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 MLE for missing data or when maximizing the likelihood function is challenging.

EM algorithm 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.

Application

For illustration, we apply EM algorithm to a simulated dataset that follows a two-component Gaussian mixture distribution. Below is an abridged version of the full report, which can be found here.

Simulated data

We use the `rnorm` function from the `stats` package to simulate 100 data points that follow $\mathcal{N}(0,1)$ and 100 data points that follow $\mathcal{N}(2,1)$.

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.

Initialization using K-means clustering

We use the `kmeans` function from the `stats` package to obtain initial values for the first EM initial iteration, as it is a common practice to use k-means clustering to find such values.

The estimates of parameters (i.e. μ, σ, π) obtained from K-means are:

cluster	mean	sd	pi
1	2.099	0.8587	0.55
2	-0.4712	0.6671	0.45

E-step: Calculate posterior probability (or soft labelling) using Bayes Rule and pass it to M-step & store log likelihood to check for convergence

In the E-step of the first iteration, we calcuate the posterior probability of the latent variable $Z_i = k$ given the observations X_i (i.e. x_i belongs to the k^{th} cluster) using the estimates of parameters obtained from K-means.

The output of the first 3 values of each distribution are:

value	post_1	post_2
-1.207	0.03252	0.9675
0.2774	0.5783	0.4217
1.084	0.9849	0.01509
2.415	1	1.54e-06
1.525	0.9989	0.001071
2.066	1	2.418e-05

In the E-step, we also compute and store the log likelihood to check for convergence.

M-step: Replace hard labelling with posterior probability (or soft labelling) and optimize the parameters using MLE & return final estimates if convergence happens

In the M-step, we replace hard labelling with posterior probability (or soft labelling) $p(Z_i = k|X_i)$ and re-estimate the parameters using MLE. In the M-step, if convergence happens, we also return the final estimates that maximize the likelihood.

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

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

Convergence: Iterate between the E-step and the M-step untill convergence

We compute and store the log likelihood at each EM iteration and compare this log likelihood to the log likehood of the previous iteration to see if the change is minimal. If the change is minimal (i.e. convergence), we stop and the estimates that the M-step returns are the final estimates. If it isn't, then we repeat another EM step.

Convergence occurs at 31st iteration with log likelihood = -351.4:

max_log_likelihood	#s of iterations
-351.4	31

Discussion

We see that the final estimates are not quite what we look for. This could be because of the initial values using K-means clustering, but the major reason is perhaps due to the simulated data. There is a lot of overlap between the 100 simulated data points that follow $\mathcal{N}(0, 1)$ and another 100 data points that follow $\mathcal{N}(2, 1)$. We expect EM algorithm to perform better if the overlap is not as wide as in our example, and it does (We re-simulate 100 data points that follow $\mathcal{N}(0, 1)$ and another 100 data points that follow $\mathcal{N}(4, 1)$ and find that the estimates are more accurate and convergence happens earlier now at 12th iteration.)

We also see how EM algorithm can be applied to two-component Gaussian mixture model. A logical next step is to expand the algorithm for k-component Gaussian mixture model and for other mixture model such as multinomial.

Reference

Fitting a Mixture Model Using the Expectation-Maximization Algorithm in R

Introduction to EM: Gaussian Mixture Models

Terminology

EM algorithm and GMM model

Maximum likelihood estimation

Mixture model

Posterior probability