A Study of Expectation-Maximization using Gaussian Mixtures in Generative Learning

Eli Uc
eliuc90@gmail.com

1. Introduction

A typical problem that utilizes this algorithm would be one where an unknown hidden variable is present. For example, a large scale taking the weight of every car that goes over a bridge. The independent and individually distributed data that is taken is only the weight, however the distribution of data suggests multiple "clusters" of unidentified classes. Intuitively, different types of cars (sedan, truck, motorcycles, etc.) would produce different ranges of weights which is reflected in the distribution produced by the data.

The Expectation-Maximization algorithm is a 2 step iterative algorithm: The Expectation Step (E-Step) and the Maximization Step (M-Step). The first E-Step is to estimate the likelihood of the complete data (the observation data as well as the hidden variable data). This is approximated through taking the statistical Expectation of the Log probability of the complete data with respect to the incomplete observed data. The second M-Step is to maximize this approximation (the likelihood of the complete data).

The principle function that this algorithm revolves around is called the Q-Function:

$$Q(\Psi; \Psi^n) = E_{Z|X;\Psi^n}[P_{X,Z}(D, D_Z; \Psi)|D] \qquad (1)$$

Variable	Description
Ψ	Distribution Parameters
Ψ^n	Distribution Parameters for the nth iteration
X	Observation Random Variable
Z	Hidden Variable Random Variable
D	Observation Dataset
D_Z	Hidden Variable Dataset

Table 1. Description of Variables

The Q-Function is the operation done in the E-Step where the Likelihood of the complete data is being approximated with the given observable data. This Q-function approximation is then maximized using Maximum Likelihood Estimation (MLE). In other forms of EM, this maximization process is done using the Maximum A-Priori (MAP) approximation.

2. Common Use - Gaussian Mixtures

Identifying and applying an effective model to fit a distribution of data is often challenging in the realm of generative learning. Real world observations are mostly complex and significantly more intricate than the common distributions available to us (Poisson, Multimodal, Uniform, etc). A good approximation of the complex structure of observational distributions is a sum of a single parameterized distribution. To put this in an analytic perspective,

$$P(x) = \sum_{c=1}^{C} P(x|z=c) \cdot P(z=c) \qquad (2)$$

where x is the random variable of the data and z is a hidden variable. This also denotes that a distribution can be described as a weighted sum of distributions. In this case, the hidden(latent) variable takes on a discrete and finite form.

Expectation-Maximization is a powerful tool in these problems to attempt to identify the hidden variable that form the components of the Gaussian mixtures. In this case, we study the instance in which we model the distribution to be a sum of Gaussians. A multivariate Gaussian distribution is parameterized by a mean vector and a covariance matrix. A Gaussian mixture also depends on the a priori probability of each hidden variable. We can now define a Gaussian mixture more specifically:

$$P(x) = \sum_{c=1}^{C} G(x, \mu_c, \Sigma_c) \cdot \pi_c \qquad (3)$$

3. Analysis

3.1. Generalized EM w/ Finite Discrete Hidden States

The math for the Q-function and subsequent MLE parameter estimation can be generalized given that the latent variables are discrete and finite and now that the type of modeled distribution is defined (in this case, Gaussian).

As seen in Equation 1, calculating the Q function requires modeling the joint probability of the complete data, $P_{X,Z}(D, D_Z; \Psi)$. This can be placed in terms of observable distributions using the definition of conditional probability:

$$P_{X,Z}(D, D_Z; \Psi) = P_{X|Z}(x|z; \Psi) \cdot P_Z(z; \Psi) \quad (4)$$

This expression becomes highly nonlinear with respect to the random variable z. Because the Expectation step is with respect to z (given x), it is desirable to have a linear relationship with z for simpler analysis. Given the assumption of discrete latent variables, there is a more convenient way to express z.

