Improvement to nls() : Solution to Test(Hard)

Arkajyoti Bhattacharjee

Theory :

The logistic sigmoid growth curve,here, is given by -

$$y = f(t) = \frac{a}{1 + be^{-ct}} \ldots (\Delta)$$

Here, the growth $y = f(t)$ is a function of time t.
b : the scale parameter of the sigmoid.
a : the curve's maximum value.
c : the logistic growth rate or steepness of the curve.

Now, given the test data, we need to estimate a logistic sigmoid growth curve of the form as in Eq. Δ using R packages - **nlsr** and **minpack.lm**.

R Code :

```
### "Improvements to nls()" - GSOC'21
## Hard

# Install and load necessary packages
install.packages("nlsr")
install.packages("minpack.lm")
library(nlsr)
library(minpack.lm)

# test data

time <- c( 5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29,
 30, 31, 32, 33, 34, 35)
y <- c( 0.0074203,  0.3188325,  0.2815891, -0.3171173, -0.0305409,  0.2266773,
  -0.0216102,  0.2319695, -0.1082007,  0.2246899,  0.6144181,  1.1655192,
  1.8038330,  2.7644418,  4.1104270,  5.0470456,  6.1896092,  6.4128618,
  7.2974793,  7.8965245,  8.4364991,  8.8252770,  8.9836204, 9.6607736,
  9.1746182,  9.5348823, 10.0421165,  9.8477874,  9.2886090,  9.3169916,
  9.6270209 )
```

```r
# Plot y vs. time
plot(time,y,xlab="t",ylab=expression(y[t]),
main="Plot of observed and fitted y vs. Time",
lwd=2,col="red",type="o",xlim=c(5,35))

# clean the data
time <- time[y>0]
y <- y[y>0]

## model based on a list of parameters
getPred <- function(parS, x) {
  return ( parS$a /(1+  parS$b*exp(-x * parS$c)) )
}

## residual function
residFun <- function(p, observed, x) {
  return(observed- getPred(p,x))
}

## starting values for parameters
parStart <- list(a=10.04212,b=1352.3,c=0.33883)

##
model<- y ~ a/(1+b*exp(-c*time))
Data=data.frame(time,y)

# the analytic jacobian function is-
jacobian<-function(x,observed,Pars){
    mat <- matrix(0,nrow=length(x),ncol=length(Pars))
      colnames(mat)<-c("a","b","c")
    mat[,"a"]<- -1/(1+Pars$b*exp(-Pars$c*x))
    mat[,"b"]<- Pars$a*exp(-Pars$c*x)/(1+Pars$b*exp(-Pars$c*x))^2
    mat[,"c"]<- -Pars$a*x*Pars$b*exp(-Pars$c*x)/(1+Pars$b*exp(-Pars$c*x))^2
      return(mat)
}

# the approximate jacobian function is-
jacobian.approx<-function(x,observed,parS){
    delta<- 5 # the smaller the better
    mat <- matrix(0,nrow=length(x),ncol=length(parS))
      colnames(mat)<-c("a","b","c")
```

```r
    mat[,"a"]<-  ((parS$a+delta) /(1+  parS$b*exp(-x * parS$c))-
              parS$a /(1+  parS$b*exp(-x * parS$c)))/delta
    mat[,"b"]<- (parS$a /(1+  (parS$b+delta)*exp(-x * parS$c))-
              parS$a /(1+  parS$b*exp(-x * parS$c)))/delta
    mat[,"c"]<- (parS$a /(1+  parS$b*exp(-x * (parS$c+delta)))-
              parS$a /(1+  parS$b*exp(-x * parS$c)))/delta
      return(mat)
}

# fitting the nonlinear model based on approximate jacobian
nls.out.approx <- nls.lm(par=parStart, fn = residFun, observed = y,
x = time, control = nls.lm.control(nprint=1),jac=jacobian.approx)
a.est.approx=nls.out.approx$par[["a"]]
b.est.approx=nls.out.approx$par[["b"]]
c.est.approx=nls.out.approx$par[["c"]]
cat("The estimate of 'a' using approximation is : ",a.est.approx,"\n")
cat("The estimate of 'b' using approximation is : ",b.est.approx,"\n")
cat("The estimate of 'c' using approximation is : ",c.est.approx,"\n")
lines(time,a.est.approx/(1 + b.est.approx * exp(-c.est.approx * time)),
col="blue",pch=16,lwd=2)

# fitting the nonlinear model based on Jacobian
nls.out <- nls.lm(par=parStart, fn = residFun, observed = y,
x = time, control = nls.lm.control(nprint=1),jac=jacobian)
a.est=nls.out$par[["a"]]
b.est=nls.out$par[["b"]]
c.est=nls.out$par[["c"]]
cat("The estimate of 'a' using analytic Jacobian is : ",a.est,"\n")
cat("The estimate of 'b' using analytic Jacobian is : ",b.est,"\n")
cat("The estimate of 'c' using analytic Jacobian is : ",c.est,"\n")
lines(time,a.est/(1 + b.est * exp(-c.est * time)),col="seagreen",pch=16,lwd=2)
legend("topleft",legend=c("observed y","fitted y(approx)","fitted y(Jacobian)"),
pch=c(16,NA,NA),lwd=2,col=c("red","blue","seagreen"),cex=0.8)
```

Output :

```
## Warning: package 'nlsr' was built under R version 4.0.5

## Warning: package 'minpack.lm' was built under R version 4.0.5

## It.    0, RSS =    8.60502, Par. =    10.0421     1352.3    0.33883
## It.    1, RSS =    8.60515, Par. =    10.0421     1352.3    0.33883

## The estimate of 'a' using approximation is :  10.04212

## The estimate of 'b' using approximation is :  1352.3
```

