UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Algoritmos de detección de comunidades en grafos y aplicación en conjuntos de datos de tránsito en aeropuertos y trasvase de alumnos entre titulaciones de la UPV

Trabajo Fin de Grado

Grado en Ciencia de Datos

AUTOR/A: Méndez Carmona, Alejandro

Tutor/a: Morillas Gómez, Samuel

Cotutor/a: Hervás Jorge, Antonio

CURSO ACADÉMICO: 2021/2022

Resum

Els avanços en teoria de grafs i anàlisi de xarxes complexes han estat fonamentals per a una millor comprensió dels sistemes complexos que poden aparèixer a la vida real. L'estructura comunitària és una de les característiques més importants dels grafs que modelen aquests sistemes; no obstant això, l'obtenció de comunitats és un problema extraordinàriament complex i encara no s'ha obtingut una solució que funcione satisfactòriament en qualsevol tipus de graf que es puga presentar en situacions reals. Durant les dues darreres dècades s'han desenvolupat multitud de mètodes per detectar comunitats en grafs, usant conceptes i tècniques de diferents àmbits científics com la física, la biologia, o les ciències socials. En aquest treball s'exposen alguns dels mètodes més emprats actualment a detecció de comunitats, i es presenten certs algorismes l'objectiu dels quals és la millora dels resultats obtinguts fins ara en grafs dirigits dèbilment connectats. A més, es presenta un nou algorisme basat en tècniques usades en tractament d'imatges per a la detecció de fronteres. Aquest nou mètode fa ús del producte de convolució per a la detecció i poda d'arestes de soroll. L'eficàcia d'aquest algorisme es comprova mitjançant la seva aplicació en dos casos reals: el trànsit de vols comercials entre aeroports dels Estats Units, i el flux d'estudiants entre graus al sistema d'accés a la universitat pública a Espanya.

Paraules clau: Teoria de grafs, Detecció de comunitats, Tractament d'imatges, Convolució

Resumen

Los avances en teoría de grafos y análisis de redes complejas han sido fundamentales para una mejor comprensión de los sistemas complejos que pueden aparecer en la vida real. La estructura comunitaria es una de las características más importantes de los grafos que modelan estos sistemas; sin embargo, la obtención de comunidades es un problema extraordinariamente complejo y aún no se ha obtenido una solución que funcione satisfactoriamente en cualquier tipo de grafo que pueda presentarse en situaciones reales. Durante las dos últimas décadas se han desarrollado multitud de métodos para detectar comunidades en grafos, usando conceptos y técnicas de distintos ámbitos científicos como la física, la biología, o las ciencias sociales.

En este trabajo se exponen algunos de los métodos más empleados actualmente en detección de comunidades, y se presentan ciertos algoritmos cuyo objetivo es la mejora de los resultados obtenidos hasta el momento en grafos dirigidos débilmente conectados. Además, se presenta un nuevo algoritmo basado en técnicas usadas en tratamiento de imágenes para la detección de bordes. Este nuevo método hace uso del producto de convolución para la detección y poda de aristas de ruido. La eficacia de este algoritmo se comprueba mediante su aplicación en dos casos reales: el tráfico de vuelos comerciales entre aeropuertos de Estados Unidos, y el flujo de estudiantes entre grados en el sistema de acceso a la universidad pública en España.

Palabras clave: Teoría de grafos, Detección de comunidades, Tratamiento de imágenes, Convolución

Abstract

The advances in graph theory and complex network analysis have been fundamental in a better understanding of real complex systems. Community structure is one of the most important characteristics in graphs modelling these systems; nevertheless, obtaining graph's communities is an extraordinarely complicated problem and has not yet been found a solution that can obtain satisfying results in any kind of graph that may appear in real scenarios. During the last two decades, a huge number of community detection methods have been developed, employing concepts and techniques from various scientific areas such as physics, biology or social sciences.

In this work, some of the most popular community detection methods are described, and are also presented new algorithms developed to improve the results in weakly connected directed graphs. A new algorithm based on image processing techniques for edge detection is described. This new method makes uso of the convolution product for detecting and pruning noise edges that may be present in the graph. The performance of this algorithm is proved in two real cases: commercial flights transit between US airports, and studient transit between univeristy degrees in the access system to the Spanish public universities.

Key words: Graph theory, Community detection, Image processing, Convolution

Índice general

Índice general v
Índice de figuras vii
Índice de tablas viii

1 Introducción 1
1.1 Objetivos . . . 2
1.2 Estructura de la memoria . . . 2
2 Estado del arte 3
2.1 Grafos y redes complejas . . . 3
2.2 Comunidades . . . 4
2.2.1 Comunidad y estructura comunitaria . . . 5
2.2.2 Algoritmos para la detección de comunidades . . . 7
3 Algoritmos basados en centros 15
3.1 Método Fdp . . . 15
3.2 Método IsoFdp . . . 16
3.2.1 IsoMap . . . 16
3.2.2 Fdp . . . 16
3.3 Otros métodos basados en centros . . . 17
3.4 Algoritmo In-Out . . . 18
4 Algoritmos basados en filtros de convolución 21
4.1 Convolución en imágenes . . . 21
4.2 Detección de bordes en imágenes . . . 21
4.2.1 Definición de borde . . . 22
4.2.2 Definición de detector de bordes . . . 23
4.2.3 Filtros de primer orden . . . 24
4.2.4 Filtros de segundo orden . . . 25
4.3 ConvGraph . . . 27
4.3.1 Grafo línea . . . 27
4.3.2 Convolución en grafo línea . . . 27
4.4 Otro algoritmo de detección de comunidades basado en convolución . . . 29
4.5 Un nuevo algoritmo de detección de comunidades basado en convolución 30
5 Caso práctico: Flujo de vuelos entre aeropuertos de Estados Unidos 33
5.1 Descripción de los datos . . . 33
5.2 Detección de comunidades . . . 34
5.2.1 Girvan-Newman . . . 34
5.2.2 *Walktrap* . . . 36
5.2.3 Convolución . . . 36
6 Caso práctico: Sistema de acceso a la universidad pública en España 39
6.1 Descripción de los datos . . . 40
6.2 Preprocesamiento de los datos . . . 41
6.3 Detección de comunidades . . . 43
6.3.1 Girvan-Newman . . . 43

6.3.2 Walktrap 43
6.3.3 Convolución 43
7 Conclusiones **53**
Bibliografía **55**

Apéndices
A Lista de grados **59**
B Objetivos de Desarrollo Sostenible **67**

Índice de figuras

2.1 Ejemplos de grafo 4
2.2 Ejemplo de grafo dirigido con estructura comunitaria 5
2.3 Grafo no dirigido 8
2.4 Grafo dirigido 8
2.5 Detección de comunidades con Girvan-Newman 9
2.6 Detección de comunidades con *fast greedy* 11
2.7 Detección de comunidades con *Walktrap* 12
2.8 Resumen resultados detección de comunidades en grafos no dirigidos . . 13
2.9 Resumen resultados detección de comunidades en grafos dirigidos 13

3.1 Detección de comunidades mediante algoritmo de centros 18
3.2 Grafo sistema de acceso a la universidad 19
3.3 Grafo IN 20
3.4 Grafo OUT 20

4.1 Ejemplo de convolución 22
4.2 Función de intensidad de la franja en rojo de la imagen 22
4.3 Ejemplo detección de bordes 24
4.4 Ejemplo detección de bordes mediante Laplaciana de la Gaussiana 27
4.5 Ejemplo de transformación de grafo a grafo línea 28
4.6 Ejemplo identificación aristas con respecto a la arista a 29
4.7 Ejemplo identificación aristas con respecto a la arista a 31
4.8 Resultado poda mediante convolución 32
4.9 Comunidades obtenidas al aplicar *Walktrap* 32

5.1 Grafo aeropuertos Estados Unidos 35
5.2 Distribución tráfico aeropuertos y viajes 35
5.3 Comunidades del grafo de aeropuertos obtenidas mediante Girvan-Newman 36
5.4 Comunidades del grafo de aeropuertos obtenidas mediante *Walktrap* . . . 37
5.5 Grafo de aeropuertos podado 38
5.6 Comunidades obtenidas mediante *Walktrap* en el grafo podado 38

6.1 Histograma frecuencia solicitudes en primera opción 41
6.2 Grafo Sistema de acceso a la universidad en la Comunidad Valenciana . . 42
6.3 Distribución grados de entrada y salida 43
6.4 Comunidades detectadas con Girvan-Newman 44
6.5 Comunidades detectadas con *Walktrap* 44
6.6 Grafo podado mediante convolución con $\sigma = 0{,}5$ 45
6.7 Detección de comunidades en el grafo podado mediante *Walktrap* 46
6.8 Comunidades pequeñas del grafo de acceso a la universidad 47
6.9 Comunidades de tamaño medio del grafo de acceso a la universidad . . . 47
6.10 Comunidades de Humanidades e Ingeniería del grafo de acceso a la universidad 48
6.11 Comunidad de Ingeniería del grafo de acceso a la universidad 49

6.12 Subcomunidades dentro de la comunidad de ingenierías 49
6.13 Subcomunidades dentro de la comunidad de ciencias 50
6.14 Subcomunidades dentro de la comunidad de ciencias sociales (Valencia y Castellón) 51
6.15 Subcomunidades dentro de la comunidad de ciencias sociales (Alicante) . 51

Índice de tablas

5.1 Extracto del conjunto de datos de aeropuertos estadounidenses 34
5.2 Extracto del conjunto de datos de viajes entre aeropuertos 34

6.1 Extracto del conjunto de datos de grados universitarios 40
6.2 Extracto del conjunto de datos de solicitudes de acceso 41

CAPÍTULO 1
Introducción

La teoría de grafos es un elemento esencial en una gran cantidad de ámbitos científicos. Su capacidad de modelar innumerables tipos de relaciones y procesos han hecho de la teoría de grafos una herramienta fundamental para la resolución de todo tipo de problemas en, por ejemplo, física, biología o ciencias de la computación. Desde la publicación en 1736 del trabajo de Leonhard Euler para la resolución del problema de los puentes de Königsberg [1], considerado el primer artículo de la historia en teoría de grafos, los métodos y aplicaciones en este área de las matemáticas se han ampliado enormemente. Los avances en las tecnologías de la información han afectado considerablemente en este ámbito, especialmente con el aumento de los datos disponibles y de los recursos computacionales para su procesamiento y análisis.

El modelo de grafo aleatorio definido por Erdös y Rényi [2] fue uno de los avances más importantes en teoría de grafos del siglo XX. Un grafo aleatorio es un grafo sin orden en los vértices, donde la probabilidad de que exista un enlace entre dos vértices es la misma para cualquier par de ellos. El estudio de grafos que modelan sistemas reales ha demostrado que en estos sistemas existe un alto nivel de orden y organización en sus vértices. Esta característica de los sistemas reales se llama estructura de comunidad. Una comunidad es un grupo de vértices que comparten propiedades comunes y tienen funciones similares dentro del grafo.

Estas comunidades aparecen en todo tipo de grafos. En redes sociales se puede encontrar diversos tipos de comunidades: familia, amigos, compañeros de trabajo... También aparecen comunidades en grafos que modelan sistemas biológicos, económicos, o políticos. La detección de estas comunidades es de mucha utilidad, y se emplea en infinidad de áreas. Algunas aplicaciones concretas puede ser la agrupación de clientes con intereses similares para una mejor y más eficiente recomendación de productos, o el análisis de los efectos de la estructura comunitaria en la red de difusión de una enfermedad.

En 2002, Girvan y Newman propusieron un nuevo algoritmo para la detección de comunidades que impulsaría un aumento de la actividad investigadora en este ámbito [3]. Muchos métodos han sido propuestos en las últimas dos décadas usando conceptos, herramientas y técnicas de diferentes áreas científicas, como son la física, la biología, la sociología o la matemática discreta. No obstante, debido a la gran variedad de grafos que pueden modelar sistemas reales, no existe ningún algoritmo que funcione bien en todos ellos. Uno de los grafos menos estudiados en este contexto son los grafos dirigidos débilmente conectados, en los que los algoritmos de detección de comunidades más empleados no obtienen generalmente buenos resultados, y para los que es necesario definir nuevos algoritmos diseñados especialmente a obtener las comunidades en este tipo de grafos.

En este proyecto se estudia la detección de comunidades en grafos, poniendo especial interés en el caso de los grafos dirigidos débilmente conectados y algunos algoritmos que han surgido con el fin de ser aplicados con éxito en este tipo de grafos. Además, partiendo de un algoritmo previo, se propone un nuevo método de detección de comunidades. Por último, se analiza la estructura comunitaria mediante el algoritmo propuesto en dos casos de uso.

1.1 Objetivos

El objetivo principal de este trabajo es la presentación y análisis de algoritmos de detección de comunidades diseñados para su aplicación en grafos dirigidos débilmente conectados, además de la definición de un nuevo método de detección de comunidades. Este proyecto puede dividirse en 3 diferentes objetivos:

- Presentación de los conceptos básicos en detección de comunidades y de los algoritmos más ampliamente usados.
- Estudio de los algoritmos de detección de comunidades surgidos para su uso en grafos dirigidos débilmente conectados.
- Definición de un nuevo algoritmo de detección de comunidades basado en convolución.

1.2 Estructura de la memoria

Este trabajo está dividido en siete capítulos. En el capítulo 2 se introducen los conceptos básicos en teoría de grafos y en detección de comunidades en grafos, además de presentar algunos de los algoritmos más conocidos. En los capítulos 3 y 4 se presentan algunos algoritmos de detección de comunidades que han sido definidos con el fin de emplearse en grafos dirigidos débilmente conectados. En el capítulo 3 se presentan dos algoritmos de detección de comunidades basados en la identificación de potenciales centros de comunidades. En el capítulo 4 se presentan otros dos algoritmos, estos basados en las técnicas de detección de bordes empleadas en tratamiento de imágenes; además de proponerse un nuevo algoritmo de detección de comunidades. En los capítulos 5 y 6 se aplican los algoritmos de detección de comunidades en dos casos reales: en el capítulo 5 se estudian las comunidades presentes en el sistema de transporte aéreo de Estados Unidos, y en el capítulo 6 se usan los métodos de detección de comunidades para estudiar el sistema de acceso a la universidad pública española. Por último, el capítulo 7 contiene las conclusiones y trabajo futuro de este trabajo.

CAPÍTULO 2
Estado del arte

2.1 Grafos y redes complejas

La teoría de grafos es la rama de las matemáticas dedicada al estudio de grafos: estructuras matemáticas empleadas para modelar relaciones entre objetos. Un grafo puede representar cualquier sistema que se pueda representar como puntos, o vértices, relacionados entre ellos (estas relaciones se representan como aristas). Un mapa de carreteras uniendo ciudades, el plano de una ciudad o una red de ordenadores puede ser representado como un grafo. El estudio de sus propiedades es de gran interés en muchas disciplinas científicas.

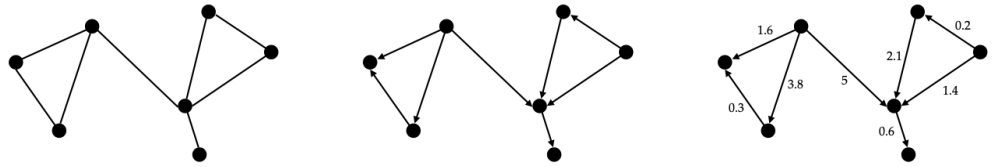
Un grafo, G, viene definido por la pareja de conjuntos (V, E), donde V es un conjunto de vértices y E un conjunto de pares no ordenados de vértices. Los elementos de E, llamados aristas, relacionan dos vértices $u, v \in V$, y se denotan como (u, v). Los vértices u y v son los extremos de la arista (u, v). Dos vértices son adyacentes si están relacionados mediante una arista. A su vez, dos aristas son adyacentes si tienen un vértice en común. Se dice que una arista incide en un vértice cuando el vértice es extremo de la arista. El grado de un vértice corresponde al número de aristas incidentes en este. El grado de un vértice v se denota como k_v.

Cuando E está formado por pares no ordenados de vértices, se dice que el grafo es no dirigido; en cambio, si está formado por pares ordenados, el grafo será dirigido. En los grafos dirigidos, al contrario que en los no dirigidos, las aristas tienen orientación. Dada una arista (u, v) de un grafo dirigido, se llama extremo inicial al vértice u y extremo final al vértice v. En este caso, un vértice v tiene asociados dos clases de grado: grado de entrada, que corresponde al número de aristas cuyo extremo inicial es v; y de salida, número de aristas cuyo extremo final es v.

Si las aristas de un grafo tienen asociado un valor numérico, o peso, se dice que el grafo es ponderado, donde el peso asociado a la arista (u, v) se denota como $w_{u,v}$. La definición de grafo no incluye bucles (aristas donde ambos extremos corresponden al mismo vértice) ni aristas múltiples (dos o más aristas con extremos idénticos). Un grafo con alguno de estos elementos se denomina multigrafo.

Los grafos se representan gráficamente mediante puntos unidos por líneas, donde cada punto está relacionado con un vértice distinto y dos puntos están unidos si existe una arista que una los dos vértices asociados. La Figura 2.1 muestra tres ejemplos de grafos.

El orden y tamaño de un grafo hacen referencia al número de vértices, N, y aristas, K, respectivamente. Dado N, el tamaño, o número de aristas, máximo de un grafo equivale a $N(N-1)/2$ en el caso de los grafos no dirigidos, o $N(N-1)$ cuando el grafo es dirigido.



(a) Grafo no dirigido **(b)** Grafo dirigido **(c)** Grafo dirigido ponderado

Figura 2.1: Ejemplos de grafo

Dado un grafo $G = (V, E)$, el grafo $G' = (V', E')$ será un subgrafo de G si $V' \subset V$ y $E' \subset E$. Cuando E' contiene todas las aristas de G que unen los vértices de V', se dice que el subgrafo G' es un subgrafo inducido.

Un camino de longitud n es un grafo $P = (V(P), E(P))$, donde $V(P) = \{v_0, v_1, \ldots, v_n\}$ y $E(P) = \{(v_0, v_1), (v_1, v_2), \ldots, (v_{n-1}, v_n)\}$. Un grafo es conexo si, para cada par de vértices, existe un camino que vaya de un vértice al otro. El camino más corto entre dos vértices es el camino de mínima longitud que una ambos vértices. La distancia entre dos vértices corresponde a la longitud del camino más corto entre estos dos vértices.

