An Examination of the Expected Maximization Algorithm Over an Eye

Daniel D. Beatty

2012-04-28

Abstract

1 Specification and Organization

For the exercise reported here, there is a digital picture of a human retina as shown in figure 1. The objective of the work is to

1. Take the gradient of the monochrome image
2. Use the Expected-Maximization (EM) algorithm to extract the edge points from the blurred optical disc boundary.

2 Background

The expected maximization algorithm is credited as first published in [1]. Various explanations have appeared in publications like [2, 3, 4].

In some explanations, EM extends properties of Maximum Likelihood Estimation (MLE) methods by determining the iterative convergence of sufficient statistics. [1] showed that the EM algorithm is applied

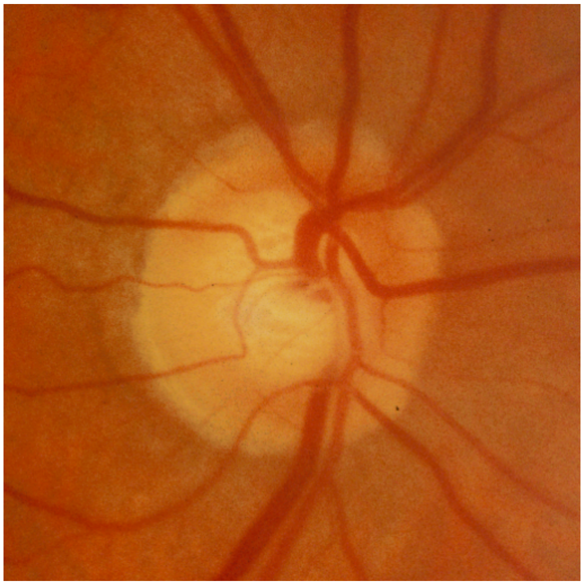


Figure 1: Original Image

effectively to two datasets. One data set is observed and the other dataset is realized from the observed dataset. This application implies that there is a mapping between the two datasets, where as the observed data set itself is deemed incomplete. The dataset realized from the observed data set is considered a complete dataset.

In general, the equation for the terminating condition of the EM algorithm [2] is stated in equation 1.

$$\vec{\Theta}^* = \arg\max_{\Theta} \sum_{\vec{x} \in \mathcal{X}^n} E[\ln p(\vec{x}|\vec{\theta}, \vec{y})] \tag{1}$$

The meaning of the terms in equation 1 are as follows:

- $\mathbf{X}$ is the true data set, $\vec{x}$ is the true data set sample,
- $\vec{\Theta}$ is the set of parameters for the statistic,
- $\vec{\theta}$ is a previous estimate of the set of parameters, and
- $\vec{y}$ is the data sample in the given data set.

Equation 1 allows a characteristic to be derived in equation 2 that identifies a convergence point, $Q(\vec{\theta}; \vec{\theta}^i)$

for the parameters identified by $\vec{\theta}$ and $\vec{\Theta}$,.

$$Q(\vec{\theta};\vec{\theta^i}) = E_{D_b}[\ln p(D_g, D_b;\vec{\theta})|D_g;\vec{\theta^i}] \tag{2}$$

where :

- $\vec{\theta^i}$ is the current (best) estimate for the full distribution.
- $\vec{\theta}$ is a candidate vector for an improved estimate.
- $Q(\vec{\theta};\theta^i)$ is a function of $\vec{\theta}$ and $\vec{\theta^i}$.
- D_b is the actual data set.
- D_g is the unknown and uncorrupted data set.
- $E_{D_b}[\ln p(D_g, D_b;\vec{\theta})|D_g;\vec{\theta^i}]$ is the expected value over the missing features. The expected value hinges on $\vec{\theta^i}$ which are the estimated true parameters.

In [5], the complete dataset is assumed to be a collection of Gaussian datasets. A Gaussian dataset is defined by its mean and variance (or its multivariate equivalent.)

