Diplomado Internacional: Modelado Matemático y Simulaciones

Módulo 3: Fundamentos teóricos de la modelación con grafos

Instructores

Dr. Jesús F. Espinoza & Dra. Rosalía G. Hernández

Departamento de Matemáticas
Universidad de Sonora
México

Descripción del módulo

- **Objetivo:** Establecer las bases teóricas de la teoría de grafos y el análisis topológico de datos (TDA) para su aplicación en modelado y simulación.

- **Estructura:**
  - Sesión 1: Fundamentos de teoría de grafos.
  - Sesión 2: Análisis topológico de datos (TDA, *Topological Data Analysis*).

Parte 1: Teoría de grafos

1. Fundamentos de grafos: ¿Qué es un grafo?

Definición: Grafo simple

Un **grafo simple** G es un par ordenado (V, E), donde:

- V es un conjunto (finito, por conveniencia en este curso), cuyos elementos se llaman **vértices** (o nodos).
- E es un conjunto de pares no ordenados de vértices distintos. Los elementos de E se llaman **aristas** (o enlaces).

Ejemplo 1

Sean:

- $V = \{1, 2, 3, 4\}$
- $E = \{\{1, 2\}, \{1, 3\}, \{2, 3\}, \{3, 4\}\}$

Representación gráfica:

Terminología básica

- **Adyacencia:** Dos vértices $u, v \in V$ son **adyacentes** (o vecinos) si existe una arista que los une, es decir, si $\{u, v\} \in E$.
- **Incidencia:** Una arista $e \in E$ es **incidente** a un vértice $v \in V$ si v es uno de los extremos de e.
- **Orden y tamaño:** El **orden** del grafo es el número de vértices, $|V|$. El **tamaño** del grafo es el número de aristas, $|E|$.

Ejemplo 2

En el ejemplo anterior:

- 1 y 2 son adyacentes. 1 y 4 no son adyacentes.
- La arista $\{1, 2\}$ es incidente a los vértices 1 y 2.
- Orden $|V| = 4$. Tamaño $|E| = 4$.

Variaciones del concepto de grafo

No todos los sistemas se modelan con grafos simples.

- **Grafo dirigido (digrafo):** Las aristas son **pares ordenados** de vértices, (u, v), representando una dirección (de u a v). Se suelen llamar **arcos**.

Ejemplo 3

Variaciones del concepto de grafo

- **Multigrafo:** Se permiten múltiples aristas entre el mismo par de vértices.

Ejemplo 4

Rutas de vuelo entre ciudades:

Variaciones del concepto de grafo

- **Pseudografo:** Se permiten **bucles** o lazos (aristas que conectan un vértice consigo mismo, $\{v, v\}$).

Ejemplo 5

Variaciones del concepto de grafo

- **Grafo ponderado (o red):** A cada arista $e \in E$ se le asigna un valor numérico $w(e)$, llamado **peso**.

Ejemplo 6

Pesos que representan distancias entre vuelos.

Actividad rápida: Modelado con grafos

Pregunta para la audiencia

En los siguientes sistemas, ¿qué representarían los vértices y las aristas?, ¿qué tipo de grafo (simple, dirigido, ponderado, etc.) sería el más adecuado?

1. La red de amigos en Facebook.
2. El sistema de calles de una ciudad.
3. Las dependencias entre tareas en un proyecto.

Actividad rápida: Modelado con grafos

Pregunta para la audiencia

En los siguientes sistemas, ¿qué representarían los vértices y las aristas?, ¿qué tipo de grafo (simple, dirigido, ponderado, etc.) sería el más adecuado?

1. La red de amigos en Facebook.
2. El sistema de calles de una ciudad.
3. Las dependencias entre tareas en un proyecto.

Posibles respuestas

1. Vértices: Personas. Aristas: Amistad (simétrica). Grafo simple (o quizás ponderado por intensidad de interacción).
2. Vértices: Intersecciones. Aristas: Calles. Grafo dirigido (si hay calles de un solo sentido) y ponderado (distancia o tiempo de viaje). Posiblemente multigrafo si consideramos carriles.
3. Vértices: Tareas. Aristas: Dependencia (Tarea A debe terminar antes que B). Grafo dirigido, acíclico.

Subgrafos

Definición: Subgrafo

Un grafo $H = (V_H, E_H)$ es un **subgrafo** de $G = (V_G, E_G)$ si $V_H \subseteq V_G$, $E_H \subseteq E_G$ y todas las aristas en E_H tienen sus extremos en V_H.

Ejemplo 7

Sea G el cuadrado con vértices 1,2,3,4 con una diagonal {1,3}.
Un subgrafo H podría tener vértices $\{1, 2, 3\} \subset V_G$ y solo la arista $\{1, 2\}$.

Subgrafos inducidos

Definición: Subgrafo inducido

Dado un subconjunto de vértices $S \subseteq V_G$, el **subgrafo inducido** por S, denotado $G[S]$, es el grafo (S, E_S) donde E_S contiene todas las aristas de E_G cuyos dos extremos están en S.

Ejemplo 8

Sea G el cuadrado con vértices 1,2,3,4 con una diagonal {1,3} del ejemplo anterior, y sea $S = \{1, 2, 3\} \subset V_G$.
El subgrafo inducido $G[S]$ tiene vértices S y las aristas $\{1, 2\}, \{2, 3\}, \{1, 3\}$ (un triángulo).

Isomorfismo de grafos

¿Cuándo son dos grafos *"esencialmente el mismo"*?

Definición: Isomorfismo

Dos grafos $G_1 = (V_1, E_1)$ y $G_2 = (V_2, E_2)$ son **isomorfos** ($G_1 \cong G_2$) si existe una función biyectiva (uno a uno y sobre) $f : V_1 \to V_2$ tal que:

$$\{u, v\} \in E_1 \Leftrightarrow \{f(u), f(v)\} \in E_2$$

La función f preserva la adyacencia.

Ejemplo 9

Un grafo que es un camino de 3 vértices (1-2-3) y un grafo que es un camino de 3 vértices (A-B-C) son isomorfos.

Isomorfismo de grafos

Ejemplo 10

Un cuadrado (4 vértices, 4 aristas) y un camino de 4 vértices (4 vértices, 3 aristas) NO son isomorfos.

Nota: *Determinar si dos grafos arbitrarios son isomorfos es un problema computacionalmente difícil. La complejidad exacta del problema es abierta; actualmente se conoce un algoritmo cuasipolinomial.*

Grado de un vértice

La medida más básica de la importancia de un nodo.

Definición: Grado

El **grado** de un vértice v en un grafo G, denotado $\deg(v)$ o $d(v)$, es el número de aristas incidentes a v. Equivalentemente, es el número de vecinos de v.

Ejemplo 11

Determinar los grados de los vértices del cuadrado con aristas 1,2,3,4 y diagonal {1,3}

Grado de un vértice

Un resultado fundamental relaciona los grados con el número de aristas.

Teorema 12

Lema del apretón de manos (Handshaking Lemma). Para cualquier grafo $G = (V, E)$, la suma de los grados de todos los vértices es igual al doble del número de aristas:

$$\sum_{v \in V} \deg(v) = 2|E|$$

Idea de la demostración: *Cada arista $\{u, v\}$ contribuye exactamente dos veces a la suma total de grados: una vez en $\deg(u)$ y otra en $\deg(v)$.*

Actividad rápida: Lema del apretón de manos

Ejercicio

¿Es posible construir un grafo simple con 5 vértices que tengan los siguientes grados: 3, 3, 3, 1, 0?

Actividad rápida: Lema del apretón de manos

Ejercicio

¿Es posible construir un grafo simple con 5 vértices que tengan los siguientes grados: 3, 3, 3, 1, 0?

Solución

Calculamos la suma de los grados: $3 + 3 + 3 + 1 + 0 = 10$.
Según el Lema del apretón de manos, la suma debe ser $2|E|$. Como 10 es par, el lema se cumple ($|E| = 5$).
Sin embargo, cumplir el lema no garantiza la existencia del grafo (es condición necesaria pero no suficiente).
Tenemos 5 vértices. Uno tiene grado 0 (aislado). Quedan 4 vértices. En este subgrafo de 4 vértices, los grados deben ser 3, 3, 3, 1.
Si un vértice tiene grado 3 en un grafo de 4 vértices, debe estar conectado a todos los demás. Supongamos V1, V2, V3 tienen grado 3. V1 se conecta a V2, V3, V4. V2 se conecta a V1, V3, V4. V3 se conecta a V1, V2, V4.
Pero entonces, V4 también tiene grado 3 (conectado a V1, V2, V3). Esto contradice que el cuarto vértice debía tener grado 1.
Por lo tanto, la respuesta correcta es: **No es posible**.

Caminos, ciclos y conectividad

- **Camino (simple):** Una sucesión de vértices distintos $v_0, v_1, \ldots, v_k$ tal que $\{v_{i-1}, v_i\}$ es una arista para todo $i = 1, \ldots, k$. La longitud del camino es k.
- **Ciclo (simple):** Un camino con $k \geq 3$ donde el primer y último vértice son el mismo ($v_0 = v_k$) y los vértices $v_1, \ldots, v_{k-1}$ son distintos.