Instead of denoting the hidden states as scalars $z_i = \{1, 2, ..., C\}$, they will now be expressed as vectors with the i^{th} hidden state being a vector of zeros with a 1 in the i^{th} element, or e_i:

$$z_i = \left\{ \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \ldots, \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \right\} \quad (5)$$

This allows for further simplification of the likelihood of the complete data:

$$\begin{aligned} P_{X,Z}(x, z; \Psi) &= P_{X|Z}(x|z; \Psi) \cdot P_Z(z; \Psi) \\ &= \prod_{j=1}^{C} [P_{X|Z}(x|e_j; \Psi)\pi_j]^{z_j} \\ \log P_{X,Z}(x, z; \Psi) &= \sum_{j=1}^{C} \log \left[P_{X|Z}(x|e_j; \Psi)\ \pi_j\right]^{z_j} \\ &= \sum_{j=1}^{C} z_j \log \left[P_{X|Z}(x|e_j; \Psi)\ \pi_j\right] \end{aligned} \quad (6)$$

Now that the log-likelihood has been established for a single observation x and corresponding hidden vector (variable) z, the log likelihood of the complete dataset can be extrapolated:

$$\begin{aligned} P_{X,Z}(D, \mathbb{Z}; \Psi) &= \prod_{i=1}^{N} P_{X,Z}(x_i, z_i; \Psi) \\ &= \prod_{i=1}^{N} \prod_{j=1}^{C} [P_{X|Z}(x_i|e_j; \Psi)\ \pi_j]^{z_{ij}} \\ \log P_{X,Z}(D, \mathbb{Z}; \Psi) &= \sum_{i=1}^{N} \sum_{j=1}^{C} z_{ij} \log \left[P_{X|Z}(x|e_j; \Psi)\ \pi_j\right] \end{aligned} \quad (7)$$

Finally, taking the expectation of equation 7 produces the Q-function that was studied earlier. The Q-function is now reduced to finding the expectation of the hidden random variable z with respect to the hidden variable conditioned on the observed variable x. This is simplified to:

$$\begin{aligned} E_{X|Z;\Psi^{(n)}}(z_{ij}|D) &= 1 \cdot P_{Z|X}(z_{ij} = 1|x_i; \Psi^{(n)}) + 0 \\ &= P_{Z|X}(e_j|x_i; \Psi^{(n)}) \\ \mathbf{h_{ij}} &= P_{Z|X}(e_j|x_i; \Psi^{(n)}) \end{aligned} \quad (8)$$

This expectation reduces to the posterior probability of an observable point given the mixture component. This therefore reduces the Q function for a Generalized EM algorithm for discrete finite hidden variable z:

$$Q(\Psi; \Psi^n) = \sum_{i,j} h_{ij} \log[P_{X|Z}(x_i|e_j; \Psi)\ \pi_j] \quad (9)$$

3.2. Convergence

While analysis has been done on estimating the likelihood of the complete data (evaluating Q-function in E-Step), it still needs to be shown that this iterative process converges. To prove this, it is useful to observe the effect of the likelihood of the complete data with respect to the likelihood of the observable data:

$$\begin{aligned} P_{X,Z}(D, \mathbb{Z}; \Psi) &= P_{Z|X}(z|D; \Psi)\ P_X(D) \\ P_X(D) &= \frac{P_{X,Z}(D, \mathbb{Z}; \Psi)}{P_{Z|X}(z|D; \Psi)} \\ \log P_X(D) &= \log P_{X,Z}(D, \mathbb{Z}; \Psi) - \log P_{Z|X}(z|D; \Psi) \end{aligned} \quad (10)$$

Taking the $E_{Z|X;\Psi^n}(\cdot)$ operator on both sides produces:

$$\log P_X(D) = Q(\Psi; \Psi^{(n)}) + H(\Psi; \Psi^{(n)}) \quad (11)$$

Because the M-Step of the EM algorithm seeks to maximize the Q function after each iteration, it can be claimed that $Q(\Psi; \Psi^{(n+1)}) \geq Q(\Psi; \Psi^{(n)})$. As seen from [1], an identical relationship can be derived for successive H function values: $H(\Psi; \Psi^{(n+1)}) \geq H(\Psi; \Psi^{(n)})$. This allows further analysis of the likelihood of the observable data:

$$\begin{aligned} &\log P_X(D; \Psi^{(n+1)}) - \\ &\quad \log P_X(D; \Psi^{(n)}) = Q(\Psi; \Psi^{(n+1)}) + H(\Psi; \Psi^{(n+1)}) - \\ &\qquad Q(\Psi; \Psi^{(n)}) - H(\Psi; \Psi^{(n)}) \\ &\quad = \Delta Q + \Delta H \\ &\quad \geq 0 \end{aligned} \quad (12)$$

Thus, log $P_X(D;\Psi^{(n+1)}) \geq \log P_X(D;\Psi^{(n)})$ and can claim convergence since the log likelihood of the observable data is always increasing or staying the same after each iteration. This does not prove, however, that the solution converges to a global maxima, but instead to a local maxima, and is thus subject to deceitfully optimal points such as saddle points or low local maxima.