La información de la topología de un grafo de orden N se encuentra en la matriz de adyacencia A, una matriz $N \times N$ donde el elemento A_{uv} es 1 si existe una arista que una los vértices u y v (en caso contrario es 0). Si el grafo es no dirigido, la matriz A es simétrica, y la suma de los elementos de la fila (o columna) i equivale al grado del vértice i. Si el grafo es dirigido, la matriz A puede no ser simétrica. En este último caso, la suma de los elementos de la fila i equivalen al grado de salida del vértice i, mientras que los de la columna i equivalen al grado de entrada. Si las aristas son ponderadas, la información de los pesos se encuentra en la matriz de pesos W del grafo, donde el elemento W_{uv} equivale al peso de la arista (u, v). La matriz de acceso Q almacena la información acerca de, dado un vértice, que vértices son alcanzables desde este. La matriz de accesibilidad R muestra qué vértices alcanza un determinado vértice.

El espectro de un grafo es el conjunto de valores propios de su matriz de adyacencia. La propiedades espectrales son fundamentales en el estudio de grafos, además de ser empleadas en métodos de detección de comunidades.

Gracias a la habilidad de los grafos de modelar gran variedad de relaciones y procesos, la teoría de grafos tiene aplicaciones en diversas áreas científicas, como biología, química, informática o ciencias sociales.

2.2 Comunidades

Una comunidad es una agrupación de vértices altamente conectados entre ellos y mal conectados con el resto. Los vértices agrupados en la misma comunidad tienen características comunes que les hacen jugar un cierto papel dentro de la red.

La detección de comunidades consiste en la identificación de grupos de vértices en un grafo según sus propiedades estructurales. Esto permite obtener información acerca de la propiedades de los vértices del grafo y de las relaciones entre ellos que no se pueden obtener mediante observaciones o mediciones directas. La detección de comunidades juega un papel muy importante en muchos ámbitos científicos. Por ejemplo, está aceptada la existencia de estructura comunitaria en muchas de los grafos en biología, como la red metabólica o la de interacción entre proteínas [4], [5]. También tiene su aplicación en ciencias de salud, por ejemplo, en la detección de cáncer de pulmón [6] o en el estudio de redes

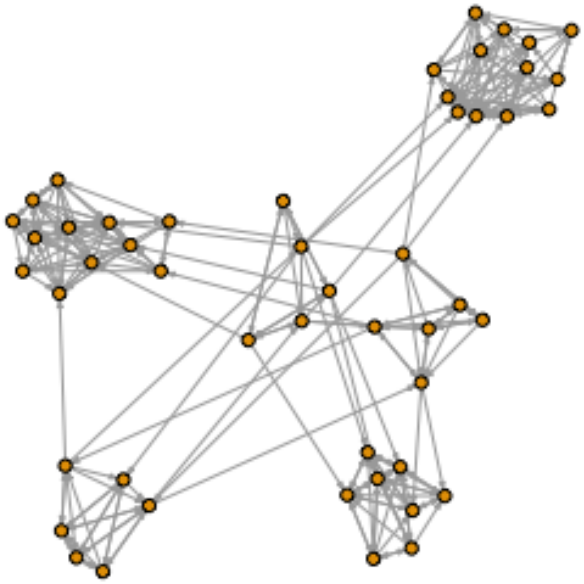


Figura 2.2: Ejemplo de grafo dirigido con estructura comunitaria

epidemiológicas [7], [8]. La detección de comunidades es también ampliamente empleada en neurociencia, donde se ha aplicado para el estudio de la topología y estructura de las redes neuronales, además de en la investigación en enfermedades relacionadas como el Alzheimer [9], demencia frontotemporal y otras enfermedades neurodegenerativas [10], o Trastorno Límite de la Personalidad [11]. El estudio de comunidades en grafo se ha aplicado en otras muchas disciplinas, por ejemplo, redes sociales, economía o redes de comunicación.

2.2.1. Comunidad y estructura comunitaria

Existe consenso en la definición de comunidad como grupo de vértices fuertemente relacionados entre ellos, y débilmente con vértices de distinto grupo [3]. No obstante, uno de los mayores problemas a la hora de obtener las comunidades de un grafo radican en la no disponibilidad de una definición cuantitativa de comunidad universalmente aceptada [12]. Muchas de las definiciones de comunidad parten de la idea intuitiva de que los nodos dentro de una misma comunidad deben estar mucho más conectados entre ellos que entre el resto de nodos fuera de esa comunidad. En la figura 2.2 se puede ver un ejemplo de grafo dirigido con una clara estructura comunitaria. En este grafo se puede observar claramente cinco comunidades distintas, donde los nodos en cada comunidad están mucho más relacionados entre ellos que con el resto de nodos del grafo.

Dado un grafo $G = (V, E)$, la obtención de comunidades consiste en la división del conjunto de vértices V en n_c comunidades, donde cada nodo debe pertenece a una comunidad. Esta división de comunidades, o partición, se define como el conjunto de subconjuntos disjuntos de vértices $C = c_1, c_2, \ldots, c_{n_c}$. Existen también métodos de detección de comunidades donde un nodo puede pertenecer a más de una comunidad. En ese caso, dos comunidades se dice que están superpuestas si tienen algún nodo en común. No obstante, este trabajo se focaliza en la detección de comunidades sin superposición.

Dado un grafo G, una partición C y una comunidad $c \in C$, se define el grado interno y externo de un vértice $v \in c$, k_v^{int} y k_v^{ext}, como el número de aristas entre v y otros vértices de c, y el número de aristas entre v y el resto de vértices del grafo, respectivamente.

Varias definiciones cuantitativas de comunidad se han propuesto, que se clasifican en tres tipos: locales, globales y basadas en similitud de vértices.

Definición local de comunidad

Las comunidades pueden considerarse, hasta cierto punto, entidades independientes del resto del grafo, por lo que es de especial interés su estudio y evaluación de manera individual. Las definiciones de comunidad locales se focalizan en el subgrafo a estudiar ignorando el resto del grafo. Estas definiciones locales están fuertemente relacionadas con el concepto de subgrupo cohesivo usado en el análisis de redes sociales [13]. En este ámbito, una comunidad puede definirse como un subgrupo donde todos sus miembros se conocen entre sí, lo que se conoce como reciprocidad completa [14]. En términos de teoría de grafos, una comunidad correspondería a un clique, un grafo donde todos sus vértices son adyacentes entre ellos. Cliques de tres vértices aparecen frecuentemente en redes reales, pero cliques más grandes son mucho menos frecuentes, lo que hace esta definición de comunidad demasiado estricta. El n-clique [15] es una generalización del concepto de clique, donde la distancia entre cualquier par de nodos del subgrafo debe ser menor o igual que n. En [13] se introducen otras formas de clique, como n-club, n-plan, k-plex o k-core. El problema con la definición de comunidad como subgrupo cohesivo radica en solo tener en cuenta cohesión interna, ignorando la posible cohesión con otros grupos; por lo que de aquí se puede concluir que es importante la comparación entre la cohesión interna y externa de un subgrafo. Otros conceptos, como el LS-Set [16], tienen en cuenta ambas cohesiones interna y externa de un subgrafo, y es la manera habitual de proceder a la hora de definir comunidades.

Modularidad

Las comunidades también pueden ser definidas con respecto al grafo en su totalidad, al ser estas partes indispensables en el grafo que no se pueden eliminar sin afectar considerablemente el funcionamiento del sistema [12].

La modularidad [17] es una función que mide la bondad de la división en un grafo en comunidades. Esta función está basada en la idea de que un grafo aleatorio está desprovisto de estructura comunitaria, por lo que una posible existencia de comunidades se puede observar mediante la comparación de la densidad de aristas de un subgrafo y la densidad esperada en el mismo subgrafo usando un modelo nulo, i.e. una copia del grafo original que mantiene todas sus propiedades estructurales excepto la estructura de comunidad.

Formalmente, la modularidad se puede definir como:

$$Q = \frac{1}{2N} \sum_{uv} \left(A_{uv} - \frac{k_u k_v}{2N} \right) \delta(C_u, C_v)$$

donde N es el número de vértices del grafo, A la matriz de adyacencia, y k_v el grado del vértice v. La función $\delta(C_u, C_v)$ devuelve uno si u y v pertenecen a la misma comunidad (cero en caso contrario). Esta modo de calcular la modularidad no se emplea en la práctica, sino que se calculan dos parámetros previamente que hacen el cálculo más eficiente.

Dado un grafo G y una partición del grafo en k comunidades, definimos e como una matriz simétrica $k \times k$, donde cada elemento e_{ij} de la matriz representa la proporción de aristas que conectan vértices de la comunidad i con vértices de la comunidad j, y se calcula de la siguiente manera:

$$e_{ij} = \frac{1}{2N} \sum_{uv} A_{uv} \delta(C_u, i) \delta(C_v, i)$$

Los elementos de la diagonal de la matriz e representan la proporción de aristas que conectan vértices dentro de una misma comunidad. A partir de esta matriz se define a_i, que se obtiene mediante la suma de las filas (o columnas) de e, y representa la proporción de aristas que conectan vértices a la comunidad i.

$$a_i = \frac{1}{2N} \sum_{v} k_v \delta(C_v, i)$$

Se puede demostrar que la modularidad equivale a:

$$Q = \sum_{I} (e_{ii} - a_i^2)$$

2.2.2. Algoritmos para la detección de comunidades

Muchos algoritmos para la detección de comunidades se han propuesto en los últimos años, usando técnicas y herramientas de diversas disciplinas como biología, física, ciencias sociales, matemática aplicada o ciencias de la computación [18]. No obstante, ninguno de los algoritmos propuestos hasta el momento ha demostrado un buen rendimiento en todos los tipos de grafos debido a la gran variedad de estos que existen [19]. El rendimiento de un algoritmo de detección de comunidades está influenciado enormemente por la topología y estructura del grafo. En esta sección se describen algunos de los algoritmos más conocidos y usados en la actualidad. Además se aplica sobre el grafo no dirigido de la figura 2.3, y sobre un grafo dirigido equivalente, representado en la figura 2.4. Estos grafos se han generado con el generador descrito en [20]. Este generador está diseñado con el propósito de obtener grafos dirigidos débilmente conectados, que es el tipo de grafos de interés en este proyecto. En estos dos grafos se pueden observar cinco comunidades bien definidas, pero con aristas de ruido relacionando vértices de distintas comunidades. Son estas aristas de ruido las que dificultan la tarea de los algoritmos de detección de comunidades.

Algoritmo de Girvan-Newman

Un mecanismo simple para la obtención de comunidades es la detección y eliminado de aristas uniendo vértices de distinta comunidad, de manera que las comunidades queden desconectadas unas de otras. Esta es la manera de proceder de los algoritmos divisivos, entre los que se encuentra el algoritmo Girvan-Newman.

El algoritmo Girvan-Newman [3], es un algoritmo de detección de comunidades basado en el concepto de *edge betweenness*. Este método tiene una gran importancia histórica, además de ser un algoritmo ampliamente empleado.

El *edge betweenness*, mencionado anteriormente, es una medida de centralidad de aristas, que pretende cuantificar la importancia de una arista según la cantidad de «tráfico» que pasa por ella. Dada una arista cualquiera de un grafo, el *edge betweenness* equivale

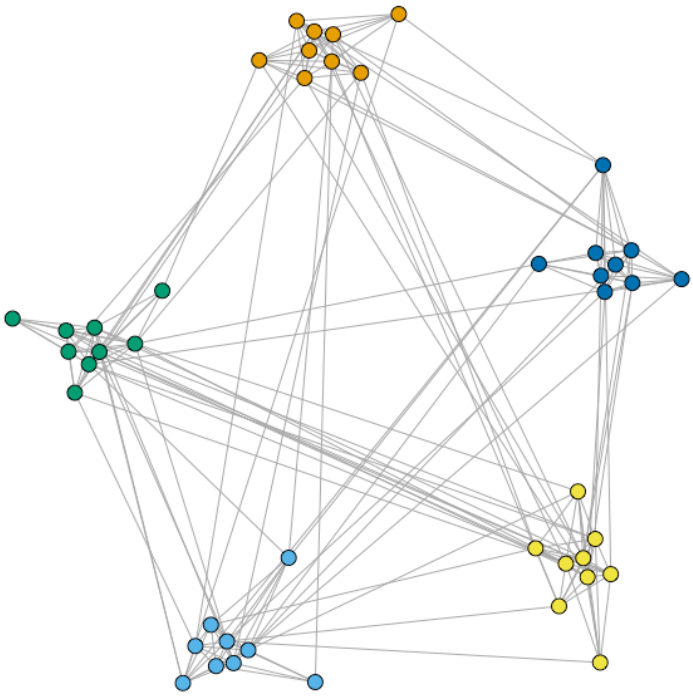


Figura 2.3: Grafo no dirigido

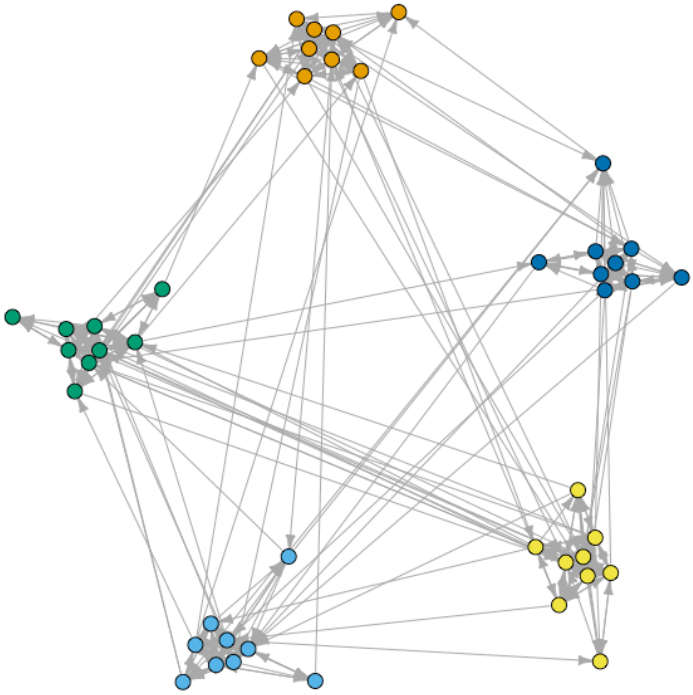


Figura 2.4: Grafo dirigido

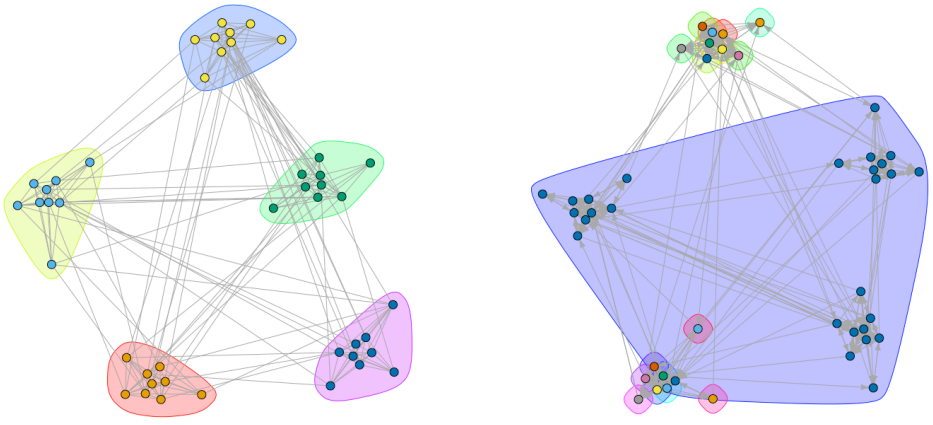
al número de caminos más cortos entre todos los pares de vértices que pasan por esta arista. Es una extensión del concepto de *site betweenness* [21], y expresa la importancia de una arista en procesos como la difusión de información, donde la información suele fluir a través de los caminos más cortos. Otras medidas de centralidad fueron consideradas por Girvan y Newman, como *random-walk betweenness* o *current-flow betweenness*; pero *edge betweenness* es el más empleado debido a ser más rápido y la medida que mejores resultados proporciona sobre el resto [17].

De manera intuitiva se puede obtener que aristas uniendo vértices de distinta comunidad tendrán valores más altos de *edge betweenness*, al pasar una parte importante de los caminos más cortos entre vértices de distintas comunidades por esta arista. El algoritmo Girvan-Newman funciona de la siguiente manera:

1. Cálculo de *edge betweenness* para todas las aristas del grafo
2. Eliminado de la arista con mayor *edge betweenness* (en caso de empate entre dos o más aristas, una de ellas es escogida al azar)
3. Cálculo de *edge betweenness* para las aristas del grafo resultante
4. Vuelta al paso 2 hasta que no queden aristas en el grafo

Diversas modificaciones se han propuesto a lo largo de los años para mejorar el algoritmo original. Por ejemplo, Tyler et al. [22], [23] propusieron una modificación que incrementaba la velocidad de cálculo, al tener que analizar la red de co-ocurrencia de genes, que era demasiado grande como para poder ser analizada mediante el algoritmo propuesto por Girvan y Newman.

El resultado de aplicar el algoritmo Girvan-Newman sobre el grafo no dirigido de la figura 2.3 se puede ver en 2.5a. Este algoritmo consigue identificar sin errores las cinco comunidades del grafo, a pesar de tener la partición obtenida una modularidad de 0.523. No obstante, este mismo algoritmo aplicado sobre el grafo dirigido de la figura 2.4 devuelve un resultado mucho menos satisfactorio. El algoritmo detecta 18 comunidades, donde una de ellas (coloreada de azul en el grafo) contiene 28 vértices, mientras que el resto son comunidades aisladas, formadas por un solo vértice. La modularidad de esta partición es de 0.178, lejos de un resultado aceptable.



(a) Grafo no dirigido **(b)** Grafo dirigido

Figura 2.5: Detección de comunidades con Girvan-Newman

Fast Greedy

Como se ha comentado en la Sección 2.2.1, la modularidad proporciona un indicador de la calidad de una partición. Varios algoritmos de detección de comunidades se han propuesto donde el objetivo es encontrar la partición que maximice esta modularidad.

Newman propuso un primer algoritmo voraz para la maximización de la modularidad [24], que sería modificado más tarde por Clauset, Newman y Moore [25], y que se conoce como *fast greedy*. Es un método de *clustering* aglomerativo jerárquico, donde grupos de vértices son unidos formando comunidades más grandes de modo que aumente la modularidad. Al contrario de lo que ocurría en el algoritmo de Girvan-Newman, este algoritmo parte de N comunidades, cada una conteniendo uno de los N vértices del grafo, que se van agrupando en búsqueda de una partición óptima.

