Tutorial: Gaussian mixture model and the EM algorithm

MAP573 – Introduction to unsupervised learning

École Polytechnique - Autumn 2019

1 Gaussian Mixture Models

We consider a collection of random variables $(X_1, \dots, X_n)$ associated with n individuals drawn from Q populations. The label of each individual describes the population (or class) to which it belongs and is unobserved. The Q classes have *a priori* distribution $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_Q)$ with $\alpha_q = \mathbb{P}(Z_i \in q)$. In other word, the latent variable $Z_i \in \{1, .., Q\}$ indicating the label follow a multinomial distribution $Z_i \sim \mathcal{M}(1, \boldsymbol{\alpha})$, such as $\sum_{q=1}^{Q} \alpha_q = 1$.

The distribution of X_i conditional on the label of i is assumed to be a univariate gaussian distribution with unknown parameters, that is, $X_i | Z_i = q \sim \mathcal{N}(\mu_q, \sigma_q^2)$.

Hence, the marginal likelihood can be written as

$$L(X) = p(X) = \prod_{i=1}^{n} p(X_i) = \prod_{i=1}^{n} \sum_{q=1}^{Q} p(X_i, Z_i = q) = \prod_{i=1}^{n} \sum_{q=1}^{Q} p(Z_i = q) p(X_i | Z_i = q) \tag{1}$$

Equivalently, the loglikelihood is

$$\log L(X) = \sum_{i=1}^{n} \log \sum_{q=1}^{Q} p(Z_i = q) p(X_i | Z_i = q)$$

Alternative (equivalent) formulation

Introducing the additional notation $Z_{iq} = \mathbb{1}_{\{Z_i = q\}}$, then the following writting will be useful:

$$\log p(Z_i) = \log \left(\prod_{q=1}^{Q} (\alpha_q)^{Z_{iq}} \right) = \sum_{q=1}^{Q} Z_{iq} \log(\alpha_q),$$

which is non zero just when q is the effective realization of Z_i. Similarly [1],

$$\log p(X_i | Z_i) = \log \left(\prod_{q=1}^{Q} (\mathcal{N}(X_i; \mu_q, \sigma_q))^{Z_{iq}} \right) = \sum_{q=1}^{Q} Z_{iq} \log(\mathcal{N}(X_i; \mu_q, \sigma_q)).$$

[1] We denote by $\mathcal{N}(X_i; \mu_\ell, \sigma_\ell)$ the density of a normal distribution at point X_i, with a slight notational abuse.

Hence, the complete data likelihood can be written

$$L(X,Z) = p(X,Z) = \prod_{i=1}^{n} p(X_i, Z_i) = \prod_{i=1}^{n} p(Z_i) p(X_i|Z_i) = \prod_{i=1}^{n}\prod_{q=1}^{Q} (\alpha_q \mathcal{N}(X_i; \mu_q, \sigma_\ell))^{Z_{iq}},$$

and then

$$\log L(X,Z) = \sum_{i=1}^{n}\sum_{q=1}^{Q} Z_{iq} \log(\alpha_q \mathcal{N}(X_i; \mu_q, \sigma_q)), \tag{2}$$

Keep in mind that the difference between the marginal likelihood and the complete likelihood is that the Z_i are given in the latter, not in the former. Then in expression (1) the summatio over the class is indeed an integration over all possible values, while in (2) all element in $\sum_q$ will be zero but the one truly observed (the q such that $Z_{iq} = 1$).

2 Questions

2.1 *Likelihood.* Write the model complete-data loglikelihood.

We denote the vector of parameters to be estimated by $\mu = (\mu_1, \ldots, \mu_Q)$, $\boldsymbol{\sigma^2} = (\sigma_1^2, \ldots, \sigma_Q^2)$, $\boldsymbol{\tau} = (\tau_{iq, i=1,\ldots,n; q=1,\ldots Q})$. The negative complete-data loglikelihood is derived as follows

$$\begin{aligned} \log L(\boldsymbol{\mu}, \boldsymbol{\sigma}^2, \boldsymbol{\tau}; \mathbf{x}) &= \sum_{i=1}^{n} \log\left(\prod_{q=1}^{Q} \alpha_q f(x_i; \mu_q, \sigma_q^2)^{Z_{iq}}\right) \\ &= \sum_{i=1}^{n}\sum_{q=1}^{Q} Z_{iq}\left(\log \frac{\alpha_q}{\sigma\sqrt{2\pi}} \exp\{\frac{1}{2\sigma^2}(x_i - \mu_q)^2\}\right) \\ &= \sum_{i=1}^{n}\sum_{q=1}^{Q} Z_{iq}\left(\log \alpha_q - \log \sigma - \log(\sqrt{2\pi}) - \frac{1}{2\sigma_q^2}(x_i - \mu_q)^2\right) \end{aligned}$$

