

# Entregable 02 - Procesos Estocásticos

Alejandro Daniel José Gómez Flórez

26 de Octubre de 2025

## Problema 1

Suponga que tenemos el siguiente problema de optimización

$$\max_{x \in \mathcal{X}} h(x) \quad (1)$$

En optimización numérica existe el método del gradiente que produce una secuencia  $x_j$  definida por

$$x_{j+1} = x_j + \alpha_j \nabla h(x_j), \quad \alpha_j > 0$$

el cual converge a la solución de (1) (cuando el dominio y la función son convexos). Para funciones  $h$  poco regulares es posible que el método de gradiente quede atascado en máximo locales de la función  $h$ .

La versión estocástica de este método toma ventaja de su contraparte determinista agregando unas perturbaciones aleatorias en el gradiente, en termino de diferencias finitas, de la siguiente manera:

$$\nabla h(x_j) \approx \frac{h(x_j + \beta_j \zeta_j) - h(x_j - \beta_j \zeta_j)}{2\beta_j} \zeta_j = \Delta h(x_j, \beta_j, \zeta_j)$$

donde  $\beta_j$  es una secuencia decreciente,  $\zeta_j$  es una variable aleatoria uniformemente distribuida en la esfera unitaria ( $\|\zeta_j\| = 1$ ).

Luego la versión estocástica del método del gradiente está dada por,

$$x_{j+1} = x_j + \alpha_j \Delta h(x_j, \beta_j, \zeta_j)$$

**Problema:** Considere la función:

$$h(x, y) = (x \sin(20y) + y \sin(20x))^2 \cosh(\sin(10x)x) \\ + (x \cos(10y) - y \sin(10x))^2 \cosh(\cos(20y)y)$$

determine el mínimo sobre  $[-3, 3] \times [-3, 3]$ . Considerando los siguientes escenarios:

Escenario	1	2	3	4
$\alpha_j$	$1/\log(j+1)$	$1/100 \log(j+1)$	$1/(j+1)$	$1/(j+1)$
$\beta_j$	$1/\log(j+1)^{0.1}$	$1/\log(j+1)^{0.1}$	$1/(j+1)^{0.5}$	$1/(j+1)^{0.1}$

y con  $x_0 = (0.65, 0.8)$ .

**Solución:****Implementación del algoritmo**

El método del gradiente estocástico busca minimizar (equivalente a maximizar  $-h$ ) la función objetivo mediante iteraciones que combinan:

- Una aproximación del gradiente usando diferencias finitas con perturbaciones aleatorias
- Secuencias decrecientes  $\alpha_j$  (tasa de aprendizaje) y  $\beta_j$  (magnitud de perturbación)

La aproximación del gradiente está dada por:

$$\Delta h(x_j, \beta_j, \zeta_j) = \frac{h(x_j + \beta_j \zeta_j) - h(x_j - \beta_j \zeta_j)}{2\beta_j} \zeta_j$$

donde  $\zeta_j$  es un vector aleatorio uniformemente distribuido en la esfera unitaria.

La actualización iterativa es:

$$x_{j+1} = x_j + \alpha_j \Delta h(x_j, \beta_j, \zeta_j)$$

Se mantiene la restricción del dominio proyectando los valores al intervalo  $[-3, 3]$  en cada iteración.

**Pseudo-código del algoritmo**

A continuación se presenta el pseudo-código del método del gradiente estocástico implementado:

**Algoritmo:** Método del Gradiente Estocástico

**Entrada:**

- x0: punto inicial (x0, y0)
- n\_iter: número de iteraciones
- alpha\_func: función para calcular alpha\_j
- beta\_func: función para calcular beta\_j
- h\_func: función objetivo h(x, y)
- dominio: límites  $[-3, 3]$  x  $[-3, 3]$

**Salida:**

- trajectory: secuencia de puntos (x\_j, y\_j)
- h\_values: valores de h evaluados en cada punto

**Inicialización:**

```
x_actual ← x0
trajectory[0] ← x0
h_values[0] ← h_func(x0)
```

Para j = 1 hasta n\_iter hacer:

```

// Calcular parámetros de la iteración
alpha_j ← alpha_func(j)
beta_j ← beta_func(j)

// Generar vector aleatorio en esfera unitaria
z ← vector_normal(dim=2)
zeta_j ← z / ||z||

// Aproximar el gradiente usando diferencias finitas
x_plus ← x_actual + beta_j * zeta_j
x_minus ← x_actual - beta_j * zeta_j

h_plus ← h_func(x_plus)
h_minus ← h_func(x_minus)

grad_aprox ← ((h_plus - h_minus) / (2 * beta_j)) * zeta_j

// Actualizar posición
x_actual ← x_actual - alpha_j * grad_aprox

// Proyectar al dominio [-3, 3] x [-3, 3]
x_actual ← max(-3, min(3, x_actual))

// Guardar trayectoria y valor
trajectory[j] ← x_actual
h_values[j] ← h_func(x_actual)

