The EM Algorithm for Gaussian Mixtures

1 Introduction

The Expectation Maximization (EM) algorithm is a general iterative method to find the maximum likelihood estimates of parameters in a statistical model. So far, we've applied the EM algorithm in the context of Bayesian Networks. In this context we were given some samples that contains Missing Completely at Random / Missing at Random data, and our goal was to infer the parameters of the Bayesian Network (tables of probabilities).

We've previously mentioned that the EM algorithm is a general method that can be applied to a variety of other problems such as: Hidden Markov Models and Gaussian Mixtures. In this homework we will apply the EM algorithm to a Gaussian Mixture model, trying to infer the underling parameters of the generative statistical model as we will further discuss.

2 Task

Given an input file that contains samples from a Gaussian Mixture, implement the Expectation-Maximization Algorithm for a d-dimensional input space to infer the underling parameters of the generative process. Test your implementation using the samples from **GMM.in**. Each line in the input file contains the coordinates of a 2-D vector separated by a white space, corresponding to samples generated from a Gaussian Mixture with 4 components.

3 Generative Process

We will begin by describing the generative process. For convenience we will discuss 1-D case and later we will formulate the problem for the general d-dimensional case.

Suppose we are given N points on the real line. Our goal is to infer the parameters that govern their generation process. We will assume that those points are drawn from multiple Gaussian Distribution (hence the name - Gaussian Mixture). For example, in Figure 1, we have 11 points drawn from two Gaussian distributions.

The generative process consists of two steps. First we will flip a coin to pick one of the two distributions. Notice that in the general case where we have

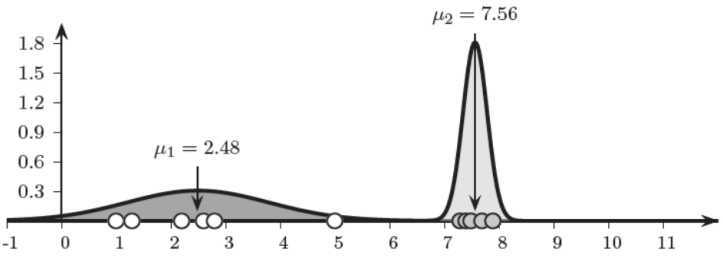

Figure 1: True distribution

K such Gaussian distributions, we can consider rolling a dice with K sides. Continuing in our simplified example, we consider that *Tails(T)* corresponds in picking the dark-grey distribution and *Heads(H)* corresponds in picking the light-grey distribution. We encode the outcome (H/T) using an indicator binary vector z. In our case, the vector $z = (1, 0)$ if the coin lands T and $z = (0, 1)$ if it lands H (the encoding can be done using a one dimensional vector, although for generality we will stick to this encoding scheme). Suppose that the coin lands on H ($z = (0, 1)$). Then the next step is draw a sample from the light-grey Gaussian distribution. We repeat the same process, flip a coin and then draw a sample from the corresponding distribution, over and over again. In this way we generate the samples (our dataset).

Having the generative process described, our task is to infer the underlying parameters. First notice that our coin flip corresponds to a Bernoulli process. For the general case where we roll a dice with K sides, we have a categorical distribution described by parameters $\alpha_0, \alpha_1, ..., \alpha_K$, such that $\forall i, \alpha_i \geq 0$ and $\sum_i \alpha_i = 1$. Second of all, each Gaussian distribution is described by two parameters, the mean μ and the variance σ. In our simplified example, both parameters, $\mu \in \mathbb{R}$ and $\sigma \in \mathbb{R}_+^*$ is a positive real number. In a d-dimensional case, $\mu \in \mathbb{R}^d$, and the variance becomes a co-variance matrix $\Sigma \in \mathbb{R}^{d \times d}$.

In order to have an one-to-one correspondence between our simplified example and the mathematical formalism that we will introduce in the following section, we use the following notation:

1. $\alpha = \{\alpha_1, \alpha_2\}$ describes the coin flip distribution. The probability of landing T is α_1 and the probability of landing H is α_2.

