complemento_Ortogonal

March 4, 2025

1 Complemento Ortogonal

1.1 Explicacion Matematica

El concepto de ortogonalidad y complementos ortogonales se remonta a los trabajos de matemáticos como Euclides y Pitágoras, quienes estudiaron las propiedades de los triángulos rectángulos y las relaciones entre sus lados. Sin embargo, el desarrollo formal del complemento ortogonal como lo conocemos hoy en día se consolidó con el avance del álgebra lineal en los siglos XIX y XX.

En términos simples, el complemento ortogonal de un subespacio

W en un espacio vectorial

V es el conjunto de todos los vectores en

V que son ortogonales a cada vector en

W. Matemáticamente, si

W es un subespacio de

V, el complemento ortogonal de

W, denotado como

W , se define como:

={v V v,w =0 para todo w W}

2 Uso exacto

2.1 1. Descomposición de Vectores

En álgebra lineal, cualquier vector en un espacio vectorial puede descomponerse en una parte que pertenece a un subespacio dado y otra parte que pertenece a su complemento ortogonal. Esto es útil en la resolución de sistemas de ecuaciones lineales y en la proyección de vectores.

2.2 2. Método de Mínimos Cuadrados

En estadística y análisis de datos, el método de mínimos cuadrados se utiliza para encontrar la mejor aproximación lineal a un conjunto de datos. La idea es minimizar la suma de los cuadrados de las diferencias entre los valores observados y los valores predichos. Esto se logra proyectando los datos en el subespacio generado por las variables explicativas, y el error de proyección pertenece al complemento ortogonal.

2.3 3. Análisis de Fourier

En el análisis de Fourier, las funciones se descomponen en una serie de senos y cosenos ortogonales. Los coeficientes de Fourier se obtienen proyectando la función original en los subespacios generados por estas funciones ortogonales.

2.4 4. Procesamiento de Señales

En el procesamiento de señales, las señales se descomponen en componentes ortogonales para facilitar su análisis y manipulación. Por ejemplo, en la Transformada Discreta de Fourier (DFT), las señales se descomponen en una suma de senos y cosenos ortogonales, lo que permite analizar las frecuencias presentes en la señal.

2.5 5. Optimización

En problemas de optimización, el complemento ortogonal se utiliza para encontrar direcciones de búsqueda que sean ortogonales a las restricciones del problema. Esto es útil en métodos como la optimización cuadrática y la programación lineal.

2.5.1 Ejemplo Práctico: Regresión Lineal (lineal regretion)

Supongamos que tenemos un conjunto de datos con n observaciones (puede ser un producto vs su precio en un ambito practico o las ventas en un año o incluso la edad vs el salario) y queremos ajustar una línea recta que minimice la suma de los cuadrados de las diferencias entre los valores observados (cuando se ingrese la edad pueda de una forma sencilla acercarse a lo que gana una persona) y los valores predichos. Este es un problema clásico de regresión lineal.

Datos Imaginemos que tenemos los siguientes datos:

x	y
1	2
2	3
3	5
4	4
5	6

Queremos ajustar una línea de la forma $y = mx + b$.

Matriz de Diseño Primero, construimos la matriz de diseño X y el vector de observaciones $\mathbf{y}$:

$$X = \begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ 1 & 5 \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} 2 \\ 3 \\ 5 \\ 4 \\ 6 \end{pmatrix}$$

Solución de Mínimos Cuadrados La solución de mínimos cuadrados se obtiene resolviendo la ecuación normal:

$$\hat{\ } = (X^T X)^{-1} X^T \mathbf{y}$$

Donde$\hat{\ }$ es el vector de coeficientes (b, m).

