Aprendizaje Automático para Datos en Grafos

Laboratorio 4

Graciana Castro
4.808.848-2
gcastro@fing.edu.uy

Julian O'Flaherty
6.285.986-9
julian.o.flaherty@fing.edu.uy

1. Introducción

Este informe presenta el Laboratorio 4 para la materia Aprendizaje Automático para Datos en Grafos. El objetivo principal es abordar el problema de inferencia de topología en grafos, es decir, la estimación de la estructura subyacente (aristas y pesos) de un grafo a partir de observaciones o medidas en sus nodos. Este problema es fundamental en muchas aplicaciones reales donde se conocen los nodos de una red pero no las conexiones entre ellos, y solo se dispone de señales o características observadas en cada nodo.

Se exploran tres metodologías principales: Graphical Lasso (Friedman, Hastie y Tibshirani, 2008), Meinshausen y Bühlmann (2006) y Kalofolias (2016). En la primera parte, descrita en la sección 2, se aplican estos métodos sobre datos sintéticos para evaluar su capacidad de recuperar estructuras conocidas. La segunda parte, sección 3, utiliza el dataset MNIST de dígitos manuscritos, donde cada imagen se interpreta como un nodo del grafo y se evalúa el desempeño mediante clustering espectral. El código implementado puede consultarse en el repositorio de GitHub `https://github.com/j-oflaherty/AA-grafos/blob/main/lab4/Lab4_AAG2025.ipynb`.

2. Inferencia de topología en datos sintéticos

2.1. Generación del grafo sintético

Para generar un grafo sintético, utilizaremos el siguiente algoritmo:

1. Sorteamos N puntos de forma uniforme en el cuadrado $[0,1] \times [0,1]$. Esos serán los vértices de nuestro grafo.

2. Para cada par de puntos i y j que sorteamos antes, tomamos como peso de la arista

$$w_{ij} = \begin{cases} 0 & \text{si } i = j \\ e^{-\frac{d(i,j)}{2\sigma^2}} & \text{si } i \neq j \end{cases}$$

   donde $d(i,j)$ es la distancia euclidea en $\mathbb{R}^2$ y σ es un parámetro fijo.

3. Descartamos las aristas cuyo peso w_{ij} sea menor que un número fijo $r > 0$.

En la figura 1 se muestra un grafo sintético generado con el algoritmo anterior para $N = 50$, $\sigma = 0{,}5$ y $r = 0{,}6$.

A cada nodo del grafo sintetico, le asignaremos una señal $x_i \sim \mathcal{N}(0, \mathbf{L}^\dagger)$, donde $\mathbf{L}^\dagger$ es la pseudoinversa de la matriz laplaciana $\mathbf{L}$ del grafo. En la figura 2 se muestra un *sample* de señales donde cada columna j es la señal asociada al nodo j del grafo. La cantidad de filas, o el tamaño de la muestra, es el parámetro n_{samples} del algoritmo, por lo que si n_{samples} es menor que N, la matriz de covarianza empírica no es invertible.

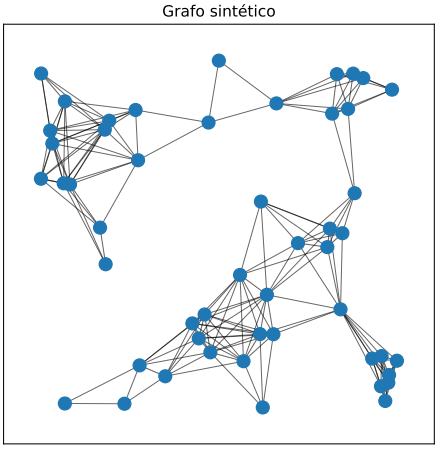


Figura 1: Grafo sintético generado con $N = 50$, $\sigma = 0{,}5$ y $r = 0{,}6$.