2. $\theta_1 = \{\mu_1, \sigma_1^2\}$ are the mean and the variance of the dark-gray distribution

3. $\theta_2 = \{\mu_2, \sigma_2^2\}$ are the mean and the variance of the light-gray distribution

Now we can define the parameters that we need to infer using the EM algorithm, namely: $\Theta = \{\alpha_1, \alpha_2, \theta_1, \theta_2\}$

The EM - algorithm will follow an iterative procedure. We start with an initial guess of the parameters (this can be a random initialization or a more sophisticate heuristic), and we try to improve our estimates by maximizing the

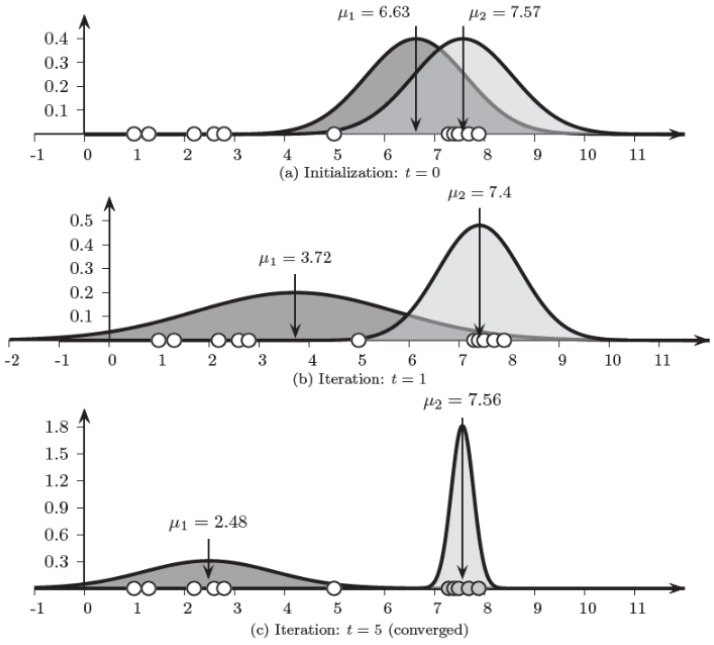

Figure 2: EM procedure.

lower-bound on the likelihood, which implicitly increases the likelihood of our data. The iterative process is illustrated in Figure 2

4 Mathematical Formalism

4.1 Finite Mixture Models

We are given a data set $D = \{\underline{x}_1, ..., \underline{x}_N\}$ where $\underline{x}_i$ is a d-dimensional vector measurement. Assume that the points are generated in an IID fashion from and underlying density $p(\underline{x})$. We further assume that $p(\underline{x})$ is defined as a finite mixture model with K components:

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

where:

- $p_k(\underline{x}|z_k, \theta_k)$ are mixture components $1 \leq k \leq K$. Each is a density or distribution defined over $p(\underline{x})$, with parameters θ_k.
- $z = (z_1, ..., z_K)$ is a vector of K binary indicator variable 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 componet that generate $\underline{x}$. It is convenient for mixture models to represent z as a vector of K indicator variables.

- $\alpha_k = p(z_k)$ are the mixture weights, representing the probability that a randomly selected $\underline{x}$ 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 $\Theta = \{\alpha_1, ..., \alpha_k, \theta_1, ..., \theta_K\}$.

4.2 Membership Weights

We can compute the "membership weight" of a data point $\underline{x}_i$ in cluster k, given parameters Θ as:

$$w_{ik} = p(z_{ik} = 1|\underline{x}_i, \Theta) = \frac{p_k(\underline{x}_i|z_k, \theta_k) \cdot \alpha_k}{\sum_{m=1}^{K} p_m(\underline{x}_i|z_m, \theta_m) \cdot \alpha_m}, 1 \leq k \leq K, 1 \leq i \leq N. \tag{2}$$

This follows from a direct application of Bayes rule.

The membership weights above reflect our uncertainty, give $\underline{x}_i$ and Θ, about which of the K components generated vector $\underline{x}_i$. Note that we are assuming in our generative mixture model that each $\underline{x}_i$ was generated by a single component - so these probabilities reflect our uncertainty about which component $\underline{x}_i$ came from, not any "mixing" in the generative process.

4.3 Gaussian Mixture Models

Fro $\underline{x} \in \mathbb{R}^d$ we can define a Gaussian mixture model by making each of the K components a Gaussian density with parameters $\underline{\mu}_k$ and Σ_k. Each component is a multivariate Gaussian density

$$p_k(\underline{x}|\theta_k) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_k|^{\frac{1}{2}}} e^{-\frac{1}{2}(\underline{x}-\underline{\mu}_k)^t \Sigma_K^{-1}(\underline{x}-\underline{\mu}_k)} \tag{3}$$

with its own parameters $\theta_k = \{\underline{\mu}_k, \Sigma_k\}$.

