Gaussian Mixture Model using Expectation Maximization

-Abhinava Sikdar

GAUSSIAN MIXTURE MODELS USING EXPECTATION MAXIMIZATION

GMM is a type of clustering algorithm. As in the name, each cluster is modelled according to a different Gaussian distribution. Unlike hard assignments into clusters(Eg. K-means), GMM assigns soft assignments to each of the data points which means that a given data point could have been generated by any of the different Gaussian Distributions but with a certain probability.

Assume that we are given a data set D = {x1, . . . , xN } where xi is a d-dimensional vector measurement. Assume that the points are generated in an IID fashion from an underlying density p(x). We further assume that p(x) is defined as a finite mixture model with K components i.e.

$$p(\underline{x}|\Theta) = \sum_{k=1}^{K} \alpha_k p_k(\underline{x}|z_k, \theta_k)$$

Where:

- The $p_k(x|z_k,\theta_k)$ are *mixture components*, $1 \le k \le K$. Each is a density or distribution defined over p(x), with parameters θ_k.
- $z=(z_1,...,z_K)$ Is a vector of K binary indicator variables that are mutually exclusive and exhaustive (i.e., one and only one of the z_k's is equal to 1, and the others are 0). z is a K-ary random variable representing the identity of the mixture component that generated x. It is convenient for mixture models to represent z as a vector of K indicator variables.
- The $\alpha_k = p(z_k)$ are the mixture weights, representing the probability that a randomly selected x was generated by component k, where $\sum_{k=1}^{K} \alpha_k = 1$.
- The complete set of parameters for a mixture model with K components is

$$\Theta = \{\alpha_1,...,\alpha_K,\theta_1,...,\theta_K\}$$

How to calculate the membership weights?

We can compute the "membership weight" of data point xi in cluster k, given parameters Θ as

$$w_{ik} = p(z_{ik} = 1|\underline{x}_i, \Theta) = \frac{p_k(\underline{x}_i|z_k, \theta_k) \cdot \alpha_k}{\sum_{m=1}^{K} p_m(\underline{x}_i|z_m, \theta_m) \cdot \alpha_m}, \quad 1 \le k \le K, \quad 1 \le i \le N.$$

This follows from a direct application of Bayes rule. The membership weights above reflect our uncertainty, given xi and Θ, about which of the K components generated vector xi. Note that we are assuming in our generative mixture model that each xi was generated by a single component—so these probabilities reflect our uncertainty about which component xi came from, not any "mixing" in the generative process.

For x ∈ Rd we can define a Gaussian mixture model by making each of the K components a Gaussian density with parameters μ(k) and Σ(k). Each component is a multivariate Gaussian density with its own parameters θ(k) ={μ(k),Σ(k)}.

$$p_k(\underline{x}|\theta_k) = \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} e^{-\frac{1}{2}(\underline{x}-\underline{\mu}_k)^t \Sigma_k^{-1}(\underline{x}-\underline{\mu}_k)}$$

The EM Algorithm for Gaussian Mixture Models

We define the EM (Expectation-Maximization) algorithm for Gaussian mixtures as follows. The algorithm is an iterative algorithm that starts from some initial estimate of Θ (e.g., random), and then proceeds to iteratively update Θ until convergence is detected. Each iteration consists of an E-step and an M-step.

E-Step: Denote the current parameter values as Θ. Compute w(ik) (using the equation above for membership weights) for all data points x(i), $1 \le i \le N$ and all mixture components $1 \le k \le K$. Note that for each data point x(i) the membership weights are defined such that $\sum_{k=1}^{K} w_{ik} = 1$. This yields an N × K matrix of membership weights, where each of the rows sum to 1.

M-Step: Now use the membership weights and the data to calculate new parameter values. Let N(k) = $\sum_{i=1}^{N} w_{ik}$, i.e., the sum of the membership weights for the kth component—this is the effective number of data points assigned to component k. Now update parameters as:

$$\alpha_k^{new} = \frac{N_k}{N}, \quad 1 \le k \le K.$$

$$\underline{\mu}_k^{new} = \left(\frac{1}{N_k}\right) \sum_{i=1}^{N} w_{ik} \cdot \underline{x}_i \quad 1 \le k \le K.$$

$$\Sigma_k^{new} = \left(\frac{1}{N_k}\right) \sum_{i=1}^{N} w_{ik} \cdot (\underline{x}_i - \underline{\mu}_k^{new})(\underline{x}_i - \underline{\mu}_k^{new})^t \quad 1 \le k \le K.$$

The equations in the M-step need to be computed in this order, i.e., first compute the K new α's, then the K new μ(k)'s, and finally the K new Σ(k)'s.

After we have computed all of the new parameters, the M-step is complete and we can now go back and recompute the membership weights in the E-step, then recompute the parameters again in the E-step, and continue updating the parameters in this manner. Each pair of E and M steps is considered to be one iteration.