2.2. Estimación de estructura con Graphical Lasso

Estimaremos la estructura del grafo a partir de la matriz de datos utilizando el método *Graphical Lasso*. Este método busca al estimador de máxima verosimilitud de la matriz de precisión $\boldsymbol{\Theta}$ que cumple:

$$\hat{\boldsymbol{\Theta}} = \arg\max_{\boldsymbol{\Theta} \succeq 0} \left\{ \log\det \boldsymbol{\Theta} - \operatorname{trace}(\hat{\boldsymbol{\Sigma}}\boldsymbol{\Theta}) - \lambda \|\boldsymbol{\Theta}\|_1 \right\} \tag{1}$$

Una de las propiedades importantes del Graphical Lasso, es que cuando $\lambda = 2\sqrt{\frac{\log N}{P}}$,

$$\|\hat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}_0\|_2 \leq \sqrt{\frac{d_{\text{máx}}^2 \log N}{P}} \quad \text{w.h.p.}$$

con el parámetro λ siendo parámetro de regularización. En la figura 3 se puede observar la estimación obtenida con *Graphical Lasso* comparada con la estimación empírica cuando el número de samples utilizado n_{samples} es menor que la cantidad de nodos N. En este caso, la estimación empírica resulta mala, dado que la matriz de covarianza empírica no es invertible. El método de *Graphical Lasso* obtiene un resultado muy similar a la matriz de precisión verdadera, donde el valor de λ se calcula mediante validación cruzada.

En la figura 4 observamos el efecto del variar el valor de λ sobre la matriz de precisión estimada. A medida que aumenta la regularización, la matriz va perdiendo ceros tendiendo a una matriz diagonal, lo cual es esperable dado que el valor de λ está asociado a la norma 1 de la matriz de precisión.

2.3. Estimación de estructura con Meinshausen y Bühlmann

2.4. Estimación de estructura con Meinshausen y Bühlmann

El método de Meinshausen y Bühlmann propone una alternativa eficiente para la estimación de la estructura de grafos mediante selección de vecindarios con Lasso. Este método es particularmente útil en problemas de alta dimensionalidad donde $p \gg n$, ya que permite identificar

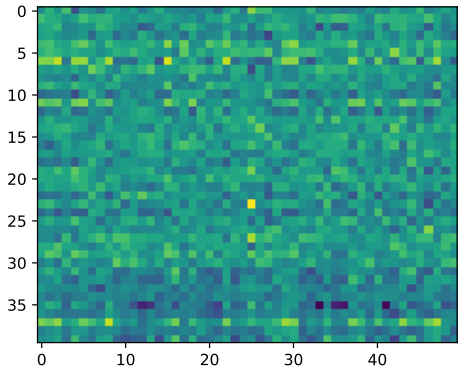
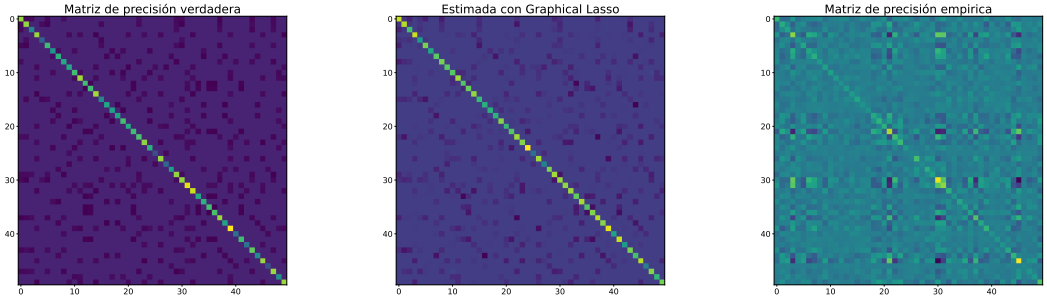


Figura 2: Sample de señales asignadas a los nodos del grafo sintético.