El algoritmo presentado por Newman necesita de tres estructuras de datos que se van actualizando a cada iteración:

- Una matriz ΔQ, donde el elemento ΔQ_{ij} representa el incremento en la modularidad que se produciría al unir las comunidades i y j (si las comunidades no están unidas mediante ninguna arista, el valor es cero).
- Un *max-heap* H que almacene el mayor elemento de cada fila de la matriz ΔQ, con las etiquetas i, j correspondientes.
- Un vector con los elementos a_i (su obtención está descrita en la Sección 2.2.1).

Una matriz ΔQ, donde el elemento ΔQ_{ij} representa el incremento en la modularidad que se produciría al unir las comunidades i y j (si las comunidades no están unidas mediante ninguna arista, el valor es cero). Un *max-heap* H que almacene el mayor elemento de cada fila de la matriz ΔQ, con las etiquetas i, j correspondientes. Un vector con los elementos a_i (su obtención está descrita en la sección mod).

El algoritmo *fast greedy* se define de la siguiente manera:

1. Cálculo de los valores iniciales de ΔQ y a_i, y obtención de H a partir de ΔQ
2. Obtención del mayor valor de ΔQ mediante H, unión de las comunidades correspondientes, y actualización de ΔQ, H y a_i.
3. Vuelta al paso 2 hasta la obtención de una única comunidad.

Al unir dos comunidades i y j en una nueva comunidad j', solo es necesaria la actualización de la fila y columna j en ΔQ y la eliminación de la fila y columna i en la misma matriz. La actualización de ΔQ consiste en:

- Si una comunidad k está conectada a ambas comunidades i y j:

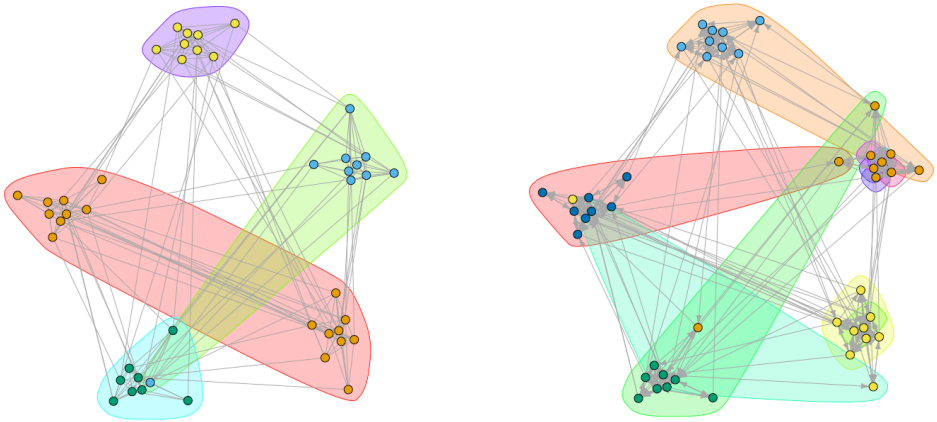
$$\Delta' Q_{jk} = \Delta Q_{ik} + \Delta Q_{jk}$$

- Si k está conectada a i pero no a j:

$$\Delta' Q_{jk} = \Delta Q_{ik} - 2a_j a_k$$

- Si k está conectada a j pero no a i:

$$\Delta' Q_{jk} = \Delta Q_{jk} - 2a_i a_k$$



(a) Grafo no dirigido **(b)** Grafo dirigido

Figura 2.6: Detección de comunidades con *fast greedy*

El elemento j del vector a se actualiza como $a'_j = a_j + a_i$, y $a_i = 0$. También es necesario actualizar el *max-heap* con los nuevos valores de ΔQ. No obstante, Clauset, Newman y Moore [25] comprobaron que era innecesario el uso de H, lo que aumentó sustancialmente la velocidad del algoritmo al poder prescindir del coste de mantenimiento del *max-heap*.

La figura 2.6 muestra las comunidades detectadas mediante el algoritmo *fast greedy* en los grafos de la figuras 2.3 y 2.4. En el grafo no dirigido, el algoritmo detecta cuatro comunidades, y obtiene una modularidad 0.46. Se puede apreciar como dos de las comunidades son detectadas como una sola, mientras que el resto de comunidades son correctamente identificadas (a excepción de un vértice que se incluido en otra comunidad). El resultado en el grafo dirigido es algo peor, con una modularidas de 0.29. En este caso, a pesar de que muchos de los vértices son correctamente agrupados en su comunidad, algunos otros son, o bien añadidos a una comunidad errónea, o agrupados en comunidades aisladas.

Walktrap

El uso de caminos aleatorios puede ser también muy conveniente para la detección de comunidades. En un grafo con una fuerte estructura comunitaria, un caminante aleatorio permanecerá más tiempo dentro de una comunidad debido a la gran densidad de aristas internas y, por consiguiente, el número de caminos que puede tomar. Pons y Lapaty [26] propusieron un método de clustering aglomerativo jerárquico basado en el método de Ward [27], donde las distancias entre nodos se obtienen a partir de caminos aleatorios.

En un camino aleatorio, la probabilidad de llegar al vértice v desde u es $P_{uv} = \frac{A_{uv}}{d(u)}$, o, de forma matricial, $P = D^{-1}A$, donde D es la matriz diagonal que contiene los grados de los vértices del grafo. Para obtener la probabilidad de ir del vértice u al vértice v en t pasos, se ha de multiplicar t veces la matriz de probabilidades P. Esta probabilidad se denota como P^t_{uv}.

En [26] se define una distancia para medir la similitud entre dos vértices. El objetivo es tener una distancia donde, si dos vértices pertenecen a una misma comunidad, esta distancia sea pequeña, mientras que si son de distinta comunidad la distancia sea grande. Esta distancia está definida como:

$$r_{ij} = \sqrt{\sum_{k=1}^{n} \frac{(P_{ik}^t - P_{jk}^t)^2}{d(k)}} = \|D^{-\frac{1}{2}} P_{i\bullet}^t - D^{-\frac{1}{2}} P_{j\bullet}^t\|$$

Esta distancia se puede generalizar para obtener la distancia entre dos comunidades. Considerando un camino aleatorio de longitud t que comienza en un vértice aleatorio escogido entre los vértices de una comunidad C, se define la probabilidad de llegar a un vértice j desde la comunidad C en t pasos como:

$$P_{Cj}^t = \frac{1}{|C|} \sum_{i \in C} P_{ij}^t$$

A partir de esta probabilidad, la distancia entre dos comunidades C_1 y C_2 se define como:

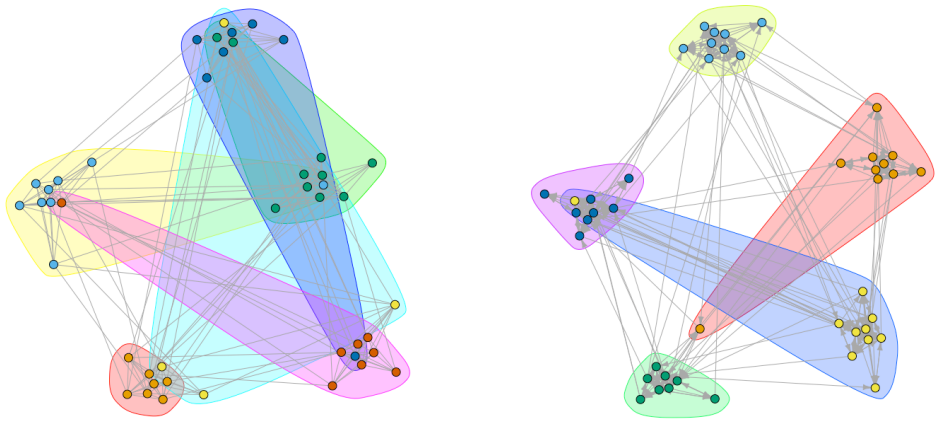
$$r_{C_1C_2} = \sqrt{\sum_{k=1}^{n} \frac{(P_{C_1k}^t - P_{C_2k}^t)^2}{d(k)}} = \|D^{-\frac{1}{2}} P_{C_1\bullet}^t - D^{-\frac{1}{2}} P_{C_2\bullet}^t\|$$

Esta distancia entre comunidades proporciona el criterio de unión de comunidades empleado en el algoritmo *Walktrap*. Dada una partición P del grafo, se busca unir aquellas dos comunidades que minimizan la media de las distancias al cuadrado entre cada vértice y su comunidad. Se usa una aproximación voraz para encontrar este mínimo, ya que el problema es NP, consistente en, para cada pareja de comunidades adyacentes C_1 y C_2, calcular la variación de la media que se produciría al unir ambas comunidades en una nueva, C_3. Esta variación se calcula como:

$$\Delta\sigma(C_1, C_2) = \frac{1}{n} \left(\sum_{i \in C_3} r_{iC_3}^2 - \sum_{i \in C_1} r_{iC_1}^2 - \sum_{i \in C_2} r_{iC_2}^2 \right)$$

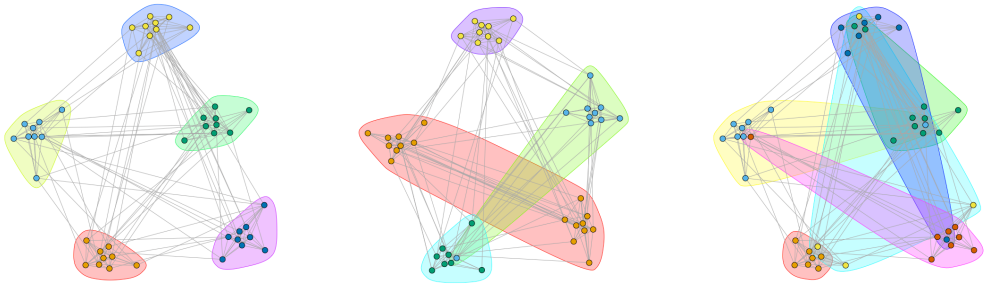
El algoritmo *Walktrap* parte de una partición P_1 de N comunidades, cada una conteniendo un único vértice del grafo. Cada iteración k consiste en:

1. Se escogen las dos comunidades C_1 y C_2 que minimicen $\Delta\sigma(C_1, C_2)$



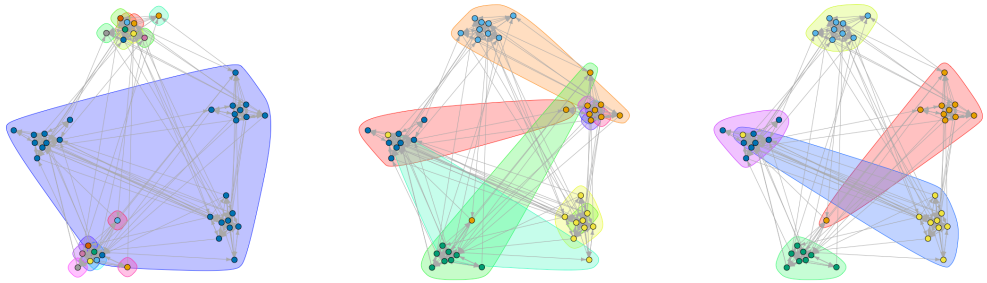
(a) Grafo no dirigido **(b)** Grafo dirigido

Figura 2.7: Detección de comunidades con *Walktrap*



(a) Girvan-Newman **(b)** Fast Greedy **(c)** Walktrap

Figura 2.8: Resumen resultados detección de comunidades en grafos no dirigidos



(a) Girvan-Newman **(b)** Fast Greedy **(c)** Walktrap

Figura 2.9: Resumen resultados detección de comunidades en grafos dirigidos

2. Se unen ambas comunidades, formando una nueva comunidad C_3, y se crea la nueva partición $P_{k+1} = (P_k \backslash \{C_1, C_2\}) \cup C_3$

Finalmente, la partición escogida será aquella que maximice la modularidad.

Las comunidades detectadas en los grafos de las figuras 2.3 y 2.4 mediante el algoritmo *Walktrap* se pueden ver en la figura 2.7. En este caso, los resultados obtenidos se comportan de manera opuesta a los obtenidos con Girvan-Newman: mientras que las comunidades obtenidas en el grafo no dirigido difieren mucho de la partición ideal, en el grafo dirigido se obtienen unos resultados muy cercanos a lo que se busca. La partición del grafo no dirigido tiene una modularidad de 0.26, mientras que la del grafo dirigido es de 0.57.

Resumen resultados obtenidos

Las figuras 2.8 y 2.9 muestran los resultados obtenidos con los tres algoritmos de detección descritos en esta sección aplicados en los grafos de la figura **??**. Como se puede observar, los resultados son muy distintos entre ellos. Esto no significa necesariamente que existan algoritmos mejores que otros, sino que, como se ha comentado antes, los algoritmos definidos hasta ahora no rinden de la misma manera en distintos tipos de grafos.

El procedimiento y criterio de detección de comunidades difiere enormemente entre unos algoritmos y otros, lo que ocasiona en muchas ocasiones que se obtengan resultados muy distintos con diferentes métodos. En este caso, esta característica de los algoritmos es muy palpable: mientras que el algoritmo Girvan-Newman ha obtenido muy buenos resultados en el grafo no dirigido y muy malos en el dirigido, el algoritmo *Walktrap* ha

obtenido resultados justo al contrario. Por tanto, aún siendo Girvan-Newman uno de los algoritmos más populares desde que fuera propuesto hace 20 años, se ha demostrado que tiene un rendimiento muy pobre en grafos dirigidos débilmente conectados. Es por ello que es necesario diseñar nuevos algoritmos y métodos con el propósito de ser aplicados en los grafos donde los algoritmos presentados hasta ahora no obtienen buenos resultados.

CAPÍTULO 3
Algoritmos basados en centros

La detección de comunidades se asemeja en muchas formas al *clustering*. En ambos problemas se busca la agrupación de objetos por similitud. Muchas técnicas de *clustering*, como K-Means o DBSCAN, pueden ser modificadas para su uso en la tarea de detectar comunidades [28], [29]. Uno de los métodos de *clustering* adaptado para su uso en grafos es el método Fdp, que se basa en la identificación de los potenciales centros de los *clusters*.

3.1 Método Fdp

En [30] se propuso el método de *clustering* Fdp, abreviatura de *Find density peaks*. Este método trata de identificar *clusters* mediante una identificación previa de sus centros. Parte de la suposición de que el centro de un *cluster* está rodeado de vecinos con una densidad local baja y está alejado de otros puntos con alta densidad local. La densidad local de un punto i se define como:

$$\rho_i = \sum_j \chi(d_{ij} - d_c)$$

donde d_ij es la distancia entre dos puntos i y j, d_c una distancia umbral, y $\chi(x)$ una función donde $\chi(x) = 1$ si $x < 0$ y $\chi(x) = 0$ en cualquier otro caso. El algoritmo es sensible únicamente a la magnitud relativa de ρ_i, lo que implica que, para conjuntos de datos grandes, el resultado del análisis es robusto con respecto a la elección del umbral d_c [30]. Este aspecto lo diferencia de otros métodos como DBSCAN, donde el umbral de densidad seleccionado afecta considerablemente los resultados obtenidos [31].

La distancia de un punto i al punto de mayor densidad más cercano se calcula como:

$$\delta_i = \min_{j:\rho_j > \rho_i} d_{ij}$$

Los centros de los *clusters* se pueden identificar como aquellos puntos con valores de ρ_i y δ_i mayores que el resto. Después de identificarse los centros, el resto de puntos son asignados al *cluster* del punto vecino con mayor densidad local.

Fdp, al igual que DBSCAN, sufre de la llamada «maldición de la dimensión», y lo hace un método no aconsejable para ser aplicado en conjuntos de datos de altas dimensiones [32]. Por desgracia, las redes tienen características que las hacen muy similares a conjuntos de datos con dimensión alta, por lo que este algoritmo no se puede aplicar directamente para la obtención de comunidades en redes.

3.2 Método IsoFdp

El método IsoFdp [33] fue propuesto como una modificación del método de *clustering* Fdp para poder ser aplicado al problema de detección de comunidades en grafos. En este método, una reducción de la dimensión del conjunto de datos es aplicada usando IsoMap, para después emplear Fdp para la detección de comunidades

3.2.1. IsoMap

Dado un grafo $G = (V, E)$, la similitud de estructura entre dos vértices u, v se define como [34]:

$$SS(u,v) = \frac{|N(u) \cap N(v)|}{\sqrt{|N(u) \times N(v)|}}$$

donde $N(v) = \{u \in V | (u,v) \in E\} \cup \{v\}$ es el conjunto de vértices adyacentes al nodo v, además del mismo vértice v. La similitud de estructura es empleada para la construcción de la matriz de similitud SS_G, donde el valor ij de la matriz corresponde a $SS(i,j)$. De esta matriz se obtiene, a su vez, la matriz de distancias D_G, donde el valor ij es $1/SS(i,j)$, si $i \neq j$ (cero en cualquier otro caso).

IsoMap genera un nuevo grafo G' a partir de la matriz de distancias D_G. De este nuevo grafo obtiene la matriz de distancias geodésicas $D_{G'}$, donde el elemento ij de la matriz corresponde a la distancia del camino más corto entre los nodos i y j. Por último, un escalamiento multidimensional es aplicado a esta matriz de distancias, de modo que se obtiene una proyección del grafo en una matriz d-dimensional, donde $d \ll N$, que trata de preservar la estructura del grafo original.

3.2.2. Fdp

Una nueva medida γ_i es propuesta en esta modificación, calculada a partir de ρ_i y δ_i como:

$$\gamma_i = \rho_i \delta_i$$

En el método original de Fdp, los centros se identifican por tener una densidad local, ρ_i, mayor que sus vecinos, además de estar alejados de otros centros, por lo que tenían un mayor valor de δ_i, lo que hace razonable el uso de la nueva medida γ_i, que permite identificar a los centros como aquellos vértices con mayor valor en esta medida.

