2-D newton-raphson

Sina Sanei

November 3, 2018

2-D newton-raphson

```
log_lik= function (theta){ ##theta=(mu,gamma)
  n = length(data1)
  ll=(-sum(data1)+ n*theta[1])/theta[2] - 2* sum(log(theta[2]+exp(-data1-theta[1])))
  return(ll)
}
```

since taking derivatives is hard to do for log-likelihood function , I will use the aproximations for first and second derivatives :

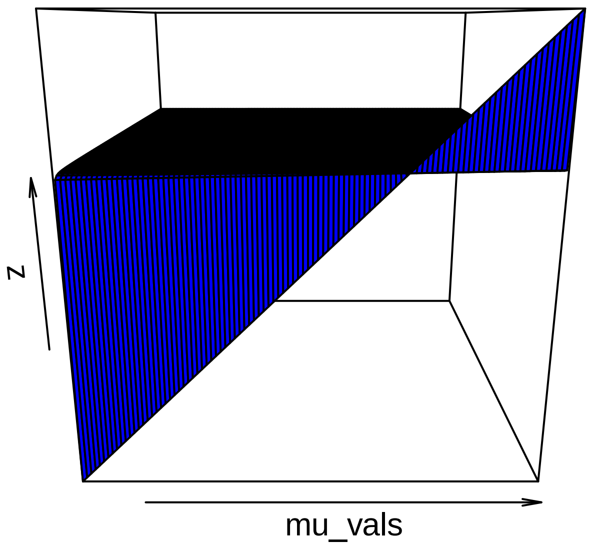
```
grad_ll =function(theta, eps, eps1){ #aprox graiant of loglikelihood
  (log_lik(theta +eps)-log_lik(theta))/eps1
}
grad2_ll = function(theta , epss , epss1){#aprox second derivatives of loglikelihood
  (grad_ll(theta+epss ,epss,epss1)-grad_ll(theta,epss,epss1))/epss1
}
```

I will take the start value of μ the median of X , as before, and by a grid search for γ will choose the value that has the largest likelihood:

```
gama=seq(0,5,length.out = 101)[-1]
mu_1=rep(median(data1),100)
lll=rep(NA,100)
for (i in 1:100){
  lll[i]=log_lik(c(mu_1[i],gama[i]))
}
theta_ini = c(median(data1),gama[which.max(lll)])
theta_ini ##initial value of theta:
```

```
## [1] 38.06589  0.15000
```

```
mu_vals=seq(-10,60, length=100)
gamma_vals=seq(0.001,5, length=100)
z=matrix(NA,100,100)
for(i in 1:100){
  for (j in 1:100) {
  if(is.na(log_lik(c(mu_vals[i],gamma_vals[j])))){z[i,j]=0}
  else {
  z[i,j]=log_lik(c(mu_vals[i],gamma_vals[j])) }
}}
persp(mu_vals,gamma_vals,z, col='blue')
```




```
## contoure and perpective does not look very informative for the function
```

```
#plot(theta_ini[1],theta_ini[2],pch=19, col="red")
#srate= 0.5
mle_nr=function(xvec,Stop_crit,srate){
  n=length(xvec);
  theta_curr=theta_ini;
  nn=0
  theta_seq = theta_curr
  ####compute first derivative of log-likelihood #####
  gradiant =c(grad_ll(theta_curr,c(0.01,0),0.01),grad_ll(theta_curr,c(0,0.01),0.01))
  ### Continue algorithm until the first derivative ###
  ### of the log-likelihood is within stop criterion ##
  #while(!is.infinite(gradiant[1]) | !is.infinite(gradiant[2]) | abs(gradiant[1])>Stop_crit){ #| abs(gr
  for ( n in 1:100){
    ####compute second derivative of log-likelihood #####
    second_gradiant_11= grad2_ll(theta_curr,c(0.01,0),0.01);
    second_gradiant_22= grad2_ll(theta_curr,c(0,0.01),0.01);
    second_gradiant_12= ((log_lik(theta_curr+c(0.01,0.01))-log_lik(theta_curr+c(0,0.01)              ))
    second_gradiant_21= ((log_lik(theta_curr+c(0.01,0.01))-log_lik(theta_curr+c(0.01,0)              ))
    hess = matrix(c(second_gradiant_11,second_gradiant_12,
                    second_gradiant_21,second_gradiant_22), 2,2)
    #### Newton-Raphson's  update of estimate of mu ####
    theta_new=theta_curr+srate*solve(hess)%*%gradiant
    theta_seq = cbind(theta_seq, theta_new);
    theta_curr=theta_new;
    ####compute first derivative of log-likelihood #####
    gradiant =c(grad_ll(theta_curr,c(0.01,0),0.01),grad_ll(theta_curr,c(0,0.01),0.01));
    nn=nn+1
    n=n+1
  }
  a=(list(thetahat=theta_curr, iterations =nn ,sequence = theta_seq))
}
```

```
s=mle_nr(data1,0.001,srate=0.1)
s$sequence
```

```
##      theta_seq
## [1,]  38.06589 38.0915 38.11453 38.13524 38.153860 38.1705899 38.18561894
## [2,]   0.15000  0.1340  0.11960  0.10664  0.094976  0.0844784  0.07503056
##
## [1,] 38.1991143 38.21122575 38.22208778 38.23182093 38.2405331 38.24832101
## [2,]  0.0665275  0.05887475  0.05198728  0.04578855  0.0402097  0.03518873
##
## [1,] 38.25527079 38.26145936 38.26695509 38.27181852 38.27610309
## [2,]  0.03066985  0.02660287  0.02294258  0.01964832  0.01668349
##
## [1,] 38.27985563 38.28311686 38.285921791 38.288300008 38.290275857
## [2,]  0.01401514  0.01161363  0.009452265  0.007507038  0.005756334
##
## [1,] 38.291868484 38.293091658 38.293953165 38.294453217 38.2945791530 NaN
## [2,]  0.004180701  0.002762631  0.001486368  0.000337731 -0.0006960421 NaN
##
## [1,] NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN
## [2,] NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN
##
## [1,] NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN
## [2,] NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN
##
## [1,] NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN
## [2,] NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN
##
## [1,] NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN
## [2,] NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN
##
## [1,] NaN NaN NaN NaN
## [2,] NaN NaN NaN NaN
```