

Analyse de données

Programmation de l'algorithme EM

Yoann Bonnet (@ENSIIE)

Semestre 3, 2022-2023

Simulation

1. Simulation d'un échantillon de $n = 100$ observations d'une loi de Poisson de paramètre $\lambda = 3$.

```
poisson_3 <- rpois(100, 3)
```

2. Simulation d'un échantillon de $n = 200$ observations d'une loi de Poisson de paramètre $\lambda = 15$.

```
poisson_15 <- rpois(200, 15)
```

3. Création d'un vecteur de 300 valeurs entières (100 "1", suivi de 200 "2").

```
X <- matrix(nrow = 1, ncol = 300)
# Les 100 premières valeurs sont des "1"
for (i in 1:100){
  X[1,i] <- 1
}
# Les 200 dernières valeurs sont des "2"
for (i in 101:300){
  X[1,i] <- 2
}
```

4. Simulation d'un mélange de lois de Poisson à deux composantes :

$$P(x) = \pi_1 \frac{e^{-\lambda_1}}{x!} \lambda_1^x + \pi_2 \frac{e^{-\lambda_2}}{x!} \lambda_2^x$$

avec $\lambda_1 = 3$ et $\lambda_2 = 15$, $\pi_1 = 0.4$.

```
lambda_1 = 3
lambda_2 = 15
pi_1 = 0.4
pi_2 = 0.6
pi_k = c(pi_1, pi_2)
X1 <- rpois(300, lambda_1)
X2 <- rpois(300, lambda_2)
Z <- sample(1:2, size = 300, replace = T, prob = pi_k)
X <- c()
for (i in 1:300){
  if (Z[i] == 1){
    X[i] = X1[i]
  }
  if (Z[i] == 2){
```

```

    X[i] = X2[i]
  }
}

```

Algorithme EM pour un mélange de lois de Poisson à K composantes

1. Programmation de l'initialisation de l'algorithme EM.

```

EM.initialisation <- function(x,k){
  pi <- rep(1/k,k)
  lambda=x[sample(length(x),k)]
  parametres=list(pi=pi,lambda=lambda)
  return(parametres)
}

```

2. Programmation de l'étape E

```

EM.etape_E <- function(x,parametres){
  K<-length(parametres$lambda)
  tik <- matrix(0,nrow=length(x),ncol=K)

  for (k in 1:K){
    pik <- parametres$pi[k]
    lambda_k <- parametres$lambda[k]
    tik[,k] <- (pik*dpois(x,lambda=lambda_k))
  }

  tik <- tik/rowSums(tik)
  return(tik)
}

```

3. Programmation de l'étape M

```

EM.etape_M <- function(x,parametres,tik){
  K <- ncol(tik)
  lambda <- rep(0,K)

  for(k in 1:K){
    lambda[k] <- sum(tik[,k]*x)/sum(tik[,k])
  }

  pik <- colSums((tik))/length(x)
  parametres <-list(pi=pik,lambda=lambda)
  return(parametres)
}

```

4. Programmation de l'algorithme au complet

```

EM.algorithme <- function(x,k){
  parametres <- EM.initialisation(x,k)
  tik <- EM.etape_E(x,parametres)
  nvx.params <- EM.etape_M(x,parametres,tik)
  while((sum(abs(unlist(nvx.params)**2)-unlist(parametres)**2))/sum(abs(unlist(parametres)**2))>(1e-6)){
    T <- EM.etape_E(x,nvx.params)
  }
}

```

```

    parametres <- nvx.params
    nvx.params <- EM.etape_M(x,parametres,T)
  }

  return(nvx.params)
}

```

Test de l'algorithme EM sur les données simulées ci-dessus :

```
EM.algorithme(X,2)
```

```

## $pi
## [1] 0.5984719 0.4015281
##
## $lambda
## [1] 15.206467 3.252632

```

Conclusion. Notre implémentation de l'algorithme EM semble correcte : en effet, la somme des deux proportions est bien égale à 1, et les valeurs de λ_1 et λ_2 sont très proches de 3 et 15, qui sont les deux paramètres de notre mélange de lois de Poisson.