Una función de densidad es definida para la evaluación de la calidad de las comunidades detectadas. La densidad de partición local dada una comunidad c se calcula como:

$$D_c = \frac{m_c - (n_c - 1)}{n_c(n_c - 1)/2 - (n_c - 1)}$$

donde n_c y m_c son el número de vértices y aristas, respectivamente, de la comunidad c. La densidad de partición del grafo completo se define como la suma ponderada de las densidades de partición local para todas las k comunidades, donde el peso asignado a cada comunidad en la suma equivale a su número de vértices.

$$D = \frac{2}{N} \sum_c n_c \frac{m_c - (n_c - 1)}{(n_c - 2)(n_c - 1)}$$

Para evitar un exceso de fragmentación a la hora de escoger la mejor partición, un término de penalización es añadido a la definición anterior.

$$D = \frac{2}{\sqrt{k}N} \sum_c n_c \frac{m_c - (n_c - 1)}{(n_c - 2)(n_c - 1)}$$

Donde k es el número de comunidades en las que se ha dividido el grafo. Esta última definición de densidad de partición es la que se emplea en el algoritmo para encontrar la partición óptima. El algoritmo IsoFdp recibe únicamente dos parámetros: la dimensión a la que se va a reducir el conjunto de datos original, d, y el número de comunidades a detectar, k. Una vez reducida la dimensión del conjunto de datos y calculado el valor γ_i para todos los vértices del grafo, los k vértices con γ_i más alto son seleccionados como centros, y el resto de vértices son asignados a la comunidad del vértice vecino con mayor densidad local (de manera similar a como se procedía en el método Fdp). Mediante el cálculo de la densidad de partición, D, se busca encontrar los valores d y k óptimos que obtengan el máximo valor de D. De manera resumida, los pasos que sigue el algoritmo son los siguientes:

1. Se transforma el grafo en una matriz de distancias usando la similitud de estructura entre dos vértices
2. Se reduce a una dimensión d la matriz de distancias obtenida en el paso anterior mediante el método IsoMap
3. Se calcula para todos los vértices γ_i a partir de la densidad local, ρ_i y la distancia al vértice más cercano con mayor densidad local, δ_i.
4. Se selecciona como centros los k vértices con mayor γ_i, y se asigna al resto de vértices la comunidad de su vecino con mayor densidad local
5. Se calcula la densidad de partición D para la evaluación de la partición obtenida
6. Se repiten los pasos 2 a 5 hasta encontrar el valor máximo de D obtenido con los valores óptimos $d*$ y $k*$

3.3 Otros métodos basados en centros

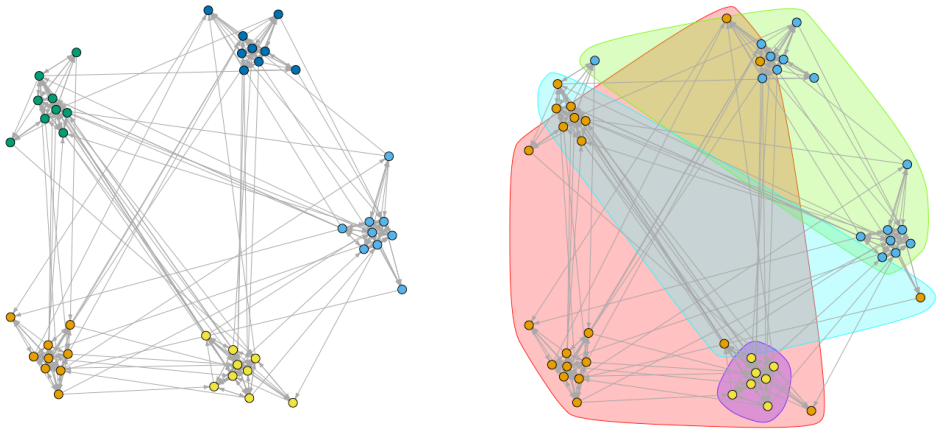
En [35] un método basado en la obtención de centros es propuesto con el objetivo de mejorar el rendimiento de otros algoritmos de detección de comunidades cuando son aplicados a grafos dirigidos ponderados débilmente conectados.

Este algoritmo define la distancia entre dos vértices u y v como la suma de la inversa de los pesos de las aristas que conforman el camino entre u y v que hace mínima esta suma.

$$d_{uv} = \frac{1}{w_{ux}} + \frac{1}{w_{xy}} + ... + \frac{1}{w_{zv}}$$

El conjunto de centros potenciales se obtiene a partir de los vértices cuya distancia media con el resto de vértices del grafo sea menor que la de los vértices de su entorno.

Una vez obtenidos los centros, se procede a podar las aristas de poca importancia en el grafo, con el objetivo de eliminar aristas uniendo vértices de comunidades distintas. En cada camino mínimo entre dos centros, se poda una única arista si esta tiene un peso



(a) Grafo original **(b)** Comunidades detectadas

Figura 3.1: Detección de comunidades mediante algoritmo de centros

menor que un cierto porcentaje del peso medio de las aristas del camino. Este proceso debe hacerse de manera ordenada, empezando por los caminos entre los centros más cercanos, y terminando por los de los centros más alejados.

La figura 3.1 muestra los resultados obtenidos de aplicar este algoritmo sobre el grafo empleado para la comparación de los algoritmos de la sección 2.2.2. Se puede observar que este algoritmo no mejora los resultados obtenidos hasta el momento. Este método junta dos comunidades en una sola, además de cometer errores en algunos vértices aislados.

3.4 Algoritmo In-Out

En [36] se propuso un algoritmo de detección de comunidades en grafos dirigidos. Este algoritmo fue desarrollado inicialmente con el fin de ser aplicado en el grafo que modela el tráfico entre grados universitarios en el sistema de acceso a universidades públicas españolas.

Este algoritmo produce dos tipos de comunidades distintas: comunidades donde al menos uno de los vértices puede alcanzar al resto de vértices de la comunidad, o bien, ser alcanzado por el resto. Dado un grafo dirigido $G = (V, E)$, los pasos que sigue el algoritmo In-Out son los siguientes:

1. Se obtienen las matrices de accesibilidad, R, y acceso, Q, de G (sección 2.1).
2. Se obtienen los grados de entrada y salida para cada vértice.
3. Se ordenan los vértices de mayor a menor en función de su grado de salida. En caso de empate, los vértices con mismo grado de salida se ordenan de menor a mayor en función del grado de entrada. Denotamos el conjunto ordenado de vértices como $\{v_1, v_2, ..., v_N\}$
4. Se genera la primera partición de G. Para cada vértice k:
   - Si $v_k \in C_1 \vee C_2 \vee ... \vee C_{k-1}$, pasamos al siguiente vértice.

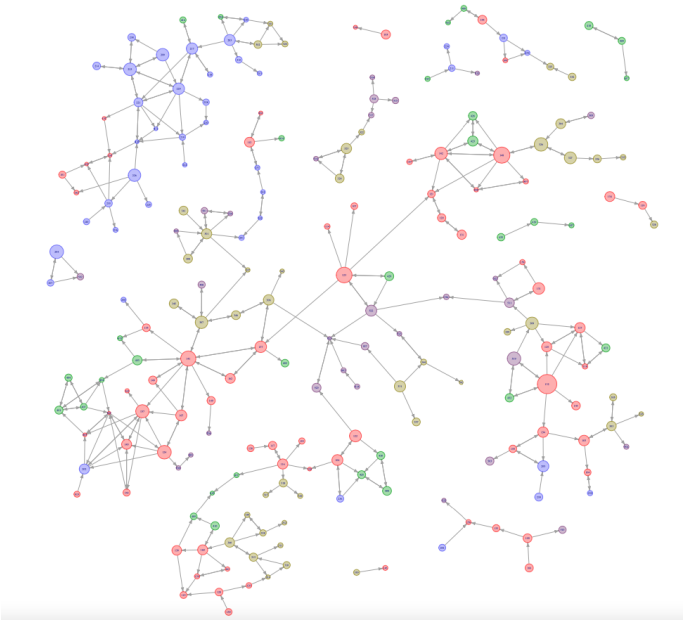


Figura 3.2: Grafo sistema de acceso a la universidad

- En otro caso, definimos una nueva comunidad como $C_k = \{v_j; v_j \in R(v_k), v_j \notin C_1 \vee C_2 \vee ... \vee C_{k-1}\}$

5. Se ordenan los vértices de mayor a menor en función de su grado de entrada. En caso de empate, los vértices con mismo grado de salida se ordenan de menor a mayor en función del grado de salida. Denotamos el conjunto ordenado de vértices como $\{v'_1, v'_2, ..., v'_N\}$
6. Se genera la segunda partición de G. Para cada vértice k:
   - Si $v'_k \in C'_1 \vee C'_2 \vee ... \vee C'_{k-1}$, pasamos al siguiente vértice.
   - En otro caso, definimos una nueva comunidad como $C'_k = \{v'_j; v'_j \in Q(v_k), v'_j \notin C'_1 \vee C'_2 \vee ... \vee C'_{k-1}\}$
7. Se genera el grafo $G_{OUT} = \bigcup_{i \in I} G_{OUT}(C_i)$ como el grafo generado por cada una de las comunidades obtenidas en 4.
8. Se genera el grafo $G_{IN} = \bigcup_{j \in J} G_{IN}(C_j)$ como el grafo generado por cada una de las comunidades obtenidas en 6.

Mediante este algoritmo se obtienen dos particiones distintas del mismo grafo. En el grafo OUT se obtienen comunidades formadas por vértices con los mayores grados de salida y los vértices a los que emite, mientras que en el grafo IN se obtienen comunidades formadas por vértices con los mayores grados de entrada y los vértices de los que recibe.

Como ejemplo de uso se usa el algoritmo en el mismo caso para el que surgió, que es el grafo que modela el sistema de acceso a la universidad pública en España. Este grafo se puede ver en la figura 3.2. Las comunidades obtenidas al aplicar el algoritmo se pueden observar en las figura **??** y 3.4. La modularidad obtenida en el grafo IN es de 0.669, mientras que en el OUT es de 0.827, ambos resultados bastante aceptables. No obstante, el uso de este algoritmo es limitado, y no en todos casos es capaz de obtener buenos resultados.

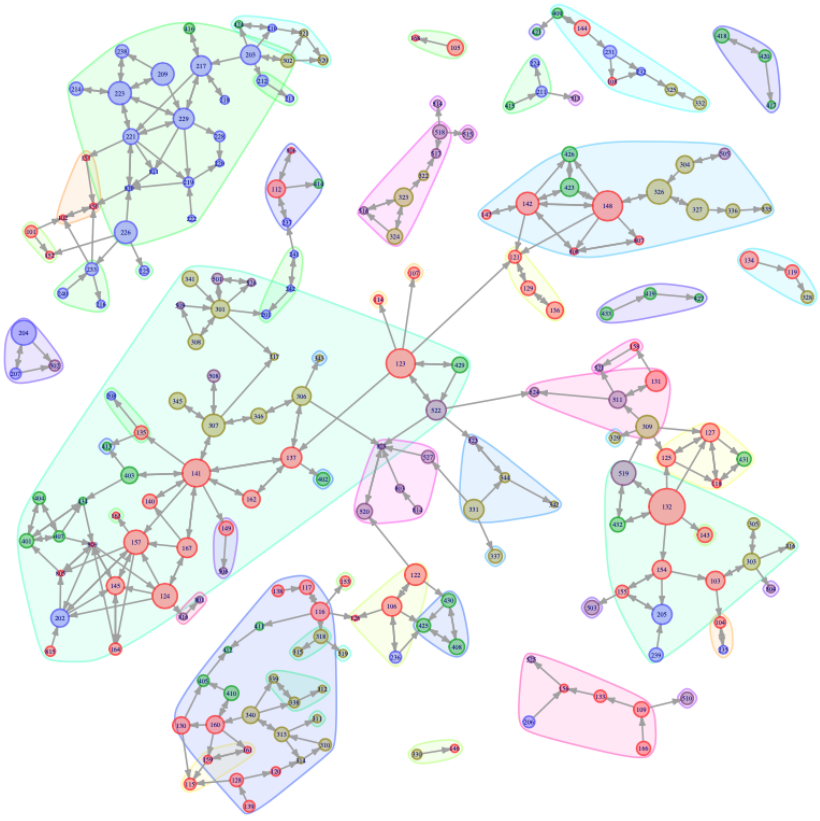


Figura 3.3: Grafo IN

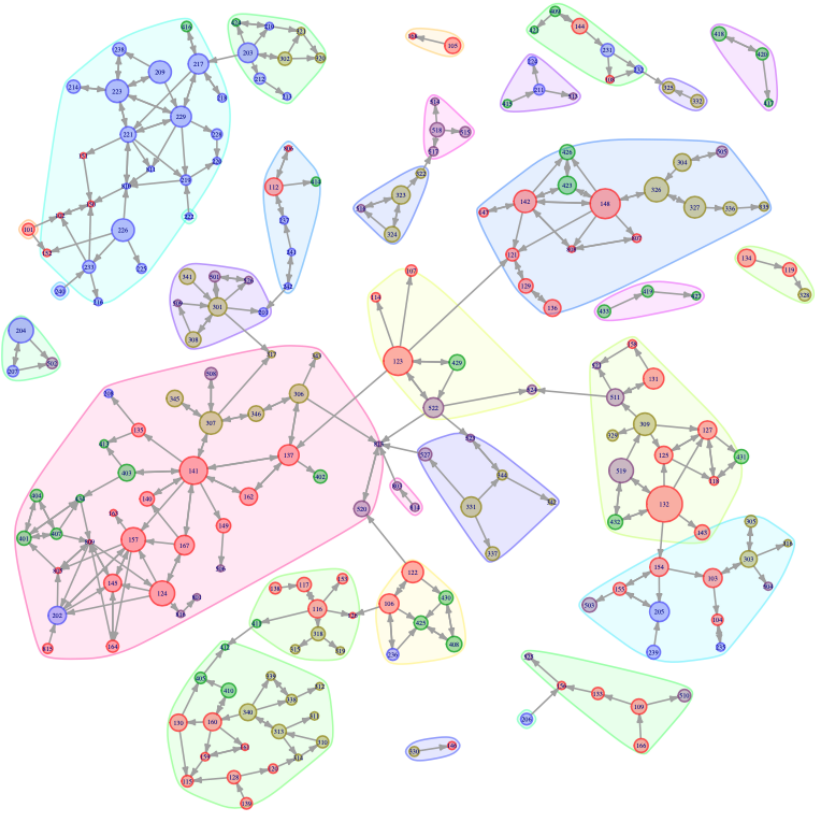


Figura 3.4: Grafo OUT

CAPÍTULO 4

Algoritmos basados en filtros de convolución

Recientemente, con los avances en el desarrollo de Redes Convolucionales sobre Grafos (*Graph Convolutional Networks*, o GCN), se ha extendido la aplicación de productos de convolución en grafos. Los filtros de convolución se emplean en tratamiento de imágenes para, entre otras cosas, detectar bordes en la imagen. De manera análoga, se han propuesto algoritmos de detección de comunidades que usan el producto de convolución con el fin de encontrar aristas que funcionan como «bordes» entre comunidades.

4.1 Convolución en imágenes

En tratamiento de imágenes, una matriz de convolución (también llamada *kernel*, máscara, o filtro de convolución) es una matriz que se usa en una imagen para desenfocar, enfocar, realzar o detectar bordes, entre otras muchas más aplicaciones, mediante la convolución entre esta matriz y la imagen.

La convolución de dos matrices consiste en realizar una multiplicación elemento a elemento de ambas matrices y sumar los productos. A continuación, un ejemplo del cálculo de la convolución de dos matrices:

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} * \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} = (a \cdot 1) + (b \cdot 2) + (c \cdot 3) + (d \cdot 4) + ... + (i \cdot 9)$$

Los *kernels* más comúnmente usados en tratamiento de imágenes son simétricos con respecto a su punto central. Dado un *kernel* de tamaño $n \times n$, se aplica la convolución sobre una imagen superponiendo el centro del *kernel* sobre el píxel a convolucionar, y se calcula el producto de convolución entre el *kernel* y la matriz de la imagen superpuesta. El resultado de esta convolución será el nuevo valor del píxel sobre el que se ha centrado el *kernel*. Este proceso se repite para todos los píxeles de la imagen. Un ejemplo se puede ver en la figura 4.1.

4.2 Detección de bordes en imágenes

Muchos de los procesos en tratamiento de imágenes necesitan la identificación previa de características relevantes en la imagen para la estimación de la estructura y propieda-

1	0	4	2	125	67
8	2	5	4	34	12
20	13	25	15	240	2
76	8	6	6	100	76
34	66	134	223	201	3
255	123	89	55	32	2

Matriz original

*

1	2	1
2	4	2
1	2	1

Kernel

=

110	120	522	984
257	198	755	1360
567	687	1312	1689
1288	1496	1911	1659

Matriz convolucionada

Figura 4.1: Ejemplo de convolución

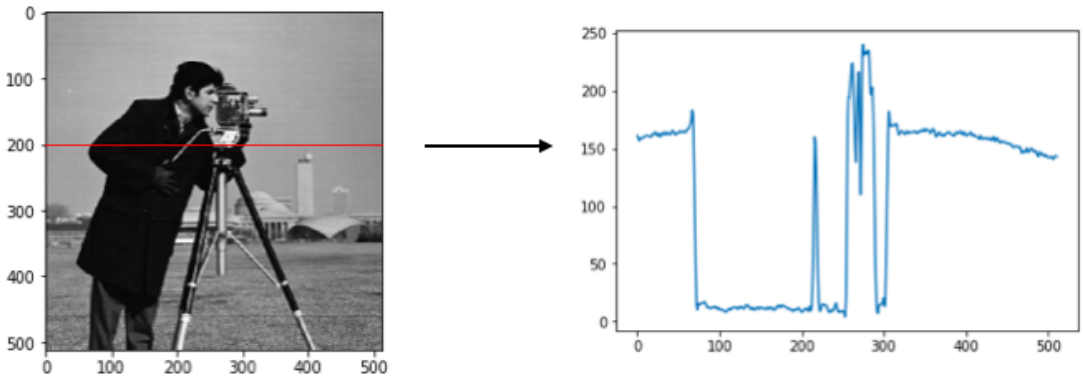


Figura 4.2: Función de intensidad de la franja en rojo de la imagen

des de los objetos en escena. Una de esas características son los bordes. Un borde se puede definir como un cambio local significativo en la imagen, que suele ocurrir en la frontera entre dos regiones distintas de la imagen. La detección de bordes es comúnmente uno de los primeros pasos cuando se pretende recuperar información de una imagen.

4.2.1. Definición de borde

Un borde en una imagen es un cambio local significativo en la intensidad de la imagen, que suele corresponder a una discontinuidad en la intensidad de la imagen o en su primera derivada. Existen distintos tipos de discontinuidades, los más comunes siendo: discontinuidad escalón (*step*, en inglés), donde la intensidad de la imagen cambia abruptamente de un valor en un lado de la discontinuidad a otro distinto en el lado opuesto; y discontinuidad linea, donde la intensidad de la imagen cambia abruptamente para después volver al valor inicial. No obstante, este tipo de bordes no se suelen ver en imágenes, ya que no suelen aparecer discontinuidades instantáneas, sino que los cambios de intensidad ocurren gradualmente en un pequeño intervalo. Un ejemplo de la función de intensidad de una imagen real se puede ver en la figura 4.2. En la gráfica se pueden apreciar cambios significativos y locales en la imagen de intensidad que corresponden a bordes en la imagen, pero también muestra cambios que no se pueden considerar bordes. Estos cambios se deben a ruido en la imagen. Por desgracia, las imágenes reales contienen mucho ruido, y es muy difícil desarrollar métodos de detección de bordes capaces de localizar bordes y ser inmunes al ruido en la imagen.

