Universidad Nacional del Altiplano
Facultad de Ingeniería Estadística e Informática

Docente:

Ing. Torres Cruz Fred

Alumno:

Ticona Miramira Roberto Angel

Trabajo 8 - Resumen del artículo Comparing the Moore–Penrose Pseudoinverse and Gradient Descent for Solving Linear Regression Problems: A Performance Analysis

El artículo compara dos métodos fundamentales para resolver problemas de regresión lineal: el **pseudoinverso de Moore–Penrose** y el **descenso de gradiente**. La regresión lineal busca encontrar los parámetros que mejor predicen una variable dependiente a partir de variables independientes, minimizando los errores cuadrados (OLS).

El pseudoinverso de Moore–Penrose ofrece una **solución analítica exacta** mediante operaciones matriciales, proporcionando el mínimo error posible. Sin embargo, puede ser computacionalmente costoso e inestable cuando la matriz de características es muy grande o está mal condicionada. En cambio, el **descenso de gradiente** es un método **iterativo** que ajusta los parámetros gradualmente en función de la dirección del gradiente negativo del error, controlado por una tasa de aprendizaje. Aunque no garantiza una solución exacta, es escalable y adaptable a grandes volúmenes de datos.

En el **marco teórico**, se explica que la estabilidad numérica y la eficiencia de cada método dependen del tamaño del conjunto de datos (n), el número de variables (d) y la condición de la matriz X. El pseudoinverso utiliza descomposición SVD, mientras que el descenso de gradiente requiere pasos controlados y puede verse afectado por la escala de las variables.

En la **metodología**, el autor realizó experimentos con datos sintéticos y reales. Los datos sintéticos permitieron manipular parámetros como el número de muestras, la dimensionalidad y el factor de condición. Para los datos reales se usaron los conjuntos *California Housing* (20,640 muestras y 8 variables) y *Diabetes* (442 muestras y 10 variables). En ambos casos se evaluaron tres métricas: **tiempo de ejecución**, **error cuadrático medio (MSE)** y, para el descenso de gradiente, el **número de iteraciones hasta la convergencia**. Los experimentos se implementaron en Python, usando la función `pinv()` para el pseudoinverso y un algoritmo de gradiente con tasa de aprendizaje $\alpha = 0.01$ y tolerancia 10^{-6}.

Los **resultados** mostraron que el pseudoinverso fue más rápido y preciso en conjuntos pequeños o moderados, manteniendo una alta estabilidad numérica. En cambio, el descenso de gradiente necesitó más iteraciones y su rendimiento dependió fuertemente del escalado de las variables y del valor de la tasa de aprendizaje. Sin embargo, en escenarios de alta dimensionalidad o grandes volúmenes de datos, el descenso de gradiente (y sus variantes como el estocástico) resultó más escalable y eficiente.

En **conclusión**, el pseudoinverso de Moore–Penrose es ideal para conjuntos de datos bien condicionados y de tamaño moderado, ofreciendo soluciones exactas con bajo error. El descenso de gradiente es preferible para problemas de gran escala, aunque requiere ajuste de hiperparámetros y un buen preprocesamiento. Ambos métodos son esenciales en el análisis de regresión, y su elección depende del equilibrio entre exactitud, estabilidad y costo computacional.

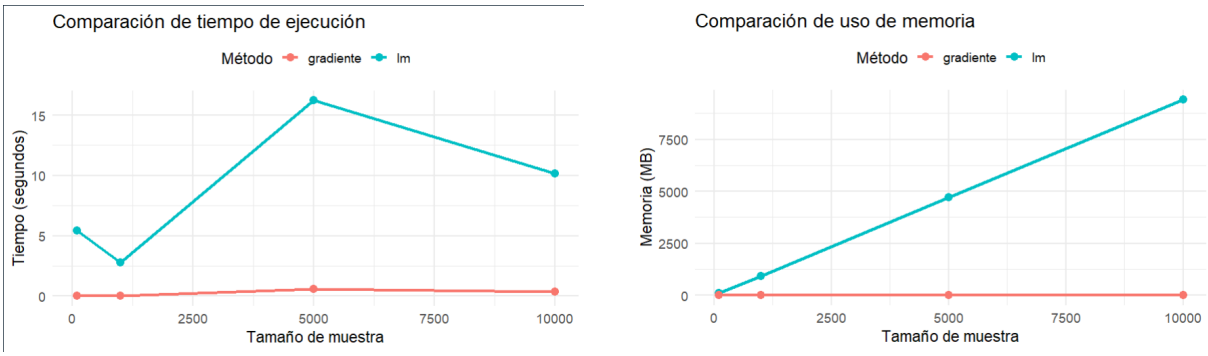
Comparación de rendimiento entre métodos de optimización

El objetivo de esta prueba fue evaluar la eficiencia computacional del método de regresión lineal de `R` (`lm`) en comparación con el algoritmo de **gradiente descendente**. Se emplearon conjuntos de datos de diferentes tamaños (100 a 10 000 observaciones) y se registraron el **tiempo de ejecución** (en segundos) y el **uso de memoria** (en bytes) mediante la librería `bench`. Ambos métodos fueron implementados bajo las mismas condiciones para garantizar una comparación justa.

Tamaño	Método	Tiempo (s)	Memoria (bytes)
100	lm	5.431	104 573 176
100	Gradiente	0.005	143 432
1000	lm	2.769	967 960 000
1000	Gradiente	0.030	800 048
5000	lm	16.215	4 934 680 000
5000	Gradiente	0.566	4 000 048
10000	lm	10.175	9 862 360 000
10000	Gradiente	0.352	8 000 048

Tabla 1: Comparación del rendimiento entre `lm()` y el método del gradiente descendente.

Resultados gráficos: A continuación se muestran las diferencias en tiempo de ejecución y consumo de memoria entre ambos métodos.



Análisis: Los resultados evidencian una ventaja clara del gradiente descendente frente a la función `lm()`. Mientras que `lm()` presenta un aumento drástico del tiempo y la memoria conforme crece el tamaño de los datos, el gradiente mantiene un incremento casi lineal y mucho menor. Esto se debe a que `lm()` realiza operaciones matriciales completas para estimar los coeficientes, lo que resulta costoso en términos computacionales, mientras que el gradiente ajusta los parámetros de forma iterativa y progresiva. En consecuencia, para grandes volúmenes de datos o tareas repetitivas, el método del gradiente descendente ofrece un rendimiento considerablemente superior, tanto en velocidad como en eficiencia del uso de recursos.