Ejemplo 13

Caminos, ciclos y conectividad

- **Conectividad:** Un grafo es **conexo** si existe un camino entre cualquier par de vértices distintos.
- **Componente conexa:** Es un subgrafo maximal conexo. Un grafo puede tener una o más componentes conexas.

Ejemplo 14

Representaciones de grafos: Lista de adyacencia

¿Cómo almacenamos un grafo en una computadora?

Lista de adyacencia

Para cada vértice, almacenamos una lista de sus vecinos.

Ejemplo 15

Grafo: $E = \{\{1,2\},\{1,3\},\{2,3\}\}$

- 1: [2, 3]
- 2: [1, 3]
- 3: [1, 2]

- **Ventaja:** Eficiente en memoria para grafos **esparsos** (pocos enlaces, $|E| \ll |V|^2$).
- **Desventaja:** Verificar si una arista $\{u, v\}$ existe puede tomar tiempo (hay que buscar en la lista de u).

Representaciones de grafos: Matriz de adyacencia

Definición: Matriz de adyacencia

Sea $G = (V, E)$ con $|V| = n$. La **matriz de adyacencia** A es una matriz $n \times n$ donde:

$$A_{ij} = \begin{cases} 1 & \text{si } \{i, j\} \in E \\ 0 & \text{en otro caso} \end{cases}$$

Ejemplo 16

Grafo: $E = \{\{1, 2\}, \{1, 3\}, \{2, 3\}\}$

$$A = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

- Para grafos simples, A es simétrica y tiene ceros en la diagonal.
- **Ventaja:** Verificar adyacencia es muy rápido (tiempo constante, $O(1)$).
- **Desventaja:** Requiere mucha memoria ($O(|V|^2)$), ineficiente para grafos esparsos.

Representaciones de grafos: Matriz de incidencia

Definición: Matriz de incidencia

Sea $G = (V, E)$ con $|V| = n$ y $|E| = m$. La **matriz de incidencia** B es una matriz $n \times m$ (vértices por aristas) donde:

$$B_{ie} = \begin{cases} 1 & \text{si el vértice } i \text{ es incidente a la arista } e \\ 0 & \text{en otro caso} \end{cases}$$

Ejemplo 17

Grafo: $E = \{e_1 = \{1, 2\}, e_2 = \{1, 3\}, e_3 = \{2, 3\}\}$

$$B = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

Nota: *Cada columna tiene exactamente dos unos.*

El Laplaciano del grafo

Una de las representaciones matriciales más importantes, fundamental en teoría espectral de grafos y aprendizaje automático.
Primero, definimos la **matriz de grados** D. Es una matriz diagonal $n \times n$ donde $D_{ii} = \deg(i)$.

Definición: Matriz Laplaciana

La **matriz Laplaciana** L de un grafo G se define como:

$$L = D - A$$

Ejemplo 18

$$A = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}, D = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix}, L =$$

Propiedades del Laplaciano

El Laplaciano tiene propiedades matemáticas muy interesantes y útiles.

Teorema 19

Propiedades espectrales de L Sea L la matriz Laplaciana de un grafo G.

1. *L es simétrica.*
2. *L es* ***semidefinida positiva****. Esto significa que todos sus autovalores (eigenvalues) λ_i son reales y no negativos ($\lambda_i \geq 0$).*
3. *La suma de cada fila (y columna) de L es 0. Por lo tanto, el vector constante $\mathbf{1} = (1, 1, \ldots, 1)^T$ es un autovector con autovalor 0. Así que siempre $\lambda_1 = 0$.*

Propiedades del Laplaciano

El siguiente resultado conecta la estructura del grafo con el espectro de L.

Teorema 20

La multiplicidad del autovalor 0 de L es igual al número de componentes conexas del grafo G.

Corolario 21

El grafo es conexo si y solo si el segundo autovalor más pequeño (λ_2, llamado conectividad algebraica o valor de Fiedler) es estrictamente positivo ($\lambda_2 > 0$).

Otras matrices y conceptos

- **Laplaciano normalizado:** Hay varias definiciones. Una común es $L_{\text{sym}} = D^{-1/2}LD^{-1/2} = I - D^{-1/2}AD^{-1/2}$ (en grafos sin vértices aislados). Sus autovalores están acotados entre 0 y 2.
- **Diámetro:** La longitud del camino más largo entre todos los caminos más cortos entre pares de vértices.
- Existe una relación entre el espectro del Laplaciano y el diámetro del grafo. Cotas como la cota de Chung establecen que el diámetro está acotado superiormente en función de los autovalores. (No entraremos en detalle, pero es importante saber que el espectro captura propiedades geométricas globales).

2. Estructura básica: Árboles

Los árboles son quizás la clase de grafos más importante, tanto en matemáticas como en ciencias de la computación.

Definición: Árbol y bosque

Un **árbol** es un grafo conexo y acíclico (no contiene ciclos).
Un **bosque** es un grafo acíclico (sus componentes conexas son árboles).

Ejemplo 22

Caracterizaciones de árboles

Existen varias formas equivalentes de definir un árbol.

Teorema 23

Sea $G = (V, E)$ un grafo con n vértices. Las siguientes afirmaciones son equivalentes:

1. *G es un árbol (conexo y acíclico).*
2. *G es conexo y tiene exactamente $n - 1$ aristas ($|E| = |V| - 1$).*
3. *G es acíclico y tiene exactamente $n - 1$ aristas.*
4. *Para cualquier par de vértices $u, v \in V$, existe un **único** camino entre u y v.*
5. *G es mínimamente conexo (si se elimina cualquier arista, el grafo deja de ser conexo).*
6. *G es máximamente acíclico (si se añade cualquier arista nueva, se crea un ciclo).*

Actividad rápida: Identificar árboles

Pregunta para la audiencia

¿Cuáles de los siguientes grafos son árboles?

$\mathbf{G_1}$: $V_1 = \{1, 2, 3, 4, 5\}$,
$E_1 = \{\{1, 2\}, \{2, 3\}, \{3, 4\}, \{4, 5\}\}$

$\mathbf{G_2}$: $V_2 = \{1, 2, 3, 4, 5\}$,
$E_2 = \{\{1, 2\}, \{2, 3\}, \{3, 4\}, \{4, 5\}, \{5, 1\}\}$

$\mathbf{G_3}$: $V_3 = \{1, 2, 3, 4, 5\}$,
$E_3 = \{\{1, 2\}, \{2, 3\}, \{3, 1\}, \{3, 4\}, \{4, 5\}\}$

$\mathbf{G_4}$: $V_4 = \{1, 2, 3, 4, 5, 6\}$,
$E_4 = \{\{1, 2\}, \{2, 3\}, \{4, 5\}\}$

Actividad rápida: Identificar árboles

Solución

- G_1 : **Sí es un árbol** (es conexo y acíclico).
- G_2 : **No es un árbol** (contiene el ciclo $1 - 2 - 3 - 4 - 5 - 1$).
- G_3 : **No es un árbol**. Es conexo pero tiene 5 vértices y 5 aristas ($n = 5$). Un grafo con n vértices y n aristas siempre contiene un ciclo (en este caso, $1 - 2 - 3 - 1$).
- G_4 : **No es un árbol** (no es conexo, ya que no hay camino entre el vértice 1 y el 4, por ejemplo). Se le conoce como *bosque*.

Conectividad: Cortes y puentes

¿Qué tan robusta es la conexión en un grafo?

- **Puente (Bridge):** Una arista cuya eliminación aumenta el número de componentes conexas del grafo. En un grafo conexo, su eliminación lo desconecta.
- **Vértice de corte (Cut vertex o Articulation point):** Un vértice cuya eliminación (junto con sus aristas incidentes) aumenta el número de componentes conexas.

Ejemplo 24

- En un árbol, toda arista es un puente y todo vértice con grado mayor a 1 es un vértice de corte.
- El grafo “mancuerna”(dos triángulos unidos por una sola arista). Esa arista es un puente.

Conectividad: Cortes y puentes

Noción de k-conectividad

Un grafo G es **k-conexo por vértices** (o simplemente k-conexo) si $|V| > k$ y G permanece conexo tras la eliminación de cualquier subconjunto de menos de k vértices. De manera similar se define la **k-conectividad por aristas**.

Esto mide la robustez de la red ante fallos de nodos o enlaces.

Teorema de Menger

Un resultado clásico y profundo sobre conectividad. Relaciona el tamaño mínimo de un corte con el número máximo de caminos disjuntos.

Teorema 25 (Teorema de Menger (versión vértices))

Sean u y v dos vértices no adyacentes en un grafo G. El número mínimo de vértices que se deben eliminar para separar u de v es igual al número máximo de caminos internamente disjuntos (no comparten vértices intermedios) entre u y v.