Los bordes son características muy importantes en una imagen, ya que pueden aportar información muy valiosa acerca de los objetos en escena. Por ejemplo, el contorno de un objeto suele producir bordes escalón en la imagen, al ser normalmente la intensidad del objeto en la imagen mayor que la del fondo.

4.2.2. Definición de detector de bordes

Gradiente

La detección de bordes consiste esencialmente en localizar cambios locales significativos en una imagen. En una imagen unidimensional, un borde está asociado a un máximo o mínimo local en su primera derivada. El gradiente mide los cambios de una función, y una imagen puede ser considerada como una matriz de muestras de una función de intensidad. El gradiente es el equivalente bidimensional de la primera derivada, y está definido como el vector:

$$G[f(x,y)] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

En imágenes, el gradiente se aproxima mediante el cálculo de diferencias. Una simple aproximación numérica del gradiente viene definida por:

$$G_x \cong f[i,j+1] - f[i,j]$$

$$G_y \cong f[i,j] - f[i+1,j]$$

Esta aproximación se puede implementar mediante los siguientes filtros de convolución:

$$G_x = \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \qquad G_y = \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}$$

Fases en la detección de bordes

El esquema general de un algoritmo de detección de bordes tiene tres fases distintas. La primera fase consiste en el filtrado, o suavizado, de la imagen, cuyo objetivo es la reducción de ruido en la imagen para facilitar la detección de bordes. No obstante, el reducir ruido puede conllevar pérdida de información en la imagen, por lo que es necesario usar un método de suavizado que consiga eliminar la mayor cantidad de ruido sin comprometer la conservación de los bordes. Uno de los métodos de suavizado más empleados es la Gaussiana, que viene definida por:

$$g(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{x^2}{2\sigma^2}}$$

donde $\sigma > 0$ es un parámetro de regularización (también llamado escala). La elección de este parámetro es importante, ya que puede afectar considerablemente a las siguientes fases del proceso.

El siguiente paso del proceso es el realce. En esta fase se determinan los cambios de intensidad en la vecindad de un punto. El realce resalta píxeles donde hay un cambio significativo local de intensidad, empleándose para ello el gradiente. Por último, se detectan

(a) Imagen original **(b)** Bordes detectados

Figura 4.3: Ejemplo detección de bordes

los puntos donde hay bordes. Distintos métodos se usan para determinan que puntos son bordes (umbralizado, por ejemplo, es uno de los métodos empledos). La figura 4.3 muestra una imagen en escala de grises y los bordes detectados después de aplicar un método de detección de bordes.

4.2.3. Filtros de primer orden

Los filtros de primer orden calculan la primera derivada para cada punto de la imagen, y, si está se encuentra por encima de cierto umbral, entonces el punto es considerado como borde. En esta sección se introducen los filtros de primer orden más empleados.

Filtro de Roberts

El filtro de Roberts calcula una aproximación al gradiente como:

$$G[f(i,i)] = |f(i,j) - f(i+1,j+1)| + |f(i+1,j) - f(i,j+1)|$$

Usando filtros de convolución esto se convierte en

$$G[f(i,i)] = |G_x| + |G_y|$$

donde G_x y G_y se calculan mediante los *kernels*

$$G_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \qquad G_y = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

El problema de usar *kernels* de tamaño 2×2 es que el gradiente se calcula en un punto intermedio $(i + \frac{1}{2}, j + \frac{1}{2})$. Este problema se puede resolver mediante el uso de *kernels* simétricos.

Filtro de Sobel

Para evitar calcular el gradiente en un punto intermedio entre píxeles, se usa un *kernel* simétrico. El filtro de Sobel usa un *kernel* de tamaño 3×3. En este caso, el gradiente se calcula como

$$M = \sqrt{s_x^2 + s_y^2}$$

Las derivadas parciales, s_x y s_y se calculan usando los siguientes filtros de convolución:

$$s_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad s_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Como se puede ver en los filtros, este método pone más énfasis en los píxeles más cercanos al centro del filtro. Este es uno de los métodos más comúnmente empleados.

Filtro de Prewitt

Este filtro es muy similar al de Sobel, pero sin dar más importancia a los píxeles más cercanos al centro. Los filtros de convolución empleados son:

$$s_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad s_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

4.2.4. Filtros de segundo orden

Los filtros de segundo orden no usan calculan el gradiente para cada punto, sino la segunda derivada. El objetivo de estos filtros es encontrar máximos locales en los valores de gradiente y considerar estos como bordes. Esto significa que en los puntos considerados como bordes habrá un pico en la primera derivada y, de manera equivalente, un cambio de signo en la segunda derivada, por lo que los bordes se detectarán mediante la localización de estos cambios de signo en la segunda derivada de la intensidad de la imagen.

Filtro de la Laplaciana

La Laplaciana es el equivalente bidimensional de la segunda derivada, y se calcula como:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

La segunda derivada de x se puede aproximar de la siguiente manera:

$$\begin{aligned} \frac{\partial^2 f}{\partial x^2} &= \frac{\partial G_x}{\partial x} \\ &= \frac{\partial [f(i,j+1) - f(i,j)]}{\partial x} \\ &= \frac{\partial f(i,j+1)}{\partial x} - \frac{\partial f(i,j)}{\partial x} \\ &= [f(i,j+2) - f(i,j+1)] - [f(i,j+1) - f(i,j)] \\ &= f(i,j+2) - 2f(i,j+1) + f(i,j) \end{aligned}$$

Esta aproximación numérica de la segunda derivada esta centrada en el píxel $(i, j+1)$, de manera que, reemplazando j por $j-1$, se obtiene:

$$\frac{\partial^2 f}{\partial x^2} = f(i, j+1) - 2f(i, j) + f(i, j-1)$$

De manera análoga se puede calcular la segunda derivada de y como:

$$\frac{\partial^2 f}{\partial y^2} = f(i+1, j) - 2f(i, j) + f(i-1, j)$$

Combinando ambas ecuaciones se puede obtener el siguiente filtro que aproxima la Laplaciana:

$$\nabla^2 \approx \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

El filtro de la Laplaciana señala la presencia de un borde cuando el resultado de la convolución cambia de signo (regiones de cero uniformes se ignoran).

Filtro de la Laplaciana de la Gaussiana

El filtro de la Laplaciana no se usa de manera frecuente ya que los filtros que usan dos derivadas se ven más afectados por el ruido en la imagen que los filtros que solo usan la primera. Es por eso que es necesario realizar un suavizado previo al realce de bordes para filtrar el ruido. Es por ello que surge la Laplaciana de la Gaussiana (LoG, por sus siglas en inglés), que combina el suavizado Gaussiano con la Laplaciana para la detección de bordes [37].

En este método, la imagen tiene que ser convolucionada previamente con un filtro Gaussiano, para suavizar la imagen y reducir ruido. De esta manera, puntos aislados de ruido y pequeñas estructuras en la imagen son filtrados. El filtro de la Laplaciana es entonces empleado como aproximación de la segunda derivada para obtener los puntos cuyo gradiente sea máximo local. La Laplaciana de la Gaussiana se define como:

$$\nabla^2 g(x, y) = \left(\frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} \right) e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Un ejemplo de una aproximación del filtro LoG de tamaño 5×5 es el siguiente:

$$\begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$

Como se ha comentado en la sección 4.2.2, el valor del parámetro de regularización σ de la Gaussiana tiene un papel importante en el resultado de la detección de bordes. Un mayor valor de σ resulta en un mejor filtrado del ruido, pero a su vez implica una mayor pérdida de información que puede afectar negativamente a la detección de bordes. Sin embargo, si se usa un valor menor de σ, la imagen puede mantener parte del ruido después del suavizado. Algo similar ocurre con el tamaño del filtro empleado: usar

(a) Imagen original **(b)** LoG con $\sigma = 2$ **(c)** LoG con $\sigma = 4$

Figura 4.4: Ejemplo detección de bordes mediante Laplaciana de la Gaussiana

un filtro pequeño tiene el peligro de detectar falsos bordes en la imagen, mientras que un filtro grande puede no detectar bordes o juntar bordes distintos si estos están lo suficientemente juntos. La figura 4.4 muestra un ejemplo de detección de bordes usando el filtro de la Laplaciana de la Gaussiana con distintos valores de σ. Se puede observar como, para $\sigma = 2$, el método de detección de bordes se ve muy afectado por el ruido en la imagen. No obstante, para $\sigma = 4$, aunque se elimina gran parte del ruido, se pierden algunos detalles, como los edificios del fondo.

4.3 ConvGraph

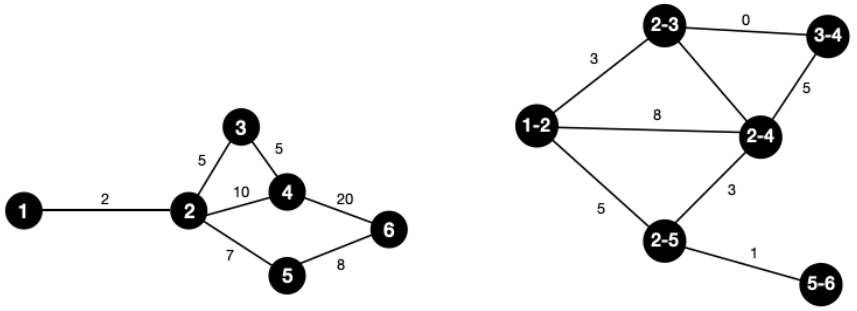
ConvGraph [38] es un algoritmo de detección de comunidades basado en las técnicas de detección de bordes. En este método, se aplica un operador de convolución sobre el grafo con el fin de detectar bordes en las comunidades de este.

4.3.1. Grafo línea

Este método incluye una transformación previa del grafo, G, en un grafo línea, H. En este último, cada vértice representa una relación (u, v) en el grafo original. Si dos de estas relaciones tienen algún vértice en común, se representa como una arista en el grafo línea. De manera formal, dado un grafo ponderado $G = (V, E)$, se define el grafo línea de G como $H = (E, R)$, donde cada vértice $e \in E$ tiene asociado un valor $f(e) \in \mathbb{R}$. Dado dos vértices $e_1 = (u_1, v_1), e_2 = (u_2, v_2) \in E$, habrá una arista entre ambos vértices si las relaciones que representan tiene algún vértice de G en común, esto es, si $\{u_1, v_1\} \cap \{u_2, v_2\} \neq \emptyset$. En la figura 4.5 se puede ver un ejemplo de la transformación de un grafo a su respectivo grafo línea. El peso de una arista (e_1, e_2) se obtiene a partir de la diferencia en valor absoluto de los pesos de las aristas e_1 y e_2 en el grafo original. En este ejemplo, la arista que une los vértices $(1, 2)$ y $(2, 3)$ en el grafo línea tiene peso $5 - 2 = 3$.

4.3.2. Convolución en grafo línea

Como se ha comentado en la sección 4.2, la convolución en imágenes nos permite detectar bordes al localizar cambios bruscos de intensidad. El método ConvGraph aplica los métodos de detección de bordes que se emplean en imágenes en grafos, al considerar equivalentes ambos objetos, donde un píxel equivaldría a un vértice, la intensidad de un píxel al peso de un vértice, y la adyacencia de un píxel por posición a la adyacencia de un vértice por arista. En este algoritmo, dado un grafo G, aplican sobre el grafo línea de



(a) Grafo **(b)** Grafo Línea

Figura 4.5: Ejemplo de transformación de grafo a grafo línea

G, H, el producto de convolución de la Laplaciana de la Gaussiana. Los pasos que sigue el algoritmo son los siguientes:

1. Se transforma el grafo G en su grafo línea H
2. Se aplica el producto de convolución de la Laplaciana de la Gaussiana sobre H
3. Se detecta el borde entre dos comunidades en los vértices en los que haya un cambio de signo
4. Se calcula para cada comunidad el ratio entre los pesos internos de los vértices de la comunidad con respecto al grafo completo, de manera que la comunidad se seleccione como válida si el valor es significativo
5. Las comunidades obtenidas en H se traducen a G

A pesar de dar este algoritmo buenos resultados, los autores remarcan que el algoritmo anterior tiende a no detectar correctamente las comunidades cuando estas son muy densas. Para solucionar este problema, proponen un segundo filtrado sobre el grafo en las comunidades con alta densidad. En este caso, se propone el coeficiente de *clustering* como medida de densidad. El coeficiente de *clustering* para un vértice i se calcula como:

$$c_i = \frac{|e_{jk}|}{k_i(k_i - 1)} \ : \ v_j, v_k \in N_i, \ e_{jk} \in E$$

donde k_i es el grado de salida del vértice i, y $N_i = v_j \ : \ e_{ij} \in E$ es el subconjunto de vértices con aristas cuyo origen es i y su destino es j. El coeficiente de *clustering* de un grafo se calcula como la media del coeficiente de *clustering* de todas sus aristas. El coeficiente de *clustering* se encuentra en un rango entre cero y uno. Los autores del método consideran grafos con un coeficiente de *clustering* como muy densos. Dado un grafo G y el conjunto de comunidades C obtenido en el algoritmo anterior, el segundo filtrado para obtener comunidades muy densas consiste en:

1. Para cada comunidad en C, calcular su densidad mediante el coeficiente de *clustering*.
2. Si la densidad de la comunidad supera un umbral, se vuelve a aplicar el algoritmo anterior únicamente sobre el subgrafo definido por la comunidad y los vértices vecinos a esta.

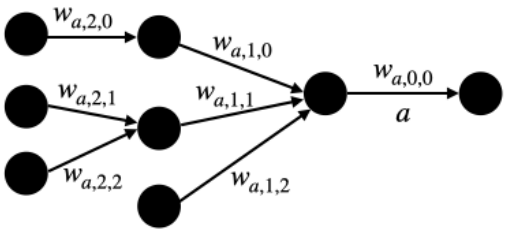


Figura 4.6: Ejemplo identificación aristas con respecto a la arista a

3. Si alguno de los vecinos es añadido en el anterior paso a la comunidad, se considera a este vértice como miembro de la comunidad.

En este método, es necesario ajustar el tamaño del *kernel*, además del parámetro de regularización de la Gaussiana. Según los resultados obtenidos en [38], un kernel de tamaño cinco y un parámetro de regularización entre 0.7 y 0.9 son los que han demostrado mejor rendimiento.

4.4 Otro algoritmo de detección de comunidades basado en convolución

En [39] se propone un método de detección de comunidades donde se emplean técnicas de convolución para la poda de aristas no significativas. Este método pretende mejorar los resultados obtenidos hasta el momento en la detección de comunidades en grafos dirigidos débilmente conectados.

El producto de convolución en grafos es definido como la suma del producto de los pesos de las aristas con un *kernel*, un vector de números reales $[k_0, k_1, ..., k_n]$. Para poder aplicar este producto, el grafo debe poder ser representado como una estructura irregular n-dimensional. Con este fin, se asigna a cada arista un conjunto de índices calculados en base a la distancia a la que está de otras aristas. Al centrarse este método en su aplicación en grafos dirigidos, es necesario recalcar que las distancias entre aristas tienen en cuenta la orientación de estas. Como un ejemplo, la figura 4.6 muestra un grafo donde los índices de las aristas están obtenidos en función de la distancia necesaria para alcanzar la arista a. Las aristas en la figura están etiquetadas como $w_{destino,distancia,id}$, donde el destino es la arista para la que es calculada la distancia, e id es un identificador para diferenciar cada arista del resto de aristas a la misma distancia de a.

El producto de convolución para una arista a consiste en la suma del producto de los pesos de las aristas a una distancia máxima n de a con su respectivo valor en el kernel. La siguiente ecuación muestra el cálculo del producto de convolución de la arista a tomando en consideración las aristas a una distancia máxima de 2.

$$result_a = w_{a,0,0} * k_0 + \sum_i w_{a,1,i} * k_1 + \sum_j w_{a,2,j} * k_2$$

Los autores del método proponen una variación del producto de convolución donde, en vez de usar los pesos de las aristas, se usa el cuadrado de los pesos. Por tanto, el producto de convolución queda de la siguiente manera:

$$result_a = w^2_{a,0,0} * k_0 + \sum_i w^2_{a,1,i} * k_1 + \sum_j w^2_{a,2,j} * k_2$$

En este método, se consideran bordes de las comunidades las aristas cuyo producto de convolución es negativo. Estas aristas son podadas, para después aplicar sobre el grafo resultante un algoritmo de detección de comunidades.

Dado un grafo dirigido ponderado $G = (V, E)$, el algoritmo sigue los siguientes pasos:

1. Para cada arista $a \in E$:
   - Obtención de las aristas a, como máximo, distancia n de la arista a.
   - Obtención del *kernel* $[k_0, k_1, ..., k_n]$. Los autores de este método recomiendan el uso de la Laplaciana de la Gaussiana o del kernel Log.
   - Cálculo del producto de convolución para la arista a mediante el método descrito anteriormente.
2. Poda de las aristas cuyo producto de convolución sea negativo
3. Obtención de las comunidades del grafo podado mediante un algoritmo de detección de comunidades

4.5 Un nuevo algoritmo de detección de comunidades basado en convolución

En este trabajo se propone un nuevo método basado en convolución. Este método es una variación del algoritmo propuesto en [39] y descrito en la sección 4.4. Este nuevo método difiere del anterior fundamentalmente en dos aspectos: la definición de producto de convolución en grafos y la obtención del filtro de convolución.

Teniendo en cuenta la direccionalidad del grafo, el método anterior tenía en cuenta únicamente las aristas a distancia n de la arista a convolucionar. En este nuevo método se amplía el número de aristas a convolucionar teniendo en cuenta también las aristas que están a distancia n desde la arista a convolucionar. Por tanto, no solo se tiene en cuenta el flujo de entrada a la arista a convolucionar, sino que también se tiene en cuenta el flujo de salida. Para aplicar el producto de convolución, se emplean las aristas a una distancia, como máximo, n de o desde la arista a convolucionar. A esta distancia n se le llama distancia de convolución. La figura 4.7 muestra de manera esquemática las arista a a convolucionar y el resto de aristas involucradas en la convolución tomando como distancia de convolución $n = 2$. La nomenclatura es la misma que se emplea en el método anterior ($w_{destino,distancia,id}$). Al igual que en detección de bordes en imágenes la elección del tamaño del *kernel* tiene un efecto considerable en el resultado final, el impacto de la distancia n escogida es importante. Si esta distancia es muy pequeña es más probable que aristas de ruido no se filtren, mientras que si es muy grande puede resultar en una excesiva generalización y, por tanto, una incorrecta identificación de las aristas de ruido.

