

Práctica Dirigida Número 1

Optimización, Sesgo-Varianza y Cross-Validation en Python

Zarit de la Cruz

26 de octubre de 2025

1. Ejercicios Propuestos

Parte I:

1. Implementación de Descenso de Gradiente

En este ejercicio, implementaremos el Descenso de Gradiente para encontrar el mínimo de una función cuadrática simple:

$$f(x) = x^2 - 4x + 5$$

Analíticamente, sabemos que el mínimo se encuentra donde la derivada es cero: $f'(x) = 2x - 4 = 0 \implies x = 2$. Usaremos este resultado para verificar nuestra implementación.

- a) **Definir la función y su gradiente:** Cree dos funciones en Python. La primera, `f(x)`, que devuelva el valor de la función. La segunda, `df(x)`, que devuelva el valor de la derivada (el gradiente).
- b) **Inicializar parámetros:** Defina los hiperparámetros del algoritmo:
 - Un punto de partida (ej. `x_inicial = 10.0`).
 - Una tasa de aprendizaje (*learning rate*, ej. `alpha = 0.1`).
 - Un número de iteraciones (ej. `n_iteraciones = 50`).
- c) **Implementar el bucle de optimización:** Cree un bucle que se repita `n_iteraciones` veces. En cada iteración, actualice el valor de `x` usando la regla del Descenso de Gradiente:

$$x_{\text{nuevo}} = x_{\text{viejo}} - \alpha \cdot f'(x_{\text{viejo}})$$

Guarde el valor de `x` en cada paso en una lista para poder visualizar la trayectoria.

- d) **Visualizar y verificar:** Genere un gráfico que muestre la curva de la función $f(x)$. Sobre este gráfico, dibuje los puntos de la trayectoria que siguió su algoritmo. ¿Converge al mínimo teórico en $x = 2$?

Parte II: Aplicación a un Problema de Machine Learning

El Conjunto de Datos: Hitters

Utilizaremos el dataset `Hitters`, que contiene estadísticas de jugadores de béisbol. Nuestro objetivo será predecir su `Salary`.

- **URL del dataset:** <https://gist.github.com/keeganhines/59974f1ebef97bbaa9raw/d9bcf657f97201394a59ffffd801c44347eb7e28d/Hitters.csv>

El Modelo: Regresión Lasso

Usaremos la **Regresión Lasso**, cuya complejidad se controla con el hiperparámetro **alpha** (α). Nuestra tarea será encontrar el α óptimo.

2. Carga y Preparación de Datos:

- Cargar el dataset desde la URL en un DataFrame de pandas.
- Realizar una exploración inicial (`.info()`, `.describe()`).
- Manejar los valores faltantes en la columna `Salary` eliminando las filas correspondientes. Imprima las dimensiones del DataFrame antes y después de la limpieza.
- Separar los datos en predictores (`X`) y objetivo (`y`).

3. Implementación de Validación Cruzada con Lasso:

- Utilice el objeto `LassoCV` de `scikit-learn` para encontrar el mejor α automáticamente.
- Configure el modelo para que use 10 pliegues (`cv=10`) y fije una semilla (`random_state=1`) para la reproducibilidad.
- Entrene el modelo `LassoCV` con sus datos.
- Imprima el valor óptimo de α encontrado, que se almacena en el atributo `.alpha_`.

4. Análisis del Trade-off Sesgo-Varianza:

- El objeto `LassoCV` entrenado almacena el MSE de cada pliegue para cada α en el atributo `.mse_path_`.
- Calcule el MSE promedio a través de los 10 pliegues para cada valor de α probado.
- Genere un gráfico con los valores de α en el eje X (escala logarítmica) y el MSE promedio en el eje Y.
- Interpretación:** Explique cómo el gráfico representa el trade-off sesgo-varianza, identificando las regiones de underfitting y overfitting.