(a) Precisión verdadera

(b) Estimación mediante Graphical Lasso

(c) Estimación empírica ($\Theta = \Sigma^{-1}$)

Figura 3: Comparación entre la matriz de precisión verdadera, la estimación mediante Graphical Lasso y la estimación obtenida con empíricamente ($\Theta = \Sigma^{-1}$), cuando $n_{\text{samples}} = 40$ y $N = 50$

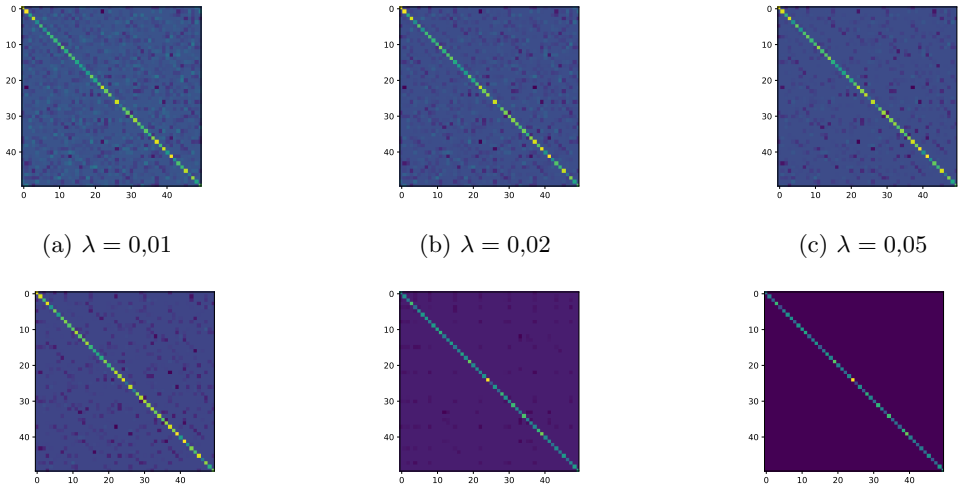
los ceros estructurales en la matriz de precisión que representan restricciones de independencia condicional entre variables.

La idea central consiste en estimar el vecindario de cada nodo de forma individual mediante regresión. Para cada nodo a, se define su vecindario como el conjunto mínimo de variables que, al ser conocidas, hacen que X_a sea condicionalmente independiente del resto. Esto se formula como un problema de regresión Lasso donde X_a es la variable respuesta y las demás son predictoras:

$$\hat{\theta}_{a,\lambda} = \underset{\theta:\theta_a=0}{\operatorname{argmin}} \left(n^{-1} \|X_a - X\theta\|_2^2 + \lambda \|\theta\|_1 \right) \tag{2}$$

El vecindario estimado de a se determina por los coeficientes no nulos de $\hat{\theta}_{a,\lambda}$. Una vez obtenidos todos los vecindarios, el conjunto de aristas del grafo se construye mediante la regla AND:

$$\hat{E}_{\lambda,\wedge} = \{(a,b) : a \in \hat{ne}_{\lambda,b} \wedge b \in \hat{ne}_{\lambda,a}\} \tag{3}$$



(a) $\lambda = 0{,}01$ (b) $\lambda = 0{,}02$ (c) $\lambda = 0{,}05$

(d) $\lambda = 0{,}1$ (e) $\lambda = 0{,}5$ (f) $\lambda = 1$

Figura 4: Estimación mediante Graphical Lasso para distintos valores de λ. La matriz de precisión verdadera se muestra en la figura 3a. No se muestran valores menores a 0,01 porque hay poca diferencia notable, y a partir de 0,005 el sistema queda mal condicionado y no puede ser resuelto.

Una propiedad importante de este método es que mantiene consistencia incluso cuando $p \gg n$, siempre que el parámetro λ se elija adecuadamente.