Note that

2.2 *E-step.*

For fixed values of $\hat{\mu}_q, \hat{\sigma}_q^2$ and $\hat{\alpha}_q$, give the expression of the estimates of the posterior probabilities $\tau_{iq} = \mathbb{P}(Z_{iq} = 1|X_i)$.

$$\tau_{iq} = \frac{\hat{\alpha}_q f(x_i; \hat{\mu}_q, \hat{\sigma}_q^2)}{\sum_{q=1}^{Q} \hat{\alpha}_q f(x_i; \hat{\mu}_q, \hat{\sigma}_q^2)}, \tag{3}$$

where f is the density of the normal distribution.

2.3 *M-step.*

The maximization step consists in solving the following optimization problem

$$\arg\max_{\sigma_q,\mu_q,\alpha_q} \underbrace{\sum_{i=1}^{n}\sum_{q=1}^{Q}\hat{\tau}_{iq}\left(\log\alpha_q-\log\sigma-\log(\sqrt{2\pi})-\frac{1}{2\sigma_q^2}(x_i-\mu_q)^2\right)}_{Q(\boldsymbol{\sigma},\boldsymbol{\mu},\boldsymbol{\alpha};\hat{\boldsymbol{\tau}})} \tag{4}$$

Consider first the mixture coefficients. We solve the above maximization problem under the constraint that the mixture coefficients sum to 1. This can be dealt with the Lagrange multiplier technique. By deriving the objective function w.r.t α_q, we get

$$\frac{\sum_i \tau_{iq}}{\alpha_q}+\lambda=0 \Leftrightarrow \alpha_q=\frac{\sum_i \tau_{iq}}{-\lambda} \tag{5}$$

where λ corresponds to the Lagrange multiplier associated with the constraint $\sum_q \alpha_q = 1$. If we sum the latter result over all q, we get that $1=\sum_q \tau_{iq}/(-\lambda)$. In other words, $\lambda=-\sum_q \tau_{iq}$ so that finally

$$\hat{\alpha}_q=\frac{\sum_{i=1}^{n}\tau_{iq}}{\sum_{i=1}^{n}\sum_{q=1}^{Q}\tau_{iq}} \tag{6}$$

Concerning, $\hat{\mu}_q$, null gradient condition leads to

$$\sum_i \frac{\tau_{iq}}{2\sigma_q^2}(x_i-\mu_q)=0 \Leftrightarrow \mu_q=\frac{\sum_i \tau_{iq}x_i}{\sum_i \tau_{iq}} \tag{7}$$

Similarly, for $\hat{\sigma}_q$, we get

$$\sum_{i=1}^{n}\tau_{iq}\left(-\frac{1}{2\sigma_q^2}+\frac{1}{2\sigma_q^4}(x_i-\mu_q)^2\right)=0 \Leftrightarrow \sigma_q^2=\frac{\sum_{i=1}^{n}\tau_{iq}(x_i-\mu_q)^2}{\sum_{i=1}^{n}\tau_{iq}} \tag{8}$$

