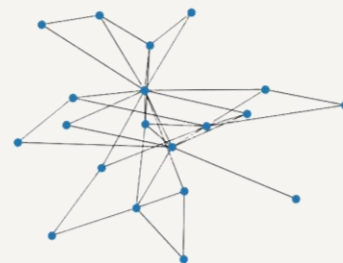


# Inferencia de Topología

ASIGNATURA: PROCESAMIENTO DE DATOS EN DOMINIOS  
IRREGULARES

JAIME CÁCERES YANES  
ADRIÁN ONECHA PRIETO  
HUGO LEÓN MARÍN

UNIVERSIDAD REY JUAN CARLOS  
CURSO 2025-2026



# Índice

- MOTIVACIÓN Y PROBLEMA
- ALGORITMOS Y MÉTODOS DE EVALUACIÓN
- DISEÑO EXPERIMENTAL Y RESULTADOS
- CONCLUSIONES

# Introducción

- En muchos problemas reales, los datos están asociados a **grafos**
- La **estructura del grafo** no siempre es conocida
- Observamos **señales en los nodos** del grafo
- El objetivo es **inferir el grafo subyacente** a partir de estas señales

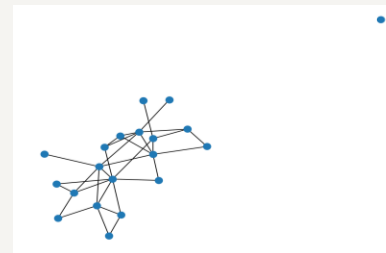


# Modelos de grafos considerados

## Erdős–Rényi (ER)

Conexiones aleatorias

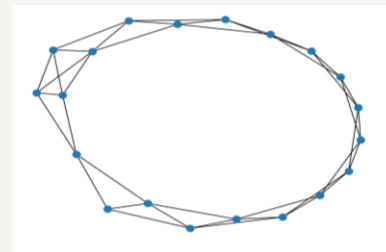
Grafo base sin estructura compleja



## Watts–Strogatz (Small-World)

Alto clustering y caminos cortos

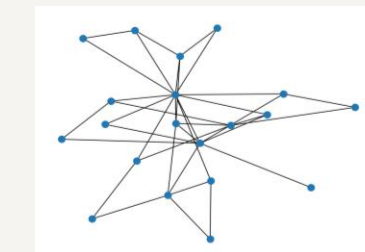
Modela redes reales



## Barabási–Albert (Scale-Free)

Distribución de grados en ley de potencia

Presencia de nodos hub (Estructura homogénea)



# Tipos de señales consideradas

- **Señales Gaussianas**
- Modeladas como variables gaussianas multivariantes  $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \Theta^{-1})$
- La estructura del grafo se refleja en la matriz de precisión  $\Theta$

- **Señales estacionarias**
- Generadas como  $\mathbf{x} = \mathbf{H}\mathbf{w}$ , con  $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  y  $\mathbf{H} = h_0\mathbf{I} + h_1\mathbf{L} + h_2\mathbf{L}^2 + \dots$
- El grafo se codifica mediante el Laplaciano
- Hemos elegido  $h_0=1.0$ ,  $h_1=-0.25$ ,  $h_2=0.0$

## Gaussiano:



$\Theta$  captura independencia condicional.  
Cero si no hay conexión.

## Estacionario:



Filtrado en dominio espectral.  $H$  transforma ruido blanco.

# Algoritmos de inferencia considerados

## Graphical Lasso (SCIKIT)

- Estima una matriz de precisión **dispersa**
- Basado en regularización L1

## GLASSO con CVXPY

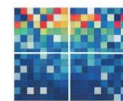
- Formulación convexa explícita
- Solución mediante optimización convexa

## Descenso por gradiente proyectado (PGD)

- Algoritmo iterativo
- Promueve dispersidad mediante soft-thresholding

## Objetivo común:

Inferir la estructura del grafo a partir de la matriz de precisión estimada



Covarianza (S)



Algoritmos  
(GLASSO / PGD)



Matriz de Precisión



Grafo  
Recuperado

# Métricas de evaluación

## Error relativo de Frobenius

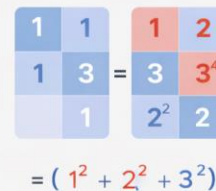
- Compara la matriz estimada con la real
- Mide precisión de la inferencia

## Esparsidad

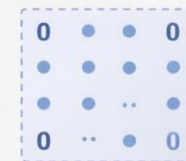
- Proporción de entradas cercanas a cero
- Refleja simplicidad del grafo inferido

## Tiempo de cómputo

- Tiempo del solver de inferencia
- Evalúa la eficiencia del método


$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 3 & 2 \end{bmatrix}$$
$$= (1^2 + 2^2 + 3^2)$$