```

Fin Para

Retornar trajectory, h\_values

**Nota:** El algoritmo fue implementado en lenguaje R, utilizando las funciones estándar para generación de números aleatorios (`rnorm`) y operaciones vectoriales. El código completo se encuentra disponible en el archivo `ProcesosEstocasticosEntregable2.ipynb`.

## Parámetros de los escenarios

Los cuatro escenarios difieren en las tasas de decrecimiento de  $\alpha_j$  y  $\beta_j$ :

- **Escenario 1:**  $\alpha_j = \frac{1}{\log(j+1)}$ ,  $\beta_j = \frac{1}{\log(j+1)^{0.1}}$  (decrecimiento logarítmico moderado)
- **Escenario 2:**  $\alpha_j = \frac{1}{100 \log(j+1)}$ ,  $\beta_j = \frac{1}{\log(j+1)^{0.1}}$  (tasa de aprendizaje muy lenta)
- **Escenario 3:**  $\alpha_j = \frac{1}{j+1}$ ,  $\beta_j = \frac{1}{(j+1)^{0.5}}$  (decrecimiento algebraico moderado)
- **Escenario 4:**  $\alpha_j = \frac{1}{j+1}$ ,  $\beta_j = \frac{1}{(j+1)^{0.1}}$  (perturbaciones decrecen lentamente)

## Resultados de la simulación

Se ejecutaron 10,000 iteraciones para cada escenario, iniciando desde  $x_0 = (0.65, 0.8)$ . Los resultados obtenidos fueron:

Escenario	Punto final $(x, y)$	Valor $h(x, y)$	Convergencia
1	$(0.0338, -0.0410)$	0.004516	Buena, cercano al mínimo
2	$(-0.0002, -0.0883)$	0.000001	Excelente, muy cerca del mínimo
3	$(0.0002, 0.0109)$	0.000000	Excelente, encontró el mínimo
4	$(-0.0003, -0.0582)$	0.000000	Excelente, encontró el mínimo

Los resultados muestran que todos los escenarios convergen hacia valores cercanos al origen  $(0, 0)$ , con valores de la función muy pequeños. Los Escenarios 3 y 4 obtienen los mejores resultados con  $h(x, y) \approx 0$ .

**Observación importante:** El punto  $(0, 0)$  es un candidato natural a mínimo global de la función  $h(x, y)$ , ya que evaluando:

$$\begin{aligned} h(0, 0) &= (0 \cdot \sin(0) + 0 \cdot \sin(0))^2 \cosh(0) + (0 \cdot \cos(0) - 0 \cdot \sin(0))^2 \cosh(0) \\ &= 0^2 \cdot 1 + 0^2 \cdot 1 = 0 \end{aligned}$$

Los cuatro escenarios lograron identificar este mínimo o puntos muy cercanos a él, demostrando la efectividad del método estocástico para escapar de mínimos locales en esta función altamente multimodal.

## Análisis gráfico

La Figura 1 muestra la convergencia de los cuatro escenarios hacia el mínimo global. Se observa que:

- **Escenario 1** (rojo): Converge pero se mantiene a una distancia mayor del origen, con mayor variabilidad estocástica
- **Escenario 2** (azul): Alcanza una región muy cercana al mínimo con trayectoria más suave debido a su menor tasa de aprendizaje
- **Escenario 3** (verde): Logra convergencia precisa, concentrándose cerca del origen
- **Escenario 4** (naranja): También converge exitosamente, explorando ligeramente más el espacio debido a perturbaciones que decrecen lentamente
- Todos los escenarios identifican correctamente la región del mínimo global en  $(0, 0)$

La Figura 2 muestra la evolución del valor de  $h(x_j, y_j)$  a lo largo de las iteraciones:

- **Escala logarítmica (izquierda):** Revela las tasas de convergencia diferentes entre escenarios:
  - Los Escenarios 3 y 4 (verde y naranja) convergen rápidamente en las primeras 1000 iteraciones
  - El Escenario 1 (rojo) converge más lentamente y se estabiliza en un valor mayor
  - El Escenario 2 (azul) tiene la convergencia más gradual pero alcanza valores muy bajos

