hw_5

Kun Zhou

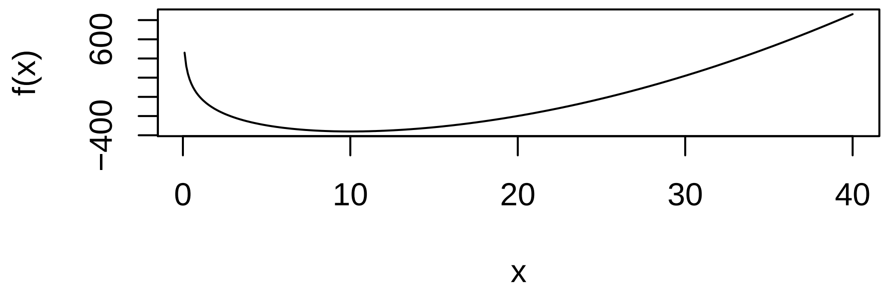
February 29, 2016

1.

$$
\begin{aligned}
&f(x) = x^n - na\log x \\
&f'(x) = nx^{n-1} - \frac{na}{x} \\
&f''(x) = n(n-1)x^{n-2} - \frac{na}{x^2} \\
&n \geq 1 \Rightarrow n(n-1) \geq 0 \Rightarrow f''(x) \geq 0 \Rightarrow f(x) \text{ is convex} \\
&f'(x) = 0 \Rightarrow x_{\min} = a^{\frac{1}{n}}
\end{aligned}
$$

```
newton <- function(par, n, a)
{
  tra = NULL
  last = par
  count = 0
  while(abs(n * last^(n-1) - n*a/last) > 1e-6)  #repeat the process until f' is almost 0
  {
    now = last - (n * last^(n-1) - n*a/last) / (n*(n-1)*last^(n-2) + n*a/(last^2))
    count = count + 1
    last = now
    tra = c(tra, last)
  }
  return(list(result = last, step = count, trace = tra))
}
# f(x)
fun <- function(x, n, a)
{
  return((x^n - n * a * log(x)))
}

# initial position = 1, n =2, a = 100
x = seq(0.1,40, 0.1)
plot(x, fun(x, 2, 100), type="l", ylab="f(x)")
```




```r
newton(1, 2, 100)
```

```
## $result
## [1] 10
##
## $step
## [1] 7
##
## $trace
## [1]  1.980198  3.810961  6.655339  9.224722  9.967527  9.999947 10.000000
```

I set $a = 100$. It takes 10 7 steps to reach the optimum point. The $\{x_k\}$ is also shown above.

2.

$v_1 = u_1$

$v_2 = u_2 - \frac{u_2^t A v_1}{v_1^t A v_1} v_1 \Rightarrow v_2^t A v_1 = (u_2 - \frac{u_2^t A v_1}{v_1^t A v_1} v_1)^t A v_1 = u_2^t A v_1 - \frac{u_2^t A v_1}{v_1^t A v_1} v_1^t A v_1 = 0 \Rightarrow v_1, v_2$are conjugate.

Assume v_k is conjugate with $v_1, \ldots, v_{k-1}$.

$v_{k+1} = u_{k+1}^t - \sum_{j=1}^{k} \frac{u_{k+1}^t A v_j}{v_j^t A v_j} v_j \Rightarrow$ For $i = 1, \ldots, k$,

$v_{k+1}^t A v_i = (u_{k+1}^t - \sum_{j=1}^{k} \frac{u_{k+1}^t A v_j}{v_j^t A v_j} v_j)^t A v_i$

$= u_{k+1}^t A v_i - \sum_{j=1}^{k} \frac{u_{k+1}^t A v_j}{v_j^t A v_j} v_j^t A v_i$

$= u_{k+1}^t A v_i - \frac{u_{k+1}^t A v_i}{v_i^t A v_i} v_i^t A v_i \qquad$ since $v_j^t A v_i = 0$ if $i \neq j$

$= 0$

Thus $\{v_1, \ldots, v_n\}$ are conjugate.