In the Gaussian case, each class has a proportion that defines how much each class contributes as determined by their mean and covariance (denoted $\alpha, \vec{\mu}, \mathbf{\Sigma}$ respectively). If one acquires a sample as an initial set, there is an EM algorithm that will find these sufficient statistics. In [5], there is an example using a matrix as the result of the expectation step. From this expectation, the sufficient statistics for the next step are computed. When an expected equation forms a matrix $\mathbf{A}$, as in equation 3, it is called an expected matrix.

$$a_{ij}^{(k)} = \frac{\alpha_j p(\vec{y_i}^{(k)}|\vec{\mu_j}^{(k)}, \Sigma_j^{(k)})}{\sum_{j=1}^{M} \alpha_j p(\vec{y_i}^{(k)}|\vec{\mu_j}^{(k)}, \Sigma_j^{(k)})} \tag{3}$$

The three sufficient statistics are computed in the maximization step. They are computed by the following equations.

$$\vec{\mu}_j^{(k+1)} = \frac{\sum_{i=1}^{N} a_{ij}^{(p)} \vec{y}_i}{\sum_{i=1}^{N} a_{ij}} \tag{4}$$

$$\mathbf{\Sigma}_j^{(k+1)} = \frac{\sum_{i=1}^{N} (\vec{y}_i \vec{\mu}_j^{(p)})^T (\vec{y}_i \vec{\mu}_j^{(p)})}{\sum_{i=1}^{N} a_{ij}^{(k)}} \tag{5}$$

$$\alpha_j^{(k+1)} = \frac{1}{N} \sum_{i=1}^{N} a_{ij}^{(k)} \tag{6}$$

As shown in equation 2, there is a convergence of probability parameters that can be determined by the Q function, shown in equation 7.

$$Q(\theta^*; \theta) = \log(\prod_{i=1}^{N} a_{ii}) \tag{7}$$

Also, the initial guess for $\mathbf{A}$ can not be the zero matrix. As such, the sufficient statistics would be rendered zero, and no convergence would occur. Typically, the guesses are for μ to be scattered for each class and for the expected matrix to be

$$\mathbf{A} = \frac{1}{N}\mathbf{I}.$$

3 Approach

To start, the image is decomposed into a two dimensional array arranged from left to right and top to bottom. This two dimensional array is then reassembled into a one dimensional array with the following mapping:

$$s[i] = t[x][y] \tag{8}$$

$$i = y(\bar{r}) + x \tag{9}$$

where

- $\bar{r}$ is the row size

- y is the row index of the pixel,
- x is the column index of the pixel, and
- i is the single dimensional array index of the pixel.

The $s[i]$ are thereby considered the observed set and complete set is assumed to be a collection of Gaussian sets. The number of Gaussian sets is left as a variable. As in the example given by [5], the complete set is assumed to be a sum of the incomplete set and complete sets. Thus, a framework is build to compute equations 3, 5, and 6.

An Matlab/Octave prototype [1] obtains the mean, variance, and proportions from the EM algorithm. A similar implementation made with the Cocoa Renaissance framework was constructed by the author of this report.

Both algorithms make effective histograms based on the most intense gray level (assuming it is not 1.0). The Cocoa version produces an 8 bit version to establish the histogram, and performs the remaining computations with the 16 version. All values are normalize between [0..1]. Both establish a set of parameters for scaling, minimum and maximum values. The log-likelihood value is determined in both expectation and maximization step for comparison. If the absolute difference between the two log-likelihood values are within a margin of error, the iteration stops.

4 Results

The Octave version can not plot images, but the output of vectors from the Octave can be plotted with GNUplot. The main histogram of the human eye is shown in figure 2. This plot shows that the composite image pixel dominance in the low range below 20 and above 50. In figure 3, the results of the EM segmentation is shown. In this case, one can see the contributions of 5 Gaussian distributions and that combined they almost fit the original. Also, we can see that certain sections of the image should be distinguishably separated by this filter.