- *Implementation.*

```
get_cloglik <- function(X, Z, theta) {
  alpha <- theta$alpha; mu <- theta$mu; sigma <- theta$sigma
  xs <- scale(matrix(X,length(x),length(alpha)),mu,sigma)
  res <- sum(Z*(log(alpha)-log(sigma)-.5*(log(2*pi)+xs^2)))
  res
}
M_step <- function(X, tau) {
  n <- length(X); Q <- ncol(tau)
  alpha  <- colMeans(tau)
  mu     <- colMeans(tau * matrix(X,n,Q)) / alpha
  sigma  <- sqrt(colMeans(tau*sweep(matrix(X,n,Q),2,mu,"-")^2)/alpha)
  list(alpha = alpha, mu = mu, sigma = sigma)
}

E_step <- function(X, theta) {
  probs <- mapply(function(alpha, mu, sigma) {
    alpha*dnorm(X,mu,sigma)
  }, theta$alpha, theta$mu, theta$sigma)
  likelihoods <- rowSums(probs)
  list(tau = probs / likelihoods, loglik = sum(log(likelihoods)))
}
```

```
EM_mixture <- function(X, Q,
                       init.cl = sample(1:Q,n,rep=TRUE), max.iter=100, eps=1e-5) {
    n <- length(X); tau <- matrix(0,n,Q); tau[cbind(1:n,init.cl)] <- 1
    loglik  <- vector("numeric", max.iter)
    Eloglik <- vector("numeric", max.iter)
    iter <- 0; cond <- FALSE

    while (!cond) {
        iter <- iter + 1
        ## M step
        theta <- M_step(X, tau)
        ## E step
        res_Estep <- E_step(X, theta)
        tau <- res_Estep$tau
        ## check consistency
        loglik[iter]  <- res_Estep$loglik
        Eloglik[iter] <- get_cloglik(X, tau, theta)
        if (iter > 1)
            cond <- (iter>=max.iter) | Eloglik[iter]-Eloglik[iter-1] < eps
    }

    res <- list(alpha = theta$alpha,  mu = theta$mu,  sigma = theta$sigma,
                tau   = tau, cl = apply(tau, 1, which.max),
                Eloglik = Eloglik[1:iter],
                loglik  = loglik[1:iter])
    res
}
```

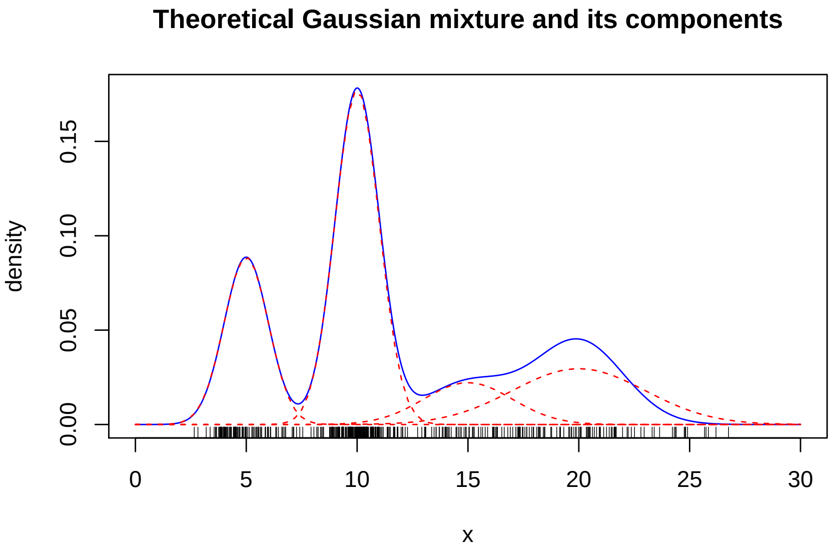
2.4 Examples

We test ICL and BIC on a simple example with 4 groups

Let us start with the data generation.

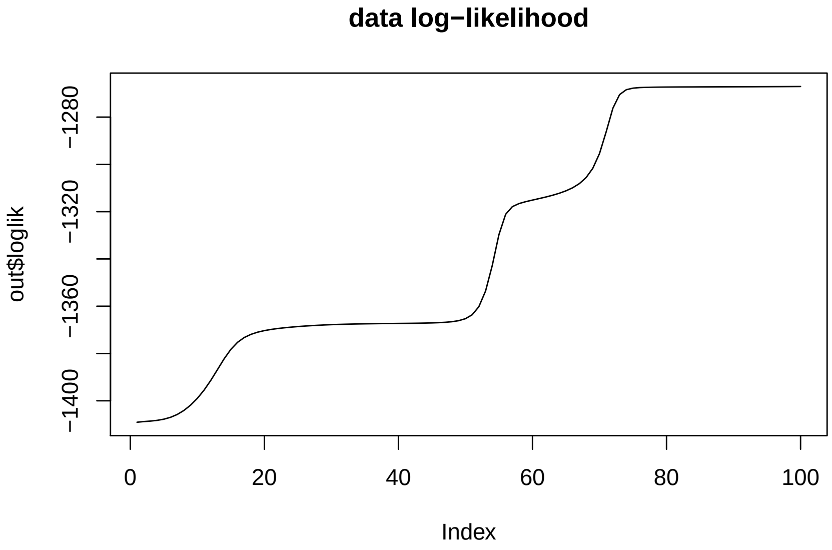
```
mu1 <- 5   ; sigma1 <- 1; n1 <- 100
mu2 <- 10  ; sigma2 <- 1; n2 <- 200
mu3 <- 15  ; sigma3 <- 2; n3 <- 50
mu4 <- 20  ; sigma4 <- 3; n4 <- 100
```

```
cl <- rep(1:4,c(n1,n2,n3,n4))
x <- c(rnorm(n1,mu1,sigma1),rnorm(n2,mu2,sigma2),
       rnorm(n3,mu3,sigma3),rnorm(n4,mu4,sigma4))
n <- length(x)

## we randomize the class ordering
rnd <- sample(1:n)
cl <- cl[rnd]
x  <- x[rnd]

alpha <- c(n1,n2,n3,n4)/n

curve(alpha[1]*dnorm(x,mu1,sigma1) +
      alpha[2]*dnorm(x,mu2,sigma2) +
      alpha[3]*dnorm(x,mu3,sigma3) +
      alpha[4]*dnorm(x,mu4,sigma3),
      col="blue", lty=1, from=0,to=30, n=1000,
      main="Theoretical Gaussian mixture and its components",
      xlab="x", ylab="density")
curve(alpha[1]*dnorm(x,mu1,sigma1), col="red", add=TRUE, lty=2)
curve(alpha[2]*dnorm(x,mu2,sigma2), col="red", add=TRUE, lty=2)
curve(alpha[3]*dnorm(x,mu3,sigma3), col="red", add=TRUE, lty=2)
curve(alpha[4]*dnorm(x,mu4,sigma4), col="red", add=TRUE, lty=2)
rug(x)
```

