Data Analysis - EM Algorithm (2023)

Richard CHEAM

2023-11-30

EM algorithm for Gaussian mixture model of K components

The overall mixture model, in this case, GMM, is given by

$$\mathbb{P}(x_i/\theta) = \sum_{k=1}^{K} \pi_k \mathbb{P}_k(x_i/\theta) = \sum_{k=1}^{K} \pi_k \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{1}{2}(\frac{x_i-\mu_k}{\sigma_k})^2}$$

Having observation:

$$X = (x_1, x_2, ..., x_n)$$

Latent variable (missing data):

$$Z = (z_1, z_2, ..., z_k)$$

Thus, the goal is to estimate the missing parameter:

$$\theta = \{\pi_{1,...,k-1}, \mu_{1,...,k}, \sigma^2_{1,...,k}\}$$

Implement the initialization of the EM algorithm

```
#k is the number of components in the mixture model
init <- function(X, k){
  #unknown parameter vector
  #theta = list()
  #proportion
  pi = rep(1/k, k) #replicate a vector contains 1/k at each value of k-length
  mu <- X[sample(1:length(X),k)]
  sigma <- rep(1, k)
  return (c(pi,mu,sigma))
}
#X[sample(1:n,k)]
```

Implement the E-step

At iteration q, E step computes $Q(\theta|\theta^{(q)})$

$$Q(\theta|\theta^{(q)}) = \mathbb{E}_{Z|X;\theta^{(q)}}[\log \mathbb{P}(X, Z; \theta)] = \mathbb{E}_{Z|X;\theta^{(q)}}[\log \prod_{i=1}^{n} \mathbb{P}(x_i, z_i; \theta)]$$

$$\sum_{i=1}^{n} \mathbb{E}_{z_i|x_i;\theta^{(q)}} \left[\log \left(\mathbb{P}(z_i) \times \mathbb{P}(x_i|z_i;\theta)\right)\right]$$

Let $z_{ik} = 1_{\{z_i = k\}}$

$$Q(\theta|\theta^{(q)}) = \sum_{i=1}^{n} \mathbb{E}_{z_i|x_i;\theta^{(q)}} \left[\sum_{k=1}^{K} z_{ik} \log \left(\mathbb{P}(z_i = k) \times \mathbb{P}(x_i|z_i = k;\theta)\right)\right]$$

$$= \sum_{i=1}^{n} \sum_{k=1}^{K} \mathbb{P}(z_i = k|x_i;\theta^{(q)}) \log \left(\mathbb{P}(z_i = k) \times \mathbb{P}(x_i|z_i = k;\theta)\right)$$

$$= \sum_{i=1}^{n} \sum_{k=1}^{K} \mathbb{P}(z_i = k|x_i;\theta^{(q)}) \log \left(\pi_k \times \mathbb{P}_k(x_i;\mu_k,\sigma_k)\right)$$

We have

$$t_{ik}^{(q)} = \mathbb{P}(z_i = k|x_i;\theta^{(q)}) = \frac{\pi_k^{(q)} \mathbb{P}_k(x_i;\mu_k^{(q)},\sigma_k^{(q)})}{\sum_{m=1}^{K} \pi_m^{(q)} \mathbb{P}_m(x_i;\mu_m^{(q)},\sigma_m^{(q)})}$$

At each iteration q, we only need to compute $t_{ik}^{(q)}, i \in [\![1,n]\!], k \in [\![1,K]\!]$

```
#theta = (pi,mu,sigma^2)
E_step <- function(X, K, theta) {
  pi <- theta[1:K]
  mu <- theta[(K+1):(2*K)]
  sigma <- theta[(2*K)+1:K]
  t <- matrix(0, nrow = length(X), ncol = K)
  for (i in 1:length(X)) {
    for (k in 1:K) {
      t[i, k] <- (pi[k] * dnorm(X[i], mu[k], sqrt(sigma[k]))) / sum(pi * dnorm(X[i], mu, sqrt(sigma)))
    }
  }
  return(t)
}
```

Implement the M-step

At iteration q, M step computes

$$\theta^{(q+1)} = \arg\max_{\theta} Q(\theta|\theta^{(q)})$$

By optimizing Q with respect to π and θ, we then have

$$\pi_k^{(q+1)} = \frac{\sum_{i=1}^{n} t_{ik}^{(q)}}{\sum_{i=1}^{n} \sum_{m=1}^{K} t_{im}^{(q)}} = \frac{\sum_{i=1}^{n} t_{ik}^{(q)}}{n}, \; k \in [\![1,K]\!]$$

$$\mu_k^{(q+1)} = \frac{\sum_i t_{ik} x_i}{\sum_i t_{ik}}$$

$$\sigma_k^{2(q+1)} = \frac{\sum_i t_{ik}(x_i - \mu_k)(x_i - \mu_k)^T}{\sum_i t_{ik}} = \frac{\sum_i t_{ik} x_i x_i^T}{\sum_i t_{ik}} - \mu_k \mu_k^T$$

```r
#(sum(t[,k]*(X*t(X)))/sum(t[,k]))-mu*t(mu)
# t : result of E step
M_step <- function(X, K, t) {
  pi <- sapply(1:K, function(k) sum(t[,k]) / length(X))
  mu <- sapply(1:K, function(k) sum(t[,k]*X) / sum(t[,k]))
  sigma <- sapply(1:K, function(k) sum(t[,k]*(X-mu[k])^2) / sum(t[,k]) )
  theta <- c(pi, mu, sigma)
  return(theta)
}
```

Test EM algorithm on the simulated data from Exercise 1 of the worksheet

The convergence is attained by taking the condition:

$$\frac{||\theta^{(q)} - \theta^{(q+1)}||_2^2}{||\theta^{(q)}||_2^2} < \epsilon = 10^{-6}$$

```r
parameters <- list()
parameters$mean = c(0, 4)
parameters$pi = c(1/3, 1-1/3)
parameters$sd = c(1,1/2)
n = 1000

Z <- rmultinom(n, 1, prob = parameters$pi)
effectifs <- rowSums(Z)

X <- c(rnorm(effectifs[1], mean = parameters$mean[1], sd = parameters$sd[1]),
       rnorm(effectifs[2], mean = parameters$mean[2], sd = parameters$sd[2]))
```

```r
K <- 2
theta <- init(X, K)

repeat {

  # E step
  t <- E_step(X, K, theta)

  # M step
  new_theta <- M_step(X, K, t)

  # convergence
  if (sum((theta - new_theta)^2) / sum((theta)^2) < 1e-6) {
    break
  }

  theta <- new_theta

}
```

```r
print(paste(c("pi_1 =", "pi_2 =", "mu_1 =", "mu_2 =", "sigma_1 =", "sigma_2 ="), theta))
```

```
## [1] "pi_1 = 0.323536083440999"    "pi_2 = 0.676463916559001"
## [3] "mu_1 = -0.00372924672227759" "mu_2 = 4.01337898301646"
## [5] "sigma_1 = 1.1094246930573"   "sigma_2 = 0.241830375748379"
```

```r
trueClass <- rep(c(1,2), effectifs)

# Comparison between true classes and EM clusters
EM_cluster <- c()
for (row in 1:nrow(t)) {
  EM_cluster <- c(EM_cluster, which.max(t[row,]))
}

table(EM=EM_cluster, True=trueClass)
```

```
##    True
## EM    1   2
##   1 321   0
##   2   2 677
```

Based on the table above, we can see that the clustering was done well by the algorithm.