Cálculo Calculamos $X^T X$ y $X^T \mathbf{y}$:

$$X^T X = \begin{pmatrix} 5 & 15 \\ 15 & 55 \end{pmatrix}, \quad X^T \mathbf{y} = \begin{pmatrix} 20 \\ 70 \end{pmatrix}$$

Luego, invertimos $X^T X$ y multiplicamos por $X^T \mathbf{y}$:

$$(X^T X)^{-1} = \begin{pmatrix} 11 & -3 \\ -3 & 1 \end{pmatrix}, \quad \hat{\ } = \begin{pmatrix} 11 & -3 \\ -3 & 1 \end{pmatrix} \begin{pmatrix} 20 \\ 70 \end{pmatrix} = \begin{pmatrix} 0.4 \\ 1.0 \end{pmatrix}$$

Por lo tanto, la línea de mejor ajuste es:

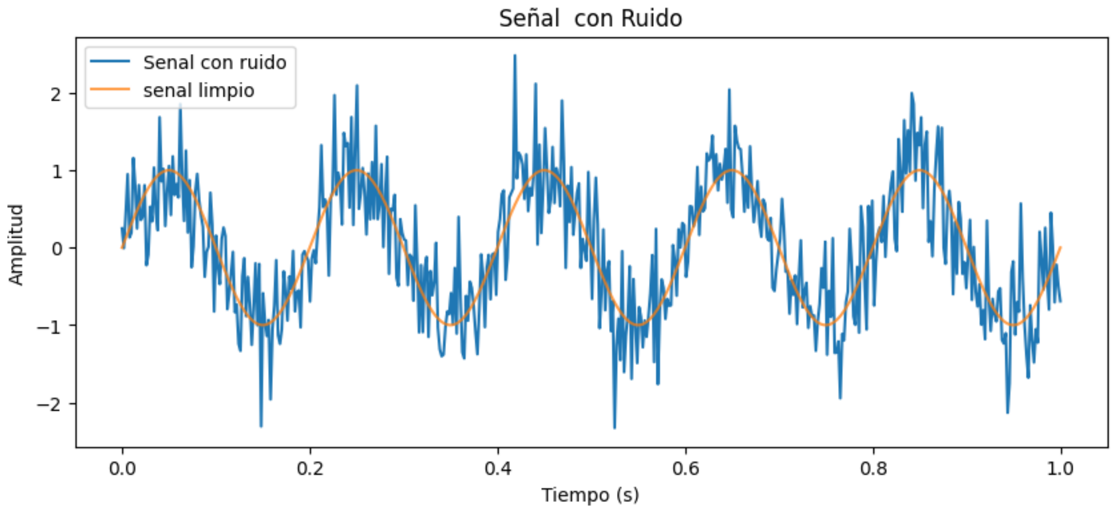
$$y = 1.0x + 0.4$$

Interpretación del Complemento Ortogonal El error de ajuste, es decir, la diferencia entre los valores observados y los valores predichos, pertenece al complemento ortogonal del subespacio generado por las columnas de X. Esto significa que los errores son ortogonales a los vectores de la matriz de diseño, lo que garantiza que hemos encontrado la mejor aproximación lineal en el sentido de mínimos cuadrados.

3 Codigo

[7]:
```python
import numpy as np
import matplotlib.pyplot as plt

# Generar una señal  simulada
np.random.seed(42)
t = np.linspace(0, 1, 500)
ecg_clean = np.sin(2 * np.pi * 5 * t)  # Señal limpia
noise = 0.5 * np.random.randn(500)  # Ruido
ecg_noisy = ecg_clean + noise  # Señal con ruido

# Graficar la señal con ruido
plt.figure(figsize=(10, 4))
plt.plot(t, ecg_noisy, label='Senal con ruido')
plt.plot(t, ecg_clean, label='senal limpio', alpha=0.75)
plt.legend()
plt.title('Señal  con Ruido')
plt.xlabel('Tiempo (s)')
```

```
plt.ylabel('Amplitud')
plt.show()
```

