

Programación Numérica – FINESI

Universidad Nacional del Altiplano
Facultad de Ingeniería Estadística e Informática

Docente: Fred Torres Cruz

Alumno: Leydy Griselda Aguilar Ccopa

Trabajo Encargado Visualización del Algoritmo de Gradiente Descendente

1. Introducción

El presente informe expone la implementación y representación gráfica del algoritmo de optimización conocido como **Gradiente Descendente**. Este método iterativo es ampliamente empleado en problemas de optimización y aprendizaje automático, ya que permite minimizar funciones de costo de forma progresiva y eficiente.

En este trabajo se desarrollan tres implementaciones utilizando el lenguaje de programación **R**. Dichas implementaciones permiten analizar el comportamiento del algoritmo tanto en una dimensión como en dos dimensiones, empleando las librerías **ggplot2** y **plotly** para generar visualizaciones claras del proceso de convergencia hacia el mínimo.

2. Caso 1: Gradiente Descendente en una Dimensión (1D)

En este primer caso se aplica el algoritmo de gradiente descendente para encontrar el mínimo de la función cuadrática:

$$F(x) = x^2.$$

La visualización se realiza mediante la librería **ggplot2**, mostrando la función original, su derivada (gradiente) y el tamaño del paso utilizado en cada iteración. Esto permite observar cómo el algoritmo se aproxima gradualmente al mínimo global ubicado en $x = 0$.

2.1 Código Fuente

```
1 library(ggplot2)
2 library(tidyr)
3
4 X0 <- 3.0
5 eta <- 0.05
6 n_iter <- 40
7
8 F <- function(x) {
9   return(x^2)
```

```

10 }
11
12 F_prime <- function(x) {
13   return(2 * x)
14 }
15
16 resultados_gradiente <- data.frame(
17   i = 0:n_iter,
18   Xi = NA,
19   F_xi = NA,
20   F_prime_xi = NA,
21   GradF_xi = NA
22 )
23
24 resultados_gradiente[1, "Xi"] <- X0
25 resultados_gradiente[1, "F_xi"] <- F(X0)
26 resultados_gradiente[1, "F_prime_xi"] <- F_prime(X0)
27 resultados_gradiente[1, "GradF_xi"] <- eta * F_prime(X0)
28
29 for (i in 1:n_iter) {
30   X_anterior <- resultados_gradiente[i, "Xi"]
31   gradiente_termino <- eta * F_prime(X_anterior)
32   X_actual <- X_anterior - gradiente_termino
33
34   resultados_gradiente[i + 1, "Xi"] <- X_actual
35   resultados_gradiente[i + 1, "F_xi"] <- F(X_actual)
36   resultados_gradiente[i + 1, "F_prime_xi"] <- F_prime(X_actual)
37   resultados_gradiente[i + 1, "GradF_xi"] <- eta * F_prime(X_actual)
38 }
39
40 datos_largos <- resultados_gradiente %>%
41   pivot_longer(
42     cols = c(F_xi, F_prime_xi, GradF_xi),
43     names_to = "Funcion",
44     values_to = "Valor"
45   )
46
47 grafico_multi_serie <- ggplot(datos_largos,
48   aes(x = Xi, y = Valor, color = Funcion)) +
49   geom_line(linewidth = 1) +
50   geom_point(size = 3) +
51   theme_minimal()
52
53 print(grafico_multi_serie)

```

2.2 Resultado Gráfico

Figura 1: Representación gráfica del proceso de descenso del gradiente aplicado a la función $F(x) = x^2$, donde se observa la convergencia progresiva hacia el mínimo.

3. Caso 2: Gradiente Descendente en Dos Dimensiones (3D) con Proyecciones

En este segundo caso se extiende el algoritmo a dos variables con el objetivo de minimizar la función:

$$F(x, y) = x^2 + y^2.$$

Se utiliza la librería **plotly** para generar una visualización tridimensional interactiva que muestra la superficie de la función y la trayectoria seguida por el algoritmo. Además, se incluyen proyecciones sobre los planos coordenados para analizar la evolución de cada variable de manera independiente.

3.1 Código Fuente

```
1 library(plotly)
2
3 X0 <- 4.0; Y0 <- 3.0
4 eta <- 0.1; n_iter <- 25
5
6 F_xy <- function(x, y) x^2 + y^2
7 Grad_F_xy <- function(x, y) c(2 * x, 2 * y)
8
9 resultados_2d <- data.frame(
10   k = 0:n_iter, x = NA, y = NA, F_xy = NA
11 )
12
13 x_actual <- X0; y_actual <- Y0
14 resultados_2d[1, "x"] <- x_actual
15 resultados_2d[1, "y"] <- y_actual
16
17 for (k in 0:n_iter) {
18   grad <- Grad_F_xy(x_actual, y_actual)
19   resultados_2d[k + 1, "F_xy"] <- F_xy(x_actual, y_actual)
20
21   if (k < n_iter) {
22     x_actual <- x_actual - eta * grad[1]
23     y_actual <- y_actual - eta * grad[2]
24     resultados_2d[k + 2, "x"] <- x_actual
25     resultados_2d[k + 2, "y"] <- y_actual
26   }
27 }
```

3.2 Resultado Gráfico

Figura 2: Visualización tridimensional interactiva del algoritmo de gradiente descendente aplicado a la función $F(x, y) = x^2 + y^2$, incluyendo proyecciones sobre los ejes.

4. Caso 3: Gradiente Descendente en Dos Dimensiones – Superficie Completa

En este último caso se presenta una visualización sobre la superficie completa del paraboloide definido por $F(x, y) = x^2 + y^2$. Esta representación permite observar el recorrido total del algoritmo desde el punto inicial hasta el mínimo global, proporcionando una visión más amplia del proceso de optimización.

4.1 Resultado Gráfico

Figura 3: Representación tridimensional del descenso del gradiente sobre la superficie completa de la función.