

Computational Statistics. Chapter 3: EM algorithm. Solution of exercises

Thierry Denoeux

9/6/2021

Exercise 1

Question 1a

We first give values to the model parameters:

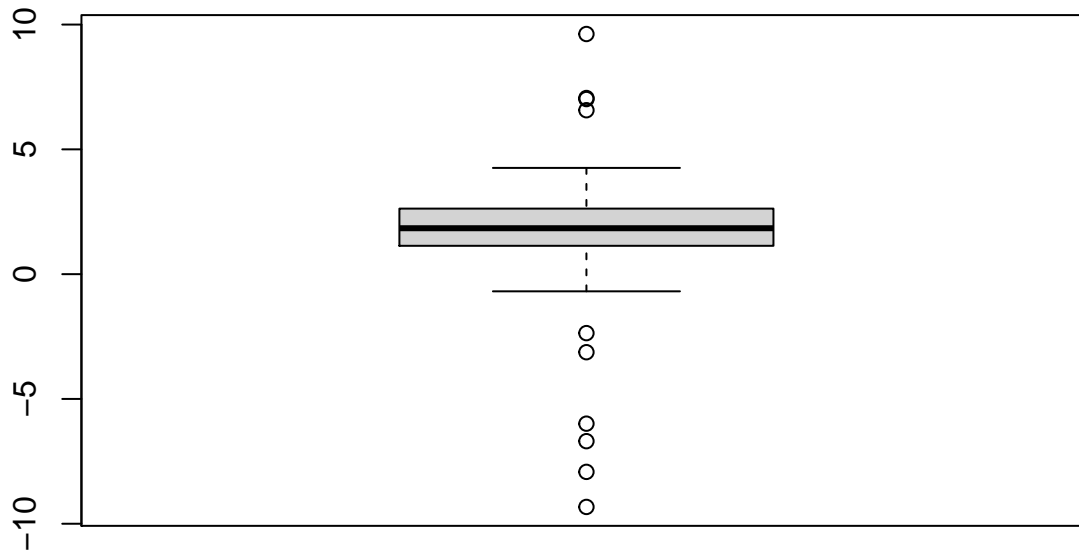
```
pi<-0.90
mu<-2
sig<- 1
a=10;
c<-1/(2*a)
n<- 100
```

We then generate the data:

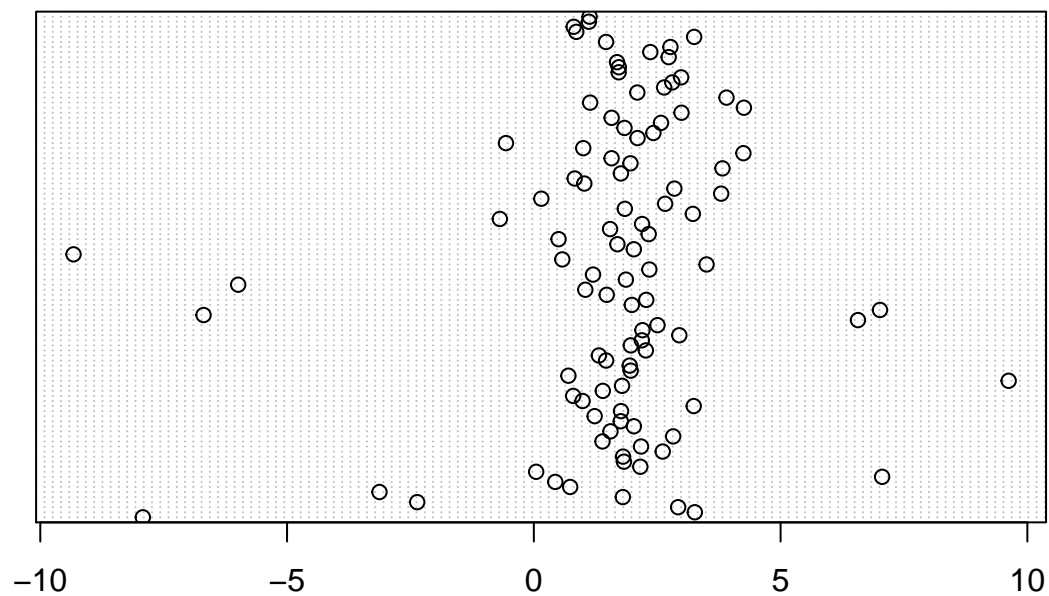
```
y<-vector("numeric",n)
z<-vector("numeric",n)
for(i in 1:n){
  z[i]=sample(c(1,0),size=1,prob=c(pi,1-pi))
  if(z[i]==1){
    y[i]<- rnorm(1,mean=mu,sd=sig)
  }
  else y[i]<-runif(1,min=-a,max=a)
}
```

Finally, we generate box and dot plots the data:

```
boxplot(y)
```



```
dotchart(y)
```



Question 1b

We first write a function that computes the observed-data log-likelihood:

```
loglik<- function(theta,y){
  phi <- sapply(y,dnorm,mean=theta[1],sd=theta[2])
  logL <- sum(log(theta[3]*phi+(1-theta[3])*c))
  return(logL)
}
```