[1] Prototype was authored by Code by Prof. Jose Vicente Manjon Herrera of the Universitat Politècnica de Valencia

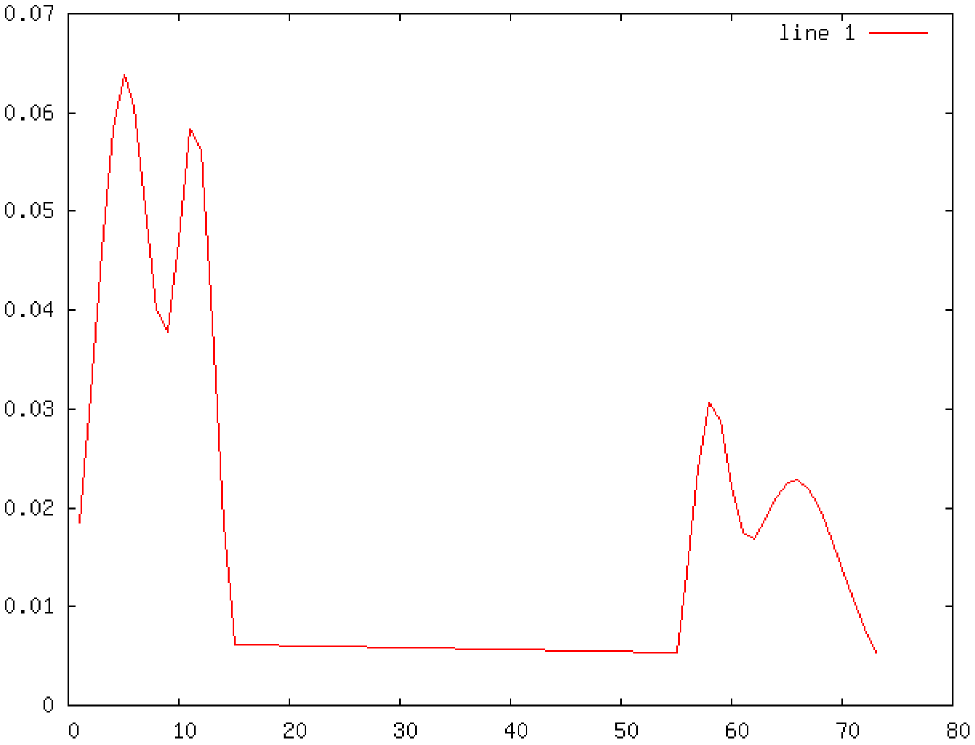


Figure 2: Histogram of the Eye image example

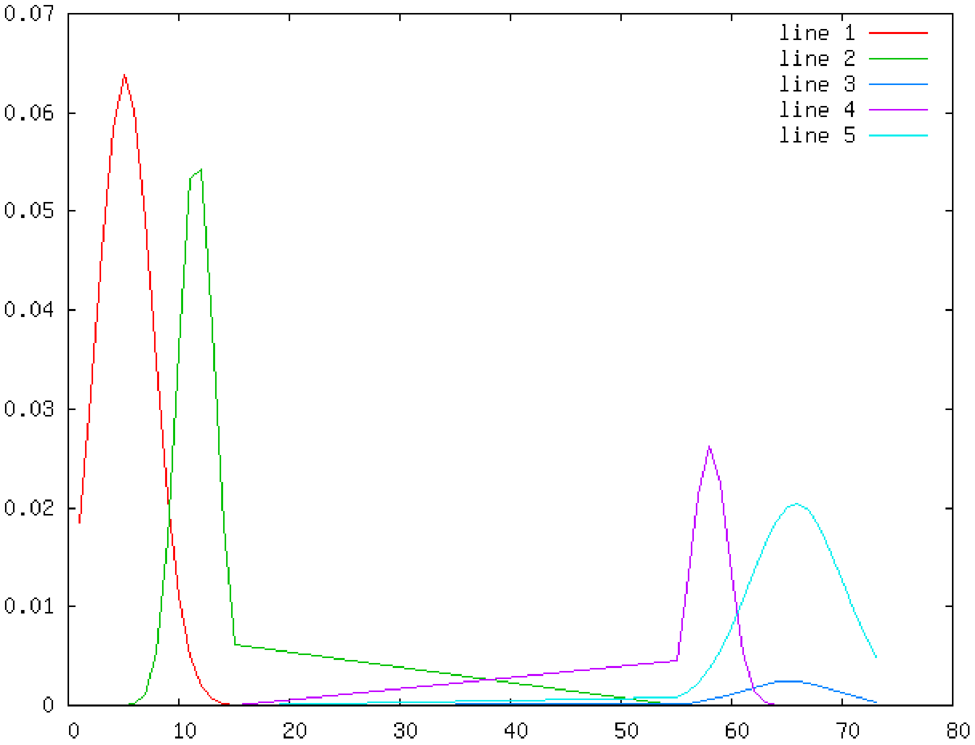
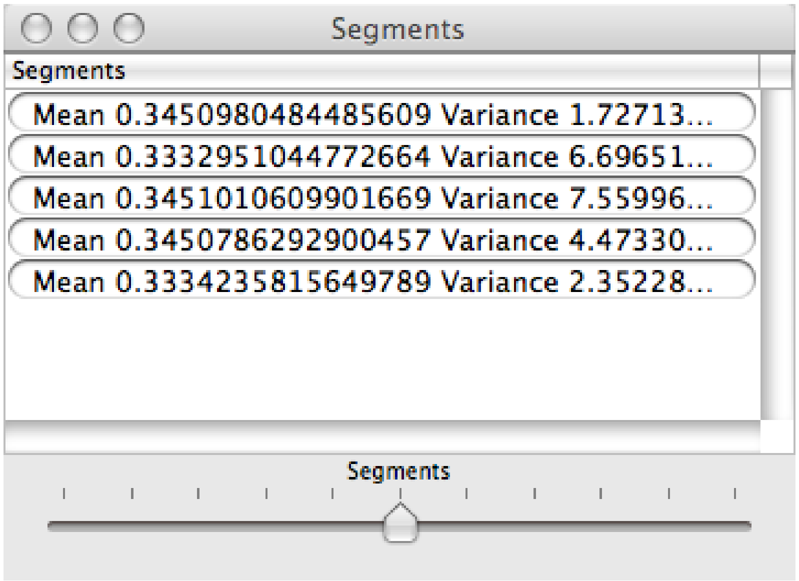


Figure 3: Histogram of the Eye image example with EM Segements

With the Cocoa implemented version we can the images themselves and play a little the images. The interface I provided for this program is as shown in 4. We can see that the user can select how many Gaussians to apply, and when the computations are performed the sufficient statistics of each Gaussian is listed. When a sufficient statistic is selected, a filter based on that statistic is applied, as in figure 5. We can see the original image of the eye along with 5 different EM based filters.



```
Mean 0.3450980484485609 Variance 1.72713...
Mean 0.3332951044772664 Variance 6.69651...
Mean 0.3451010609901669 Variance 7.55996...
Mean 0.345078629290457 Variance 4.47330...
Mean 0.3334235815649789 Variance 2.35228...
```

Figure 4: EM Segment Control

5 Conclusions

Both iteratively proceed through expectation and maximization step. The Cocoa version omits adding an ϵ to the scaled probability calculation, which may be throwing off the Cocoa version by a shift on the x domain.

Also, one filter was difficult to see since does not favor the higher intensity pixels found in the center, and the rest appears dark without that center.

A closer match in the Cocoa version may alleviate some distinction troubles. Also, a OpenCL plug-in equivalent to other more powerful image editors may be provide a better approach allowing the other filters to be applied to examine the potential of this EM segment filter.