3.3. Gaussian Mixtures

After deriving the iterative equations for a generic EM problem with finite discrete hidden components, and after verifying that this algorithm converges, computing the algorithms is a straightforward application of the observable data Gaussian distribution. The E-Step derived in equation 8 denotes that the h_{ij} factor in the Q-function is the only value that needs to be updated after each iteration (as well as the updated distributive parameters). This factor is the posterior probability of the sample x_i. For a Gaussian distribution:

$$P_{Z|X}(e_j|x_i;\Psi^{(n)}) = \frac{P_{X|Z}(x_i|e_j;\Psi^{(n)})\ P_Z(e_j)}{P_X(x_i)}$$
$$\mathbf{h_{ij}} = \frac{G(x_i,\mu_j^{(n)},\Sigma_j^{(n)})\ \pi_j^{(n)}}{\sum_{k=1}^{C} G(x_i,\mu_k^{(n)},\Sigma_k^{(n)})\ \pi_k^{(n)}} \tag{13}$$

Having h_{ij} defined, the M-step simply maximizes Equation 9 with respect to the mean, covariance, and weight. This is derived in [2], and the update equations of each parameter are:

$$\mu_j^{(n+1)} = \frac{\sum_i h_{ij}\ x_i}{\sum_i h_{ij}} \tag{14}$$

$$\sigma_j^{2(n+1)} = \frac{\sum_i h_{ij}\ (x_i-\mu_j)^2}{\sum_i h_{ij}} \tag{15}$$

$$\pi_j^{(n+1)} = \frac{1}{N}\sum_i h_{ij} \tag{16}$$

4. Experiments and Results

4.1. Problem Definition

EM will be used in to model the distribution of the Discrete Cosine Transform (DCT) segments of a cheetah image in an attempt to classify the pixel as background (not part of the cheetah) or foreground (part of the cheetah). The training set is provided by Nuno Vasconcelos of the ECE Department at UC San Diego. The training set consists of 1053 background DCT samples and 250 foreground DCT samples. The samples are 8x8 vectorized blocks.