Ejemplo 26

Nota: *Este teorema es la base de muchos algoritmos de flujo en redes (por ejemplo, Ford-Fulkerson).*

3. Métricas y análisis de redes: Centralidad

En una red, no todos los nodos son igualmente importantes. Las medidas de **centralidad** buscan cuantificar la "importancia" de un nodo según diferentes criterios.

1. Centralidad de grado (Degree Centrality).
2. Centralidad de cercanía (Closeness Centrality).
3. Centralidad de intermediación (Betweenness Centrality).
4. Centralidad de vector propio (Eigenvector Centrality) y PageRank.

Centralidad de grado y cercanía

Centralidad de grado

Simplemente el grado del nodo. Mide popularidad inmediata. $C_D(v) = \deg(v)$.

Centralidad de cercanía (Closeness)

Mide qué tan cerca está un nodo de todos los demás nodos, en un grafo conexo. Es el inverso de la distancia promedio a los demás nodos. Sea $d(u, v)$ la longitud del camino más corto entre u y v.

$$C_C(v) = \frac{1}{\sum_{u \neq v} d(v, u)}$$

(A menudo se normaliza multiplicando por $(N - 1)$).
Interpretación: Nodos centrales para la difusión rápida de información.

Ejemplo 27

Un grafo "estrella" (un nodo central conectado a todos los demás). El nodo central tiene máxima centralidad de grado y de cercanía.

Centralidad de intermediación (Betweenness)

Mide la frecuencia con la que un nodo actúa como "puente" en los caminos más cortos entre otros pares de nodos.
Sea σ_{st} el número total de caminos más cortos entre s y t. Sea $\sigma_{st}(v)$ el número de esos caminos que pasan por v.

$$C_B(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

Interpretación: Nodos que controlan el flujo de información, "guardianes" (gatekeepers).

Ejemplo 28

Un grafo donde dos comunidades están conectadas por un solo nodo. Ese nodo tendrá alta intermediación, aunque quizás no tenga el grado más alto.

Actividad rápida: Comparando centralidades

Discusión

Considere el grafo “CORBATA DE MOÑO”, con vértices {1,2,3,4,5} y aristas que forman el triangulo 1-2-3-1 y el triángulo 3-4-5-3. El vértice 3 es el nudo.
¿Qué nodo esperarían que tenga la mayor centralidad de grado, cercanía e intermediación?

Actividad rápida: Comparando centralidades

Discusión

Considere el grafo "CORBATA DE MOÑO", con vértices {1,2,3,4,5} y aristas que forman el triangulo 1-2-3-1 y el triángulo 3-4-5-3. El vértice 3 es el nudo. ¿Qué nodo esperarían que tenga la mayor centralidad de grado, cercanía e intermediación?

Análisis

Grados: deg(1)=2, deg(2)=2, deg(3)=4, deg(4)=2, deg(5)=2.

- **Grado:** El nodo 3 (máximo grado).
- **Cercanía:** El nodo 3 está a distancia 1 de todos los demás. Los otros están más lejos entre sí (por ejemplo, d(1,5)=2). El nodo 3 es el más cercano.
- **Intermediación:** Para ir de {1, 2} a {4, 5}, se debe pasar por 3. El nodo 3 tiene la máxima intermediación.

Es importante notar que estas medidas no siempre coinciden. Un nodo puede tener bajo grado pero alta intermediación.

Centralidad de vector propio (Eigenvector Centrality)

Centralidad de vector propio

La importancia de un nodo depende de la importancia de sus vecinos. Es una definición recursiva.
La centralidad x_v del vértice v es proporcional a la suma de las centralidades de sus vecinos:

$$x_v = \frac{1}{\lambda} \sum_{u \sim v} x_u$$

En forma matricial, esto es $A\mathbf{x} = \lambda\mathbf{x}$. Es la ecuación de autovalores de la matriz de adyacencia A.

- Se utiliza el autovector asociado al autovalor dominante (el más grande) de A. El Teorema de Perron-Frobenius garantiza que sus componentes son positivas, bajo ciertas condiciones. De hecho, para garantizar que todas las componentes del autovector dominante sean estrictamente positivas, la matriz de adyacencia debe ser irreducible, lo que equivale a que el grafo sea (fuertemente) conexo.

- **Interpretación:** Mide la influencia a largo plazo en la red.
- **Cálculo (Método de potencias):** Se puede calcular iterativamente. Se empieza con un vector inicial $\mathbf{x}^{(0)}$ y se itera $\mathbf{x}^{(k+1)} = A\mathbf{x}^{(k)}$ (normalizando en cada paso). Esto converge al autovector dominante.

PageRank (Google) es una variante de esta idea para grafos dirigidos, añadiendo "saltos aleatorios" para evitar quedar atrapado en sumideros.

Coeficiente de agrupamiento (Clustering Coefficient)

Mide la tendencia de los nodos a formar grupos densamente conectados (la idea de que "mis amigos son amigos entre sí").

Coeficiente de agrupamiento local

Para un vértice v, mide qué fracción de los pares de sus vecinos están conectados entre sí.

$$C(v) = \frac{\text{Número de triángulos que incluyen a } v}{\text{Número de posibles triángulos centrados en } v}$$

El denominador es $\binom{\deg(v)}{2}$. Para $deg(v) < 2$, se define $C(v) := 0$ por convención.

Ejemplo 29

Si un nodo v tiene 4 vecinos, hay $\binom{4}{2} = 6$ posibles conexiones entre ellos. Si solo 3 de esas conexiones existen, $C(v) = 3/6 = 0.5$.

Coeficiente de agrupamiento promedio

Es el promedio de los coeficientes de agrupamiento locales de todos los vértices.

$$C_{global} = \frac{1}{|V|} \sum_{v \in V} C(v)$$

Fenómeno de “pequeño mundo” (Small World)

Muchas redes reales (sociales, biológicas, tecnológicas) exhiben la propiedad de “pequeño mundo”.

Propiedad de pequeño mundo

Una red tiene la propiedad de pequeño mundo si presenta simultáneamente:

1. **Alto coeficiente de agrupamiento global:** Los nodos tienden a formar comunidades locales (como en una red regular o lattice).
2. **Baja distancia promedio (baja longitud de camino promedio):** Es posible ir de cualquier nodo a cualquier otro en pocos pasos (como en una red aleatoria).

Relación cualitativa: El alto clustering refleja la estructura local, mientras que la existencia de “atajos” aleatorios reduce drásticamente las distancias globales.

Detección de comunidades y modularidad

Detección de comunidades

Es la tarea de particionar los vértices del grafo en grupos (comunidades, clusters, módulos) tal que los nodos dentro de un grupo están más densamente conectados entre sí que con nodos fuera del grupo.

Modularidad (Q)

Es una métrica que mide la calidad de una partición dada. Compara la densidad de aristas dentro de las comunidades con la densidad esperada si las aristas se distribuyeran aleatoriamente (manteniendo los grados).

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j)$$

Donde $m = |E|$, k_i es el grado de i, c_i es la comunidad de i, y δ es la delta de Kronecker.

Valores altos de Q (cercanos a 1) indican una fuerte estructura de comunidad. Muchos algoritmos buscan maximizar la modularidad por ejemplo, Louvain, Leiden).

4. Grafos bipartitos

Definición: Grafo bipartito

Un grafo $G = (V, E)$ es **bipartito** si su conjunto de vértices V puede dividirse en dos conjuntos disjuntos V_1 y V_2 ($V = V_1 \cup V_2$, $V_1 \cap V_2 = \emptyset$) tal que toda arista conecta un vértice en V_1 con un vértice en V_2. No hay aristas dentro de V_1 ni dentro de V_2.

Ejemplo 30

- Redes de afiliación (Personas-Organizaciones)
- Mercados (Compradores-Productos)
- Sistemas de recomendación (Usuarios-Ítems)
- Biología (polinizador-planta)

Teorema 31 (Caracterización de grafos bipartitos)

Un grafo es bipartito si y sólo si no contiene ciclos de longitud impar.

Actividad rápida: Identificar grafos bipartitos

Pregunta para la audiencia

¿Cuáles de los siguientes grafos son bipartitos?

G_1: $V_1 = \{1, 2, 3, 4, 5\}$,
$E_1 = \{\{1, 2\}, \{2, 3\}, \{3, 4\}, \{4, 5\}\}$

G_2: $V_2 = \{1, 2, 3, 4, 5\}$,
$E_2 = \{\{1, 2\}, \{2, 3\}, \{3, 4\}, \{4, 5\}, \{5, 1\}\}$

G_3: $V_3 = \{1, 2, 3, 4\}$,
$E_3 = \{\{1, 2\}, \{2, 3\}, \{3, 1\}, \{3, 4\}\}$

G_4: $V_4 = \{1, 2, 3, 4\}$,
$E_4 = \{\{1, 2\}, \{2, 3\}, \{3, 4\}, \{4, 1\}\}$

Solución

- G_1 : **Sí es bipartito**. No tiene ciclos. La partición es $V_A = \{1, 3, 5\}$ y $V_B = \{2, 4\}$.
- G_2 : **No es bipartito**. Es un ciclo de longitud 5 (impar).
- G_3 : **No es bipartito**. Contiene el ciclo impar $1 - 2 - 3 - 1$.
- G_4 : **Sí es bipartito**. Es un ciclo de longitud 4 (par). La partición es $V_A = \{1, 3\}$ y $V_B = \{2, 4\}$.