Initialization and Convergence for EM

The EM algorithm can be started by either initializing the algorithm with a set of initial parameters and then conducting an E-step, or by starting with a set of initial weights and then doing a first M-step. The initial parameters or weights can be chosen randomly (e.g. select K random data points as initial means and select the covariance matrix of the whole data set for each of the initial K covariance matrices) or could be chosen via some heuristic method (such as by using the k-means algorithm to cluster the data first and then defining weights based on k-means memberships).

Convergence is generally detected by computing the value of the log-likelihood after each iteration and halting when it appears not to be changing in a significant manner from one iteration to the next. Note that the log-likelihood (under the IID assumption) is defined as follows:

$$\log l(\Theta) = \sum_{i=1}^{N} \log p(\underline{x}_i|\Theta) = \sum_{i=1}^{N} \left(\log \sum_{k=1}^{K} \alpha_k p_k(\underline{x}_i|z_k, \theta_k) \right)$$

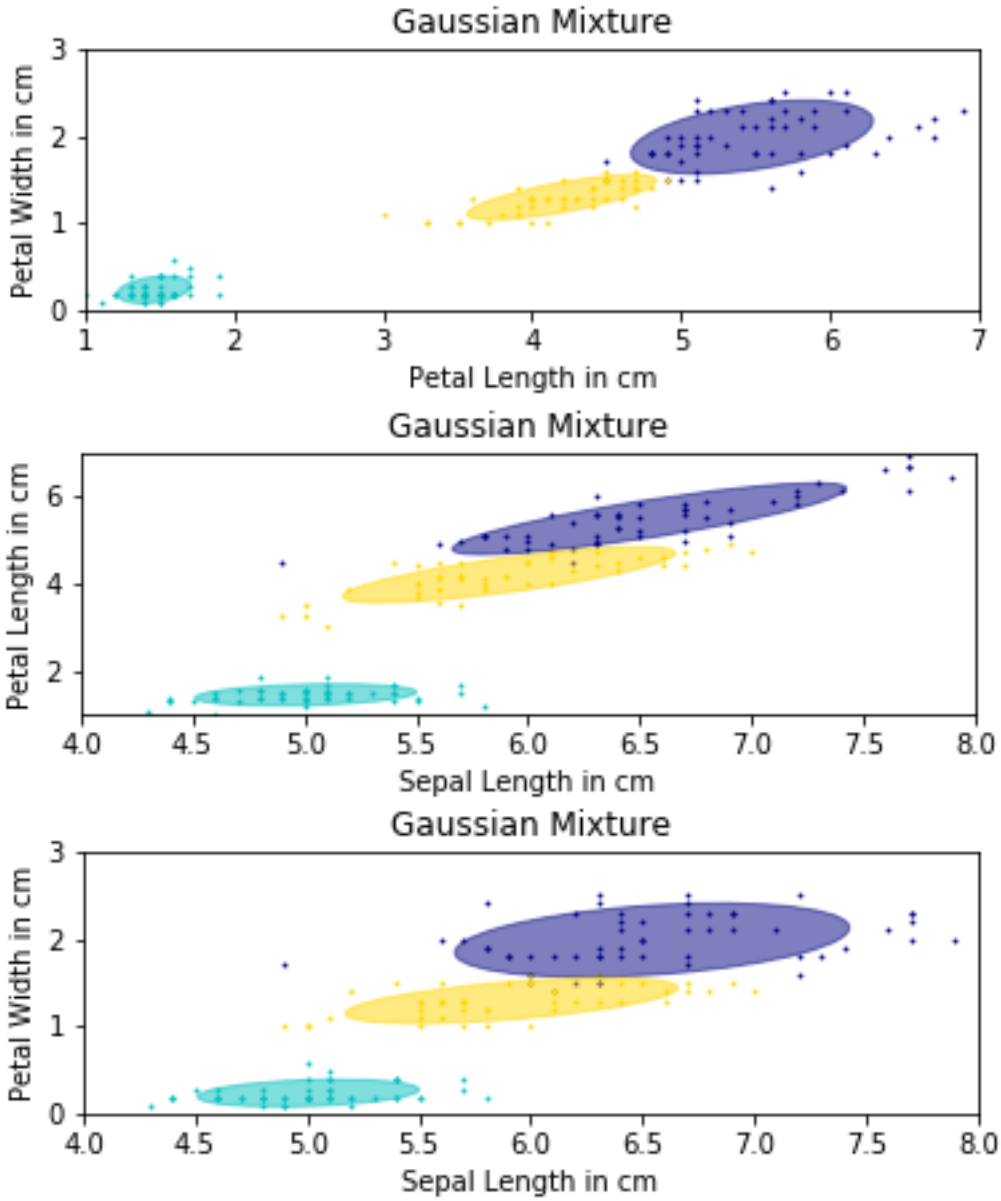
where pk(x(i)|z(k), θ(k)) is the Gaussian density for the kth mixture component.

