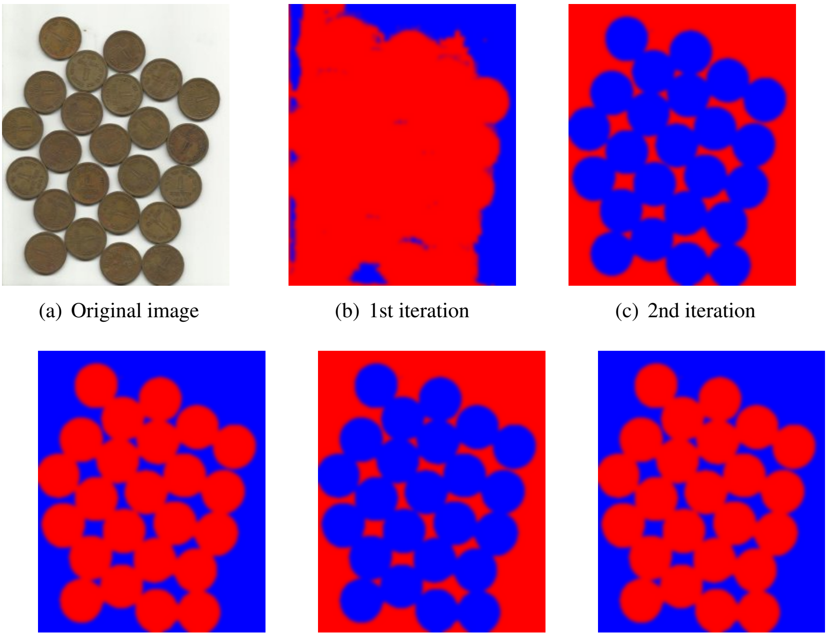
[1] attributes like mu(mean) and sigma(standard deviation)
[2] maximum likelihood / maximum a-posteriori estimates are achieved
[3] For first iteration we will take an initial guess
[4] pixel intensities in this case

M-step

We take an initial guess on the latent variables of the distribution for each segment/cluster, as unobserved. This is done before the first iteration as we have no estimate of these latent variables. However if we have data points assigned to each cluster (after M-step), we can use them to re-evaluate the latent variables. Here we took our initial guess as 1/nSegments [5] and take the "weight" of pixel in the segment into account when re-evaluating them. The latent variables for each distribution here are mean(mu) and probability (pi).[2] Note that these are 3-dimensional vectors as we are working with 3 channel RGB images.
For more details on how the the latent variables are updated check Algorithm 2 M-step

Here the distributions are normal **multi-variate gaussian distributions**.

We then go back to the E-step to re-evaluate the parameters of the distribution based on the pixels we assigned to each cluster.



(a) Original image (b) 1st iteration (c) 2nd iteration

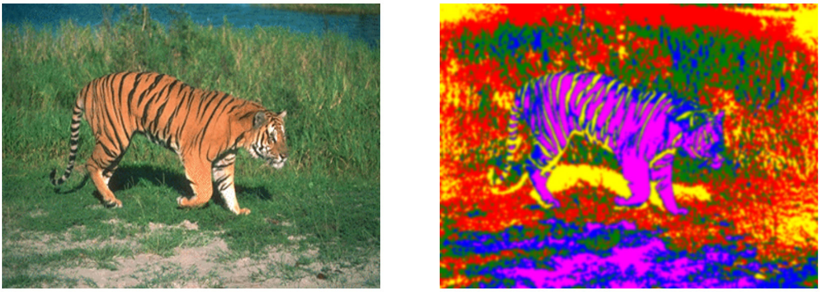
(d) 3rd iteration (e) 4th iteration (f) 5th iteration

Figure 1: Intermediate Stages of EM algorithm with 2 segments

[5] nSegments=Number of segments also, we add some noise to the value

(a) Original image (b) Segmented image

Figure 2: Image segmented into 4 components



(a) Original image (b) Segmented image

Figure 3: Image segmented into 5 components

Algorithm 1 E-step

Initialize weight of each pixel to 1 for each segment
Assuming n segments
We have values of μ and π before this step commences, (They are n-dimensional)

for $each pixel p \in image$ **do**
    $logAj|n| \leftarrow 0$ ▷ Calculating Aj, here this is a n- dimensional vector initialised
    for $segment \leq n$ **do**
        $logAj|segment| \leftarrow log(\pi|segment|) - 0.5 \times ||p - \mu|segment|||^2$ ▷ p = pixel intensity
    end for
    $Log(Amax) \leftarrow \max(logAj)$ ▷ Store the maximum value out of all the segments
    for $segment \leq n$ **do**
        $thirdterm \leftarrow thirdterm + \exp(logAj|segment| - logAmax)$
    end for
    for $segment \leq n$ **do**
        $\log Y \leftarrow logAj|segment| - logAmax - \log thirdterm$
        $W|p, segment| \leftarrow \exp \log Y$ ▷ "Weight" of the pixel in each segment is computed in this step
    end for
end for

Algorithm 2 M-step

Require: Weight of each pixel in each segment.$i.e$W
for $segment \leq n$ **do**
    $Sum \leftarrow 0$
    for $each pixel p \in image$ **do**
        $\mu|segment| \leftarrow \mu|segment| + W|p, segment| \times p$ ▷ p = pixel intensity
        $Sum \leftarrow Sum + W|p, segment|$
    end for
    $\mu|segment| \leftarrow \mu|segment| \div Sum$ ▷ nPixel= number of pixels present in the image
    $\pi|segment| \leftarrow Sum \div nPixels$ ▷ We use these updated values back in the E-step
end for

We have taken **4 assumptions**:

- The distribution is that of a multi-variate gaussian one with a co-variance matrix = Identity matrix
- Maximum iterations required for a decent result is 20.
- The number of segments/cluster are pre-defined and is not dynamic, during the progression of the algorithm.
- Whatever the case, the initial guess we have taken for μ[6] and π[7] is 1/(number of segments) with some added noise For each segment.

3 Results and Observations

1. Water coins
   - The image is segemented optimally when we choose 2 clusters. It distinguishes mainly between the 2 features namely coins and water.
   - The result we have obtained is ideal as if we ask a human, he too will say that two components are present.
   - The red cluster defines the coins and the blue cluster defines the water present in the image
2. jump
   - The image is optimally segmented when we define 4 clusters.
   - The red cluster represents the human.
   - blue cluster - Mountain
   - yellow cluster - Troposphere
   - pink cluster - Stratosphere
3. tiger
   - The image is best segmented when we choose to have 5 clusters
   - However as we see from Figure 3 due to the distribution of pixel intensities present in the picture, the image isn't optimally segmented. For example the light orange color sand and the dark orange tiger skin are under the same cluster.
   - As seen from the above 2 cases, more cluster/distributions doesn't mean more optimal result. But it does offer more flexibilty and in the case of this picture may have a better output.

[6] mean
[7] likelihood

4 Conclusion and Takeaway

- We need an idea of how many components/segments are present in the image before running the EM algorithm for the optimal result
- A good guess on the parameters of each distribution of pixels present in the image will lead to early convergence
- The algorithm is susceptible to outliers.

References

[1] Wolfgang Jank. Orcs. *Perspectives in Operations Research*, 36:367–392.

[2] Lucas Parra and Harrison H.Barret. Ieee trans med imaging. *List-Mode Likelihood: EM Algorithm and Image Quality Estimation Demonstrated on 2-D PET*, pages 228–235, 1998.

[3] Wikipedia. Em algorithm. *https://en.wikipedia.org/wiki/Expectation-maximization*$_a$*lgorithm*.