Representación matricial de grafos bipartitos

Si ordenamos los vértices poniendo primero los de V_1 y luego los de V_2, la matriz de adyacencia tiene una estructura de bloques característica.

Forma matricial por bloques

$$A = \begin{pmatrix} \mathbf{0} & B \\ B^T & \mathbf{0} \end{pmatrix}$$

Donde B es la matriz de **biadyacencia** (tamaño $|V_1| \times |V_2|$) que describe las conexiones entre V_1 y V_2. Las matrices $\mathbf{0}$ reflejan que no hay conexiones internas.

Emparejamientos (Matchings)

Definición: Emparejamiento

Un **emparejamiento** (matching) M en un grafo G es un subconjunto de aristas $M \subseteq E$ tal que no hay dos aristas en M que compartan un vértice común.

- **Emparejamiento máximo:** Un emparejamiento con el mayor número de aristas posible.
- **Emparejamiento perfecto:** Un emparejamiento que cubre todos los vértices del grafo. (Solo posible si $|V|$ es par).

Ejemplo 32

- Asignación de tareas a trabajadores
- Asignación de estudiantes a proyectos

Teorema de Hall

¿Cuándo existe un emparejamiento perfecto en un grafo bipartito? El Teorema de Hall da las condiciones necesarias y suficientes.

Sea $N(S)$ el conjunto de vecinos de un subconjunto de vértices S.

Teorema 33

Sea $G = (V_1 \cup V_2, E)$ un grafo bipartito. Existe un emparejamiento que cubre todos los vértices de V_1 si y sólo si para todo subconjunto $S \subseteq V_1$, se cumple que:

$$|N(S)| \geq |S|$$

Interpretación intuitiva.

Para poder emparejar todos los vértices de S, debe haber al menos $|S|$ vecinos disponibles en V_2. Si esta condición falla para algún S, es imposible encontrar un emparejamiento completo. Lo sorprendente es que si se cumple para todos los S, el emparejamiento siempre existe. □

5. Modelos generativos de grafos

¿Cómo podemos generar grafos que se parezcan a las redes reales? Los *modelos generativos* intentan capturar los mecanismos subyacentes que dan lugar a la estructura observada.

Estudiaremos tres modelos clásicos:

1. Erdős–Rényi (Redes aleatorias)
2. Watts–Strogatz (Pequeño mundo)
3. Barabási–Albert (Libre de escala)

Modelo Erdős–Rényi (ER)

El modelo más simple. Fija el número de vértices n y conecta cada par de vértices independientemente con una probabilidad p.

Modelo $G(n, p)$

Se tienen n vértices. Para cada uno de los $\binom{n}{2}$ posibles pares de vértices, se lanza una moneda con probabilidad p de éxito. Si sale éxito, se añade la arista.

- El número esperado de aristas es $p\binom{n}{2}$.
- La distribución de grados sigue una distribución binomial (aproximadamente Poisson para n grande y p pequeño). La mayoría de los nodos tienen un grado cercano al promedio.
- **Propiedades:** Bajo coeficiente de agrupamiento, diámetro pequeño.
- *Limitación: No captura la estructura de comunidades ni la distribución de grados observada en redes reales.*

Transiciones de fase en $G(n, p)$

El modelo ER exhibe comportamientos interesantes cuando $n \to \infty$, conocidos como transiciones de fase, dependiendo del valor de p.
Consideremos el grado promedio $c = p(n-1)$.

Teorema 34

Umbrales en Erdős–Rényi

- ***Aparición de la componente gigante:*** *Si $c > 1$ (i.e., $p > 1/n$), emerge una* ***componente gigante*** *(una componente conexa que contiene una fracción significativa de los vértices). Si $c < 1$, todas las componentes son pequeñas (tamaño logarítmico).*
- ***Conectividad:*** *Si $p > (\ln n)/n$, el grafo es conexo con alta probabilidad. Si $p < (\ln n)/n$, el grafo está desconectado con alta probabilidad.*

Esto muestra cómo propiedades globales (componente gigante, conectividad) pueden emerger abruptamente al cambiar un parámetro simple (p).

Modelo Watts–Strogatz (WS)

Diseñado para explicar el fenómeno de pequeño mundo (alto clustering, bajo diámetro).

Construcción del modelo WS

1. Empezar con una red regular (lattice): n vértices en un anillo, cada uno conectado a sus k vecinos más cercanos. (Alto clustering, alto diámetro).
2. Reconexión (Rewiring): Para cada arista, con probabilidad β, se reconecta uno de sus extremos a un vértice elegido al azar.

- El parámetro β controla la aleatoriedad.
- Para valores intermedios de β (e.g., 0.01 a 0.1), el modelo mantiene un alto clustering (la estructura local se preserva mayormente) pero el diámetro disminuye drásticamente debido a los "atajos" creados.

Modelo Barabási–Albert (BA)

Diseñado para explicar la distribución de grados observada en muchas redes reales: la ley de potencias (Scale-Free).

Propiedad libre de escala (Scale-Free)

La distribución de grados $P(k)$ sigue una ley de potencias: $P(k) \sim k^{-\gamma}$. Esto significa que la mayoría de los nodos tienen grado bajo, pero existe un número significativo de nodos con grado muy alto (**hubs**).

El modelo BA utiliza el mecanismo de **conexión preferencial** (preferential attachment): "los ricos se hacen más ricos".

Construcción del modelo BA

1. El grafo crece con el tiempo. Empezar con un pequeño grafo inicial.
2. En cada paso, se añade un nuevo vértice con m aristas.
3. Estas aristas se conectan a los vértices existentes con una probabilidad proporcional a su grado actual.

Resultado: Genera redes libres de escala con $\gamma = 3$.

Definición: Caminata aleatoria (Random Walk)

Es un proceso estocástico que describe una trayectoria en un grafo. Desde un vértice actual v, se mueve a un vecino u elegido uniformemente al azar.

La probabilidad de transición de v a u es:

$$P(v,u) = \frac{1}{\deg(v)} \text{ si } u \sim v, \text{ y } 0 \text{ en otro caso.}$$

- Las caminatas aleatorias son fundamentales para muchos algoritmos: PageRank (simula un navegador web aleatorio), algoritmos de muestreo, y más recientemente, para aprender representaciones de nodos (embeddings).

Node Embeddings

Objetivo de Node Embeddings

Aprender una función $f : V \to \mathbb{R}^d$ que mapea cada nodo a un vector de baja dimensión d (por ejemplo, $d = 128$), de tal manera que la similitud en el espacio vectorial (por ejemplo, producto punto) refleje la similitud en el grafo.

¿Por qué? Permite usar técnicas estándar de Machine Learning (clasificación, clustering) sobre los vectores en lugar de trabajar directamente sobre el grafo.

Ideas de DeepWalk y Node2Vec

Estos métodos utilizan caminatas aleatorias para definir la "similitud" entre nodos.

1. Generar muchas caminatas aleatorias cortas empezando desde cada nodo.
2. Los nodos que co-ocurren frecuentemente en las mismas caminatas se consideran similares.

El objetivo tipo Skip-Gram

Inspirado en el procesamiento de lenguaje natural (Word2Vec).

- Tratar las caminatas aleatorias como "oraciones" y los nodos como "palabras".
- El objetivo es aprender embeddings de nodos tal que maximicen la probabilidad de predecir el "contexto" (nodos vecinos en la caminata) dado un nodo central.

$$\max_f \sum_{v \in V} \sum_{c \in \text{Contexto}(v)} \log P(c|f(v))$$

Diferencia clave entre DeepWalk y Node2Vec

- **DeepWalk:** Utiliza caminatas aleatorias uniformes.
- **Node2Vec:** Introduce parámetros (p y q) para sesgar las caminatas aleatorias, permitiendo un balance entre exploración local (BFS-like) y global (DFS-like).

Correspondencia heurística: Se ha demostrado que la optimización de estos objetivos basados en caminatas está relacionada implícitamente con la factorización de ciertas matrices que capturan la proximidad espectral en el grafo.

7. Aprendizaje en grafos: Redes Neuronales de Grafos (GNN)

Los embeddings como Node2Vec son métodos "transductivos": aprenden representaciones para los nodos vistos durante el entrenamiento, pero no pueden generalizar fácilmente a nuevos nodos o grafos.

Las **Redes Neuronales de Grafos (GNNs)** son una clase de modelos de aprendizaje profundo diseñados para operar directamente sobre datos estructurados en grafos.

