Descenso de Gradiente en Ciencias Biomédicas

Regresión lineal múltiple y regresión logística

PhD. Pablo Eduardo Caicedo Rodríguez

2025-10-25

Objetivos de la sesión

- Entender el **principio del descenso de gradiente (GD)** como método de optimización.
- Aplicar GD a **regresión lineal múltiple** (objetivo continuo) y **regresión logística** (clasificación 0/1).
- Analizar decisiones de ingeniería: **tasa de aprendizaje**, **escalado/normalización**, **batch vs. mini-batch vs. SGD**.
- Interpretar resultados en **contextos biomédicos** (diagnóstico, pronóstico y evaluación de riesgo).

Agenda

1. Fundamentos de GD y funciones de costo
2. Caso 1: **Regresión lineal múltiple** con GD
3. Caso 2: **Regresión logística** con GD
4. Buenas prácticas, diagnósticos y discusión aplicada

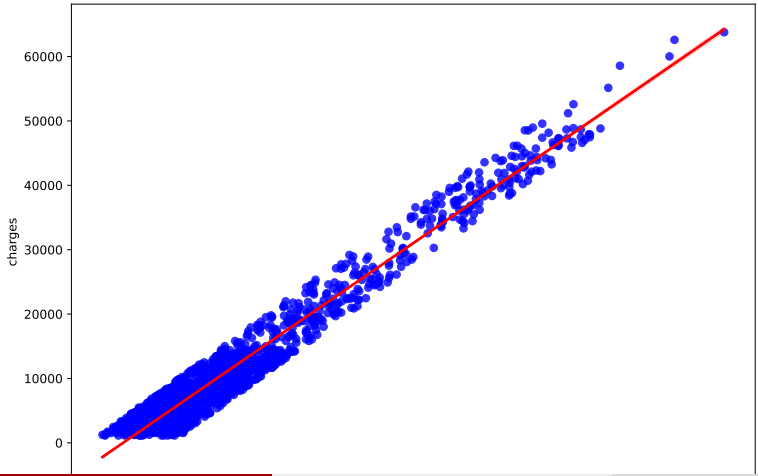
Sección 1

1. Fundamentos de descenso de gradiente

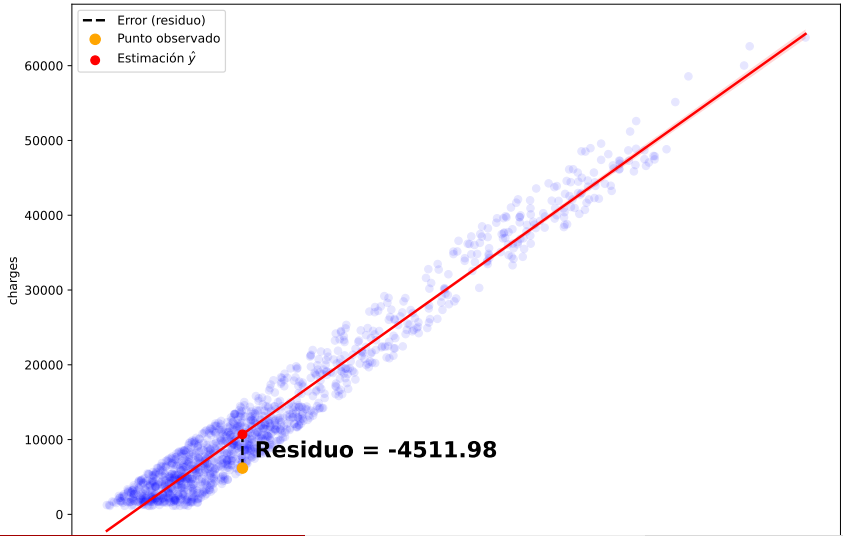
Motivación biomédica

- Ajustar parámetros de un **modelo fisiológico** o un **predictor clínico** con datos reales.
- Ejemplos típicos:
  - Estimar **gasto energético** a partir de IMC, edad y FC.
  - Estimar **edad vascular** con variables de laboratorio.
  - **Clasificar** presencia/ausencia de una condición (0/1) con biomarcadores.

