Week 2 Homework

Brian Ritz

Saturday, January 17, 2015

This project helps understanding how non-linear optimization of log-likelihood function by the Newton-Raphson method works

1. Create a test function that needs to be optimized

One-dimensional optimization – one variable

This function is really the derivative of the function we are looking for the optimum of. Because my.Optimizer will find the root. The root of the derivative is a candidate for the optimum of a derivatives parent function.

```
my.Function <- function(my.X){
#   ((my.X+4)^3) - 600
  my.X^2*3-my.X*5-5
}
```

2. Write the optimizer

```
my.Optimizer<- function(Start.Value, Function.To.Optimize, Epsilon, max.iterations=1000){

  x.1<-Start.Value
  i<-1
  repeat{

    slope.estimate <- (Function.To.Optimize(x.1+Epsilon) - Function.To.Optimize(x.1-Epsilon)) / (2*Epsilon)
```

```
    x.2 <- x.1 - (Function.To.Optimize(x.1) / slope.estimate)

    if(abs(x.2-x.1)<Epsilon){
      return(x.2)
    }else{
      x.1<-x.2
    }

    if( i == max.iterations){
      return(paste("Max Iterations reached. An optimum was not found. The algorithms last value was", as.ch
aracter(x.2)))
    }else{
      i <- i+1
    }
  }
}
```

3. Test the optimizer

The optimizer should have found a root of the parent function, let's test it out…

```
my.root<-my.Optimizer(0,my.Function, 0.00001)
paste("The optimum we found was:", as.character(my.root), sep=" ")
```

```
## [1] "The optimum we found was: -0.703257409548815"
```

```
# does my.Function(my.optimum) == 0?
```

```
all.equal(my.Function(my.root), 0)
```

```
## [1] TRUE
```

My function successfully found a root of my function.

We will try it out against the uniroot and optim, functions.

```
uniroot.root<-uniroot(my.Function, c(-1, 1))

paste("Did the uniroot function find the same root as my function? ", as.character(all.equal(uniroot.root$root, my.root)))
```

```
## [1] "Did the uniroot function find the same root as my function?  Mean relative difference: 7.877214e-06
"
```

Very Close!

```
optim.optimum<-optim(0, my.Function, method="L-BFGS-B", lower=-100, upper=100)

# the optimum and the uniroot do not equal eachother:
all.equal(optim.optimum$par, uniroot.root$root)
```

```
## [1] "Mean relative difference: 1.843902"
```

This difference can be explained because the optim finds the optimum of the function, while uniroot finds the root of the

function. To reconcile both, we should find the root of the derivative of my.function:

```
my.Derivative<-function(my.X){
  6*my.X - 5
}

derivative.root<-uniroot(my.Derivative, c(-100, 100))

all.equal(derivative.root$root, optim.optimum$par)
```

```
## [1] TRUE
```

We see that the root of the derivative is equal to the optimum found from the optimum function.