El cálculo del producto de convolución es idéntico al empleado en el método anterior, excepto en el modo de obtención de los pesos. El producto de convolución se define como

$$\kappa_a(n) = k_0 * w_{a,0,0} + \sum_{i=1}^{n} \sum_{j=1}^{m} k_i * w_{a,i,j}$$

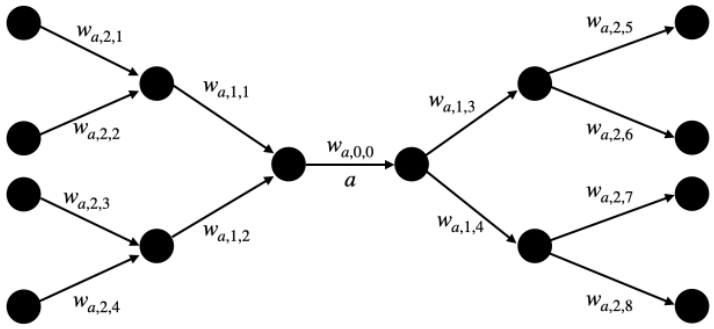


Figura 4.7: Ejemplo identificación aristas con respecto a la arista a

donde κ_a es el producto de convolución de la arista a, y donde $k_i = \nabla^2 g(i,0)$, $\forall i$, $1 \leq i \leq n$, la Laplaciana de la Gaussiana para un parámetro de regularización, σ, dado. Al igual que ocurre con la distancia n, la elección del parámetro σ tiene un efecto importante en el producto de convolución. El peso k_0 se calcula de manera que la suma de todos los pesos k sea igual a cero. Por tanto, k_0 se calcula como

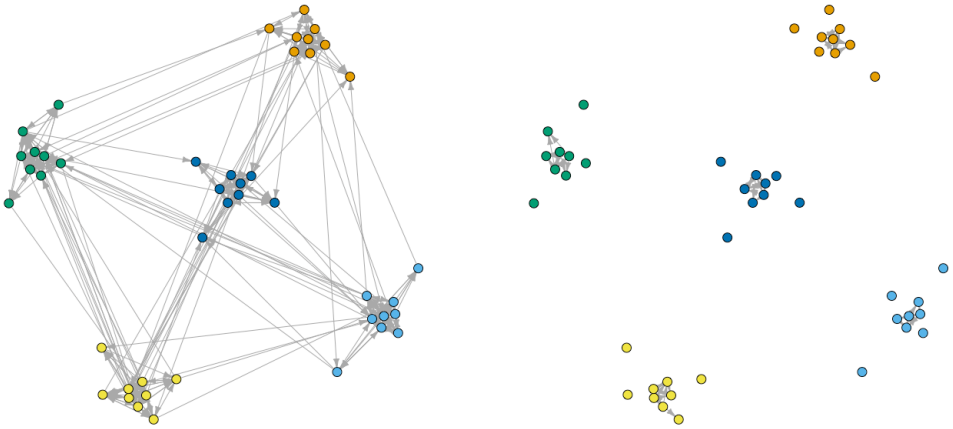
$$k_0 = \sum_{i=1}^{n} \sum_{j=1}^{m} -1 * k_i$$

donde n es la distancia definida anteriormente, y m es el número de aristas a distancia n de, o desde, a. Mediante esta restricción, se trata de emular el funcionamiento de los métodos de detección de bordes empleados en imágenes, ya que los pesos de todos los tipos de filtros de convolución vistos en la sección 4.2 suman cero. Si no fuera así, regiones de la imagen completamente uniformes (sin cambios en su intensidad) tendrían, después de la convolución, un valor distinto de cero, lo que va en contra de la definición de gradiente. Es por eso que esta es una modificación sustancial con respecto al método anterior, donde no se tenía en cuenta este aspecto.

Al igual que en el método anterior, se podan las aristas cuyo producto de convolución sea negativo. Por tanto, dado un grafo dirigido $G = (V, E)$, los pasos que sigue el algoritmo son los siguientes:

1. Se selecciona la distancia de convolución, n, y el parámetro de regularización, σ.
2. Se calcula el producto de convolución $\kappa_a(n)$ para toda arista $a \in E$, calculando los pesos del producto de convolución mediante la Laplaciana de la Gaussiana, usando el parámetro de regularización definido en el paso anterior.
3. Se podan las aristas cuyo producto de convolución sea negativo.
4. Se emplea un algoritmo de detección de comunidades para obtener una partición del grafo podado.

Como ejemplo, se aplica el algoritmo sobre el grafo dirigido de la figura 2.4. Se aplica el producto de convolución, fijando la distancia de convolución como $n = 2$ y el parámetro de regularización como $\sigma = 0{,}55$. La figura 4.8 muestra el grafo después de la poda. En este grafo tan pequeño no es necesario aplicar el algoritmo de detección de comunidades después de la poda, ya que serán las mismas componentes conexas las que compondrán las comunidades. Esto se puede ver en el grafo de la figura 4.9, que muestra las comunidades obtenidas después de aplicar el algoritmo *Walktrap* sobre el grafo podado.



(a) Grafo original **(b)** Grafo podado

Figura 4.8: Resultado poda mediante convolución

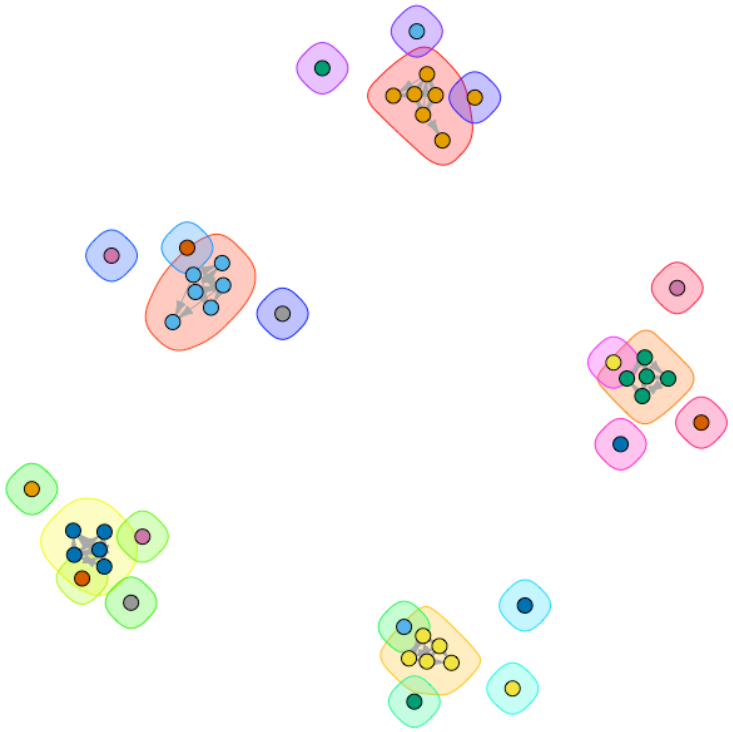


Figura 4.9: Comunidades obtenidas al aplicar *Walktrap*

CAPÍTULO 5
Caso práctico: Flujo de vuelos entre aeropuertos de Estados Unidos

Los sistemas de transporte, junto con otras infraestructuras como la red eléctrica o la de comunicación, son elementos fundamentales en la sociedad y economía actual. Garantizan la disponibilidad de movilidad vital para la calidad de vida de los ciudadanos, además de producir crecimiento económico y empleo. Al igual que el resto de sistemas de transporte, el aéreo tiene un gran impacto a nivel local, nacional, e internacional. Este sistema de transporte es usado diariamente por una gran cantidad de personas. No obstante, el sistema de transporte aéreo es muy vulnerable a factores económicos, políticos o meteorológicos. Por ejemplo, un día de nubes bajas puede reducir el número de aterrizajes por hora en el Aeropuerto Internacional O'Hare (Chicago) de 100 a 72, lo que resulta en retrasos y vuelos cancelados en todo Estados Unidos, con los costes económicos que esto conlleva [40].

El estudio de la estructura de la red aérea de transporte es crucial a la hora de analizar sus características y mecanismo de funcionamiento. No obstante, esta estructura está definida por varios factores: no solo está determinada por las decisiones que toman las aerolíneas para maximizar beneficios, sino también por factores geográficos, políticos y económicos. Esto hace muy complejo el análisis de la estructura actual del sistema de transporte aéreo.

En este proyecto se analiza el sistema de transporte aéreo estadounidense. Los datos que se emplean son una recolección de los vuelos nacionales de pasajeros producidos en diciembre de 2010. Los vértices del grafo que modela el sistema aéreo son los distintos aeropuertos estadounidenses, mientras que las aristas entre ellos representan los vuelos directos entre ambos. Mediante los algoritmos descritos en este proyecto se detectan y analizan las comunidades presentes en este grafo.

5.1 Descripción de los datos

Se dispone de dos conjuntos de datos: un conjunto de datos con información acerca de los aeropuertos, y otro conjunto con los enlaces entre aeropuertos. El primer conjunto está compuesto por cuatro variables: el nombre del aeropuerto, sus coordenadas geográficas (latitud y longitud) y el código de identificación de la Asociación Internacional de Transporte Aéreo (IATA, por sus siglas en inglés), que es único para cada aeropuerto. Un extracto de este conjunto de datos se puede ver en la tabla 5.1. El otro conjunto de datos

Código IATA	Nombre	Latitud	Longitud
ATL	Hartsfield Jackson Atalnta Int. Airpt.	33.6367	-84.428
DEN	Denver Int. Airpt.	39.8617	-104.67
ORD	Chicago O'Hare Int. Airpt.	41.978	-87.904
LAX	Los Angeles Int. Airpt.	33.942	-118.408

Tabla 5.1: Extracto del conjunto de datos de aeropuertos estadounidenses

Código IATA Salida	Código IATA Llegada	Viajes
AUS	HOU	447
PDX	LAX	447
ATL	CHO	419
DFW	RAP	406

Tabla 5.2: Extracto del conjunto de datos de viajes entre aeropuertos

contiene datos acerca de los vuelos entre aeropuertos. Este conjunto de datos solo contiene tres variables: el aeropuerto de salida, el aeropuerto de destino, y el número de vuelos ocurridos en diciembre de 2010. Los aeropuertos se identifican con su código IATA. Un extracto de este otro conjunto se puede ver en la tabla 5.2.

Dado estos conjuntos de datos, la construcción del grafo es muy sencilla. Sea $G = (V, E)$ el grafo que modela el tránsito entre aeropuertos de Estados Unidos, el conjunto de vértices V está formado por los distintos aeropuertos desde donde se produjo algún vuelo en diciembre de 2010; y, dado dos aeropuertos $a_i, a_j \in V$, existe una arista dirigida $(a_i, a_j) \in E$ si se produjo algún vuelo partiendo de a_i con destino a_j. El peso de la arista (a_i, a_j) equivale al número de veces que se ha repetido en el mes el viaje desde el aeropuerto a_i hasta a_j. El grafo resultante se puede ver en la figura 5.1.

El grafo de tráfico aéreo en Estados Unidos está formado por 633 vértices y 7595 aristas. No obstante, no todos los aeropuertos generan el mismo tráfico, sino que de muy pocos aeropuertos son desde los que salen y a los que llegan la mayoría de los vuelos. Esto se puede ver en la distribución de grados, representada en la figura 5.2, que sigue una distribución de ley de potencias, donde la mayoría de vértices tienen un grado muy pequeño, y muy pocos vértices tienen un grado mucho mayor que la media. Lo mismo pasa con los viajes: algunos itinerarios se realizan muchas más veces que otros. Por ejemplo, los aeropuertos de Atlanta o Denver generan mucho más tráfico que los aeropuertos regionales; y el viaje entre los aeropuertos de Austin y Houston es mucho más frecuente que muchos otros itinerarios. Por tanto, el grafo tiene algunas aristas con un peso importante en el sistema de transporte, a la vez que tiene muchas aristas de peso mínimo que no son relevantes en el conjunto global del grafo.

5.2 Detección de comunidades

A continuación, se aplican diversos algoritmos de detección de comunidades sobre el grafo de aeropuertos.

5.2.1. Girvan-Newman

Los resultados obtenidos con Girvan-Newman se pueden observar en la figura 5.3. La modularidad de esta partición es de 0.2, lo que no es un buen resultado. Este algoritmo divide el grafo en 114 comunidades, de las cuales únicamente tres están compuestas por

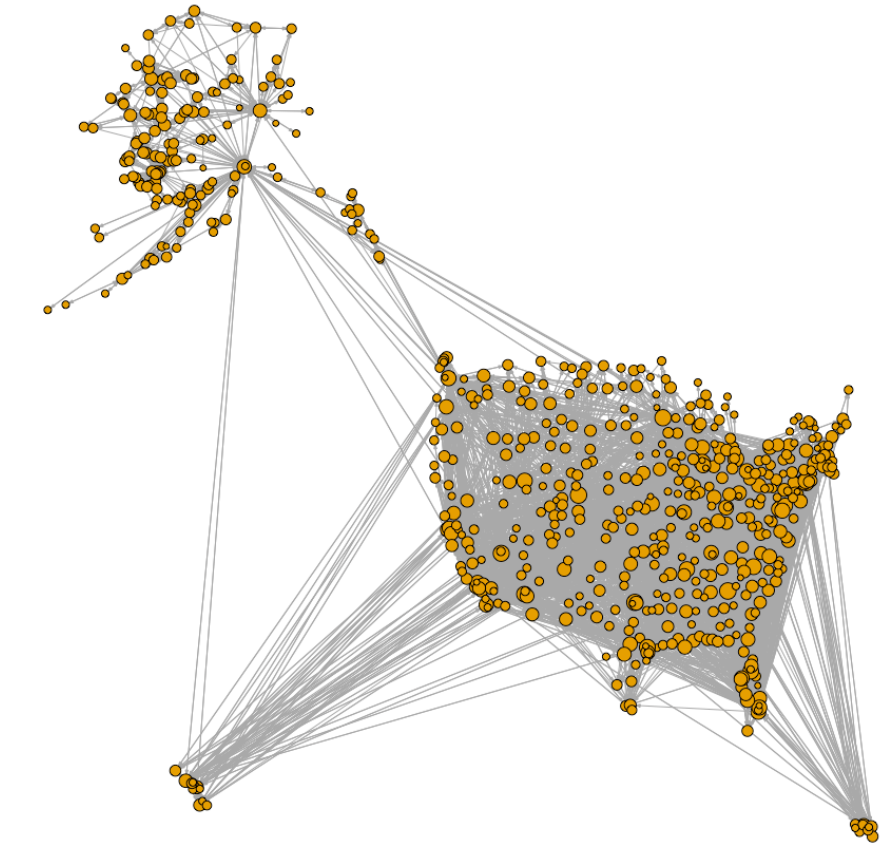
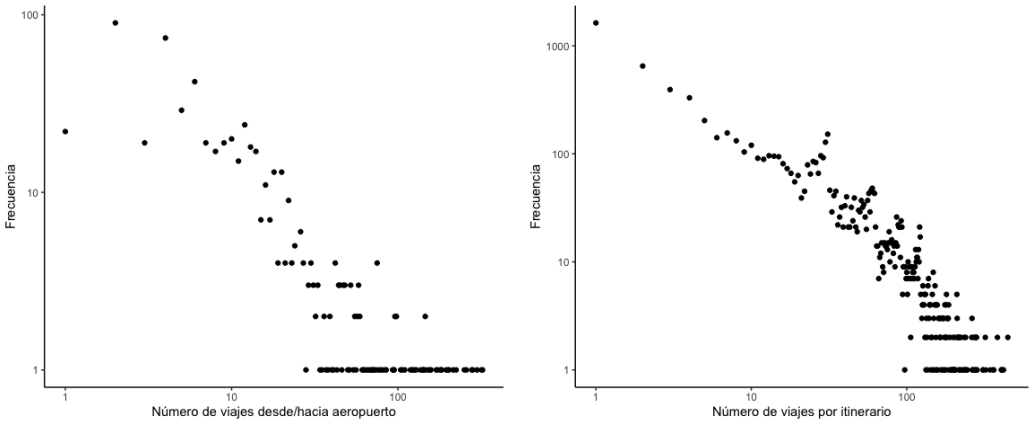


Figura 5.1: Grafo aeropuertos Estados Unidos



(a) Distribución grado **(b)** Distribución itinerarios

Figura 5.2: Distribución tráfico aeropuertos y viajes

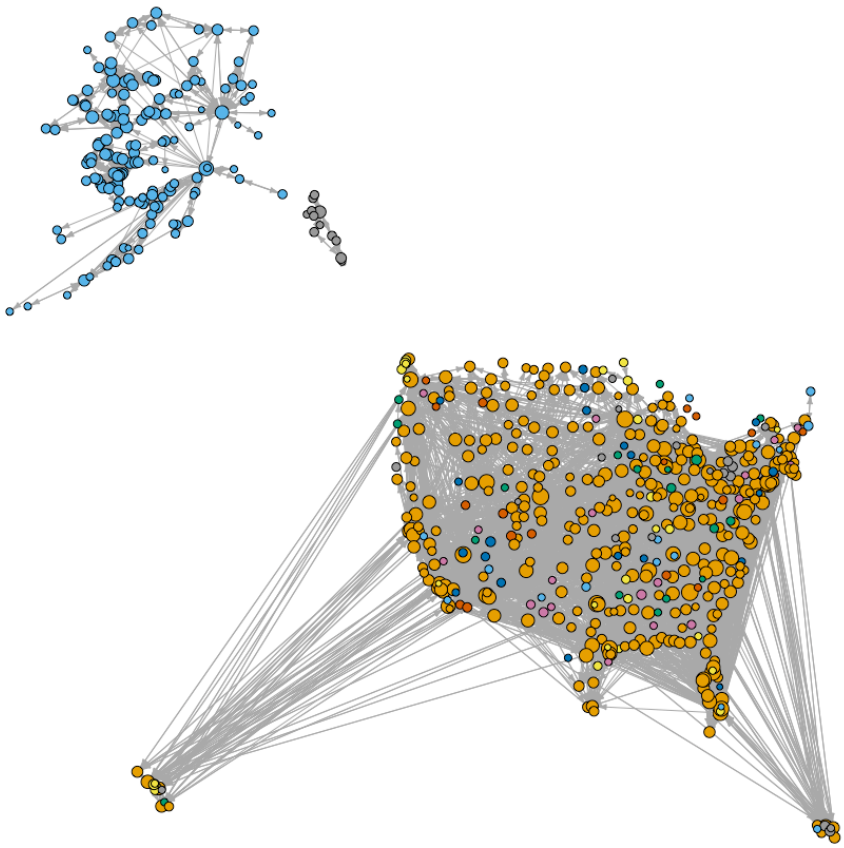


Figura 5.3: Comunidades del grafo de aeropuertos obtenidas mediante Girvan-Newman

más de diez vértices. Las dos comunidades más grandes, de 325 y 141 vértices, dividen los aeropuertos de los 48 estados contiguos (además de Hawái y Puerto Rico) de los aeropuertos ubicados en el estado de Alaska. También detecta una pequeña comunidad al sur de este estado. El resto de comunidades están formadas por vértices aislados, que corresponden a aeropuertos con muy poco tránsito. Aunque está partición del grafo ofrece información, sería de interés obtener comunidades que ofrecieran una división con mayor detalle.