Regresión Lineal



Regresión Lineal



Función de costo (error)

Regresión (continuo):

$$J(\mathbf{w}) \;=\; \frac{1}{2m}\sum_{i=1}^{m}\left(y_j - \hat{y}_j\right)^2$$

donde $\hat{y}_j = \mathbf{w}^\top \mathbf{x}_j$.

Clasificación binaria: Entropía cruzada (log-loss):

$$J(\mathbf{w}) \;=\; -\frac{1}{m}\sum_{i=1}^{m}\left[\, y_i \log \hat{p}_i + (1 - y_i)\right.$$

donde $\hat{p}_i = \sigma(\mathbf{w}^\top \mathbf{x}_i) = \frac{1}{1+e^{-\mathbf{w}^\top \mathbf{x}_i}}$.

Función de costo (error)

! ... Para el caso de ejemplo (un modelo tipo lineal)

$$\mathbf{w_j}^\top = \left[w_{j,1}, w_{j,0}\right]$$

$$salary_i = x_i$$

$$\hat{y}_j = w_{j,1} * x_j + w_{j,0}$$

$$J(\mathbf{w}) = \frac{1}{2m} \sum_{i=1}^{m} \left(y_j - w_{j,1} * x_i - w_{j,0}\right)^2$$

Idea central del GD

- Partimos de $\mathbf{w}_{(0)}$ (p. ej., aleatorio).
- Iteramos:

$$\mathbf{w}_{(t+1)} \;=\; \mathbf{w}_{(t)} \;-\; \alpha\, \nabla_{\mathbf{w}} J(\mathbf{w}_{(t)})$$

- $\alpha =$ **tasa de aprendizaje**: grande $\rightarrow$ rápido pero inestable; pequeña $\rightarrow$ estable pero lento.
- Convergencia: buscamos $\nabla J \approx \mathbf{0}$.

Visualización conceptual: "valle" del error y trayectoria en zig-zag hacia el mínimo.

Algoritmo de Descenso de Gradiente

Para la estimación de $w_{j,0}$

$$w_{j+1,0} = w_{j,0} - \alpha \frac{\partial J}{\partial w_{j,0}}$$

$$\frac{\partial J}{\partial w_{j,0}} = \frac{\partial}{\partial w_{j,0}} \left(\frac{1}{2m} \sum_{i=1}^{m} \left(y_i - w_{j,1} x_i - w_{j,0} \right)^2 \right)$$

$$\frac{\partial}{\partial w_{j,0}} \left(\frac{1}{2m} \sum_{i=1}^{m} \left(y_i - w_{j,1} x_i - w_{j,0} \right)^2 \right) = \frac{1}{m} \sum_{i=1}^{m} \left(w_{j,1} x_i + w_{j,0} - y_i \right)$$

Para la estimación de $w_{j,1}$

Variantes: batch, mini-batch, SGD

- **Batch GD:** usa todo el conjunto en cada actualización (costo alto por iteración).
- **Mini-batch GD:** usa lotes pequeños (compromiso eficiencia/ruido).
- **SGD (estocástico):** actualiza con una sola muestra por paso (barato, ruidoso, puede escapar de óptimos pobres).

Práctica recomendada: mini-batch (p. ej., 32–256).

Preprocesamiento y escalado

- **Estandarizar** o **normalizar** las x_j acelera y estabiliza GD.
- Centrar: $x_j \leftarrow (x_j - \mu_j)/\sigma_j$.
- Manejo de outliers y transformaciones (log, Box–Cox) cuando aplique.

1.5 EDA

Empezar a usar jupyter.

Sección 2

2. Caso 1 · Regresión lineal múltiple con GD

Planteamiento

Objetivo biomédico (ejemplo): predecir **gasto energético** (kcal) a partir de **edad**, **IMC** y **FC**.