Tareas comunes en aprendizaje de grafos

- **Clasificación de nodos:** Predecir la etiqueta de un nodo (por ejemplo, detectar usuarios fraudulentos).
- **Predicción de enlaces:** Predecir si existe una arista entre dos nodos (por ejemplo, recomendación de amigos).
- **Clasificación de grafos:** Predecir una propiedad de todo el grafo (por ejemplo, determinar si una molécula es tóxica).

Desafíos del aprendizaje en grafos

¿Por qué no podemos usar redes neuronales estándar (MLP) o convolucionales (CNN)?

1. **Estructura irregular:** A diferencia de las imágenes (cuadrículas) o el texto (secuencias), los grafos tienen una estructura topológica arbitraria. El número de vecinos varía.
2. **Invariancia a permutaciones:** Un grafo puede representarse de muchas maneras (e.g., cambiando el orden de los nodos en la matriz de adyacencia). El modelo debe producir el mismo resultado independientemente del orden.

Requisito de invariancia/equivariancia a permutaciones

Una GNN debe respetar la simetría del grafo. Si P es una matriz de permutación:

- Clasificación de grafos (Invariancia): $\text{GNN}(PAP^T) = \text{GNN}(A)$.
- Clasificación de nodos (Equivariancia): $\text{GNN}(PAP^T) = P \cdot \text{GNN}(A)$.

El esquema de paso de mensajes (Message Passing)

La mayoría de las GNNs modernas siguen el paradigma de **paso de mensajes**. Los nodos intercambian información iterativamente con sus vecinos para actualizar sus propias representaciones (embeddings).
Una capa de GNN consta de dos fases:

1. **Agregación (AGGREGATE):** Cada nodo v recolecta información (mensajes) de su vecindario $N(v)$.

$$m_v^{(k)} = \text{AGGREGATE}^{(k)}\left(\{h_u^{(k-1)} : u \in N(v)\}\right)$$

2. **Actualización (UPDATE/COMBINE):** El nodo v actualiza su representación $h_v^{(k)}$ combinando su representación anterior $h_v^{(k-1)}$ con la información agregada $m_v^{(k)}$.

$$h_v^{(k)} = \text{UPDATE}^{(k)}\left(h_v^{(k-1)}, m_v^{(k)}\right)$$

La función de agregación debe ser invariante a permutaciones (por ejemplo: suma, promedio, máximo).
Después de K capas, la representación de un nodo captura información de su vecindario a K saltos de distancia.

Variantes de GNN: GCN y GraphSAGE

Diferentes modelos definen las operaciones de AGGREGATE y UPDATE de distintas maneras.

Red Convolucional de Grafos (GCN) (Kipf & Welling, 2017)

Utiliza una forma de agregación basada en el promedio normalizado por los grados.

$$h_v^{(k)} = \sigma\left(W^{(k)} \sum_{u \in N(v) \cup \{v\}} \frac{1}{\sqrt{\deg(u)\deg(v)}} h_u^{(k-1)}\right)$$

(σ es una función de activación, W son pesos aprendibles).

Variantes de GNN: GCN y GraphSAGE

GraphSAGE (Hamilton et al., 2017)

Proporciona un marco más flexible para la agregación (Mean, LSTM, Pooling). Ejemplo con agregador Mean:

$$h_v^{(k)} = \sigma\left(W^{(k)} \cdot \text{CONCAT}\left(h_v^{(k-1)}, \text{MEAN}(\{h_u^{(k-1)} : u \in N(v)\})\right)\right)$$

PinSage (usado en Pinterest) *es una extensión de GraphSAGE altamente escalable para grafos masivos.*

Conexión con el Laplaciano (Convolución espectral)

¿Por qué se llaman "convolucionales" ?

- La operación de convolución en el procesamiento de señales se define en el dominio de Fourier.
- En grafos, la transformada de Fourier se define utilizando la descomposición espectral del Laplaciano ($L = U\Lambda U^T$).
- La convolución espectral implica filtrar las señales del grafo en el dominio espectral.

Idea clave

La capa GCN original (Kipf & Welling) puede derivarse como una aproximación de primer orden de la convolución espectral.
Esencialmente, la operación de GCN actúa como un filtro paso bajo sobre el grafo, suavizando las representaciones de los nodos a través de sus vecindarios (relacionado con el operador Laplaciano L).

Cierre de la Sesión 1

Hasta ahora hemos analizado los grafos usando **métricas locales** (grados, clustering) y globales (diámetro, conectividad).

La topología algebraica ofrece herramientas para caracterizar la "forma" del grafo de una manera más robusta.

Pregunta clave

¿Cómo podemos cuantificar el número de "agujeros" o "ciclos" en un grafo de manera sistemática?

Esto nos lleva al concepto de *homología*, que veremos en detalle en la Sesión 2.

Fin de la Sesión 1

Parte 2: Análisis Topológico de Datos (TDA)

8. Introducción al Análisis Topológico de Datos (TDA)

Motivación

Los datos modernos suelen ser de alta dimensión, ruidosos y complejos. El TDA proporciona herramientas matemáticas basadas en la topología algebraica para descubrir la "forma" subyacente de los datos de manera robusta.

Principios del TDA

1. **Invarianza topológica:** Las propiedades topológicas se preservan bajo homeomorfismos (deformaciones continuas sin roturas).
2. **Estabilidad (enunciado informal):** Para construcciones adecuadas (por ejemplo, Čech/Vietoris-Rips o subniveles de funciones "tame"), pequeñas perturbaciones de los datos producen cambios acotados en los descriptores (diagramas/códigos de barras).
3. **Representaciones comprimidas:** La homología persistente resume estructuras en descriptores concisos (diagramas o barcodes).

Flujo de trabajo del TDA

Datos (nube de puntos) $\longrightarrow$ *Complejo simplicial (filtración)* $\longrightarrow$ *Homología persistente* $\longrightarrow$ *Descriptores topológicos (diagramas o barcodes)*

9. Simplejos: Los bloques de construcción

Los simplejos son generalizaciones de triángulos a dimensiones superiores.

Definición: k-simplejo (geométrico)

Un **k-simplejo** σ en $\mathbb{R}^d$ $(0 \leq k \leq d)$ es la envolvente convexa de $k+1$ puntos afínmente independientes en $\mathbb{R}^d$. *(Intuitivamente: los puntos no deben estar en un subespacio de dimensión menor. Por ejemplo, 3 puntos no deben ser colineales).*

- 0-simplejo: Punto (vértice).
- 1-simplejo: Segmento de línea (arista).
- 2-simplejo: Triángulo (relleno).
- 3-simplejo: Tetraedro (sólido).

Ejemplo 35

0-simplejo 1-simplejo 2-simplejo 3-simplejo

- **Cara:** Si $\sigma = \{v_0, \ldots, v_k\}$, entonces cualquier subconjunto **no vacío** $\tau \subseteq \sigma$ define una cara τ de σ.

Ejemplo 36

cara de un 1-simplejo	cara de un 2-simplejo	cara de un 3-simplejo

Complejos simpliciales

Un complejo simplicial es una colección de simplejos pegados de manera coherente.

Definición: Complejo simplicial (geométrico)

Un **complejo simplicial** K es una colección (finita o localmente finita) de simplejos en $\mathbb{R}^d$ tal que:

1. Si $\sigma \in K$, entonces toda cara de σ también pertenece a K.
2. Si $\sigma_1, \sigma_2 \in K$, entonces $\sigma_1 \cap \sigma_2$ es vacía o una cara común de ambas.

Ejemplo 37

- Dos triángulos unidos por una arista forman un complejo simplicial
- Dos triángulos que se intersecan parcialmente, **no** forman un complejo simplicial

Complejos simpliciales abstractos

Podemos definir complejos simpliciales de manera puramente combinatoria, sin referencia a una realización geométrica.

Definición: Complejo simplicial abstracto (C.S.A.)

Un complejo simplicial abstracto es un par (V, K), donde V es un conjunto de vértices y la estructura simplicial K es una colección de subconjuntos finitos no vacíos de V (llamados *simplejos*) tal que:

1. Para todo $v \in V$, se tiene que $\{v\} \in K$ (los vértices son simplejos).
2. Si $\sigma \in K$ y $\tau \subseteq \sigma$ con $\tau \neq \emptyset$, entonces $\tau \in K$ (propiedad hereditaria: cerrado bajo toma de caras).

Relación: Todo complejo geométrico induce un C.S.A. subyacente. Todo C.S.A. finito admite una *realización geométrica* $|K|$ y se puede encajar en $\mathbb{R}^m$ para algún m suficientemente grande.

Complejos simpliciales abstractos

Ejemplo 38

Sean $V = \{1, 2, 3\}$ y $K = \{\{1\}, \{2\}, \{3\}, \{1, 2\}, \{2, 3\}\}$. La realización geométrica de este C.S.A. corresponde a dos aristas unidas en el vértice 2.

Si añadimos $\{1, 3\}$, el par (V, K) sigue siendo un C.S.A., pero ahora tenemos un ciclo hueco.

