Midterm Exam 2019

3/13/2019

Your Name:

Xinyi Lin

Your UNI:

xl2836

Disease D is a chronic neurological condition that leads to fast deterioration of motor and cognitive functions and eventually leads to death. Based on a theoretical model, the survival time of a patient suffered from disease D very much depends on his or her disease onset age. To be specific, let Y as the survival time of a patient, and X be the disease onset age, the conditional distribution for Y given $X = x$ is exponential with failure rate $0.01x$.

$$f(Y = t \mid X = x) = 0.01x \exp\{-0.01xt\}.$$

Since D is a chronic condition, its actual onset times are often unobserved. Suppose the disease onset ages in a population also follows an exponential with failure rate θ, where $\theta > 0$ is an unknown parameter. Suppose $\{Y_i,\, i = 1, ..., n\}$ are observed survival times of n patients with disease D in a population. The public health researchers are interested in estimating the parameter θ in the population so that they could design disease prevention policies on target ages.

1. Write out the marginal distribution of Y, and the observed likelihood function of $\{Y_i,\, i = 1, ..., n\}$.

Answer:

This is the answer of problem 1.

2. Design a univariate optimization algorithm (e.g. Golden search or Newton's method) to find the MLE of the observed likelihood in (1), and specify each step of your algorithm. Implement the algorithm into an R function.

Answer:

R codes:

```
loglike = function(theta){
  res = sum(log(0.01*theta)-2*log(0.01*Y+theta))
  return(res)
}
```

```
golden_max = function(func, a, b){
  w = 0.618
  theta0 = a+(b-a)*(1-w)
  theta1 = theta0 + (b-a)*(1-w)*w
  tol = 1e-4
  i = 0

  #rlist = c("a", "b", "theta0", "theta1")
  rlist = c(i, a, b, theta0, theta1)
  while(abs(b-a)>tol){
    i = i+1
    if(func(theta1) > func(theta0)){
      a=theta0;
      theta0 = theta1
      theta1 = theta0 + (b-a)*(1-w)*w
      }
    else{
      b=theta1;
      theta0 = a+(b-a)*(1-w)
      theta1 = theta0 + (b-a)*(1-w)*w
      }
    rlist = rbind(rlist, c(i, a, b, theta0, theta1))
  }
  #tail(rlist)
  return(tail(rlist))
}

#golden_max(fx, 0, 1.5)
```

3. Write out the joint distribution of (Y, X), and design an EM algorithm to find the MLE of θ. Clealy write out the E-steps and M-steps in each iteration, and implement the algorithm into an R function.

Answer:

R codes:

```
# E-step evaluating conditional means E(Z_i | X_i , pars)
# pars: parameters list
```

```r
delta <- function(X, theta){
  return(1/(0.01*x))
}
```

```r
# M-step - updating the parameters
mles <- function(Z, X) {
  n <- length(X)
  thetahat <- n/sum(Z)
  return(thetahat)
}
```

```r
EMmix <- function(X, start, nreps=10) {
  i <- 0
  Z <- delta(X, start)
  newpars <- start
  res <- c(0, t(as.matrix(newpars)))
  while(i < nreps) {
  # This should actually check for convergence
    i <- i + 1
    newpars <- mles(Z, X)
    Z <- delta(X, newpars)
    res <- rbind(res, c(i, t(as.matrix(newpars))))
  }
  return(res)
}
```

4.Simulate data sets with true $\theta = 0.025$, and apply the optimization functions you deleveped in (2) and (3) to estimate θ, which algoirhtm is more efficient (comparing the numbeter of iterations and computing times)?

Answer:

R codes:

```r
simy = function(theta){
  n = 20
  true_theta = theta
  X = rexp(n, true_theta)
  Y = vector(mode = "numeric", n)
  for (i in 1:n) {
    Y[i] = rexp(1, X[i])
  }
  return(list(X=X, Y=Y))
}
```

```r
set.seed(123)
res = simy(0.025)
res$X
```

```
##  [1]  33.738290  23.064411  53.162195   1.263094   2.248439  12.660049
##  [7]  12.569092   5.810672 109.049459   1.166138  40.193202  19.208589
## [13]  11.240545  15.084713   7.531362  33.991445  62.528142  19.150417
## [19]  23.637393 161.640468
```

```r
res$Y
```

```
##  [1] 0.024990885 0.041877125 0.027938572 1.067255562 0.519706766
##  [6] 0.126844089 0.119081228 0.270304795 0.000291315 0.512674959
## [11] 0.053935482 0.026374437 0.023091213 0.172153893 0.163187720
## [16] 0.023261199 0.010063950 0.065515076 0.024904804 0.006986431
```

```r
# method 1
Y = res$Y
golden_max(loglike, 0, 1)
```

```
##  [,1]         [,2]         [,3]         [,4]         [,5]
##    15 0.0000000000 0.0007338853 0.0002803442 0.0004535969
##    16 0.0002803442 0.0007338853 0.0004535969 0.0005606671
##    17 0.0004535969 0.0007338853 0.0005606671 0.0006268364
##    18 0.0004535969 0.0006268364 0.0005197744 0.0005606721
##    19 0.0005197744 0.0006268364 0.0005606721 0.0005859469
##    20 0.0005197744 0.0005859469 0.0005450523 0.0005606740
```

5.Show that θ is $0.01\times$the median of Y, and hence (the sample median of $Y_i)\times 0.01$ is a consistent estimation of θ as well.

Answer:

6. Now that you have two estimates of θ, the MLE estimate and the one using the sample median of Y_i's, Carry out a simulation study to compare the estimation efficiency of the two estimates. Based on your simulation results, which estimate should be recommended?

Answer:

Step 1: simulation Y; Step 2: calculate θ given by MLE and median θ; Step 3: repeat step1-2 n times and calculate corresponding MSE; Step 4: compare MSEs.

R codes:

```r
# a function to calculate MLE of theta
mle_theta = function(Y){
  return()
}
```

```r
compare <- function(N=10000) {
  SSEmle <- SSEmedian <- 0
  for(i in 1:N){
    Y = simy(0.025)
    SSEmle <- SSEmle + mle_theta(Y)^2
    SSEmedian <- SSEmedian + median(Y)^2
  }
  return(list(Y=Y, MSEmean = SSEmle / N,
   MSEmedian = SSEmedian / N))
}
```

```r
res <- NULL
for(i in 1:length(pvec))
  res <- rbind(res, as.numeric(compare(N=5000)))
res <- data.frame(res)
names(res) <- c("Y", "MSEmean", "MSEmedian")
print(res)
```