Modelo lineal:

$$\hat{y} = \mathbf{w}^\top \mathbf{x} = w_0 + w_1 x_1 + \cdots + w_p x_p$$

Costo (MSE):

$$J(\mathbf{w}) = \frac{1}{2m} \sum_{i=1}^{m} (y_i - \hat{y}_i)^2$$

Gradiente:

$$\frac{\partial J}{\partial w_j} = -\frac{1}{m} \sum_{i=1}^{m} (y_i - \hat{y}_i)\, x_{ij}$$

Actualización:

$$w_j \leftarrow w_j - \alpha \frac{\partial J}{\partial w_j}$$

Pseudocódigo (mini-batch)

```
in: X (m×p), y (m),  , batch_size, epochs
preprocess: X ← standardize(X)

initialize w ← zeros(p+1)  # incluye sesgo w0 si se usa X con col

for epoch in 1..epochs:
    for B in iterate_minibatches(X, y, batch_size, shuffle=True):
        Xb, yb ← B
        yhat ← Xb · w
        grad ← (1/|B|) · (Xb  · (yhat - yb))
        w ← w -   · grad

return w
```

Diagnóstico y validación

- Curva J vs. iteraciones (entrenamiento y validación).
- Errores residuales: homocedasticidad, estructura vs. predicción.
- Interpretación clínica de coeficientes w_j y unidades.
- Comparar con **ecuaciones normales** (solución cerrada) y discutir condicionamiento numérico.

Sección 3

3. Caso 2 · Regresión logística con GD (clasificación 0/1)

Planteamiento

Objetivo biomédico (ejemplo): clasificar **riesgo de enfermedad** (0/1) con panel de biomarcadores.

Modelo:

$$\hat{p} = \sigma(\mathbf{w}^\top \mathbf{x}), \quad \sigma(z) = \frac{1}{1+e^{-z}}$$

Costo (entropía cruzada):

$$J(\mathbf{w}) = -\frac{1}{m}\sum_{i=1}^{m}\Big[y_i \log \hat{p}_i + (1-y_i)\log(1-\hat{p}_i)\Big]$$

Gradiente:

$$\frac{\partial J}{\partial w_j} = \frac{1}{m}\sum_{i=1}^{m}(\hat{p}_i - y_i)\, x_{ij}$$

Actualización:

$$\frac{\partial J}$$

Pseudocódigo (mini-batch)

```
in: X (m×p), y {0,1}^m,  , batch_size, epochs
preprocess: X ← standardize(X)

initialize w ← zeros(p+1)

for epoch in 1..epochs:
    for B in iterate_minibatches(X, y, batch_size, shuffle=True):
        Xb, yb ← B
        z ← Xb · w
        p ← sigmoid(z)
        grad ← (1/|B|) · (Xb  · (p - yb))
        w ← w -   · grad

return w
```

Métricas y curvas

- **AUC-ROC**, **AUPRC**, **sensibilidad**, **especificidad**, **F1**.
- Elección del **umbral** τ por criterio clínico (p. ej., maximizar sensibilidad bajo límite de FPs).
- Calibración: curvas de confiabilidad.

Visualización de la frontera de decisión

- Con dos características (x_1, x_2), la frontera es una **línea** (hiperplano en general).
- Durante GD, la frontera rota/traslada hasta estabilizarse.
- Añadir **términos polinomiales** o **bases** para fronteras no lineales; GD sigue aplicando.

Sección 4

4. Buenas prácticas y discusión aplicada

Hiperparámetros y trucos prácticos

- **Tasa de aprendizaje (α)**: búsqueda en rejilla o **programación de tasa** (decay).
- **Inicialización**: pequeña aleatoria (cero puede estancar con ciertas variantes).
- **Barajado** por época y **mini-batches** estratificados si la clase es rara.
- **Regularización** (L2/L1) para estabilidad e interpretabilidad:

$$J_\lambda = J + \frac{\lambda}{2}\|\mathbf{w}\|_2^2 \quad \text{(Ridge)}$$

- **Detección de fuga de datos** y validación por **sujeto** en estudios clínicos.

Checklist de la sesión (rápido)

- ☐ Estandarizaste variables de entrada.
- ☐ Definiste costo adecuado (MSE vs. CE).
- ☐ Elegiste mini-batch y α razonables.
- ☐ Verificaste convergencia con curva de J.
- ☐ Evaluaste con métricas adecuadas al objetivo clínico.
- ☐ Documentaste supuestos y limitaciones.