Para implementar este método, seguimos el siguiente algoritmo:

1. Normalizamos la matriz de datos $\mathbf{X}$ por columnas.
2. Inicializamos una matriz $\mathbf{B}$ de ceros de tamaño $p \times p$.
3. Para cada nodo j del grafo:
   a) Extraemos la columna j de $\mathbf{X}$ como variable objetivo $\mathbf{y}_j$.
   b) Utilizamos las demás columnas como matriz de predictores $\mathbf{X}_j$.
   c) Resolvemos el problema de regresión Lasso para obtener los coeficientes $\hat{\boldsymbol{\theta}}_j$.
   d) Insertamos los coeficientes en la fila j de $\mathbf{B}$, colocando un cero en la posición j.
4. Devolvemos la matriz $\mathbf{B}$ que contiene los coeficientes de regresión para cada nodo.

En la figura 5 se observa que el método de Meinshausen y Bühlmann logra capturar la estructura principal de la matriz de coeficientes verdadera, identificando correctamente las conexiones más relevantes. Sin embargo, aparecen algunos falsos positivos y negativos, lo que refleja el compromiso entre detección y ruido inherente al uso de Lasso.

2.5. Estimación de estructura con Kalofolias

El método propuesto por Kalofolias plantea aprender la estructura de un grafo a partir de señales que residen en sus nodos, asumiendo que estas señales cambian suavemente entre nodos conectados. La contribución clave consiste en reformular el problema tradicional de minimización

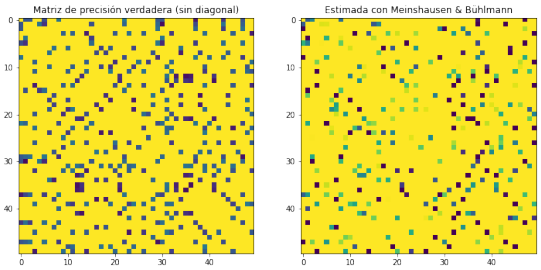


Figura 5: Comparación entre la matriz de coeficientes verdadera y la matriz de coeficientes estimada con el método de Meinshausen y Bühlmann.

del término de suavidad $\text{tr}(X^T LX)$ mostrando que es equivalente a una norma ℓ_1 ponderada aplicada a la matriz de adyacencia W:

$$\text{tr}(X^T LX) = \frac{1}{2}\|W \circ Z\|_{1,1} \tag{4}$$

donde $Z_{i,j} = \|x_i - x_j\|_2$ es la matriz de distancias por pares entre las señales. Esta equivalencia permite formular el problema de aprender el grafo como:

$$\min_{W \in \mathcal{W}_m} \|W \circ Z\|_{1,1} - \alpha \mathbf{1}^T \log(W\mathbf{1}) + \beta \|W\|_F^2 \tag{5}$$

El término logarítmico $-\alpha \mathbf{1}^T \log(W\mathbf{1})$ actúa sobre el vector de grados de los nodos, promoviendo la conectividad del grafo, mientras que el término $\beta\|W\|_F^2$ controla la dispersión de los pesos. Los parámetros $\alpha > 0$ y $\beta \geq 0$ permiten balancear estos objetivos.

En la figura 6 se presentan los resultados obtenidos con diferentes combinaciones de α y β, comparados con la matriz verdadera. Se aprecia cómo estos parámetros influyen en la estructura del grafo aprendido, permitiendo ajustar el balance entre conectividad y dispersión de pesos según las características deseadas.

2.6. Comparación y visualización de los grafos aprendidos

En la figura 7 se presenta una comparación visual entre el grafo verdadero y los grafos aprendidos mediante los métodos de Graphical Lasso, Meinshausen y Bühlmann, y Kalofolias. Cada subfigura muestra la estructura de conexiones entre nodos, permitiendo observar las similitudes y diferencias en la topología inferida por cada método en relación con el grafo original.

La comparación visual de la figura 7 muestra que Graphical Lasso recupera bien la estructura principal del grafo, aunque introduce algunas conexiones espurias; Meinshausen y Bühlmann produce un grafo más disperso, capturando solo las conexiones más fuertes y perdiendo muchas débiles; y Kalofolias logra un balance intermedio, con una topología similar pero no idéntica al grafo original.