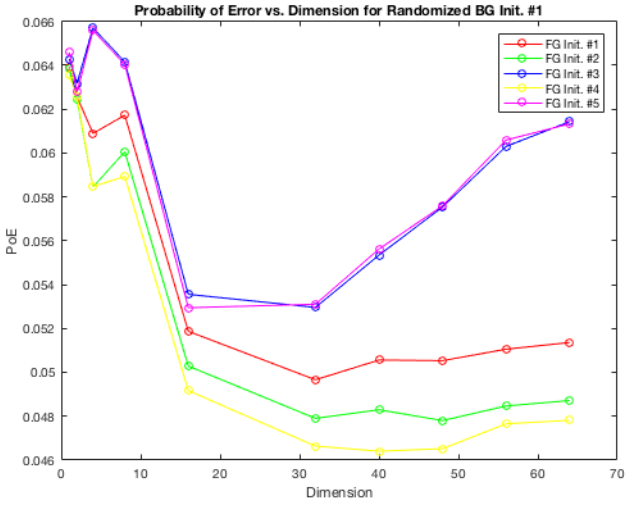
4.2. Approach

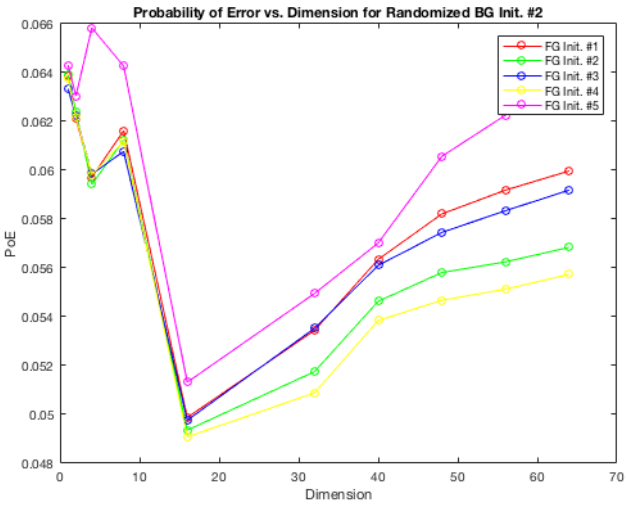
Assuming a 0/1 Loss function, Baye's Decision Rule (BDR) dictates that the class that maximizes the posterior distribution is the classified class. Here, the distribution modeled will be a Gaussian mixture with C components. As described earlier, a sum of Gaussians may be intricate enough to properly define the complex distributions that exist in nature, in this case, the DCT blocks of cheetah images.

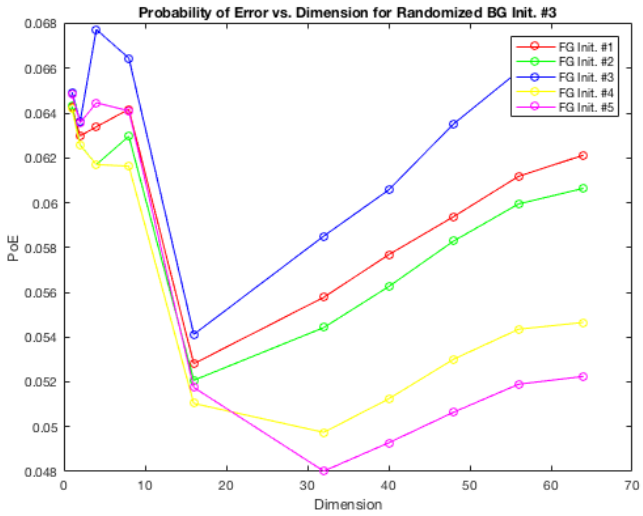
We will explore the effects of random initialization on EM to see its impact on performance using C=8 components. Also, we will study performance for varying dimensions during BDR (EM parameters will be solved with the full 64 dimensional data). We will also be experimenting performance based on number of mixture components.

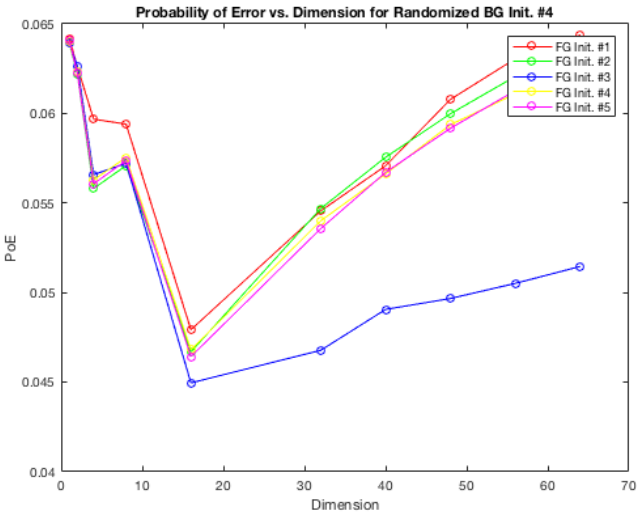
4.3. Results

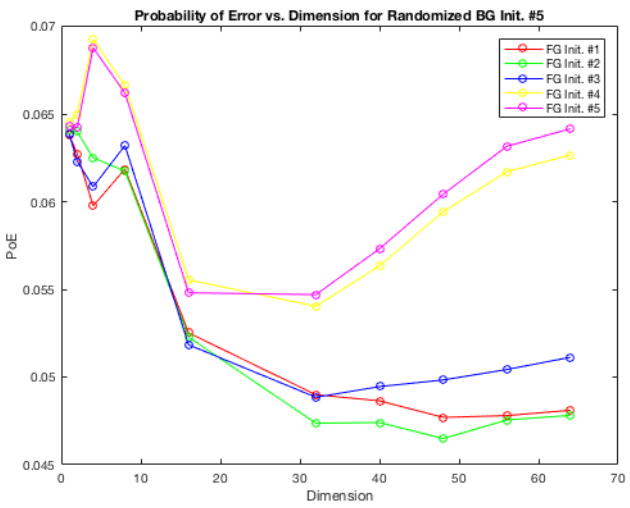
Below are the probability of error for 25 different initializations across increasing dimensions.





