

## RESUMEN DEL ARTICULO:

### "COMPARING THE MOORE - PENROSE PSEUDOINVERSE AND GRADIENT DESCENT FOR SOLVING LINEAR REGRESSION PROBLEMS: A PERFORMANCE ANALYSIS"

El artículo presenta una investigación exhaustiva sobre métodos para resolver regresión lineal por mínimos cuadrados ordinarios OLS, cuyo objetivo es encontrar el vector de parámetros  $\beta$  que minimiza la suma de errores al cuadrado, expresada como  $\arg \min_{\beta} \|X\beta - y\|_2^2$ .

#### 1: 20 Pseudoinversa de Moore-Penrose (pinv).

- Es una solución analítica directa y cerrada. Generaliza la inversa de una matriz permitiendo encontrar una solución incluso si la matriz  $X$  no es cuadrada.
- la solución se obtiene con  $\beta_{pinv} = X^+ y$ .

#### 2 EL DESCENSO DE GRADIENTE (GD) es

Es un método iterativo de optimización que encuentra la solución de forma incremental, actualizando los parámetros para minimizar errores.

Para realizar la comparación se emplea:

- Se generaron conjuntos de datos donde se controlaban sistemáticamente el número de muestras ( $n$ ), el número de características ( $d$ ) y factor de condición de matriz  $X$ . Un factor de condición bajo que simula datos mal condicionados.
- Se usó conjunto de datos "California Housing" ( $n \approx 20,640$ ,  $d = 8$ ) y "Oci diabetes" ( $n = 442$ ,  $d = 10$ ) para validar Hallazgos.
- El rendimiento se midió usando (1) tiempo de ejecución error cuadrático medio para medir precisión, iteraciones hasta la convergencia.



## RESULTADOS

### 3 EFICIENCIA COMPUTACIONAL (TIEMPO DE EJECUCIÓN)

- Para los rangos probados hasta  $n=5000$ ,  $d=50$  la pseudo inversa fue consistentemente más rápida a menudo por órdenes de magnitud, completando los milisegundos.
- El tiempo de ejecución del descenso de gradiente (Batch GD) fue notablemente más lento, ya que su costo por iteración escala con el número de muestras ( $n$ ) y características ( $d$ ).

### 2 PRECISIÓN Y ESTABILIDAD NUMÉRICA

La pseudoinversa proporciona una solución exacta con MSE bajo y estable en todos los condicionamientos y el descenso de gradiente solo alcanza una precisión comparable (MSE bajo) cuando los datos están bien acondicionados.

### 3 SENSIBILIDAD DEL CONDICIONAMIENTO

- En el descenso de gradiente demostró ser altamente sensible al condicionamiento de los datos.
- Los datos mal acondicionados hicieron que GD requiriera más iteraciones y fallara en encontrar solución precisa, mientras que la pseudoinversa al ser solución analítica no se vio afectada de la misma manera y encontró la solución correcta de mínimos cuadrados de norma mínima.

## CONCLUSIÓN

Usar la Pseudo inversa para conjunto de datos pequeños donde  $n$  y  $d$  son grandes, la pseudo inversa de Moore-Penrose es generalmente preferible.

Usar descenso de gradiente: los métodos iterativos con el descenso de gradiente se vuelven "necesarios" o "más prácticos" para conjunto de datos extremadamente grande.



## Reporte de Comparación de Rendimiento

Se realizó un análisis comparativo entre la función Lm y una implementación de gradiente evaluando tiempo de ejecución y uso de memoria de 100 a 1000000 usando bench:mark y siendo perfilado por Profvis para análisis detallado

Tamaño	Lm	Gradiente
100	0,001357	0,002312
1000	0,0014827	0,0220196
10,000	0,0074827	0,1641378
100,000	0,0553629	1,4426852
500,000	0,2189633	7,1380826
1000,000	0,5600950	14,8881908

- Lm() muestra escalabilidad lineal con pendiente suave
- Gradiente descendente escala de forma menos eficiente
- la brecha aumenta a medida que n aumenta

## Uso de memoria (Bytes)

Tamaño	Lm	Gradiente
100	64680	1830776
1000	619048	17619260
10000	6213928	176019260
100000	61851432	1760019200
500000	308197160	8800019200
1000000	616391464	17600019200

Contrario a intuición inicia el gradiente descendente usa 28x mas memoria que Lm durante la ejecución



FileEditCodeViewPlotsSessionBuildDebugProfileToolsHelp

RunSource

Untitled1 x

Analisis detallado con profvis

RunSource

```
69 # uso de memoria
70 p2 <- ggplot(resultados, aes(x = Tamaño, y = Memoria_B / 1e6, color = Metodo)) +
71   geom_point(size = 2.5) + # CORREGIDO: linewidth en lugar de size
72   scale_x_log10(labels = scales::comma) +
73   labs(title = "Comparación de uso de memoria",
74        x = "Tamaño de muestra (escala logarítmica)",
75        y = "Memoria (MB)") +
76   theme_minimal(base_size = 13)
77
78
79 print(p2)
80
81 # ---- Analisis detallado con profvis ----
82 n <- 1000000
83 x <- matrix(rnorm(n * 10), ncol = 10)
84 y <- rnorm(n)
85
86 cat("\n==== PERILANDO CON PROFVIS (n = 1,000,000) =====\n")
87
88 p <- profvis({
89   modelo_lm <- lm(y ~ X)
90   gradiente_desc(X, y)
91 })
92
93 print(p)
94
```

Console

Terminal

Background Jobs

R • R 4.5.1 • ~ /

Analisis detallado con profvis

R Script

Procesando tamaño: 1e+06

RESULTADOS COMPLETOS

Tamaño	Metodo	Tiempo_s	Memoria_B
1e+02	lm_model	0.0013578	64680
1e+02	Gradiente	0.0023126	1830776
1e+03	lm_model	0.0014827	619048
1e+03	Gradiente	0.0220196	17619200
1e+04	lm_model	0.0074224	6213928
1e+04	Gradiente	0.1641378	176019200
1e+05	lm_model	0.0535629	61851432
1e+05	Gradiente	1.4426852	1760019200
5e+05	lm_model	0.2189633	308197160
5e+05	Gradiente	7.1380826	8800019200
1e+06	lm_model	0.5600950	616391464
1e+06	Gradiente	14.8881908	17600019200