3. Inferencia de topología en MNIST

Para esta parte trabajaremos el dataset MNIST, que contiene imágenes de dígitos escritos a mano. El objetivo es aprender la estructura del grafo que conecta las imágenes basándonos en

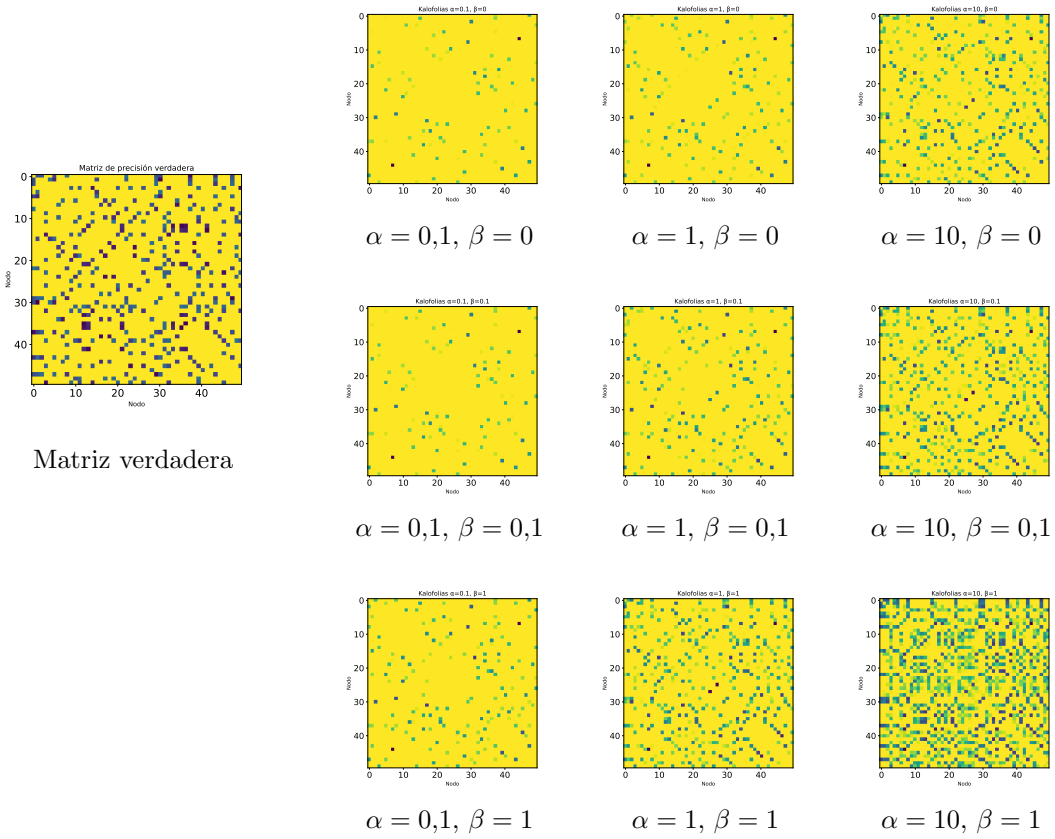


Figura 6: Resultados del método de Kalofolias para distintos valores de α y β, comparados con la matriz verdadera (izquierda) y una grilla 3x3 de variaciones (derecha).

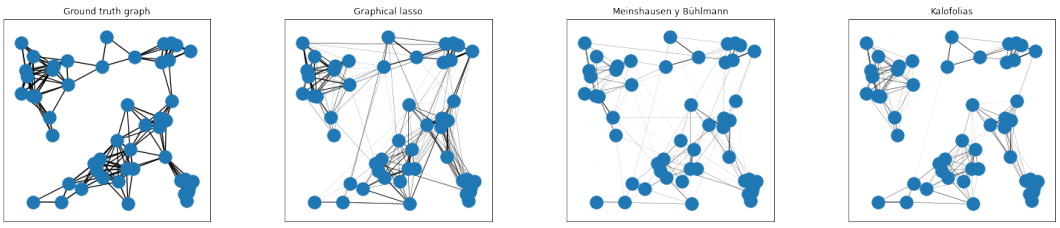
sus características visuales, y luego utilizar esta estructura para realizar clustering espectral y evaluar su desempeño.