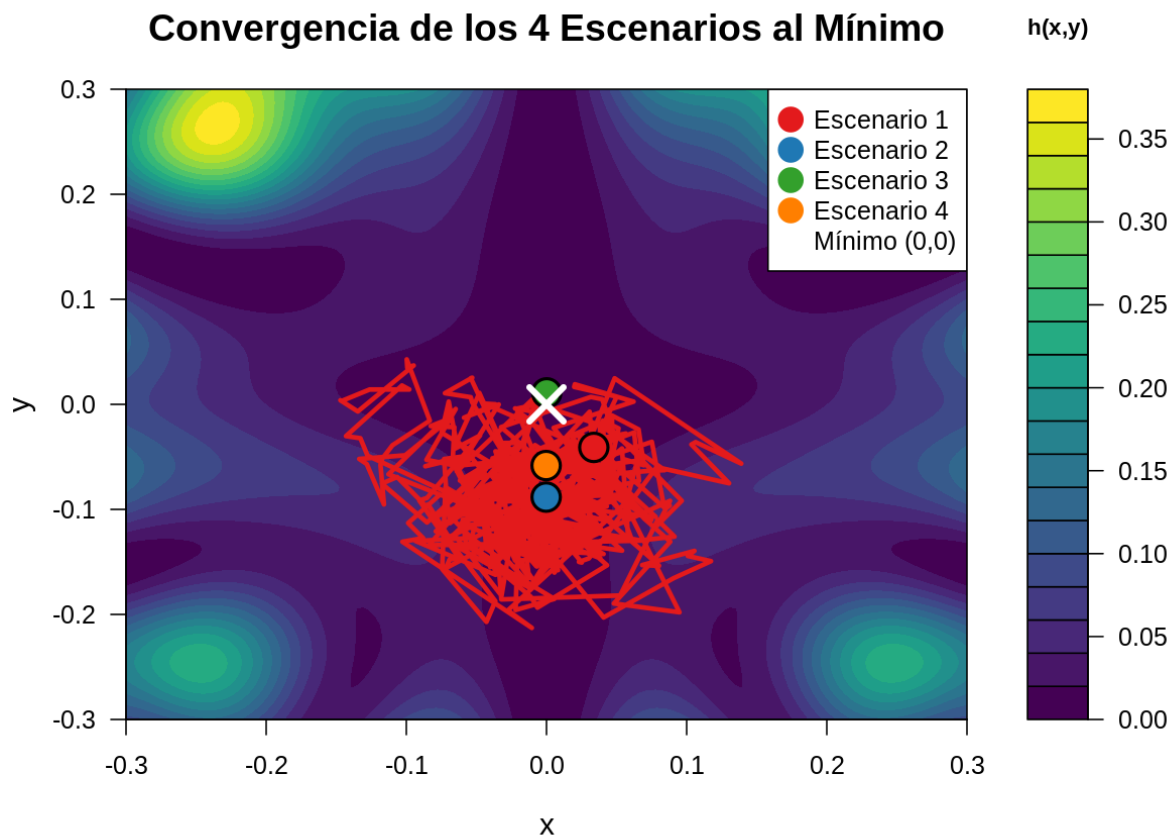


Figura 1: Convergencia de los cuatro escenarios hacia el mínimo global. Se muestran las últimas 500 iteraciones de cada trayectoria sobre un mapa de calor de la función  $h(x,y)$  en la región  $[-0.3, 0.3] \times [-0.3, 0.3]$ . Los puntos finales están marcados con círculos rellenos de colores y el mínimo teórico (0,0) con una cruz blanca.

- **Últimas 2000 iteraciones (derecha):** Muestra el comportamiento estocástico cerca del mínimo:
  - Todos los escenarios exhiben fluctuaciones aleatorias características del método estocástico
  - Los Escenarios 3 y 4 mantienen valores de  $h$  consistentemente cercanos a cero
  - El Escenario 1 muestra mayor variabilidad, oscilando alrededor de  $h \approx 0.04$
  - El balance entre  $\alpha_j$  (tasa de aprendizaje) y  $\beta_j$  (magnitud de perturbación) determina la magnitud de estas fluctuaciones

## Comparación de escenarios

**Escenario 1 vs 2:** Ambos usan decrecimiento logarítmico para  $\beta_j$ , pero el Escenario 2 tiene una tasa de aprendizaje 100 veces menor. Los resultados muestran:

