Data 609 Module 3 HW

Bryan Persaud

3/14/2021

Ex.1

Write down Newton's formula for finding the minimum of $f(x) = (3x^4 - 4x^3)/12$ in the range [-10, 10]. Then, implement it in R.

Solution:

First find the first two derivatives of f(x)

$f'(x) = \frac{12x^3 - 12x^2}{12} = x^2(x-1) = x^3 - x^2$

$f''(x) = 3x^2 - 2x$

Then we know Newton's formula to be $x_{k+1} = x_k - \frac{x_k^3 - x_k^2}{3x_k^2 - 2x_k}$

```r
f <- function(x) {
 x - (x^3 - x^2) / (3 * x^2 - 2 * x)
}
f1 <- function(X_0) {
  values <- c()
  for(i in 1:10) {
    if(i == 1){
      values[[i]] <- f(X_0)
    }
    else{
      values[[i]] <- f(values[[i - 1]])
    }
  }
  print(values)
}
f1(2)
```

```
## [[1]]
## [1] 1.5
##
## [[2]]
## [1] 1.2
##
## [[3]]
## [1] 1.05
##
## [[4]]
```

```
## [1] 1.004348
##
## [[5]]
## [1] 1.000037
##
## [[6]]
## [1] 1
##
## [[7]]
## [1] 1
##
## [[8]]
## [1] 1
##
## [[9]]
## [1] 1
##
## [[10]]
## [1] 1
```

```
f1(5)
```

```
## [[1]]
## [1] 3.461538
##
## [[2]]
## [1] 2.445307
##
## [[3]]
## [1] 1.782962
##
## [[4]]
## [1] 1.36611
##
## [[5]]
## [1] 1.127755
##
## [[6]]
## [1] 1.023598
##
## [[7]]
## [1] 1.00104
##
## [[8]]
## [1] 1.000002
##
## [[9]]
## [1] 1
##
## [[10]]
## [1] 1
```

We see that the minimum is 1 when multiple starting values are used.

Ex.2

Explore optimize() in R and try to solve the previous problem.

Solution:

```
f <- function(x) {
  (3*x^4 - 4*x^3) / 12
}
xmin <- optimize(f, interval = c(-10,10), tol = 0.0001)
xmin
```

```
## $minimum
## [1] 0.9999986
##
## $objective
## [1] -0.08333333
```

We see that the minimum is close to 1 with a value for f(x) at -0.083.

Ex.3

Use any optimization algorithm to find the minimum of $f(x, y) = (x - 1)^2 + 100(y - x^2)^2$ in the domain $-10 \leq x, y \leq 10$. Discuss any issues concerning the optimization process.

Solution:

Newton method for a higher dimension will be used to answer the question.

$x_{t+1} = x_t - H^{-1} \bigtriangledown f(x, y)$

$x_0 = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ Starting value

$\bigtriangledown f(x, y) = \begin{bmatrix} (x-1)^2 \\ 100(y - x^2)^2 \end{bmatrix}$

find the Hessian matrix by finding second derivative and plugging in starting values

$f_{xx} = 1200x^2 + 2 - 400y$

$f_{xy} = -400x$

$f_{yy} = 200$

$f_{yx} = -400x$

$H = \bigtriangledown^2 f(x, y) = \begin{bmatrix} 2 & -0 \\ -0 & 200 \end{bmatrix}$

$H^{-1} = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.005 \end{bmatrix}$

With t = 0 $x_1 = x_0 - H^{-1} \bigtriangledown f(x, y)$

$x_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0.5 & 0 \\ 0 & 0.005 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

$x_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0.5 \\ 0 \end{bmatrix}$

$x_1 = \begin{bmatrix} -0.5 \\ 0 \end{bmatrix}$

From here f(x,y) will converge into a single point as you go through more values of t. The issues I faced was optimizing for a multivariate function. I ended up choosing this optimization algorithm as it was the only one I knew how to use.

Ex.4

Explore the optimr package for R and try to solve the previous problem.

Solution:

```
library(optimr)
```

```
## Warning: package 'optimr' was built under R version 4.0.4
```

```
fn <- function(x, y){
  (x - 1)^2 + 100 * (y - x^2)^2
}
optimr(c(-10,10), fn, method = "L-BFGS-B")
```

```
## Error in fn(par, ...) : argument "y" is missing, with no default
```

```
## $convergence
## [1] 9999
##
## $par
## [1] NA NA
##
## $counts
## [1] NA NA
##
## $message
## [1] "optim method failure\n"
```