3.1. Descripción del dataset y preprocesamiento

Para trabajar, nos quedaremos con un subconjunto de 2000 imágenes (doscientas por cada dígito) de dimensión 28×28 píxeles. Cada imagen la convertimos en un vector de dimensión 784 (aplanando la matriz) y normalizamos los valores de píxeles al rango $[0, 1]$.

Para dimensionar la dificultad del problema, se realiza un análisis preliminar utilizando t-SNE para reducir la dimensionalidad de los datos a 2D y visualizar la distribución de las imágenes. En la figura 8 se observa que los dígitos forman grupos relativamente bien definidos, aunque con cierta superposición, y algunas muestras quedan lejos de su respectivo grupo (*outliers*) lo que indica que el clustering no será trivial.

Se trabajará con cada imagen como un nodo del grafo, y se buscará aprender las conexiones entre ellas basándose en si son similares en algún sentido.



(a) Grafo verdadero (b) Graphical Lasso (c) Meinshausen y Bühlmann (d) Kalofolias

Figura 7: Comparación visual de los grafos: verdadero y aprendidos por distintos métodos.

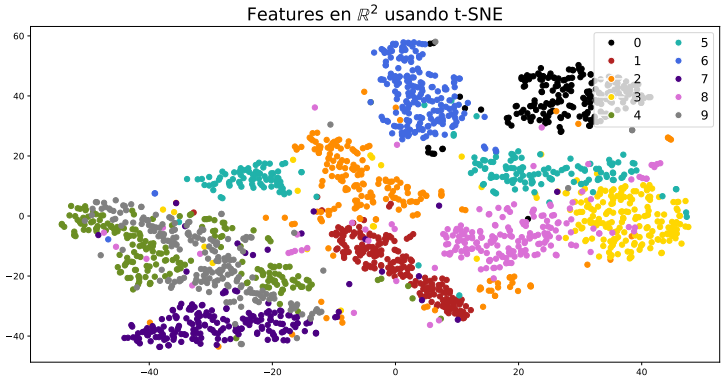


Figura 8: Visualización t-SNE del subconjunto de 2000 imágenes del dataset MNIST.

3.2. Grafo aprendido con Meinshausen y Bühlmann

Utilizando el método de Meinshausen y Bühlmann implementado previamente, aprendemos la estructura del grafo a partir de las imágenes del dataset MNIST. Se muestra en la figura 9 el grafo resultante. Cada dígito se representa como un subgrafo circular para facilitar la visualización. Se puede ver que dentro de cada subgrafo los nodos están densamente conectados, reflejando la similitud entre imágenes del mismo dígito. Además, hay conexiones entre diferentes dígitos, lo que indica que algunas imágenes comparten características visuales a pesar de pertenecer a clases distintas. Por ejemplo, se puede ver una fuerte tendencia a conectar los dígitos 4 y 9 o el 3 y el 8.

3.3. Grafo aprendido con Kalofolias y umbralización

Realizamos el aprendizaje del grafo utilizando el método de Kalofolias. Dado que este método produce una matriz de adyacencia densa, aplicamos una umbralización para eliminar las conexiones más débiles y obtener un grafo más interpretable. En la figura 10 se muestra el grafo resultante tras aplicar Kalofolias y umbralizar los pesos de las aristas. Se puede observar que para cada dígito, los nodos están fuertemente conectados (se distinguen subgrafos más densos en comparación con el método anterior). Sin embargo, la tendencia a conectarse entre dígitos

