

Cómputo científico para probabilidad y estadística. Tarea 2.

Descomposición QR y mínimos cuadrados.

Juan Esaul González Rangel

Septiembre 2023

1. Implementar el algoritmo de Gram-Schmidt modificado 8.1 del Trefethen (p. 58) para generar la descomposición QR .

En el archivo `QR.py`, el algoritmo está implementado en la función `QR`.

2. Implementar el algoritmo que calcula el estimador de mínimos cuadrados en una regresión usando la descomposición QR .

En el archivo `QR.py`, el algoritmo está implementado en la función `LSQR`.

3. Generar \mathbf{Y} compuesto de $y_i = \sin(x_i) + \varepsilon_i$ donde $\varepsilon_i \sim N(0, \sigma)$ con $\sigma = 0.11$ para $x_i = \frac{4\pi i}{n}$ para $i = 1, \dots, n$.

Hacer un ajuste de mínimos cuadrados a \mathbf{Y} , con descomposición QR , ajustando un polinomio de grado $p - 1$.

- Considerar los 12 casos: $p = 3, 4, 6, 100$ y $n = 100, 1000, 10000$.
- Graficar el ajuste en cada caso.
- Medir tiempo de ejecución de su algoritmo, comparar con descomposición QR de `scipy` y graficar los resultados.

4. Hacer $p = 0.1n$, o sea, diez veces más observaciones que coeficientes en la regresión, ¿Cual es la n máxima que puede manejar su computadora?

Hasta 1560×156 obtenemos resultados no nulos. En 2800×280 obtenemos `Runtime error` y en 3000×300 obtenemos `nan` como resultado.