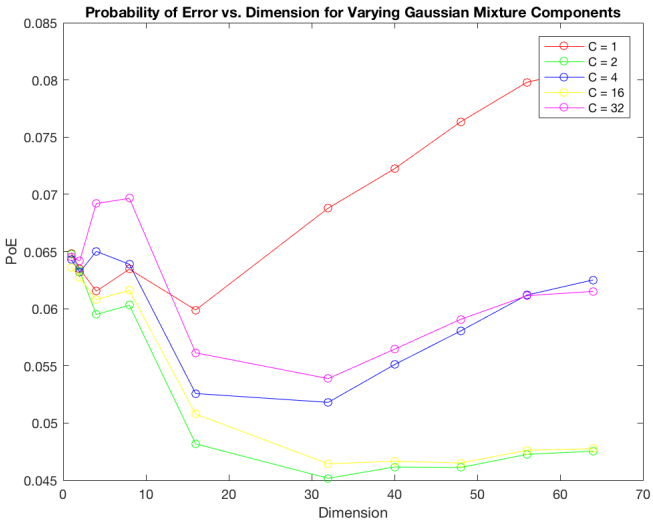


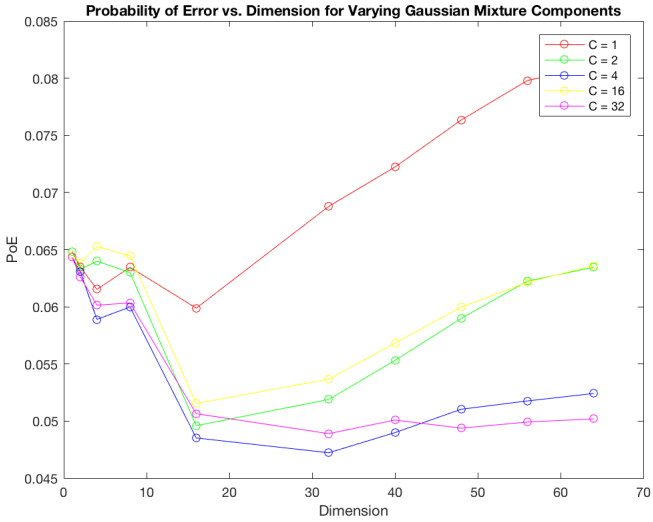




Each image has the Probability of Error (PoE) for varying dimensions for 5 different foreground initializations for the same sole background initialization. The total of five images therefore indicates 25 separate combinations of initializations.

The following images are varying PoE for different Gaussian mixture compoenents. This was done twice to briefly study the effects of initialization.





5. Conclusion

In worst case scenario, there is a about a 1.5% swing between PoE for the ”best” and ”worst” initializations. This demonstrates the robustness of EM towards random initializations. Across all performances, there appears to be a notable improvement starting at 16 dimensions, suggesting that the data is more appropriately modeled as an 8 component Gaussian mixture at 16 dimensions. Afterwards performances are not consistently improving or worsening with more dimensions, making it difficult to make claims about the data since there exists random initialization as well as local extrema that that different initializations of EM may converge to.

A few observations can be made about the last two figures. A single component gaussian mixture produces the worse PoE. The 2 and 16 component mixtures produced similar results despite the varying initializations. The same can be said about the 4 and 32 component mixtures. It is difficult to make claims about which number of components in the model produce the best performance, as the data found in this experiment can oscillate in performance based on initializations or local maxima convergences. Either way the variation between these performances is less than 1%, suggesting that there is not a lot of differences between how many components the Gaussian mixture is modeled as.

References

[1] N.Vasconcelos *”Expectation-Maximization”*. Statistical Learning, 2017, www.svcl.ucsd.edu/courses/ece271A/ece271A.htm

[2] N.Vasconcelos *”Expectation-Maximization 2”*. Statistical Learning, 2017, www.svcl.ucsd.edu/courses/ece271A/ece271A.htm

[3] N.Vasconcelos *”Expectation-Maximization 3”*. Statistical Learning, 2017, www.svcl.ucsd.edu/courses/ece271A/ece271A.htm