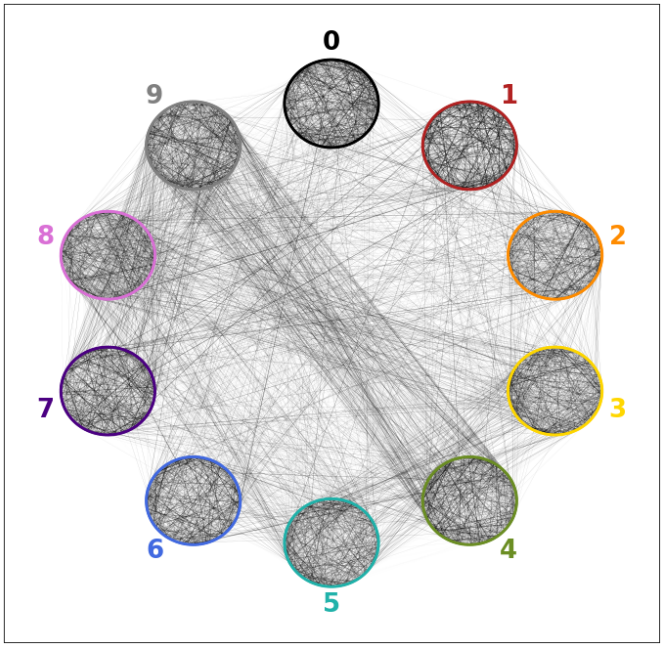


Figura 9: Grafo aprendido con el método de Meinshausen y Bühlmann sobre el dataset MNIST.

distintos todavía se mantiene fuerte para el caso entre el 4 y 9.

3.4. Clustering espectral sobre grafos aprendidos

Una vez obtenidos los grafos mediante los diferentes métodos de aprendizaje, aplicamos clustering espectral para agrupar las imágenes y evaluamos su desempeño utilizando métricas estándar. Los resultados se presentan en la tabla 1, donde se comparan las diferentes estrategias de aprendizaje de grafos junto con el clustering en el espacio original y con representación t-SNE.

Método	**V-Measure**	**Rand**	**Fowlkes-Mallows**
Clustering en espacio original	0.608977	0.414621	0.487145
Meinshausen	0.591794	0.322577	0.432388
Kalofolias	0.704810	0.484090	0.562653
Clustering con t-SNE	0.627884	0.475135	0.532973

Tabla 1: Resultados de clustering espectral para diferentes métodos de aprendizaje de grafos evaluados con métricas V-Measure, Rand Index y Fowlkes-Mallows.

Los resultados muestran que el método de Kalofolias obtiene el mejor desempeño en todas las métricas evaluadas, superando incluso al clustering en el espacio original y con t-SNE. El método de Meinshausen y Bühlmann presenta el desempeño más bajo, lo que sugiere que la estructura de grafo aprendida por este método no captura adecuadamente las relaciones relevantes para el clustering de dígitos. El clustering con t-SNE logra un desempeño intermedio, confirmando que la reducción de dimensionalidad mejora la separabilidad de los grupos pero sin alcanzar los resultados del método de Kalofolias.

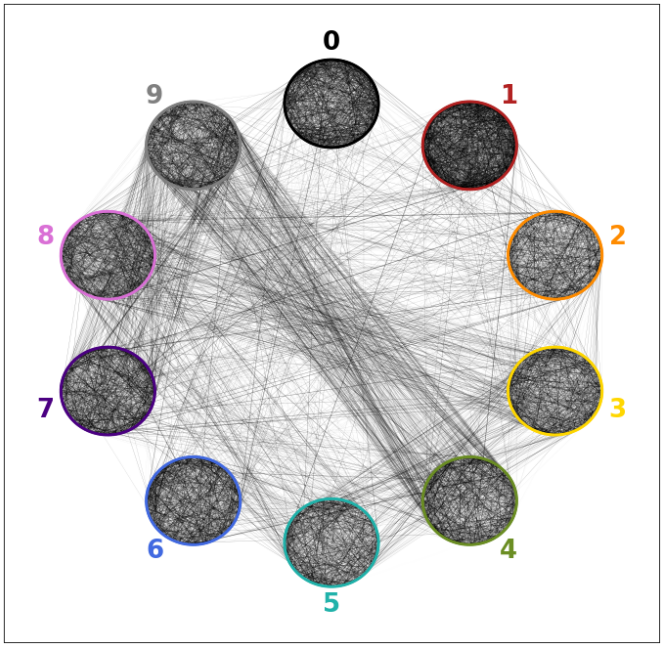


Figura 10: Grafo aprendido con el método de Kalofolias y umbralización sobre el dataset MNIST.