5.2.2. *Walktrap*

Las comunidades obtenidas mediante el algoritmo *Walktrap* se pueden ver en la figura 5.4. Al contrario de lo que ocurría con el algoritmo Girvan-Newman, las comunidades obtenidas mediante *Walktrap* si dividen el territorio de estados contiguos, a pesar de que dividen los aeropuertos de Alaska en varias comunidades. En este caso se han obtenido 77 comunidades. Las dos comunidades más grandes dividen los aeropuertos dependiendo de si se encuentran al este o al oeste del país. Otra comunidad importante se forma con los aeropuertos de la Gran Llanura (comunidad azul claro). La modularidad vuelve a ser baja, de 0.29, por lo que se prueba con otros algoritmos para tratar de mejorar los resultados.

5.2.3. Convolución

Con el fin de eliminar aristas de poca importancia en el grafo que puedan estar afectando negativamente en los resultados de detección de comunidades, se aplica la poda

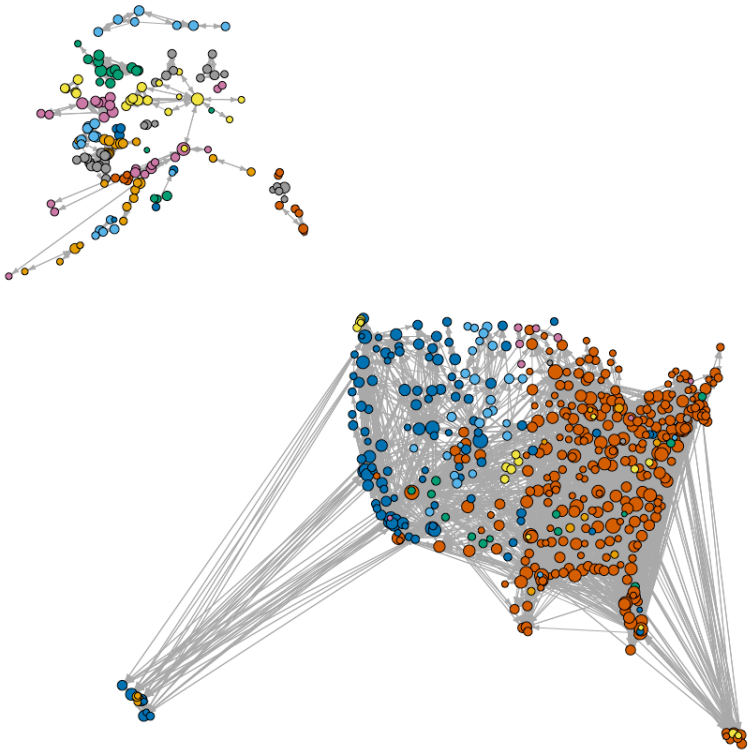


Figura 5.4: Comunidades del grafo de aeropuertos obtenidas mediante *Walktrap*

mediante convolución utilizando el método descrito en la sección 4.5. Para aplicar el producto de convolución sobre el grafo se fija el parámetro de regularización para el cálculo de la Laplaciana de la Gaussiana en $\sigma = 0{,}6$ y la distancia de convolución en $n = 2$, ya que es con estos parámetros con los que mejor resultados se han obtenido. El grafo después de aplicar la convolución y podar las aristas cuyo producto de convolución sea negativo se puede ver en la figura 5.5. En total se han podado 5407 aristas, un 71.2 % del total.

Las comunidades obtenidas aplicando el algoritmo *Walktrap* sobre el grafo podado se pueden ver en la figura 5.6. A pesar de haber podado el grafo, los resultados obtenidos no difieren demasiado de los conseguidos en el grafo original. Se sigue apreciando la diferencia entre ambas costas, y, al igual que ocurría con *Walktrap*, los aeropuertos del estado de Alaska se disgregan en varias comunidades. No obstante, hay algunos resultados de interés. Por ejemplo, se pueden ver algunas comunidades pequeñas de aeropuertos muy juntos entre sí, que corresponden a aeropuertos regionales donde hay mucho más tráfico entre ellos que entre aeropuertos más lejanos. Entre ellos está, por ejemplo, la comunidad formada por el *Alliance Municipal Airport*, el *Chadron Municipal Airport*, el *Dickinson Theodore Roosevel Regional Airport* y el *Pierre Regional Airport*, todos ellos aeropuertos ubicados en el estado de Nebraska y pertenecientes al *Essential Air Service*, un programa gubernamental que trata de garantizar el servicio de vuelos comerciales en pequeñas comunidades del país. También se pueden ver en el grafo comunidades que no tienen sentido geográficamente: algunas de estas comunidades están formadas por aeropuertos privados o bases militares aéreas.

La modularidad obtenida con esta partición es de 0.3. Aunque mejora ligeramente los resultados obtenidos con los dos métodos anteriores, lo cierto es que se queda lejos de un resultado satisfactorio. No obstante, algunos resultados interesantes se han obtenido, que pueden ser de utilidad para las administraciones y aerolíneas.

Figura 5.5: Grafo de aeropuertos podado

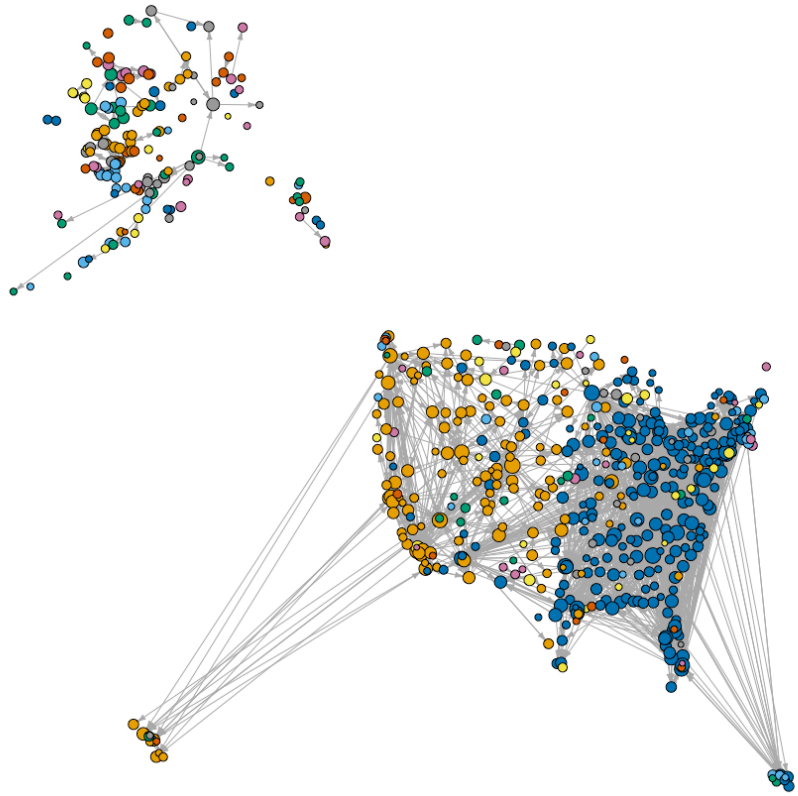


Figura 5.6: Comunidades obtenidas mediante *Walktrap* en el grafo podado

CAPÍTULO 6

Caso práctico: Sistema de acceso a la universidad pública en España

Los interesados en estudiar un grado en alguna de las universidades públicas españolas, han de pasar por un complejo sistema de acceso. Existen diversas formas de acceder a la universidad, que dependen de las características el solicitante (por ejemplo, no es el mismo proceso de acceso para los solicitantes que recién han acabado sus estudios de Bachillerato que para los solicitantes con una titulación universitaria previa). No obstante, este trabajo se centra en el sistema de acceso para los solicitantes que acaban Bachillerato, al ser este el tipo de solicitante más común y para el que más plazas se reserva en los grados en universidades públicas.

Los estudiantes que desean continuar sus estudios en el sistema universitario público español han de realizar un examen que evalúa sus conocimientos en áreas comunes para todos los solicitantes, y en algunas áreas específicas de interés para la rama de estudios que pretenden cursar en la universidad (es necesario recalcar que el proceso para acceder a universidades privadas depende de la propia universidad y no está ligado al sistema de acceso a universidades públicas). La nota de este examen, junto a las notas obtenidas durante los dos años de Bachillerato, conforman la nota de acceso a la universidad.

Cada curso, las universidades públicas ofertan un número limitado de plazas para cada uno de los grados. Por tanto, si un grado tiene más solicitudes que plazas disponibles, se acepta a los solicitantes con mayor nota de acceso hasta agotar las plazas. Es por ello que en la solicitud de acceso los solicitantes deban cumplimentar una lista ordenada de grados según su preferencia, de modo que, si no es admitido en el primer grado en orden de preferencia, se pase al segundo, y así sucesivamente hasta encontrar un grado en el que el solicitante sea aceptado. No obstante, el solicitante es añadido a una lista de espera en cada uno de los grados en los que es rechazado. De esta forma, si se liberaran plazas en alguno de estos, los solicitantes rechazados inicialmente podrían reclamar una plaza en ese grado. Es importante recalcar que, aunque la nota de acceso sea válida para estudiar en cualquier comunidad autónoma, es necesario realizar una solicitud por cada comunidad autónoma en la que el solicitante esté interesado en estudiar (por ejemplo, un solicitante puede solicitar el acceso en grados ofertados en las universidades públicas de la Comunidad Valenciana y, en otra solicitud, hacer lo mismo para universidades de la Comunidad de Madrid). Esta es una de las razones por las que las listas de espera se muevan tanto después de la primera asignación: los solicitantes que han solicitado el acceso a universidades en distintas comunidades autónomas se les es asignado un grado en cada una de estas, por lo que eventualmente tienen que rechazar su plaza en los grados que finalmente no vayan a cursar.

Código	Universidad	Nombre
116	18	Grado en Historia
117	18	Grado en Historia del Arte
118	18	Grado en Enfermería
119	18	Grado en Matemáticas
120	18	Grado en Filología Catalana

Tabla 6.1: Extracto del conjunto de datos de grados universitarios

Es de interés para las universidades y administraciones públicas el análisis del flujo de estudiantes entre grados, es decir, en que grados acaban matriculados los que solicitaron un determinado grado como primera opción. Esto puede ayudar a la decisión del número de plazas a ofertar en cada grado, reestructurar los grados ofertados, ofrecer nuevos grados, o incluso eliminar grados obsoletos. En este trabajo se estudia este problema bajo el enfoque de la teoría de grafos. A partir de los datos disponibles sobre la adjudicación de plazas en el año 2014 en la Comunidad Valenciana, se construye un grafo de grados universitarios y el tráfico de solicitantes entre ellos. Después, empleando los algoritmos de detección de comunidades presentados en capítulos anteriores, se estudia la estructura comunitaria en el grafo resultante y se obtienen conclusiones a partir de los resultados obtenidos.

6.1 Descripción de los datos

Para este trabajo se parte de dos conjuntos de datos. El primero contiene una descripción de los distintos grados ofertados en las universidades públicas de la Comunidad Valenciana en el año 2014. Este conjunto de datos contiene tres datos básicos para identificar los grados. Estas tres variables son: un código único de identificación, un código de identificación de la universidad en la que es ofertada el grado, y el nombre del grado. Un extracto del conjunto de datos se puede ver en la tabla 6.1. Un listado con todos los grados ofertados en 2014, la universidad en las que fueron ofertados y el código de identificación se puede encontrar en el Apéndice A.

Las cinco universidades públicas de la Comunidad Valenciana se identifican con los códigos 1, 18, 27, 40 y 55, que corresponden a, respectivamente, la Universidad de Alicante (UA), la Universidad de Valencia (UV), la Universidad Politécnica de Valencia (UPV), la Universidad "Jaume I" de Castellón (UJI) y la Universidad "Miguel Hernández" de Elche (UMH).

El segundo conjunto de datos contiene información acerca de las solicitudes que se realizaron el año 2014 para realizar estudios universitarios en las universidades de la Comunidad Valenciana. Las variables son las siguientes: un código de identificación de la solicitud, cinco variables que indican el código de las titulaciones escogidas en los primeros cinco puestos de la lista de preferencias, y la nota de acceso a la universidad, entre 5 y 10. Un extracto de este conjunto de datos se puede ver en la tabla 6.2. Aunque es posible escoger más de cinco grados en la lista de preferencia, en este conjunto de datos solo se tienen en cuenta los cinco primeros ya que el número de alumnos a los que se les asigna un grado fuera de los cinco primeros es su lista de preferencias es insignificante.

En 2014 se ofertaron en total 215 grados en las cinco universidades públicas de la Comunidad Valenciana, y se recibieron 16054 solicitudes de acceso. No todos los grados ofertados son igual de solicitados: unos pocos grados reciben muchas solicitudes, mientras que algunos no reciben ninguna. El histograma de la figura 6.1 muestra la distribución de la frecuencia con la que un grado es solicitado en primera opción. El número

Id	Código 1	Código 2	Código 3	Código 4	Código 5	Nota acceso
1	231	108	232	421	325	6.1
2	223	427	216	221	238	7.8
3	137	226	236	106	157	6.2
4	240	233	226	217	221	7.1
5	124	202	145	167	148	9.6

Tabla 6.2: Extracto del conjunto de datos de solicitudes de acceso

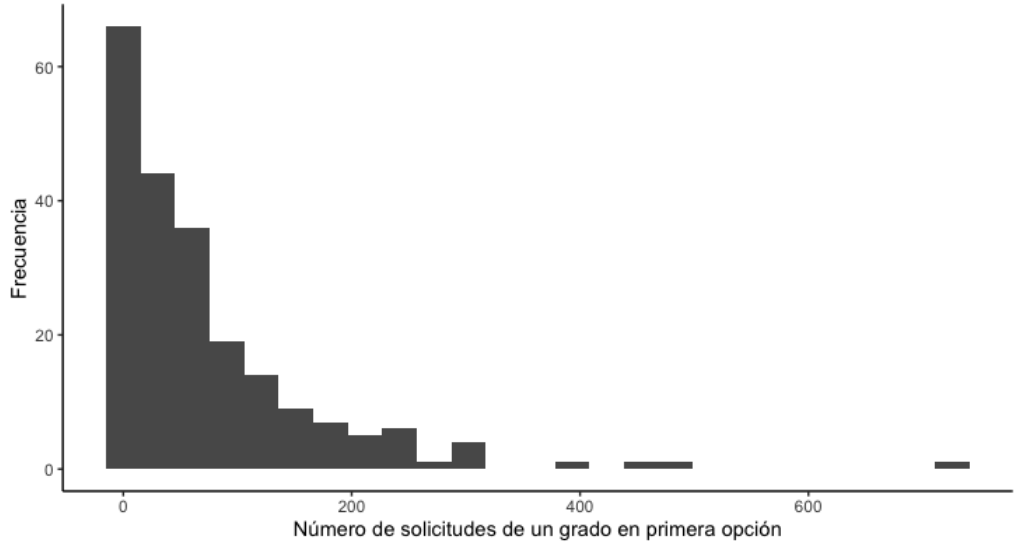


Figura 6.1: Histograma frecuencia solicitudes en primera opción

de grados disminuye exponencialmente con el aumento de solicitudes: muchos grados reciben muy pocas solicitudes en primera opción, mientras que muy pocos grados son muy solicitados. Entre estos grados están el grado en Medicina, Psicología o Derecho. Algunos de estos grados reciben muchísimas más solicitudes que plazas ofertadas (por ejemplo, en 2014 se recibieron 726 solicitudes para el grado en Medicina en la Universidad de Valencia para 320 plazas ofertadas), lo que hace que muchos de estos solicitantes tengan que matricularse finalmente en alguno de los grados escogidos en un orden más bajo de preferencia.

6.2 Preprocesamiento de los datos

Es necesario realizar un preparación de los datos para poder construir el grafo. Este grafo será dirigido ponderado, donde los vértices serán los grados ofertados por las universidades públicas de la Comunidad Valenciana, y una arista que sale del grado A y llega al grado B representa el flujo de estudiantes que escogieron A como opción preferente y que finalmente acabaron matriculados en B, donde B es el siguiente grado a A en orden de preferencia del solicitante. En [36] se propone un algoritmo para la generación del grafo que modela el sistema de acceso a la universidad.