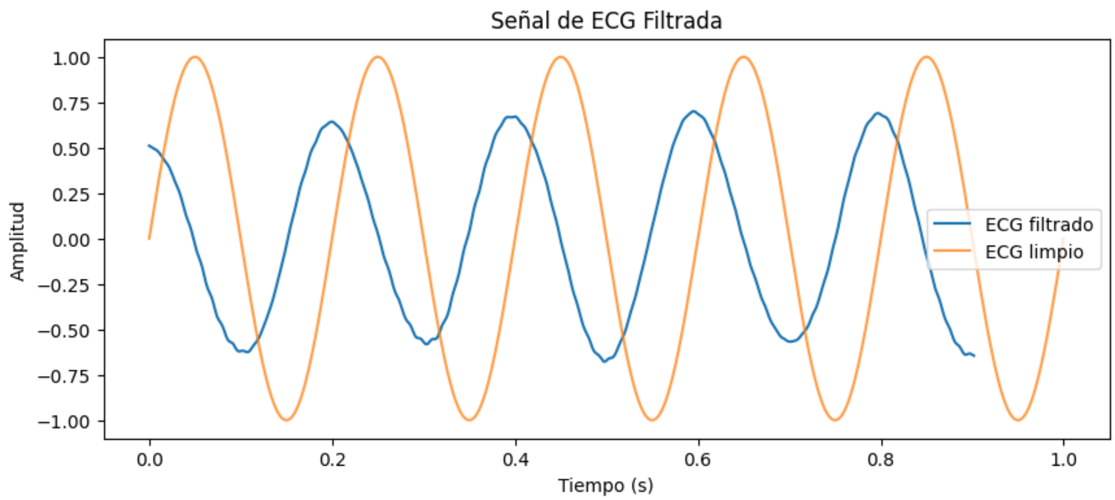


La anterior imagen es dentro lo amarillo es la senal emitida desde un sistema scada, en el cual puede ser la trasmiscion de un elemento de oil and gas donde puede pasar un elemento como el agua, la cual se puede contaminar la salida, por temas externos(fallas en el cable, contaminacion de h20 con otros elmento como lo es la tierra)

[8]:
```
from scipy.linalg import svd

# Crear una matriz de Hankel para aplicar SVD
def hankel_matrix(signal, L):
    N = len(signal)
    return np.array([signal[i:i+L] for i in range(N-L+1)])

L = 50  # Tamaño de la ventana
H = hankel_matrix(ecg_noisy, L)

# Aplicar SVD
U, s, Vt = svd(H, full_matrices=False)
```

[9]:
```
# Reconstruir la señal utilizando los primeros k componentes principales
k = 5
H_denoised = np.dot(U[:, :k], np.dot(np.diag(s[:k]), Vt[:k, :]))

# Promediar las filas de la matriz de Hankel reconstruida para obtener la señal
 ↪filtrada
ecg_denoised = np.mean(H_denoised, axis=1)

# Graficar la señal filtrada
```

```
plt.figure(figsize=(10, 4))
plt.plot(t[:len(ecg_denoised)], ecg_denoised, label='ECG filtrado')
plt.plot(t, ecg_clean, label='ECG limpio', alpha=0.75)
plt.legend()
plt.title('Señal de ECG Filtrada')
plt.xlabel('Tiempo (s)')
plt.ylabel('Amplitud')
plt.show()
```



Con los datos puedes limpiar las diferentes senales, que te permitiran de una forma mas claara que es lo que esta pasadno dentro de lo cual, el ruido se elimina con los componentes ortogonales, correspondiendo a el ruido captado o trasmitido por el SCAda(elemento de captura de datos)

4 Enlaces

http://matematicas.uam.es/~fernando.chamizo/asignaturas/2021alglin/sections/4_3.pdf

https://www.britannica.com/science/

https://mathshistory.st-andrews.ac.uk/Biographies/Grassmann/

https://machinelearningmastery.com/article/11751456084/

https://programmerclick.com/article/6115655816/