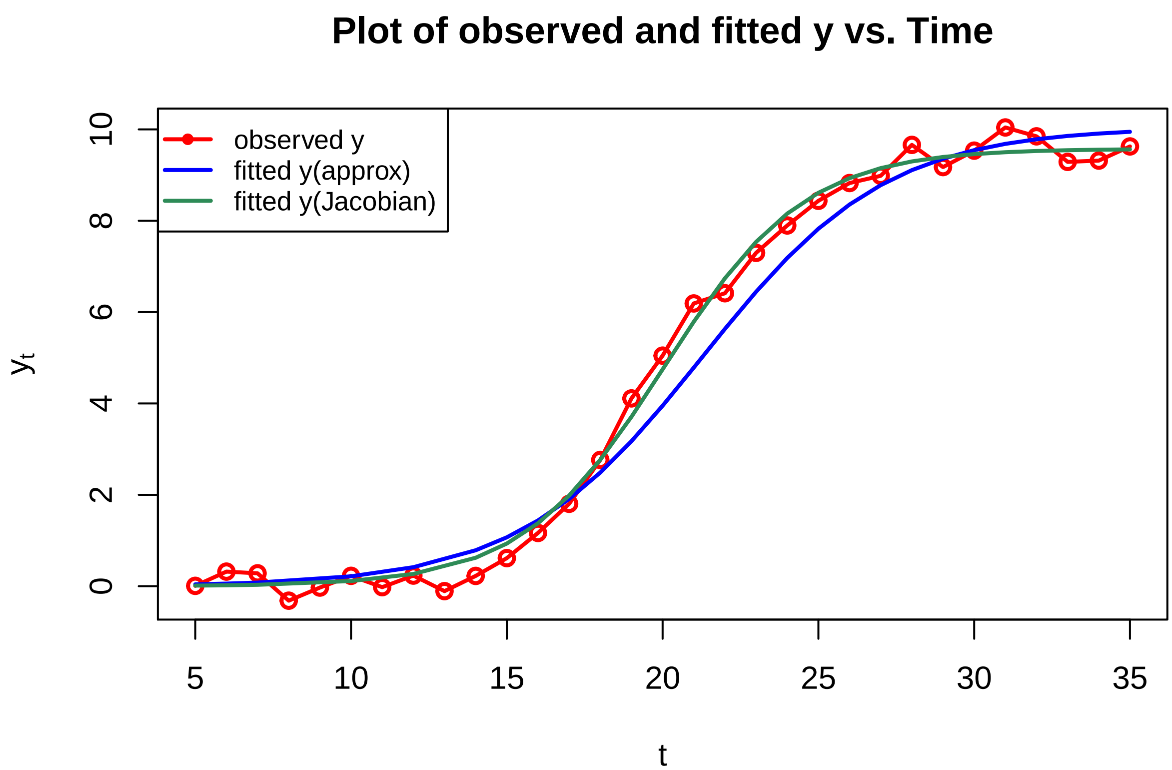
```
## The estimate of 'c' using approximation is :  0.33883

## It.    0, RSS =    8.60502, Par. =    10.0421     1352.3    0.33883
## It.    1, RSS =    5.15516, Par. =    9.45938    2788.39   0.410808
## It.    2, RSS =    4.16758, Par. =     9.5727    5271.29   0.440427
## It.    3, RSS =     2.0015, Par. =    9.55814    7007.12   0.444442
## It.    4, RSS =    1.92912, Par. =    9.57046    7167.74   0.443121
## It.    5, RSS =    1.92877, Par. =    9.57491    6972.44   0.441669
## It.    6, RSS =    1.92875, Par. =    9.57521    6976.21   0.441661
## It.    7, RSS =    1.92875, Par. =    9.57527    6973.06   0.441637
## It.    8, RSS =    1.92875, Par. =    9.57527    6973.09   0.441637

## The estimate of 'a' using analytic Jacobian is :  9.575273

## The estimate of 'b' using analytic Jacobian is :  6973.085

## The estimate of 'c' using analytic Jacobian is :  0.4416373
```



Explanation :

I first plot the graph to know how the given $y's$ look as a function of time. Since, the logistic function is strictly positive, I first discard the negative y values from the data and the corresponding time points.
Here, two methods have been used for estimation of the parameters - analytic Jacobian and a numerical

approximation. For the given logistic growth function, we have the following -

$$\partial f/\partial a = -\frac{1}{1+be^{-ct}}$$

$$\partial f/\partial b = \frac{ae^{-ct}}{(1+be^{-ct})^2}$$

$$\partial f/\partial c = -\frac{tabe^{-ct}}{(1+be^{-ct})^2}$$

Hence, the Jacobian is given by

$$J = [\partial f/\partial a \;\; \partial f/\partial b \;\; \partial f/\partial c]_{n\times 3}$$

where $n =$ number of observations. Again, the approximate Jacobian is given by -

$$J_{approx} = [\tilde{\partial} f/\partial a \;\; \tilde{\partial} f/\partial b \;\; \tilde{\partial} f/\partial c]_{n\times 3}$$

where,

$$\tilde{\partial} f/\partial a = \frac{f(t;a+\delta,b,c)-f(t;a,b,c)}{\delta}$$

$$\tilde{\partial} f/\partial b = \frac{f(t;a,b+\delta,c)-f(t;a,b,c)}{\delta}$$

$$\tilde{\partial} f/\partial c = \frac{f(t;a,b,c+\delta)-f(t;a,b,c)}{\delta}$$

Using the above procedures, my estimated logistic growth functiona, using Jacobians in the above order, are -

$$\hat{y}_{=}\frac{9.575273}{1+6973.085e^{-0.4416373t}}$$

and

$$\hat{y}_{approx} = \frac{10.04212}{1+1352.3e^{-0.33883t}}$$