Si además añadimos $\{1, 2, 3\}$, el nuevo C.S.A. es un triángulo relleno.

Actividad rápida: Complejos simpliciales

Ejercicio

Considere el complejo simplicial abstracto:
$K = \{\{a\}, \{b\}, \{c\}, \{d\}, \{a, b\}, \{b, c\}, \{c, a\}, \{a, d\}\}$.

1. ¿Cuál es la dimensión de K?
2. ¿Contiene algún 2-simplejo (triángulo)?
3. Describa la forma de K.

Recordatorio: La **dimensión** de K es $\max\{k : K \text{ contiene un } k\text{-simplejo}\}$.

Actividad rápida: Complejos simpliciales

Solución

1. La dimensión máxima de los simplejos es 1 (aristas). Dimensión de K es 1.
2. No. Para tener el 2-simplejo $\{a, b, c\}$, necesitaríamos que $\{a, b, c\} \in K$, pero no está.
3. Es un ciclo formado por las aristas {a,b}, {b,c}, {c,a}, y una arista adicional {a,d} que sale del ciclo.

El Teorema del Nervio (Nerve Theorem)

¿Cómo construimos complejos simpliciales a partir de datos? A menudo usamos cubiertas por subconjuntos abiertos del espacio.

Definición: Nervio de una cubierta

Sea $\mathcal{U} = \{U_i\}_{i \in I}$ una cubierta por abiertos de un espacio topológico X (una colección de conjuntos cuya unión es X). El **nervio** de $\mathcal{U}$, denotado $N(\mathcal{U})$, es un complejo simplicial abstracto donde:

- Los vértices son los índices I.
- Un subconjunto $\{i_0, \ldots, i_k\}$ forma un k-simplejo si la intersección de los conjuntos correspondientes es no vacía: $U_{i_0} \cap \cdots \cap U_{i_k} \neq \emptyset$.

El Teorema del Nervio (Nerve Theorem)

Teorema 39

Teorema del Nervio (Borsuk). Si la cubierta $\mathcal{U}$ es "buena" (good cover), es decir, todos los conjuntos U_i y todas sus intersecciones finitas no vacías son contractibles, entonces el nervio $N(\mathcal{U})$ tiene el mismo tipo de homotopía que la unión $\bigcup_{i \in I} U_i$.

Ejemplo de cubierta buena: bolas convexas en $\mathbb{R}^d$ cuyas intersecciones finitas son también convexas (y por tanto contractibles).

Importancia: Permite pasar de geometría continua a estructura combinatoria preservando la topología.

10. Filtraciones: De los datos al complejo

En TDA, rara vez trabajamos con un solo complejo simplicial. En su lugar, construimos una secuencia de complejos que crecen, llamada filtración.

Motivación

Si tenemos una nube de puntos, ¿cómo construimos un complejo que capture su forma? Si conectamos puntos muy cercanos, perdemos estructura global. Si conectamos puntos muy lejanos, obtenemos un complejo demasiado denso. La solución es analizar todas las escalas.

Filtraciones

Definición: Filtración

Una **filtración** de un complejo simplicial K es una sucesión anidada de subcomplejos:

$$K_0 \subseteq K_1 \subseteq K_2 \subseteq \cdots \subseteq K_m = K$$

Ejemplo 40

Filtraciones a partir de nubes de puntos

Dado un conjunto de puntos en un espacio métrico (e.g., $\mathbb{R}^d$).

Idea general: Filtración por distancia

1. Elegir un parámetro de escala $\epsilon \geq 0$.
2. Construir un complejo simplicial basado en la proximidad definida por ϵ.
3. Aumentar ϵ para obtener la filtración.

Dos construcciones clásicas son el complejo de Čech y el complejo de Vietoris–Rips.

Complejo de Čech

Definición: Complejo de Čech

Dado un conjunto de puntos X y un radio $r \geq 0$. Sea $\mathcal{B}_r(X)$ la colección de bolas de radio r centradas en los puntos de X. El **complejo de Čech** $C_r(X)$ es el nervio de esta cubierta de bolas: un k-simplejo $\{x_0, \ldots, x_k\}$ existe si las $k+1$ bolas de radio r tienen una intersección común no vacía.

- **Propiedad:** Por el Teorema del Nervio, $C_r(X)$ captura correctamente la topología de la unión de las bolas.
- **Desventaja:** Computacionalmente costoso de construir.

Ejemplo 41

Complejo de Vietoris–Rips (VR)

Definición: Complejo de Vietoris–Rips

Dado un conjunto de puntos X y un radio $r \geq 0$. El **complejo de Vietoris–Rips** $VR_r(X)$ se define como: un k-simplejo $\{x_0, \ldots, x_k\}$ existe si todas las distancias por pares $d(x_i, x_j)$ son $\leq 2r$. (i.e., las bolas de radio r se intersecan por pares).

- **Ventaja:** Computacionalmente eficiente (solo depende de distancias por pares).
- **Desventaja:** Puede no capturar correctamente la topología de la unión de bolas (puede "rellenar" agujeros prematuramente).

Ejemplo 42

Comparación entre Čech y Vietoris-Rips

Teorema 43

Relación de Intercalado (Interleaving). Para cualquier radio $r > 0$, se tienen las siguientes inclusiones (válidas en cualquier espacio métrico):

$$C_r(X) \subseteq VR_r(X) \subseteq C_{2r}(X)$$

Nota: En espacios euclidianos existen cotas más ajustadas (e.g., $VR_r(X) \subseteq C_{\alpha r}(X)$ con $\alpha < 2$), pero la relación anterior es universal.

En la práctica, VR es el más utilizado debido a su eficiencia, y el intercalado garantiza que aproxima a Čech (con un factor de 2).

Filtración de VR y de Čech

Al aumentar r desde 0 hasta infinito, obtenemos la filtración de Vietoris-Rips (o Čech).

Ejemplo 44

Otras filtraciones

Podemos cambiar la métrica en $\mathbb{R}^d$ para generar distintas filtraciones de Vietoris-Rieps o de Čech.

No solo usamos distancia:

- **Filtraciones por densidad:** Si queremos enfocarnos en regiones densas de los datos, podemos modificar la escala ϵ localmente en función de la densidad estimada.
- **Filtraciones de cliques en grafos ponderados:** Si tenemos un grafo con pesos en las aristas (e.g., correlaciones), podemos crear una filtración umbralizando los pesos.
- **Alpha-complexes:** Basados en Delaunay/Alpha-shapes; capturan mejor la geometría con menor complejidad que Čech.

11. Homología: Contando agujeros formalmente

La homología es la herramienta de la topología algebraica para contar formalmente el número de agujeros de diferentes dimensiones en un espacio topológico (en nuestro caso, un complejo simplicial). Para definirla, necesitamos maquinaria algebraica.

Trabajamos usualmente con coeficientes en el campo finito $\mathbb{Z}_2$ (enteros módulo 2), lo que simplifica los cálculos y evita problemas de orientación.

Grupos de cadenas

Grupos de cadenas (Chain groups)

Sea K un complejo simplicial. El **k-ésimo grupo de cadenas** $C_k(K)$ es el espacio vectorial generado por los k-simplejos de K.
Un elemento de $C_k(K)$ (una k-cadena) es una suma formal de k-simplejos:

$$c = \sum_i a_i \sigma_i$$

con coeficientes $a_i \in \mathbb{Z}_2$ (es decir, a_i es 0 o 1).

Interpretación con $\mathbb{Z}_2$: Una k-cadena es simplemente un subconjunto de k-simplejos.

Ejemplo 45

Para el C.S.A. 2-dimensional

$$K = \{\{a\},\{b\},\{c\},\{d\},\{a,b\},\{a,c\},\{a,d\},\{b,c\},\{b,d\},\{c,d\},\{a,b,d\},\{a,c,d\}\}$$

la suma formal $c = \{a,b\} + \{b,d\} + \{c,d\}$ es una 1-cadena.

El operador frontera (Boundary operator)

El operador frontera conecta los grupos de cadenas de diferentes dimensiones.

Definición: Operador frontera (con coeficientes $\mathbb{Z}_2$)

El **operador frontera** $\partial_k : C_k(K) \to C_{k-1}(K)$ es una transformación lineal que mapea un k-simplejo a la suma (módulo 2) de sus caras de dimensión $(k-1)$.
Para un k-simplejo $\sigma = \{v_0, \ldots, v_k\}$:

$$\partial_k(\sigma) = \sum_{i=0}^{k} \{v_0, \ldots, \hat{v}_i, \ldots, v_k\}$$

Donde $\hat{v}_i$ significa omitir el vértice v_i.

Ejemplo 46

- Sea $\sigma = \{a, c, d\} \in K$ el 2-simplejo del C.S.A. K del ejemplo anterior. Observe que σ (triángulo relleno) es una 2-cadena (es de hecho un generador de $C_2(K)$). Su frontera es la 1-cadena $\partial_2(\sigma) = \{a, c\} + \{c, d\} + \{a, d\}$ (la suma formal de sus aristas frontera).
- Para $\tau = \{1, 2\} \in K$ (una arista), su frontera es la 0-cadena $\partial_1(\tau) = \{1\} + \{2\}$ (la suma formal de sus vértices frontera).