Error de Frobenius



Dispersidad

Tiempo de  
cómputo

# Configuración experimental

Número de nodos fijo:  $N = 20$

Número de señales: 50

Algoritmos: Ridge, GLASSOS, PGD

Ridge:  $\hat{\Theta}_{\text{Ridge}} = (S + \gamma I)^{-1}$

**Resultados:** Descartamos el algoritmo ridge para nuevos experimentos por sus malos resultados

```

=== ERDOS_RENYI | GAUSSIAN | Single run (same plotted graph) ===
ridge      | Fro(0)=1.942 | sparsity=0.000 | solver_time=0.08 ms
glasso_skl | Fro(0)=0.550 | sparsity=0.389 | solver_time=100.40 ms
glasso_cvx | Fro(0)=0.550 | sparsity=0.389 | solver_time=271.84 ms
pgd        | Fro(0)=0.758 | sparsity=0.089 | solver_time=23.00 ms

=== ERDOS_RENYI | STATIONARY | Single run (same plotted graph) ===
ridge      | Fro(L)=17.866 | sparsity=0.000 | solver_time=0.08 ms
glasso_skl | Fro(L)=0.771 | sparsity=0.537 | solver_time=7.58 ms
glasso_cvx | Fro(L)=0.771 | sparsity=0.537 | solver_time=81.51 ms
pgd        | Fro(L)=0.722 | sparsity=0.447 | solver_time=22.65 ms

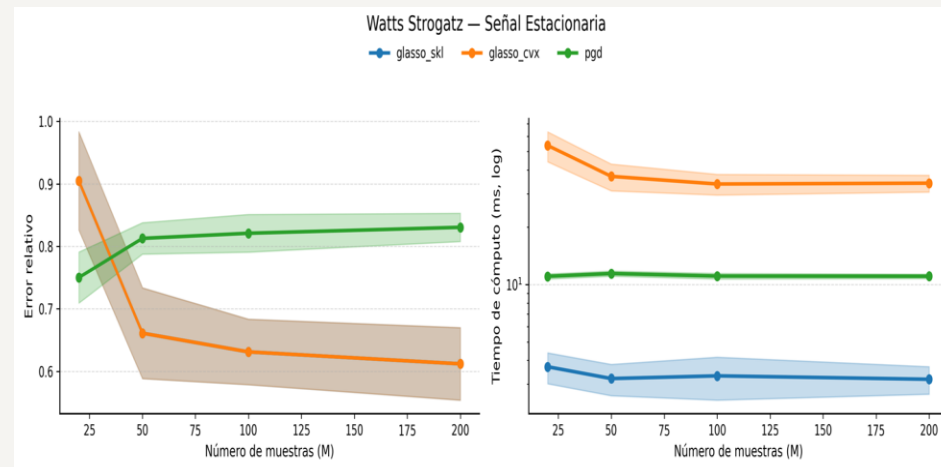
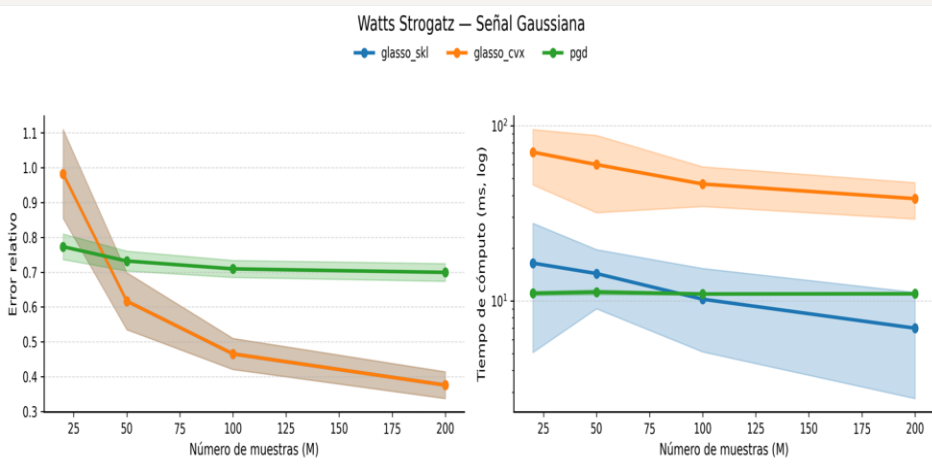
=== WATTS_STROGATZ | GAUSSIAN | Single run (same plotted graph) ===
ridge      | Fro(0)=2.218 | sparsity=0.000 | solver_time=0.07 ms
glasso_skl | Fro(0)=0.550 | sparsity=0.595 | solver_time=14.38 ms
glasso_cvx | Fro(0)=0.549 | sparsity=0.595 | solver_time=479.79 ms
pgd        | Fro(0)=0.688 | sparsity=0.321 | solver_time=22.86 ms
  
```



# Resultados variando el número de señales

Número de nodos fijo:  $N = 20$   
Tres topologías: ER, Small World, Scale-Free

Número de señales:  $M \in \{10, 20, 50, 100, 200\}$   
Evaluación mediante error y tiempo