If v_n is the combination of $v_1, \ldots, v_{n-1}, \exists i, v_n^t A v_i = (a_1 v_1 + \ldots a_{n-1} v_{n-1})^t A v_i \neq 0$ which is not true.

Thus $\{v_1, \ldots, v_n\}$ provide a basis.

3.

```r
#conjugate gradient
CG<-function(ini.x, A, b)
{
  x = matrix(ini.x)
  v = A %*% x + b
  step = 0
  for(i in 1:nrow(A))
  {
    t = as.numeric(-t(A %*% x + b) %*% v / (t(v) %*% A %*% v))
    x = x + t * v
    step = step + 1
    print(list(step=step, v=v, x=x))
    alpha = as.numeric(t(A %*% x + b) %*% A %*% v / (t(v) %*% A %*% v))
    v = -(A %*% x + b) + alpha * v
  }
}
#BFGS
BFGS <- function(ini.x, ini.H, A, b)
{
  x = ini.x
  H = ini.H
  step = 0

  for(i in 1:nrow(A))
  {
    v = -solve(H, A %*% x + b)
    t = -as.numeric(t(A %*% x + b) %*% v / (t(v) %*% A %*% v))
    x2 = x + t * v
    s = t * v
    y = A %*% (x2 - x)
    H = H + y %*% t(y) / as.numeric(t(y) %*% s) - H %*% s %*% t(s) %*% H /
      as.numeric(t(s) %*% H %*% s)
    x = x2
    step = step + 1
    print(list(step=step, H=H, v=v, x=x))
  }
}

A = matrix(c(2,1,1,1),nrow=2)
b = matrix(c(1,1))
x=matrix(c(0,0))
H = diag(2)
v = matrix(c(-1, -1))

CG(x, A, b)
```

```
## $step
## [1] 1
##
## $v
##      [,1]
## [1,]    1
```

```
## [2,]    1
##
## $x
##      [,1]
## [1,] -0.4
## [2,] -0.4
##
## $step
## [1] 2
##
## $v
##       [,1]
## [1,]  0.16
## [2,] -0.24
##
## $x
##               [,1]
## [1,]  5.551115e-17
## [2,] -1.000000e+00
```

```r
BFGS(x, H, A, b)
```

```
## $step
## [1] 1
##
## $H
##      [,1] [,2]
## [1,]  2.3  0.7
## [2,]  0.7  1.3
##
## $v
##      [,1]
## [1,]   -1
## [2,]   -1
##
## $x
##      [,1]
## [1,] -0.4
## [2,] -0.4
##
## $step
## [1] 2
##
## $H
##      [,1] [,2]
## [1,]    2    1
## [2,]    1    1
##
## $v
##       [,1]
## [1,]  0.16
## [2,] -0.24
##
## $x
```

```
##                 [,1]
## [1,]  1.665335e-16
## [2,] -1.000000e+00
```

From the aforementioned results, the two consequences of iterates coincinde. The following is the results from *optim*

```r
fun <- function(x, A, b)
{
  x = matrix(x)
  result = 0.5 * t(x) %*% A %*% x + t(x) %*% b
  return(as.numeric(result))
}
optim(par=x, fn = fun, A = A, b = b, method = "CG")
```

```
## $par
##                 [,1]
## [1,]  9.939219e-07
## [2,] -1.000003e+00
##
## $value
## [1] -0.5
##
## $counts
## function gradient
##       55       27
##
## $convergence
## [1] 0
##
## $message
## NULL
```

```r
optim(par=x, fn = fun, A = A, b = b, method = "BFGS")
```

```
## $par
##                 [,1]
## [1,] -3.798632e-07
## [2,] -9.999994e-01
##
## $value
## [1] -0.5
##
## $counts
## function gradient
##       16       10
##
## $convergence
## [1] 0
##
## $message
## NULL
```

The function *optim* gets the same result that $(0, -1)^\top$ is the point minimizes $f(x)$.