El complejo de cadenas y la propiedad fundamental

La sucesión de grupos de cadenas y operadores frontera forma un **complejo de cadenas**:

$$\ldots \xrightarrow{\partial_{k+1}} C_k(K) \xrightarrow{\partial_k} C_{k-1}(K) \xrightarrow{\partial_{k-1}} \ldots \xrightarrow{\partial_1} C_0(K) \xrightarrow{\partial_0} 0$$

Teorema 47

(Propiedad fundamental). La composición de dos operadores frontera consecutivos es cero: $\partial_{k-1} \circ \partial_k = 0$; es decir, $\partial^2 = 0$.

Ejemplo 48

Para $\sigma = \{a, c, d\}$ del ejemplo anterior, por linealidad de los operadores frontera,

$$\begin{aligned} \partial_1(\partial_2(\{a,c,d\})) &= \partial_1(\{a,c\} + \{c,d\} + \{a,d\}) \\ &= (\{a\} + \{c\}) + (\{c\} + \{d\}) + (\{a\} + \{d\}) \\ &= 2\{a\} + 2\{b\} + 2\{c\} \end{aligned}$$

En $\mathbb{Z}_2$, esto es $0 + 0 + 0 = 0$.

Ciclos y Fronteras

Ahora podemos definir formalmente qué es un ciclo y qué es una frontera.

Ciclos (Cycles)

Una k-cadena c es un **k-ciclo** si su frontera es cero: $\partial_k(c) = 0$. El conjunto de k-ciclos forma el **grupo de ciclos** $Z_k(K) := \text{Ker}(\partial_k)$.

Fronteras (Boundaries)

Una k-cadena b es una **k-frontera** si es la frontera de alguna $(k+1)$-cadena d: $b = \partial_{k+1}(d)$. El conjunto de k-fronteras forma el **grupo de fronteras** $B_k(K) := \text{Im}(\partial_{k+1})$.

Consecuencia de $\partial^2 = 0$: Toda frontera es un ciclo. $B_k(K) \subseteq Z_k(K)$.

Grupos de homología

Los ciclos detectan posibles agujeros. Pero algunos ciclos son triviales porque son la frontera de algo (están "rellenos").
Queremos identificar los ciclos que **NO** son fronteras. Estos representan los verdaderos agujeros topológicos.

Definición: Grupo de homología

El **k-ésimo grupo de homología** $H_k(K)$ se define como el grupo cociente (espacio vectorial cociente) de los ciclos módulo las fronteras:

$$H_k(K) = \frac{Z_k(K)}{B_k(K)} = \frac{\text{Ker}(\partial_k)}{\text{Im}(\partial_{k+1})}$$

- Los elementos de $H_k(K)$ se llaman **clases de homología**. Dos ciclos son homólogos si su diferencia es una frontera.
- $H_k(K)$ captura las características topológicas de dimensión k que no pueden ser trivializadas.

Números de Betti

Definición: Números de Betti

El **k-ésimo número de Betti** $\beta_k(K)$ es la dimensión (rango) del k-ésimo grupo de homología:

$$\beta_k(K) = \dim(H_k(K))$$

Teorema 49

Interpretación de los números de Betti

- β_0: *Número de componentes conexas.*
- β_1: *Número de bucles o lazos independientes (agujeros 1D).*
- β_2: *Número de cavidades o vacíos independientes (agujeros 2D).*

Números de Betti

Ejemplo 50

Toro (la "dona" hueca):

- $\beta_0 = 1$ (es conexo),
- $\beta_1 = 2$ (tiene dos bucles independientes: uno alrededor del agujero central, otro a través del tubo),
- $\beta_2 = 1$ (tiene una cavidad interna).

12. Topología de grafos (puente al TDA)

Definición: Complejo de cliques

Dado un grafo $G = (V, E)$, el **complejo de cliques** (o *flag complex*) $Cl(G)$ es el complejo simplicial abstracto donde:

- Los vértices son V.
- Un subconjunto $\sigma \subseteq V$ es un simplejo en $Cl(G)$ si y solo si σ forma un clique, esto es, todos los pares de vértices en σ están conectados por aristas.

Notemos que $Cl(G)$ es un C.S.A. porque si un conjunto forma un clique, cualquier subconjunto también forma un clique, de modo que es cerrado bajo contenciones.

Complejos de cliques (Flag complex)

Ejemplo 51

Cuadrado con vértices $\{1, 2, 3, 4\}$ y diagonal $\{1, 3\}$.

Cliques:

- vértices $\{1\}, \{2\}, \{3\}, \{4\}$;
- aristas $\{1, 2\}, \{2, 3\}, \{3, 4\}, \{4, 1\}, \{1, 3\}$;
- triángulos $\{1, 2, 3\}$ y $\{1, 3, 4\}$.

El complejo de cliques $Cl(G)$ contiene **ambos** triángulos rellenos $\{1, 2, 3\}$ y $\{1, 3, 4\}$, además de las caras inferiores correspondientes.

Números de Betti e interpretación

Ejemplo 52

Ejemplos canónicos

- **Grafo: Árbol.** Un árbol es conexo ($\beta_0 = 1$) y no tiene ciclos ($\beta_1 = 0$). El complejo de cliques de un árbol coincide con el propio grafo como 1-complejo.
- **Grafo: Ciclo simple (C_n).** Es conexo ($\beta_0 = 1$). Intuitivamente, tiene un agujero. $\beta_1 = 1$. El complejo de cliques (si $n > 3$) sigue siendo el ciclo sin rellenar.
- **Grafo: Triángulo (K_3).** Conexo ($\beta_0 = 1$). Aunque tiene un ciclo, en el complejo de cliques este ciclo está "relleno" por el clique {1,2,3}. Por lo tanto, no hay agujero topológico. $\beta_1 = 0$.

Esta distinción entre ciclos combinatorios y agujeros topológicos es crucial en TDA.

Complejo de cliques con pesos

Funciones de peso/escala

En aplicaciones, los vértices o aristas pueden tener pesos (e.g., tiempos, distancias). Podemos incorporar esta información en el complejo simplicial asignando valores a los simplejos. Esto será crucial para las filtraciones.

13. Homología persistente: La herramienta central de TDA

Hemos aprendido a calcular la homología de un complejo simplicial fijo. Pero en TDA tenemos una filtración, una sucesión de complejos que crecen.

$$K_0 \subseteq K_1 \subseteq K_2 \subseteq \cdots \subseteq K_m$$

Objetivo de la homología persistente (PH)

Rastrear cómo cambian los grupos de homología a medida que avanzamos en la filtración. Queremos saber cuándo aparece una característica topológica (nace) y cuándo desaparece (muere).

Idea clave: Las características topológicas que persisten durante un largo intervalo de la filtración son consideradas "reales" o significativas, mientras que las que aparecen y desaparecen rápidamente son consideradas "ruido".

Nacimiento y muerte de clases de homología

A medida que añadimos simplejos en la filtración (e.g., al aumentar ϵ en la filtración de Vietoris-Rips):

- **Nacimiento:** Un nuevo ciclo independiente puede formarse. Una nueva clase de homología nace en $H_k(K_i)$.
- **Muerte:** Un ciclo existente puede convertirse en una frontera (se rellena). La clase de homología muere en $H_k(K_j)$ (se fusiona con la clase trivial).

Ejemplo 53

- Nacimiento de la última arista que completa un bucle.
- Muerte de un bucle, pues se añade un triángulo que lo rellena.

Persistencia

Persistencia

Para cada característica topológica, registramos su tiempo (escala) de nacimiento b y su tiempo de muerte d. La **persistencia** es la diferencia $d - b$.

Ejemplo 54

Códigos de barras (Barcodes)

La homología persistente de una filtración se resume en una colección de intervalos $[b, d)$, llamados códigos de barras.

Definición: Código de barras

Es una representación visual de la homología persistente. Cada barra corresponde a una clase de homología independiente, indicando su intervalo de existencia a lo largo del parámetro de filtración.

Códigos de barras (Barcodes)

Ejemplo 55

- En H_0, las barras representan componentes conexas. Todas nacen en tiempo 0 (si empezamos con puntos individuales). A medida que ϵ aumenta, las componentes se fusionan (mueren). Una barra persistirá hasta el final (la componente final).
- En H_1, las barras representan bucles.

Diagramas de persistencia (Persistence Diagrams)

Una representación alternativa y equivalente a los códigos de barras.

Definición: Diagrama de persistencia

Es una colección de puntos en el plano 2D, donde cada punto (b, d) corresponde a una barra $[b, d)$.