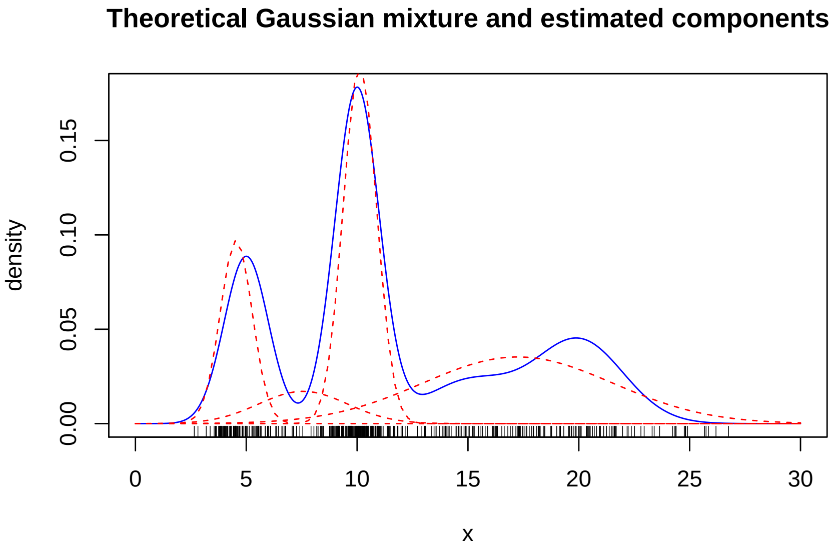


Suppose that we know the number of components, i.e. 4.

```
out <- EM_mixture(x, Q = 4)
plot(out$loglik, main = "data log-likelihood", type="l")
```



```
curve(alpha[1]*dnorm(x,mu1,sigma1) +
      alpha[2]*dnorm(x,mu2,sigma2) +
      alpha[3]*dnorm(x,mu3,sigma3) +
      alpha[4]*dnorm(x,mu4,sigma3), col="blue",
      lty=1, from=0,to=30, n=1000,
      main="Theoretical Gaussian mixture and estimated components",
      xlab="x", ylab="density")
curve(out$alpha[1]*dnorm(x,out$mu[1],out$sigma[1]), col="red", add=TRUE, lty=2)
curve(out$alpha[2]*dnorm(x,out$mu[2],out$sigma[2]), col="red", add=TRUE, lty=2)
curve(out$alpha[3]*dnorm(x,out$mu[3],out$sigma[3]), col="red", add=TRUE, lty=2)
curve(out$alpha[4]*dnorm(x,out$mu[4],out$sigma[4]), col="red", add=TRUE, lty=2)
rug(x)
```