3.5. Visualización en el plano t-SNE por método

Para visualizar cómo los diferentes métodos de aprendizaje de grafos afectan la distribución de las imágenes en el espacio reducido, aplicamos t-SNE a las representaciones obtenidas por cada método. En la figura 11 se presentan las visualizaciones t-SNE correspondientes a cada enfoque.

Analizando las diferencias entre el método de Kalofolias y el clustering con t-SNE, lo primero que salta a la vista es que Kalofolias logra correctamente muestras que se encuentran más aisladas de su cluster principal, mientras que t-SNE asigna clusters más compactos perdiendo las muestras atípicas. Esto sugiere que Kalofolias es más robusto a la presencia de *outliers* y puede capturar mejor la variabilidad dentro de cada clase.

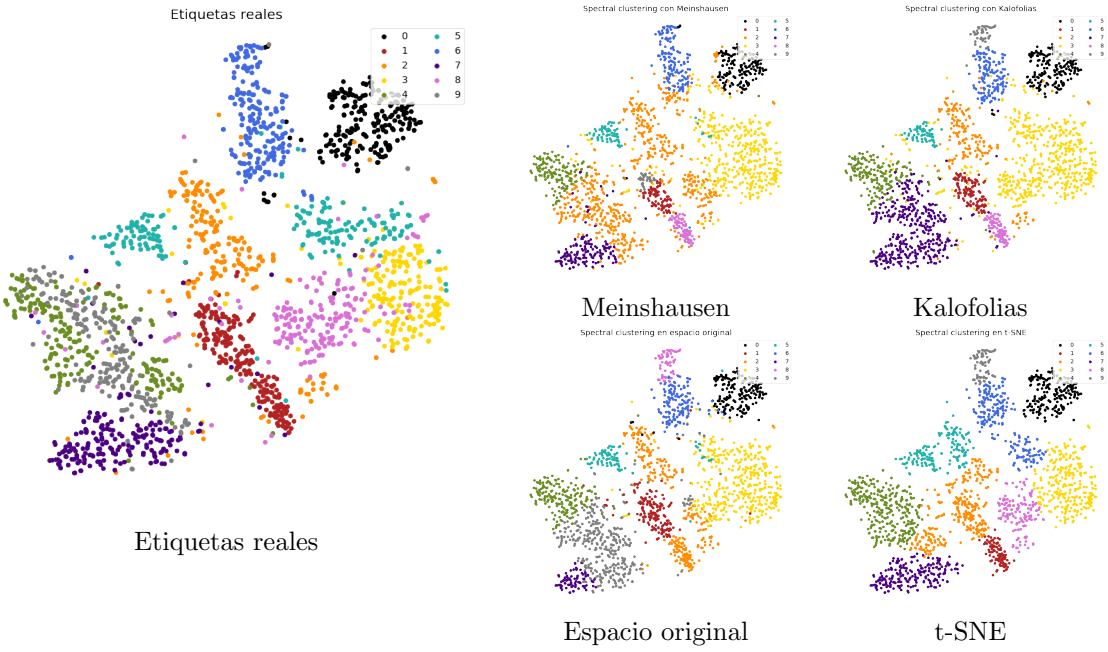


Figura 11: Visualización t-SNE de las imágenes MNIST. Izquierda: etiquetas reales. Derecha: resultados de clustering espectral usando los grafos aprendidos por cada método y en el espacio original.