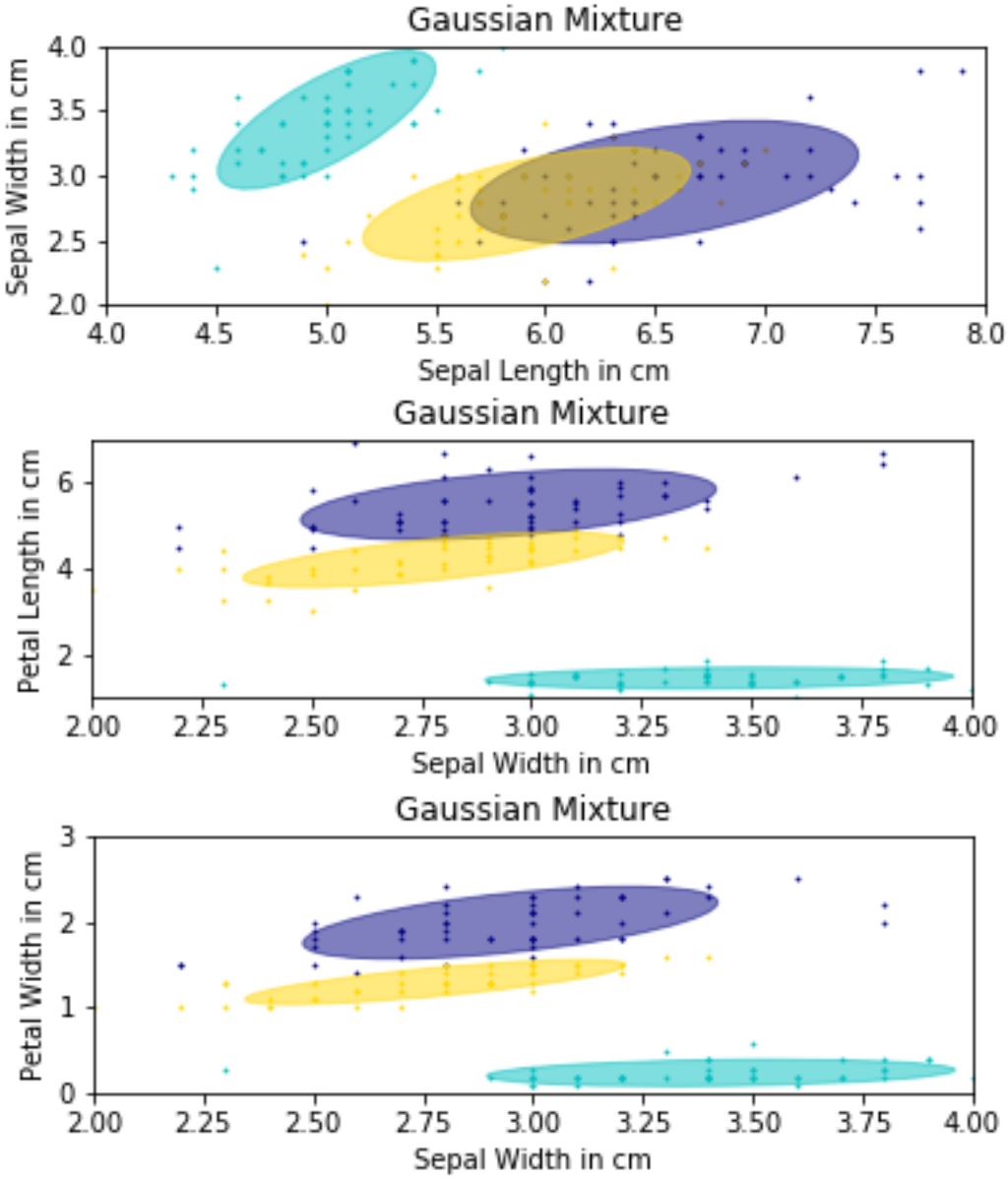
Which Data Set am I choosing & Initializations?

I am choosing the Iris data set which contains 4 features – petal length, sepal length, petal width and petal length. Based on these 4 features, there are 3 classes – Sertosa, Vesicolor and Virginica. I am choosing K=3 i.e. I will be trying to cluster the data with 3 different Gaussian Distributions. I have further chosen the initial mean array to be the mean of entire data + a random number from (0,1) as each of its entry. I have further initialised each of the covariance matrix as the identity matrix plus a random matrix with entries from (0,1). I have also set my tolerance as 1e-4 which means our implementation stops once the difference between successive log likelihoods have reduced beyond the tolerance.

Results:

With a log likelihood of -180, The density approximation was able to perform extremely well, with correctly clustering all the points for Sertosa and Virginica and about 45/50 of Vesicolor using one Gaussian component for each of the classes. The graphs are as follows:




Gaussian Mixture
Sepal Width in cm
Sepal Length in cm
4.0
3.5
3.0
2.5
2.0
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
Gaussian Mixture
Petal Length in cm
Sepal Width in cm
6
4
2
2.00
2.25
2.50
2.75
3.00
3.25
3.50
3.75
4.00
Gaussian Mixture
Petal Width in cm
Sepal Width in cm
3
2
1
0
2.00
2.25
2.50
2.75
3.00
3.25
3.50
3.75
4.00