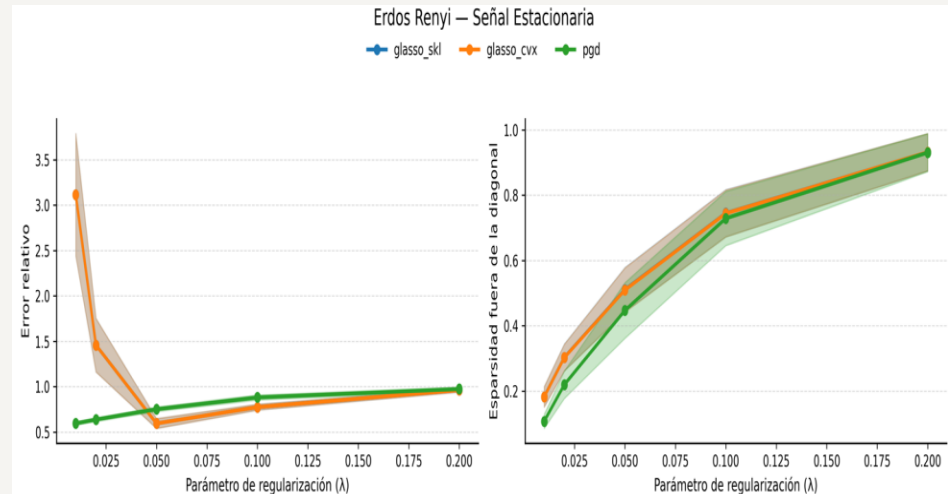
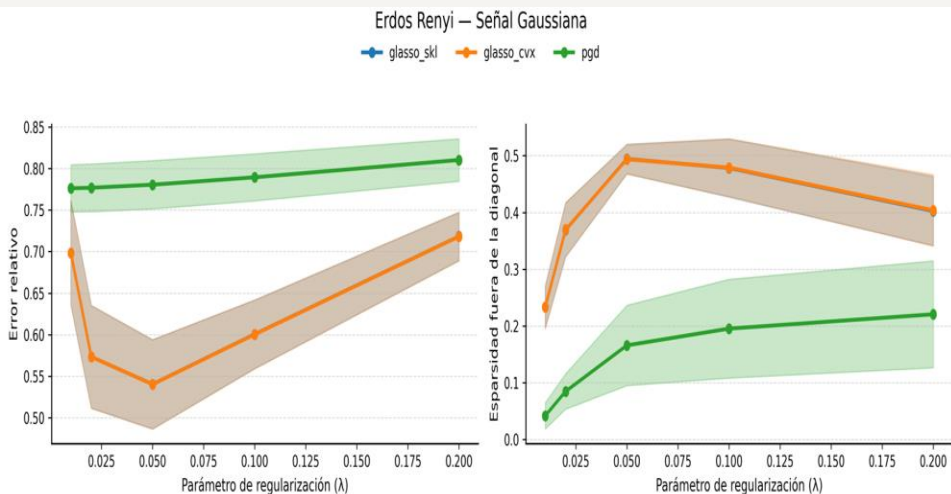
# Resultado variando el parámetro de regularización

Número de nodos fijo:  $N = 20$

Tres topologías: ER, Small World, Scale-Free

$\Lambda \in \{0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2\}$

Evaluación mediante error y esparsidad



# Resultados con $N=100$

Número de nodos fijo:  $N = 100$

Número de señales:  $M=500$

Parámetro de regularización:  $\text{lam}=0.05$

Algoritmos: GLASSOS, PGD

**Resultados:** El número de nodos influye en el tiempo de resolución

```
=== BIG RUN | ER | N=100 | M=500 | avg_degree=6 | signal=gaussian | lam=0.05 ===  
glasso_skl | Fro(0)=0.592 | sparsity(0)=0.756 | solver_time=641.14 ms  
pgd        | Fro(0)=0.909 | sparsity(0)=0.279 | solver_time=145.86 ms  
glasso_cvx | Fro(0)=0.592 | sparsity(0)=0.756 | solver_time=13728.40 ms
```

```
=== BIG RUN | ER | N=100 | M=500 | avg_degree=6 | signal=stationary | lam=0.05 ==  
glasso_skl | Fro(L)=0.569 | sparsity(0)=0.727 | solver_time=167.86 ms  
pgd        | Fro(L)=0.713 | sparsity(0)=0.710 | solver_time=111.08 ms  
glasso_cvx | Fro(L)=0.569 | sparsity(0)=0.727 | solver_time=5081.03 ms
```

# Conclusiones

- Ridge no es adecuado para la inferencia de grafos
- Graphical Lasso es el mejor en precisión en señales gaussianas
- En cambio en señales estacionarias PGD compete a Glasso
- Sklearn y CVXPY dan el mismo error y sparsity siendo sklearn mucho más rápido
- El número de muestras no influye apenas, mientras que el número de nodos sí lo hace
- Link a repositorio: [https://github.com/aonecha/trabajo\\_PDDI](https://github.com/aonecha/trabajo_PDDI)