Find the Maximizing or Minimizing Point Given Some Objective Function

Fan Wang

2022-07-24

Contents

1   Find Maximum By Iterating Over Grids

Go to the **RMD**, **R**, **PDF**, or **HTML** version of this file. Go back to fan's REconTools research support package, R4Econ examples page, PkgTestR packaging guide, or Stat4Econ course page.

1.1   Single Parameter Optimization

We have a function $f(\mu)$, we know that $a \leq \mu \leq b$, and we want to find the value of μ that maximizes $f(\mu)$ within the bounds. The same idea here is used in various aspects of solving the dynamic equilibrium borrowing and savings problem in Wang (2022) (preprint pdf).

First, we create a simple quadratic function. the minimum of the function is where $\mu = -2$

```r
# Define Function
ffi_quad_func <- function(fl_mu) {
  1 + (fl_mu + 2)^2
}
# Test Function
print(paste0("ffi_quad_func(-3)=", ffi_quad_func(-3)))
```

```
## [1] "ffi_quad_func(-3)=2"
```

```r
print(paste0("ffi_quad_func(-2)=", ffi_quad_func(-2)))
```

```
## [1] "ffi_quad_func(-2)=1"
```

```r
print(paste0("ffi_quad_func(-1)=", ffi_quad_func(-1)))
```

```
## [1] "ffi_quad_func(-1)=2"
```

Second, we develop the maximizer function given grid.

```r
# Function
ffi_find_min <- function(fl_min = -4, fl_max = 2, it_grid_len = 7) {

  # Construct grid where to evaluate the function
  ar_fl_mu <- seq(fl_min, fl_max, length.out = it_grid_len)

  # Evaluate likelihood
```

```r
  ar_obj <- sapply(ar_fl_mu, ffi_quad_func)

  # Find min grid
  it_min_idx <- which.min(ar_obj)
  fl_min_val <- ar_obj[it_min_idx]

  # Find lower and upper bound
  fl_min_new <- ar_fl_mu[max(it_min_idx - 1, 1)]
  fl_max_new <- ar_fl_mu[min(it_min_idx + 1, it_grid_len)]

  # return
  return(list(
    fl_min_val = fl_min_val,
    fl_min_new = fl_min_new,
    fl_max_new = fl_max_new
  ))
}
# Test Function
print("ffi_find_min(-3,-1,10)")
```

```
## [1] "ffi_find_min(-3,-1,10)"
```

```r
print(ffi_find_min(-3, -1, 10))
```

```
## $fl_min_val
## [1] 1.012346
##
## $fl_min_new
## [1] -2.333333
##
## $fl_max_new
## [1] -1.888889
```

```r
# Test function if lower bound is actual min
print("ffi_find_min(-2,-1,10)")
```

```
## [1] "ffi_find_min(-2,-1,10)"
```

```r
print(ffi_find_min(-2, -1, 10))
```

```
## $fl_min_val
## [1] 1
##
## $fl_min_new
## [1] -2
##
## $fl_max_new
## [1] -1.888889
```

```r
# Test function if upper bound is actual min
print("ffi_find_min(-3,-2,10)")
```

```
## [1] "ffi_find_min(-3,-2,10)"
```

```r
print(ffi_find_min(-3, -2, 10))
```

```
## $fl_min_val
```

```
## [1] 1
##
## $fl_min_new
## [1] -2.111111
##
## $fl_max_new
## [1] -2
```

Third, we iterately zoom-in to ever finer grid around the point in the last grid where the objective function had the lowest value.

```r
# Initialize min and max and tolerance criteria
fl_min_cur <- -10
fl_max_cur <- 10
it_grid_len <- 10
fl_tol <- 1e-5
it_max_iter <- 5

# Initialize initial gaps etc
fl_gap <- 1e5
fl_min_val_last <- 1e5
it_iter <- 0

# Iteratively loop over grid to find the maximum by zooming in
while ((fl_gap > fl_tol) && it_iter <= it_max_iter) {

  # Iterator counts up
  it_iter <- it_iter + 1
  print(paste0("it_iter=", it_iter))

  # build array
  ls_find_min <- ffi_find_min(
    fl_min = fl_min_cur, fl_max = fl_max_cur, it_grid_len = it_grid_len
  )

  # Min objective value current
  fl_min_val <- ls_find_min$fl_min_val
  # Find new lower and upper bound
  fl_min_cur <- ls_find_min$fl_min_new
  fl_max_cur <- ls_find_min$fl_max_new
  print(paste0("fl_min_cur=", fl_min_cur))
  print(paste0("fl_max_cur=", fl_max_cur))

  # Compare
  fl_gap <- abs(fl_min_val - fl_min_val_last)
  fl_min_val_last <- fl_min_val
  print(paste0("fl_gap=", fl_gap))
}
```

```
## [1] "it_iter=1"
## [1] "fl_min_cur=-3.33333333333333"
## [1] "fl_max_cur=1.11111111111111"
## [1] "fl_gap=99998.2098765432"
## [1] "it_iter=2"
## [1] "fl_min_cur=-2.34567901234568"
```

```
## [1] "fl_max_cur=-1.35802469135802"
## [1] "fl_gap=0.768175582990399"
## [1] "it_iter=3"
## [1] "fl_min_cur=-2.12620027434842"
## [1] "fl_max_cur=-1.90672153635117"
## [1] "fl_gap=0.0216769123947906"
## [1] "it_iter=4"
## [1] "fl_min_cur=-2.02865416857186"
## [1] "fl_max_cur=-1.97988111568358"
## [1] "fl_gap=0.00025274863560476"
## [1] "it_iter=5"
## [1] "fl_min_cur=-2.00697725617707"
## [1] "fl_max_cur=-1.99613879997968"
## [1] "fl_gap=1.57853178373024e-05"
## [1] "it_iter=6"
## [1] "fl_min_cur=-2.00095589162296"
## [1] "fl_max_cur=-1.99854734580132"
## [1] "fl_gap=2.36575822887275e-06"
```