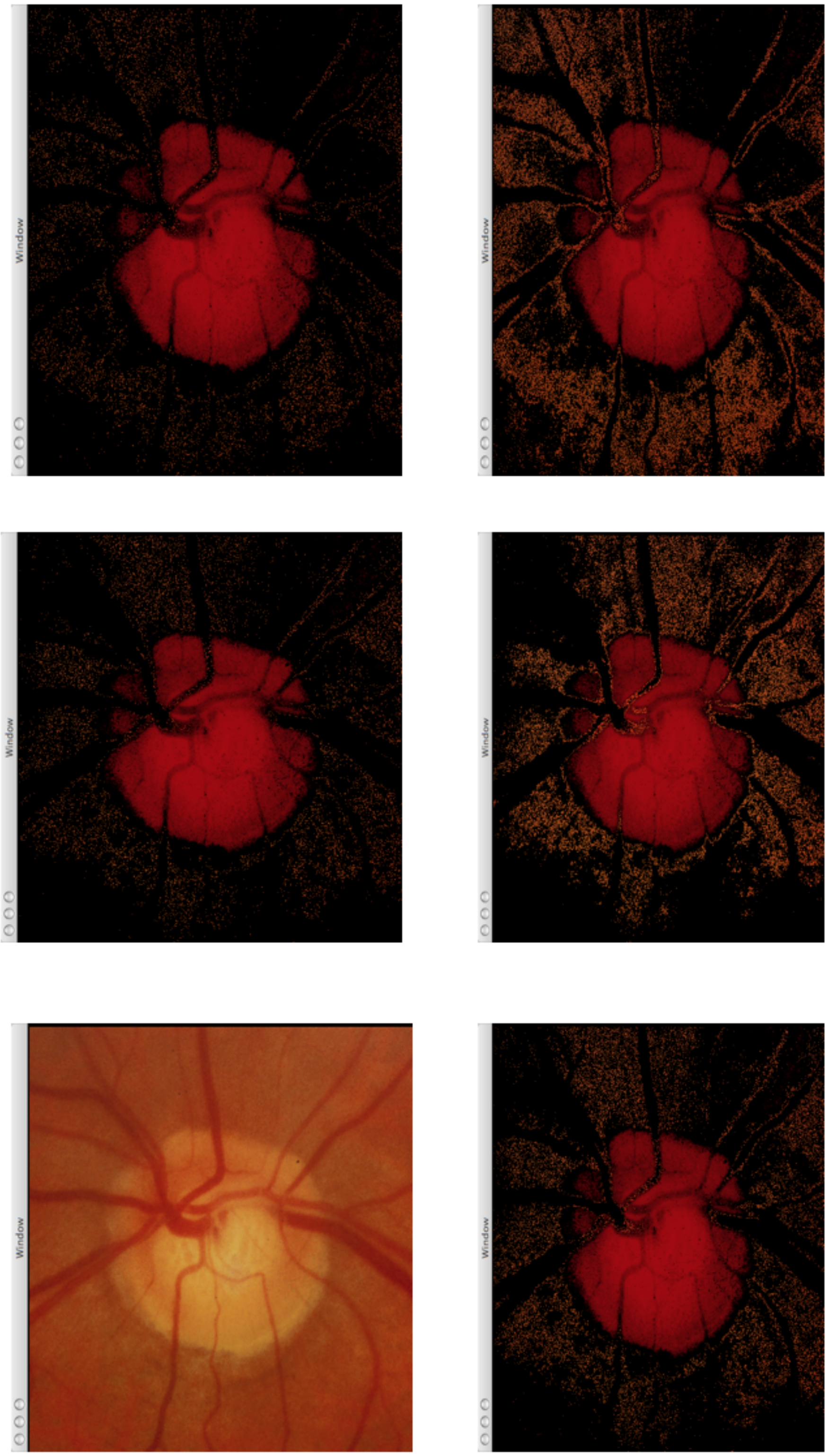


Figure 5: Five EM Segments of the Eye and the Original

References

[1] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 39(1):1–38, 1977.

[2] Richard O. Duda, Peter E. Hart, and David E. Stork. *Pattern Classification*. Wiley and Sons, 2nd edition, 2000.

[3] Todd K Moon. The expectation-maximization algorithm. In *IEEE Signal Processing Magazine*, pages 47–60, 1996.

[4] Seong wook Joo. Expectation maximization for mixture models.

[5] T. Yamazaki. Introduction of em algorithm into color image segmentation, 1998.