Diagramas de persistencia (Persistence Diagrams)

Ejemplo 56

- Todos los puntos se sitúan *en o por encima* de la diagonal ($d \geq b$).
- Los puntos lejos de la diagonal corresponden a características de alta persistencia (significativas).
- Los puntos cerca de la diagonal corresponden a características de baja persistencia (ruido).

Actividad rápida: Interpretación de diagramas

Pregunta para la audiencia

Considere el siguiente diagrama de persistencia para H_1. ¿Qué nos dice sobre la estructura de los datos?

Actividad rápida: Interpretación de diagramas

Pregunta para la audiencia

Considere el siguiente diagrama de persistencia para H_1. ¿Qué nos dice sobre la estructura de los datos?

Interpretación

- Los dos puntos lejos de la diagonal indican la presencia de dos bucles (agujeros 1D) muy prominentes y persistentes en los datos.
- Los puntos cerca de la diagonal representan pequeños bucles que se forman y cierran rápidamente debido al ruido o a la textura local de los datos.

Estabilidad de la homología persistente

Una de las razones principales del éxito de TDA es la robustez de la homología persistente.

Idea de estabilidad

Pequeñas perturbaciones en los datos de entrada (e.g., ruido, errores de medición) solo causan pequeñas perturbaciones en el diagrama de persistencia resultante.

Para formalizar esto, necesitamos una métrica para comparar diagramas de persistencia.

Estabilidad de la homología persistente

Distancia Bottleneck (Cuello de botella)

Mide la distancia máxima entre los puntos correspondientes en dos diagramas de persistencia, permitiendo que algunos puntos se mapeen a la diagonal.

Teorema 57

Teorema de estabilidad. La distancia bottleneck entre diagramas de persistencia es estable respecto a perturbaciones en la entrada.

Para funciones *(e.g., filtraciones por densidad o subnivel):*

$$d_B(Dgm(f), Dgm(g)) \leq \|f - g\|_\infty$$

Para nubes de puntos *(e.g., filtración de Vietoris-Rips):*

$$d_B(Dgm(X), Dgm(Y)) \leq 2 \cdot d_H(X, Y)$$

Donde d_H es la distancia de Hausdorff entre las nubes de puntos X e Y.

Esto garantiza formalmente la robustez del TDA al ruido.

14. Vectorización para aprendizaje automático

Los diagramas de persistencia son objetos matemáticos útiles, pero no pueden usarse directamente como entrada para la mayoría de los algoritmos de Machine Learning (que esperan vectores de características).
Necesitamos transformar los diagramas en vectores.

- **Imágenes de persistencia (Persistence Images):** Discretizar el diagrama en una cuadrícula (imagen) aplicando un kernel gaussiano sobre cada punto.
- **Paisajes de persistencia (Persistence Landscapes):** Transformar el diagrama en una secuencia de funciones continuas y suaves.
- **Siluetas (Silhouettes):** Funciones similares a los paisajes, pero ponderadas por la persistencia.

Propiedades de los paisajes

Los paisajes son funciones en un espacio de Banach, lo que permite realizar estadística (promedios, varianzas) y usar algoritmos de ML. Además, la transformación es estable.

Métricas entre diagramas y estadística en TDA

Ya mencionamos la distancia Bottleneck. Es robusta pero computacionalmente costosa y difícil de usar en optimización.

Distancias de Wasserstein

La distancia p-Wasserstein (W_p) entre diagramas es similar a la bottleneck, pero considera la suma de las distancias de los emparejamientos elevados a la potencia p, en lugar del máximo.
Es menos sensible a outliers que la distancia bottleneck.

Métricas sobre vectorizaciones

Una vez transformados los diagramas en vectores o funciones (e.g., paisajes), podemos usar métricas estándar.
Normas L^p de paisajes: Podemos calcular la norma L^p (e.g., L^2 euclidiana) de la diferencia entre paisajes.

Integración en aprendizaje automático

Integración con pipelines de aprendizaje

El flujo de trabajo completo es:

1. Datos → Filtración → Diagrama de persistencia.
2. Diagrama de persistencia → Vectorización (e.g., Paisaje).
3. Vectorización → Modelo de ML (SVM, Random Forest, Red Neuronal).

La estabilidad de las vectorizaciones favorece la robustez del pipeline, aunque el desempeño final depende también del modelo de ML y la validación.

14. Aplicación: Series de tiempo

¿Cómo aplicamos TDA a series de tiempo que son datos 1D? Necesitamos transformarlas en nubes de puntos de mayor dimensión que capturen la dinámica subyacente.

Encaje de Takens (Retardos temporales)

Dada una serie $x(t)$, definimos $v_t = (x(t), x(t+\tau), \ldots, x(t+(d-1)\tau)) \in \mathbb{R}^d$ donde d se denomina la dimensión de encaje y τ es el retardo.

Ejemplo 58

Reconstrucción de la dinámica

Motivación

Si la serie de tiempo proviene de un sistema dinámico (e.g., un oscilador), el encaje de Takens puede reconstruir el atractor del sistema.

Teorema 59

Teorema de Takens (Enunciado informal). Bajo ciertas condiciones (suficiente dimensión d y elección genérica de τ), la nube de puntos generada por el encaje de retardos es topológicamente equivalente (difeomorfa) al atractor original del sistema dinámico.

- **Aplicación de TDA:** Calculamos la homología persistente de la nube de puntos de Takens.
- Si el sistema es periódico, el atractor es un ciclo, y veremos una alta persistencia en H_1.
- Si el sistema es caótico, el atractor puede tener una topología más compleja.

Análisis de señales biomédicas

Aplicación en EEG/ECG

Las señales biomédicas (Electroencefalograma, Electrocardiograma) son complejas y ruidosas. El TDA puede ayudar a caracterizarlas.

Lectura de PH a lo largo del tiempo

1. Dividir la señal en ventanas temporales deslizantes.
2. En cada ventana, realizar un encaje de Takens.
3. Calcular la homología persistente (o sus vectorizaciones).
4. Analizar cómo cambian las características topológicas (β_k, persistencias) a lo largo del tiempo.

Uso: En varias aplicaciones, las características topológicas pueden mostrar *robustez* frente a ruido y variaciones locales, como complemento de rasgos frecuenciales o estadísticos tradicionales.

15. Mapper: Visualización topológica de datos

Mapper es un algoritmo de TDA para visualizar la forma global de datos de alta dimensión, similar a un clustering pero preservando la estructura topológica.

Objetivo de Mapper

Construir un grafo (o complejo simplicial) que resuma la estructura de una nube de puntos, donde los nodos representan clusters locales y las aristas indican que estos clusters están cerca o se solapan.

Construcción de Mapper

El algoritmo Mapper tiene 4 pasos principales:

1. **Filtro (Filter):** Proyectar los datos de alta dimensión usando una función de filtro $f : X \to \mathbb{R}^k$ (e.g., PCA, densidad).
2. **Cubierta (Cover):** Cubrir el rango de la función de filtro con un conjunto de intervalos (o cajas) solapados $\mathcal{U}$.
3. **Clustering local:** Calcular la preimagen $f^{-1}(U)$ para cada intervalo o caja $U \in \mathcal{U}$. Aplicar un algoritmo de clustering estándar (e.g., DBSCAN) a estos conjuntos de puntos locales.
4. **Nervio (Nerve):** Construir el grafo de Mapper. Cada cluster local se convierte en un nodo. Se añade una arista entre dos nodos si los clusters correspondientes tienen puntos en común (intersección no vacía).

Interpretación de Mapper

Interpretación cualitativa

El grafo resultante proporciona una visualización intuitiva de la forma global de los datos.

- **Bucles** en el grafo Mapper pueden indicar ciclos o estructuras periódicas en los datos.
- **Ramificaciones** (flares) pueden indicar diferentes subpoblaciones o transiciones en los datos.

- Los nodos del grafo pueden colorearse según el valor promedio de la función de filtro, o cualquier otra propiedad de interés (e.g., etiquetas de clase).
- Mapper es una herramienta poderosa para la exploración interactiva de datos, detección de subgrupos y generación de hipótesis.

Caso de estudio clásico: Con datos de cáncer de mama, Mapper identificó subgrupos con diferencias de pronóstico que no eran evidentes con técnicas estándar.

Resumen del módulo

Sesión 1: Teoría de grafos

- Fundamentos: Definiciones, representaciones (Laplaciano), propiedades estructurales (árboles).
- Análisis de redes: Centralidades, clustering, detección de comunidades.
- Modelos generativos: ER, WS, BA.
- Aprendizaje en grafos: Node embeddings (DeepWalk) y GNNs (Message Passing).

Sesión 2: Análisis Topológico de Datos (TDA)

- Fundamentos topológicos: Complejos simpliciales, homología, números de Betti.
- Homología persistente: Filtraciones (Vietoris-Rips), diagramas de persistencia, teorema de estabilidad.
- TDA en práctica: Vectorización (Landscapes), aplicaciones (series de tiempo), visualización (Mapper).

¡Gracias por su atención!

Dra. Rosalía G. Hernández
Dr. Jesús F. Espinoza

Universidad de Sonora