

# Lab 7

## Rishabh

### Checking the convexity of a function

Given function :

$$f(x, \gamma_1, \gamma_2) = 5(x_1^2 + x_2^2) + (\gamma_1 + \gamma_2)x_1x_2$$

where  $x^T = (x_1, x_2) \in [-10, 10] \times [-10, 10]$  and  $\gamma_1, \gamma_2 \in \mathbb{R}$ . To verify the convexity of  $f$  for

1.  $\gamma_1 = \gamma_2 = 3$
2.  $\gamma_1 = \gamma_2 = 7$
3.  $\gamma_1 = -3, \gamma_2 = 3$

```
# The given function
func_f <- function(x1, x2, gamma1, gamma2){
  return (5*(x1^2 + x2^2) + (gamma1 + gamma2)*x1*x2)
}
```

The above code is the aforementioned function.

### Method 1:

Fixing the points and varying  $\theta$ .

```
# Function to check whether the given function is not convex or not
check_convex <- function(g1, g2){

  # Generating the points x1, x2, y1, y2
  x1 <- 20*runif(1, min = 0, max = 1) - 10
  x2 <- 20*runif(1, min = 0, max = 1) - 10
  y1 <- 20*runif(1, min = 0, max = 1) - 10
  y2 <- 20*runif(1, min = 0, max = 1) - 10
```

```

# Various values of theta in (0,1)
theta <- seq(from = 0, to = 1, by = 0.001)

# Varying over theta to check the convexity condition
for(i in theta){
  if(func_f(x1*i + y1*(1-i), x2*i + y2*(1-i), g1, g2) >
    func_f(x1, x2, g1, g2)*i + func_f(y1, y2, g1, g2)*(1-i)){

    # Returning the corresponding points and theta values
    # ... for which the function is not convex
    val <- c(c(x1, x2), c(y1, y2), i)
    return (c("Not Convex", val))
  }
}
return(TRUE)
}

```

The above function checks if the given function follows the convexity property or not. If not, it returns the corresponding points and value of theta such that this does not happen and the hence the function is not convex.

1.

```

# Function Call
check_convex(3, 3)

```

```
[1] TRUE
```

Here, we can see that the function returns TRUE , which does not give any inference about the convexity of the function for  $\gamma_1 = \gamma_2 = 3$  .

2.

```

# Function Call
check_convex(7, 7)

```

```

[1] "Not Convex"          "5.97628940828145"    "-0.0969720771536231"
[4] "-1.65411869063973"   "4.68565955292434"    "0.001"

```

Here, we can see that the function returns **Not Convex** and the corresponding values of  $x_1, x_2, y_1, y_2, \theta$  for which the given function is not convex for  $\gamma_1 = \gamma_2 = 7$  .

3.

```
# Function Call  
check_convex(-3, 3)
```

```
[1] TRUE
```

Here, we can see that the function returns TRUE , which does not give any inference about the convexity of the function for  $\gamma_1 = -3, \gamma_2 = 3$  .

### Method 2:

Fixing  $\theta$  and varying the points:

```
# Function to count the number of pairs of points  
#... which do not satisfy the convexity condition  
convex_checker <- function(gamma1, gamma2){  
  
  # Generating random points  
  x1 <- 20*runif(1e3, min = 0, max = 1) - 10  
  x2 <- 20*runif(1e3, min = 0, max = 1) - 10  
  y1 <- 20*runif(1e3, min = 0, max = 1) - 10  
  y2 <- 20*runif(1e3, min = 0, max = 1) - 10  
  
  # To store count of pairs  
  count <- 0  
  
  # Some pre-determined values of theta  
  theta <- c(0.001, 0.01, 0.1, 0.5, 0.9, 0.99, 0.999)  
  
  # Checking the condition  
  for (i in 1:1e3){  
    for (j in theta){  
      lhs <- func_f(j*x1[i] + (1-j)*y1[i], j*x2[i] +  
                    (1-j)*y2[i], gamma1, gamma2)  
      rhs <- j*func_f(x1[i], x2[i], gamma1, gamma2) +  
            (1-j)*func_f(y1[i], y2[i], gamma1, gamma2)  
      if (lhs > rhs){  
        count <- count + 1  
      }  
    }  
  }  
}
```

```

}
# Returning the count of such pairs
return(count)
}

```

The above function counts the number of pairs for which the convexity condition is not satisfied. If the count is positive i.e. we get at least one pair of points which do not satisfy the convexity condition, hence the function will not be convex.

1.

```

# Function Call
conc_count <- convex_checker(3,3)
if (conc_count != 0){
  print("NOT CONVEX")
}

```

Here, we see that the count is 0, but we cannot comment about the convexity directly.

2.

```

# Function Call
conc_count <- convex_checker(7,7)
if (conc_count != 0){
  print("NOT CONVEX")
}

```

```
[1] "NOT CONVEX"
```

Here we can see that the count is some positive integer and hence the function is **not convex**.

3.

```

# Function Call
conc_count <- convex_checker(-3,3)
if (conc_count != 0){
  print("NOT CONVEX")
}

```

Here, we see that the count is 0, but we cannot comment about the convexity directly.