We then write the EM algorithm for this problem. The inputs are the data y , the initial parameter value θ_0 , the constant a and the threshold ϵ used in the stopping criterion. The outputs are the maximum observed-data log-likelihood, the corresponding MLE of θ , and the vector z of estimated probabilities.

```

em_outlier <- function(y,theta0,a,epsi){
  go_on<-TRUE
  logL0<- loglik(theta0,y)
  t<-0
  c<-1/(2*a)
  n<-length(y)
  print(c(t,logL0))
  while(go_on){
    t<-t+1
    # E-step
    phi <- sapply(y,dnorm,mean=theta0[1],sd=theta0[2])
    z<- phi*theta0[3]/(phi*theta0[3]+c*(1-theta0[3]))
    # M-step
    S<- sum(z)
    pi<-S/n
    mu<- sum(y*z)/S
    sig<-sqrt(sum(z*(y-mu)^2)/S)
    theta<-c(mu,sig,pi)
    logL<-loglik(theta,y)
    if (logL-logL0 < epsi) go_on <- FALSE
    logL0 <- logL
    theta0<-theta
    print(c(t,logL))
  }
  return(list(loglik=logL,theta=theta,z=z))
}

```

Question 1c

Let us now run the above function with our data. We initialize parameters μ and σ with the mean and standard deviation of the data, and we set $\pi_0 = 0.5$:

```

mu0<-mean(y) # +rnorm(1,mean=0,sd=0.5)
sig0<-sd(y)
pi0<-0.5
theta0<-c(mu0,sig0,pi0)

```

We then run function em_outlier:

```

estim<-em_outlier(y,theta0,a,epsi=1e-6)

```

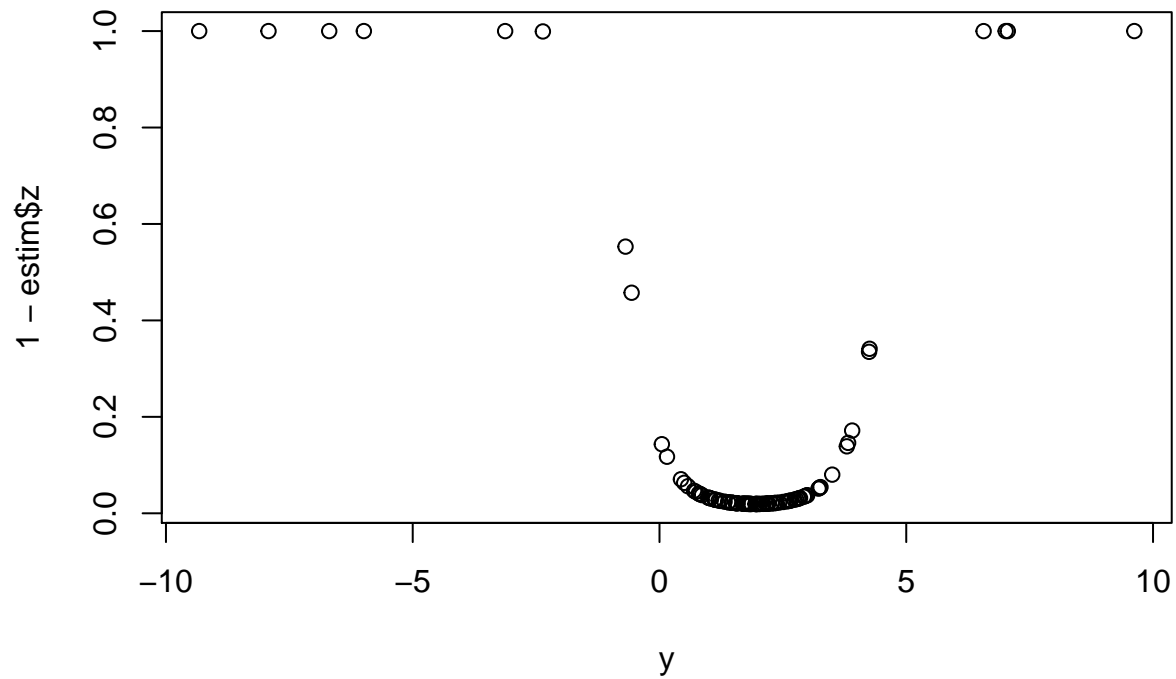
```

## [1] 0.0000 -243.1278
## [1] 1.0000 -194.9653
## [1] 2.0000 -184.2405
## [1] 3.0000 -183.827
## [1] 4.0000 -183.8055
## [1] 5.0000 -183.802
## [1] 6.0000 -183.8013
## [1] 7.0000 -183.8011
## [1] 8.0000 -183.8011
## [1] 9.0000 -183.8011
## [1] 10.0000 -183.8011
## [1] 11.0000 -183.8011

```

Finally, we plot the estimates probabilities $1 - z_i$ against the inputs y_i :

```
plot(y, 1-estim$z)
```



We can see that the outliers have a high estimated probability of being drawn from the uniform distribution, as expected.