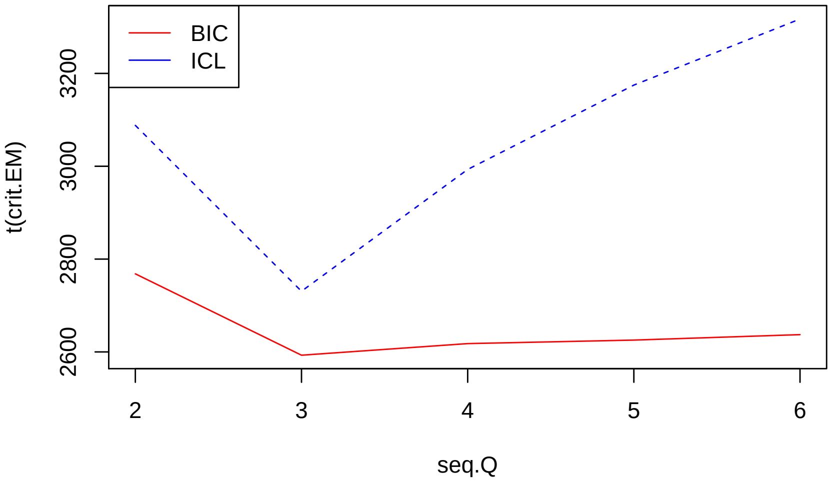


```
## the confusion table gives rather good results
table(out$cl,cl)
```

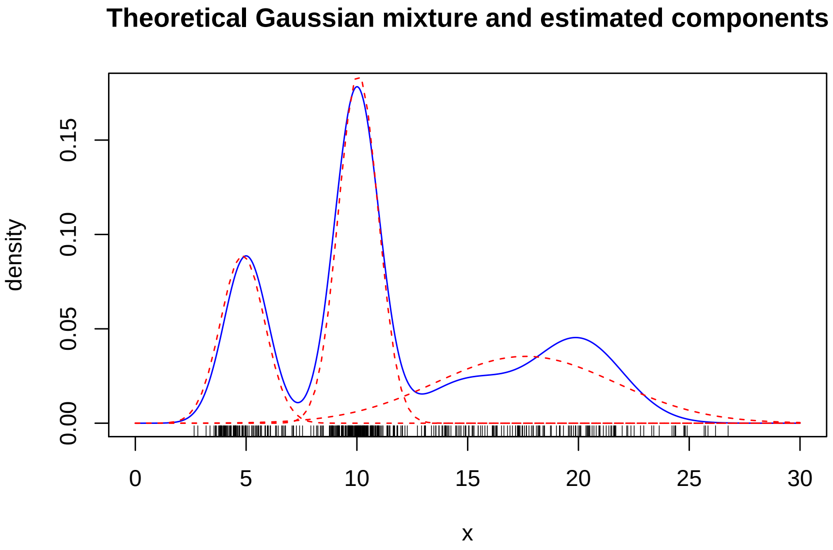
```
##    cl
##       1   2   3   4
##   1   0 179   1   1
##   2  13  10   0   0
##   3  87   0   0   0
##   4   0  11  49  99
```

The number of component mixture is hard to recover because of the last two mixed components

```r
seq.Q <- 2:6
crit.EM <- sapply(seq.Q, function(Q) {
    out <- EM_mixture(x, Q, kmeans(x,Q)$cl)
    df <- Q - 1 + 2 * Q
    return(c(BIC = -2*tail(out$loglik ,1) + log(n)*df,
             ICL = -2*tail(out$Eloglik,1) + log(n)*df ))
})
matplot(seq.Q, t(crit.EM), type="l", col=c("red", "blue"))
legend("topleft", c("BIC", "ICL"), col=c("red", "blue"),lty=1)
```



```r
Q.hat <- seq.Q[which.min(crit.EM[1, ])]
out <- EM_mixture(x, Q = Q.hat, kmeans(x, Q.hat)$cl)
par(mfrow=c(1,1))
curve(alpha[1]*dnorm(x,mu1,sigma1) +
      alpha[2]*dnorm(x,mu2,sigma2) +
      alpha[3]*dnorm(x,mu3,sigma3) +
      alpha[4]*dnorm(x,mu4,sigma3), col="blue",
      lty=1, from=0,to=30, n=1000,
      main="Theoretical Gaussian mixture and estimated components",
      xlab="x", ylab="density")
for (q in 1:Q.hat) {
    curve(out$alpha[q]*dnorm(x,out$mu[q],out$sigma[q]), col="red", add=TRUE, lty=2)
}
rug(x)
```


Theoretical Gaussian mixture and estimated components
density
0.00
0.05
0.10
0.15
0
5
10
15
20
25
30
x