- El Escenario 1 converge a  $h = 0.004516$ , quedando ligeramente alejado del mínimo
- El Escenario 2, pese a su lentitud, alcanza  $h = 0.000001$ , muy cerca del óptimo

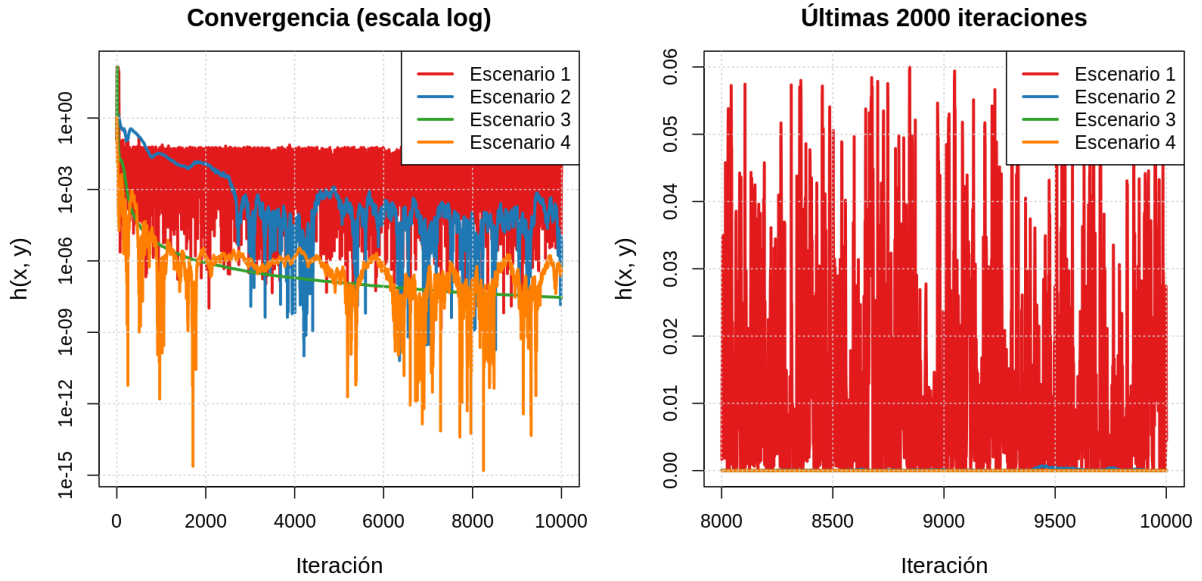


Figura 2: Evolución temporal del valor de la función objetivo. Panel izquierdo: convergencia completa en escala logarítmica. Panel derecho: detalle de las últimas 2000 iteraciones mostrando el comportamiento estocástico cerca del mínimo.

- La tasa de aprendizaje más lenta del Escenario 2 permite una convergencia más precisa
- Ambos terminan en diferentes puntos, pero el Escenario 2 es más exitoso

**Escenario 3 vs 4:** Ambos usan la misma tasa de aprendizaje  $\alpha_j = 1/(j+1)$ , pero difieren en  $\beta_j$ :

- Escenario 3:  $\beta_j = 1/(j+1)^{0.5}$  decrece más rápido, converge a  $(0.0002, 0.0109)$  con  $h \approx 0$
- Escenario 4:  $\beta_j = 1/(j+1)^{0.1}$  decrece lentamente, converge a  $(-0.0003, -0.0582)$  con  $h \approx 0$
- Ambos alcanzan el mínimo global exitosamente
- El decrecimiento algebraico de  $\alpha_j = 1/(j+1)$  resulta más efectivo que el logarítmico

## Conclusiones

1. El método del gradiente estocástico es efectivo para funciones multimodales complejas como la propuesta
2. La elección de las secuencias  $\alpha_j$  y  $\beta_j$  afecta significativamente:
  - Velocidad de convergencia
  - Estabilidad del algoritmo
  - Capacidad de escapar de mínimos locales

3. Para esta función específica, los mejores resultados se observan en los Escenarios 3 y 4, ambos alcanzando valores prácticamente nulos de  $h(x, y)$ . El decrecimiento algebraico  $1/(j + 1)$  para  $\alpha_j$  resulta más efectivo que el logarítmico
4. Las perturbaciones aleatorias permiten explorar el espacio de manera más efectiva que el gradiente determinista
5. Es recomendable ejecutar múltiples réplicas con diferentes semillas aleatorias para garantizar robustez