

EM (Expectation-Maximization) Algorithm with An Application for Gaussian Mixture Model

Frances Lin

Fall 2020

Introduction

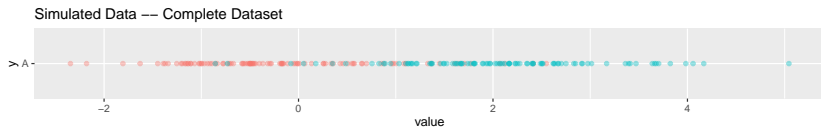
EM algorithm, short for “Expectation-Maximization” algorithm is an iterative method that is particular useful for finding maximum likelihood estimate (MLE) for missing data or when maximizing the likelihood function is challenging.

It is an umbrella term for a class of algorithm that iterates between expectation and maximization. Applications include EM algorithm for missing data, for censored data, for finite mixture models, etc.

Simulated data: 2-component Gaussian mixture

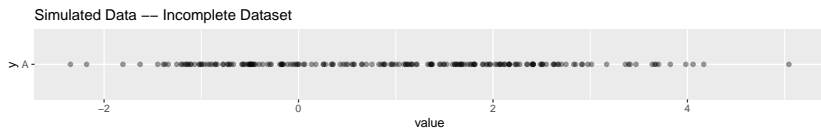
For illustration, we apply EM algorithm to a simulated dataset that follows a two-component Gaussian mixture distribution.

We simulate 100 data points that follow $\mathcal{N}(0, 1)$ and 100 data points that follow $\mathcal{N}(2, 1)$.



Simulated data: 2-component Gaussian mixture

Now suppose that we do not know which distribution each data point is from (i.e. the latent (or missing) variable Z is involved.) This is when the EM algorithm can be of use.



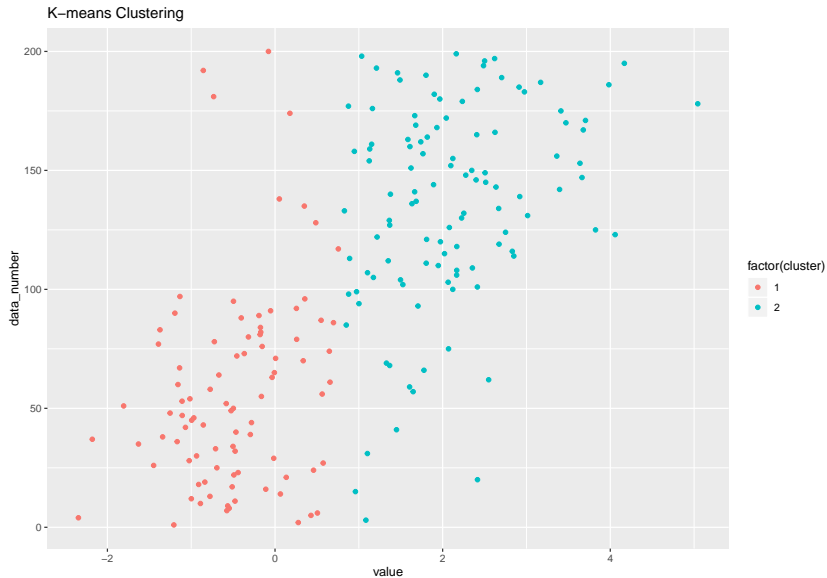
EM algorithm for Gaussian mixture

The algorithm consists of three steps:

1. Initialization
 2. The E-step
 3. The M-step
- Iteration between E-step and M-step untill convergence

Step 1: Initialization using K-means clustering

Set initial estimates for the first EM iteration using the `kmeans` function from the `stats` package



Step 2: The E-step

E-step in literature

$$Q(\theta, \theta^0) = E_{Z|X, \theta^0} \log(p(X, Z|\theta)) = \sum_Z p(Z|X, \theta^0) \log(p(X, Z|\theta))$$

Implementing

1. Calculate posterior probability (or soft labelling) $p(Z|X)$ using data X and initial estimates θ^0 and pass it to M-step
2. Compute and store the log likelihood to check for convergence

Step 3: The M-step

M-step in literature

$$\hat{\theta} = \theta^{0+1} = \operatorname{argmax}_{\theta} Q(\theta, \theta^0)$$

Implementing

1. Replace hard labelling N_k with posterior probability (or soft labelling) $p(Z|X)$ and optimize the parameters using MLE
2. Return the final estimates that maximize the likelihood, if convergence happens

Convergence: Iterate between E-step and M-step until convergence

Compare log-likelihood of current iteration to log-likelihood of previous iteration to see if the change is minimal

If the change is minimal (i.e. convergence), we stop and return the final estimates. If it isn't, then we repeat another EM step.

```
...
```

```
...
```

```
if (convergence){
```

```
    break
```

```
} else {
```

```
    1. set  $\text{log-likelihood}^{0+1}$  to  $\text{log-likelihood}^0$ 
```

```
    2. repeat E-step and M-step
```

```
}
```

Convergence: Iterate between E-step and M-step until convergence

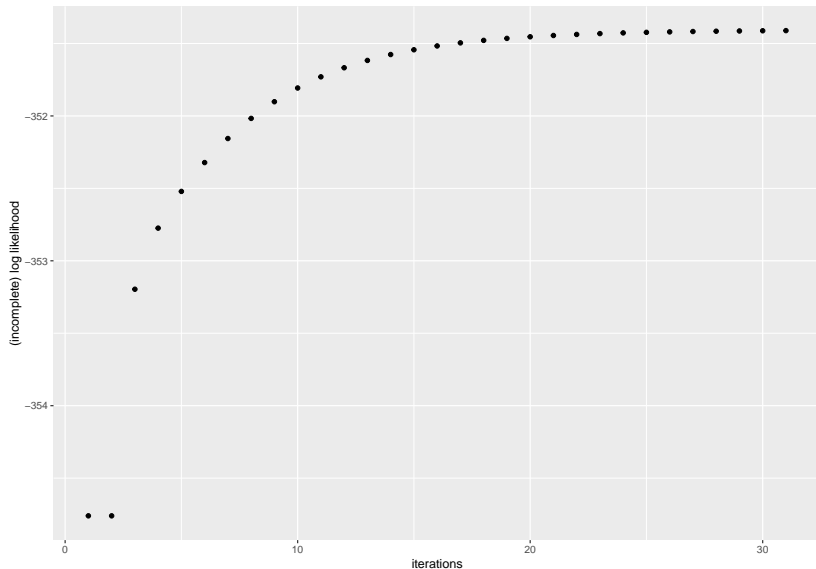
The final estimates of parameters (i.e. μ, σ, π) are:

cluster	mean	sd	pi
1	-0.6611	0.591	0.3447
2	1.786	1.078	0.6553

Convergence happens at:

max_log_likelihood	#s of iterations
-351.4	31

Convergence: Iterate between E-step and M-step until convergence



Thank you!

EM (Expectation-Maximization) Algorithm with An Application for
Gaussian Mixture Model

Frances Lin

MS student, Dept. of Statistics, Oregon State University

GitHub project repository