4.4 The EM Algorithm for Gaussian Mixture Models

We define the EM (Expectation-Maximization) algorithm for Gaussian mixtures as follows. The algorithm is an iterative procedure that starts from some initial estimates 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 $\underline{x}_i$, $1 \leq i \leq N$ and all mixture components $1 \leq k \leq K$. Note that for each data point $\underline{x}_i$ the membership weights are defined such that $\sum_{k=1}^{K} w_{ik} = 1$. This yields an $N \times K$ 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 k-th component -this is the effective number of data points assigned to component k.

Specifically,

$$\alpha_k^{new} = \frac{N_k}{N}, 1 \leq k \leq K. \tag{4}$$

These are the new weights mixture weights.

$$\underline{\mu}_k^{new} = \frac{1}{N_k} \sum_{i=1}^{N} w_{ik} \cdot \underline{x}_i, 1 \leq k \leq K. \tag{5}$$

The update mean is calculated in a manner similar to how we could compute a standard empirical average except that the i-th data vector $\underline{x}_i$ has a fractional weight w_{ik}. Note that this is a vector equation since $\underline{\mu}_k^{new}$ and $\underline{x}_i$ are both d-dimensional vectors.

$$\Sigma_k^{new} = \frac{1}{N_k} \sum_{i=1}^{N} w_{ik} \cdot (\underline{x}_i - \underline{\mu}_k^{new})(\underline{x}_i - \underline{\mu}_k^{new})^t, 1 \leq k \leq K. \tag{6}$$

Again we get an equation that is similar in form of how we would normally compute an empirical covariance matrix, except that the contribution of each data point is weighted by w_{ik}. Note that this is a matrix is a matrix equation of dimensionality $d \times d$ on each side.

The equation in the M-step need to be computed in this order, i.e., first compute the K new α's, then the K new $\underline{\mu}_k$'s, and finally the K new Σ_k's.

4.5 Initialization and Convergence Issues for EM

The EM algorithm can be started by either initializing the algorithm with a set of initial parameters and the conduction 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 points as initial means and select the covariance matrix of the whole dataset for each of the initail K covarience matrices) or could be chose 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).

Converge 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:

$$\log l(\Theta) = \sum_{i=1}^{N} \log p(\underline{x}_i|\Theta) = \sum_{i=1}^{N} (\log \sum_{k=1}^{K} \alpha_k p_k(\underline{x}_i|z_k, \theta_k)) \tag{7}$$

where $p_k(\underline{x}_i|z_k, \theta_k)$ is a Gaussian density for the k-th mixture component.

4.6 Reference values

The EM algorithm should provide estimates closed to the reference values in the Table 1.

μ_1	Σ_1	
0.0	1.0	0.0
0.0	0.0	1.0

μ_2	Σ_2	
5.0	2.0	1.0
0.0	1.0	2.0

μ_1	Σ_3	
-2.0	4.0	-1.3
-5.0	-1.3	5.0

μ_4	Σ_4	
-3.0	2.3	-1.7
7.0	-1.7	4.2

α			
1	2	3	4
0.15	0.1	0.5	0.25

Table 1: Reference parameters.

The visualization of the data points and the true labels are depicted in Figure3.

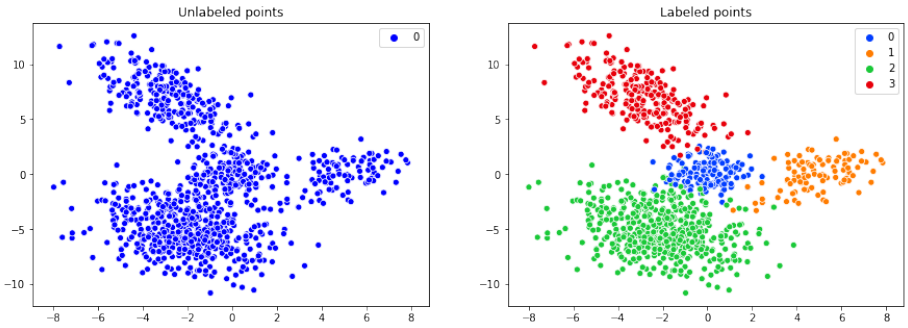

Figure 3: Data visualization.