El conjunto de datos no dispone de información acerca de en que grado se matriculó cada uno de los solicitantes. En cambio, si se dispone de las probabilidades de que un solicitante se matricule en el grado en la posición i en orden de preferencia (por ejemplo, la probabilidad de ser aceptado en el grado en primer orden de preferencia es del 57 %). La probabilidad de ser aceptado en uno de los grados en los cinco primeros puestos de la lista de preferencia es mayor que el 90 %, por lo que solo se tienen en cuenta los grados

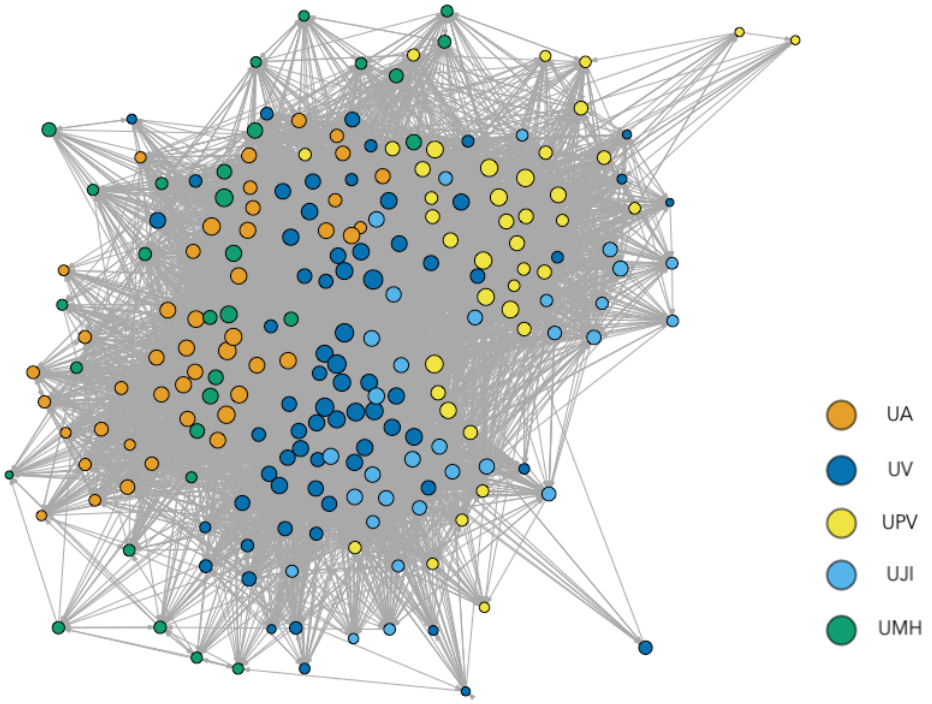


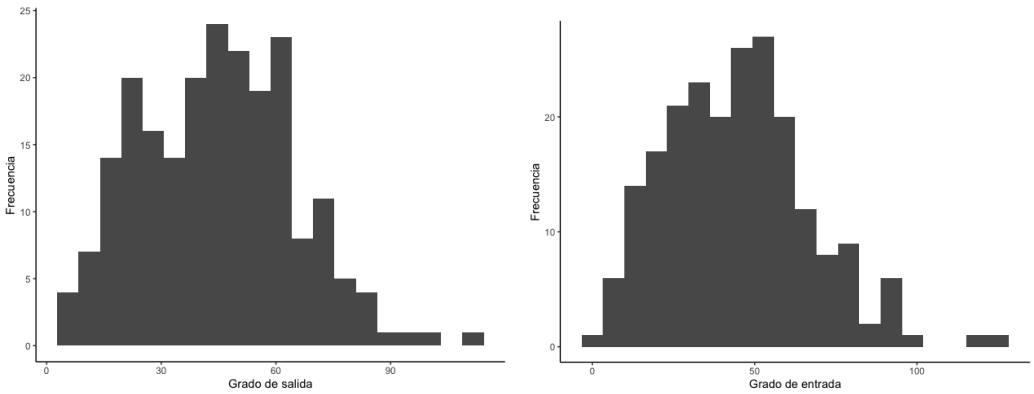
Figura 6.2: Grafo Sistema de acceso a la universidad en la Comunidad Valenciana

de cada solicitud en estos cinco primeros puestos. Los pesos de las aristas se obtienen a partir de estas probabilidades. Sea el grafo $G = (V, E)$ el grafo que modela el sistema de acceso a la universidad, y sea el vector $f = (f_1, f_2, ..., f_5)$ el vector de probabilidades de que una solicitud sea aceptada en el grado en orden n en su lista de preferencia, los pasos que sigue el preprocesamiento son los siguientes:

1. Se define V como el conjunto de grados ofertados por las universidades públicas de la Comunidad Valenciana.
2. Se inicializa el conjunto de vértices E como conjunto vacío.
3. Para cada solicitud $i \in S$, donde S es el conjunto de solicitudes
   - Dados los grados $(v_{i,1}, v_{i,2}, ..., v_{i,5})$, donde el grado $v_{i,n}$ es el grado escogido en orden n en la lista de preferencia
   - Si la arista $(v_{i,n}, v_{i,n+1}) \in E$, se incrementa su peso en f_{n+1}
   - Si la arista $(v_{i,n}, v_{i,n+1}) \notin E$, se añade al conjunto de aristas y se inicializa su peso en f_{n+1}

Una representación del grafo obtenido se puede ver en la figura 6.2. El tamaño de cada vértice viene determinado por el número de solicitudes en primera opción recibidas en el grado asociado.

El grafo resultante contiene 215 vértices (uno por cada grado ofertado) y 9615 aristas. El grafo está compuesto por una única componente conexa. El diámetro del grafo, que equivale a la mayor distancia entre dos vértices del grafo, es de 3; y la longitud media de los caminos más cortos entre vértices es de 1.723. La distribución de los grados de entrada y salida se pueden ver en los histogramas de la figura 6.3. El grado medio de



(a) Grados de entrada **(b)** Grados de salida

Figura 6.3: Distribución grados de entrada y salida

entrada y salida es el mismo, 44.72. Como se puede ver en los histogramas de la figura 6.3, la distribución de ambos grados está centrada en la media. Los grados en Psicología y en Maestro de Educación Primaria son los que mayor grado de entrada y salida tienen en el grafo.

6.3 Detección de comunidades

En este trabajo se aplican diversos algoritmos de detección de comunidades sobre el grafo modelando el sistema de acceso a la universidad.

6.3.1. Girvan-Newman

Las comunidades obtenidas mediante el algoritmo Girvan-Newman se pueden ver en la figura 6.4. Mediante este algoritmo se detectan 146 comunidades. La comunidad más grande contiene 64 vértices, mientras que el resto de comunidades son mucho más pequeñas, siendo la mayoría comunidades aisladas formadas por un solo vértice. La modularidad de esta partición es de 0.027. Los resultados obtenidos con este algoritmo no son aceptables, por lo que es necesario buscar otros métodos de detección de comunidades que devuelvan mejores resultados.

6.3.2. Walktrap

El algoritmo *Walktrap* devuelve mejores resultados que los obtenidos mediante Girvan-Newman. La figura 6.5 muestra los resultados obtenidos con este algoritmo. La modularidad de esta partición es de 0.328, que mejora la obtenida con Girvan-Newman, aunque es un resultado mejorable. Este método divide el grafo en cuatro comunidades, formadas por 71, 63, 43 y 40 vértices, respectivamente. Estas cuatro comunidades corresponden a cuatro áreas de estudios distintas: ingenierías, ciencias naturales y de la salud, ciencias sociales y humanidades. Estos resultados muestran un correcto funcionamiento del algoritmo.

6.3.3. Convolución

Dados los resultados obtenidos con los métodos de detección de comunidades anteriores, se puede intentar una mejora de estos aplicando una convolución previa al grafo

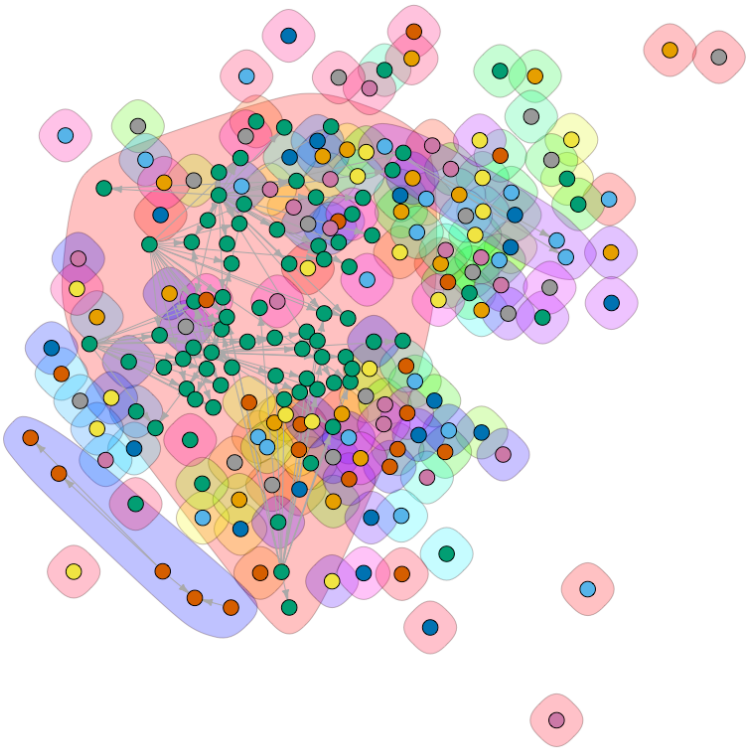


Figura 6.4: Comunidades detectadas con Girvan-Newman

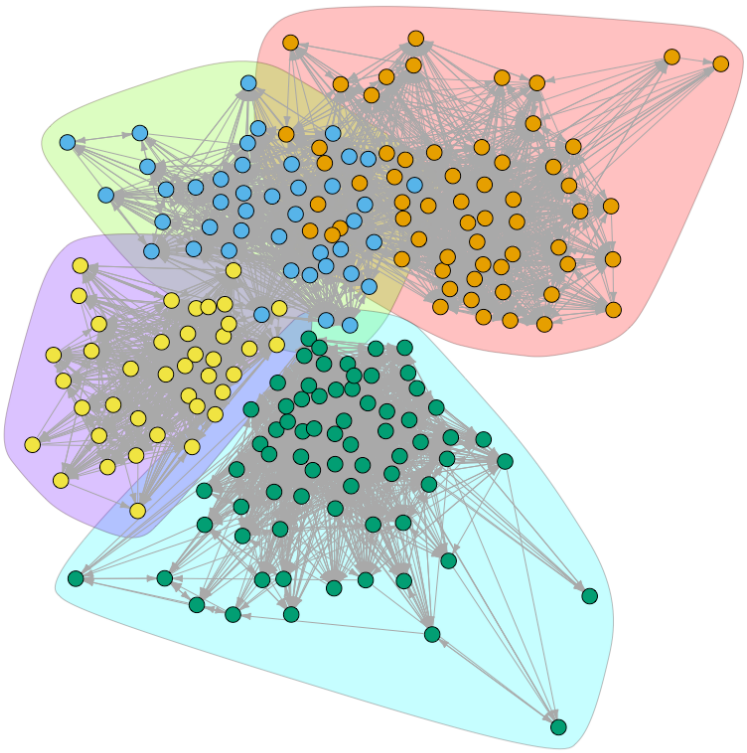


Figura 6.5: Comunidades detectadas con *Walktrap*

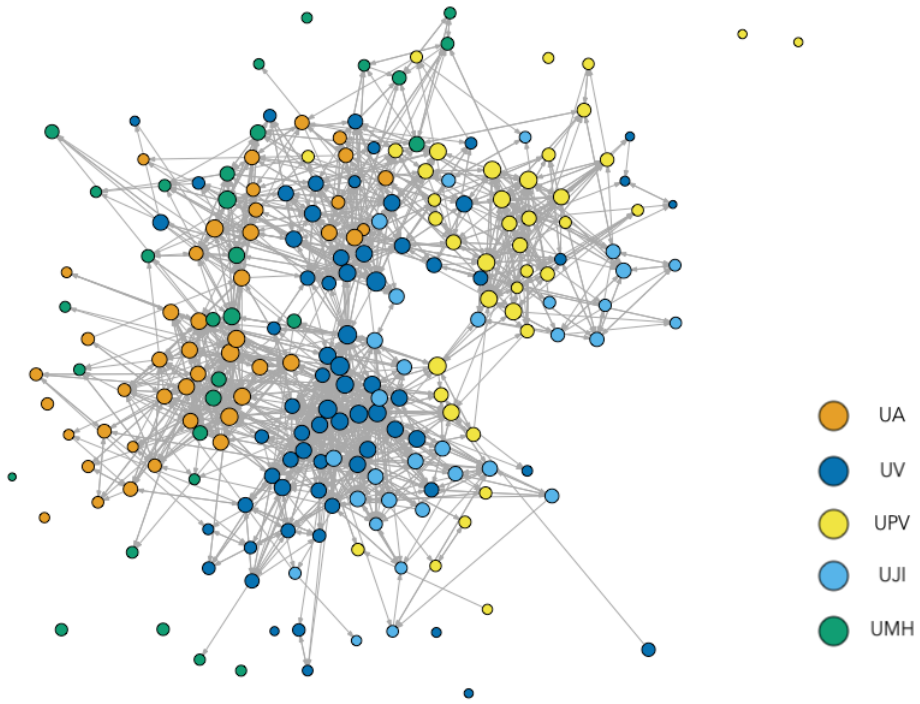


Figura 6.6: Grafo podado mediante convolución con $\sigma = 0{,}5$

para la poda de las aristas de ruido. Para realizar esto, usamos el método propuesto en la sección 4.5. Para el cálculo de la Laplaciana de la Gaussiana necesario para la obtención del *kernel* para la convolución, usamos como parámetro de regularización $\sigma = 0{,}5$, ya que es con el que mejores resultados se han obtenido. La distancia de convolución se fija en $n = 2$. Después de aplicar la convolución, se podan las aristas cuyo producto de convolución es negativo. El grafo podado después de aplicar la convolución se puede ver en la figura 6.6. De las 9615 aristas en el grafo original, se podan 7924 aristas, esto es, se eliminan un 82.41 % de las aristas del grafo. Esta poda divide el grafo original, que se componía por una única componente conexa, en 11 componentes distintas, donde la más grande está formada por 204 vértices, y el resto son componentes de uno o dos vértices.

El algoritmo *Walktrap* es aplicado sobre el grafo podado, ya que según los resultados previos es el método que mejor funciona sobre este grafo. La partición del grafo podado obtenida mediante este algoritmo se pueden ver en la figura 6.7. La modularidad de esta partición calculada con respecto al grafo original es de 0.24, pero con respecto al grafo podado es de 0.59, lo que se considera un resultado aceptable.

El algoritmo detecta 21 comunidades, de las cuales 12 son comunidades compuestas por uno o dos vértices. Estas comunidades pequeñas se pueden ver en la figura 6.8, donde los vértices están coloreados según la universidad en la que se imparte. Estas comunidades están formadas por grados muy poco solicitados. Entre ellos están, por ejemplo, el grado en Ingeniería en Geomática y Topografía de la UPV, o el grado en Información y Documentación de la UV, que recibieron 6 y 2 solicitudes en primera opción, respectivamente. Muchos de estos grados se imparten en otros campus fuera de los principales. Por ejemplo, los grados en Maestro de Educación Infantil y en Maestro de Educación Primaria que aparecen señalados en la figura 6.8 son los impartidos en el Centro Florida Universitaria, y no en la Facultad de Magisterio del campus de Tarongers en Valencia. Lo mismo ocurre con los grados en Administración y Dirección de Empresas (grado 805) y en Finanzas y Contabilidad (grado 809). Que el método genere comunidades aisladas

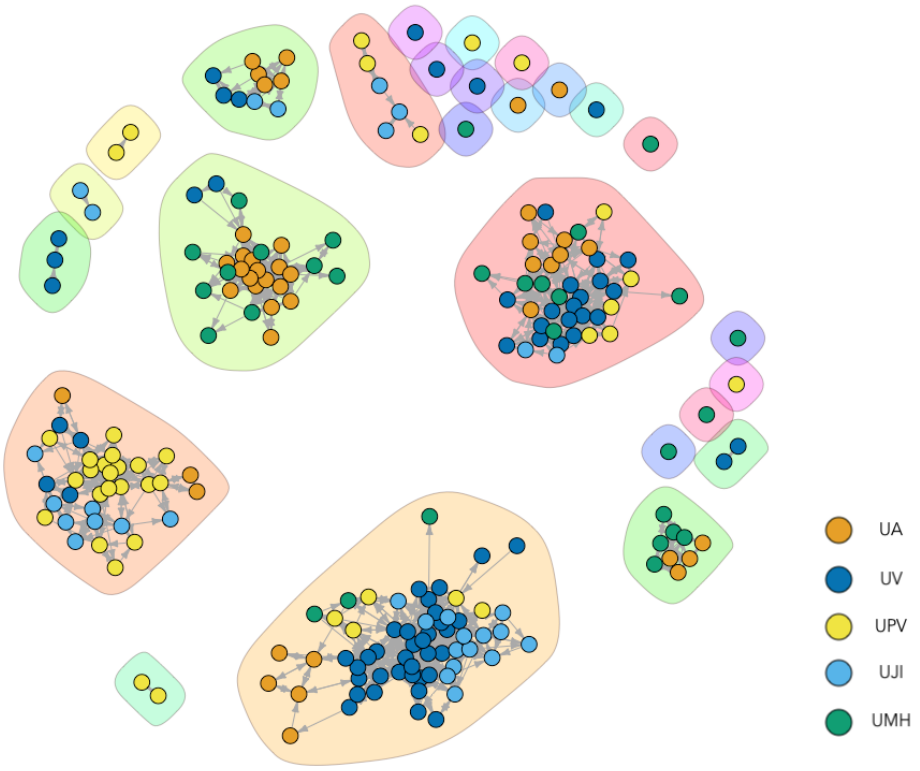


Figura 6.7: Detección de comunidades en el grafo podado mediante *Walktrap*

para estos grados indica que el tráfico de solicitantes en estos grados es muy pequeño, sino nulo, por lo que se podría considerar la viabilidad de su conservación en el sistema universitario o la abolición de estos grados.

El algoritmo detecta también algunas comunidades un poco más grandes, formadas por entre 3 y 5 vértices. Estas comunidades se pueden ver en la figura 6.9. Una de estas comunidades está formada por los grados en Ingeniería Forestal y Agroalimentaria que se ofertan en la Comunidad Valenciana, entre los que se encuentran, por ejemplo, el grado en Ingeniería Forestal y del Medio Natural (grado 224) y en Ingeniería Agroalimentaria y del Medio Rural (grado 211) ofertados por la UPV, o el grado en Ingeniería Agroalimentaria y Agroambiental (513) ofertado por la UMH. Otra de las comunidades se conforma por los grados en Administración y Dirección de Empresas y Gestión Turística ofertados por la UPV y la UV. No obstante, estos grados son los que se imparten fuera de los campus de ambas universidades en Valencia. El último grupo es el compuesto por los grados en Ingeniería Electrónica y Telemática impartidos en la UV. Como ocurría con los grados de las comunidades anteriores, todos los grados que componen las comunidades recién descritas son grados poco solicitados en primera opción.

La figura 6.10 muestra otras dos comunidades de mayor tamaño que las descritas hasta ahora. Una de ellas está compuesta por los grados del área de Filología, Traducción y Comunicación que se ofertan en la UV, la UJI y la UA. Entre estos grados se encuentra el grado en Traducción y Mediación Interlingüística (grado 160) o el grado en Estudios Ingleses (130), ambos impartidos en la UV. La otra comunidad la componen las ingenierías ofertadas en las universidades alicantinas, esto es, en la UA y la UMH.

Las cuatro comunidades restantes son las más grandes de las detectadas. Cada una de estas comunidades engloba los grados de distintas áreas de estudio: ingeniería, ciencias de la naturaleza y de la salud, y ciencias sociales y humanidades (dos de las comunida-

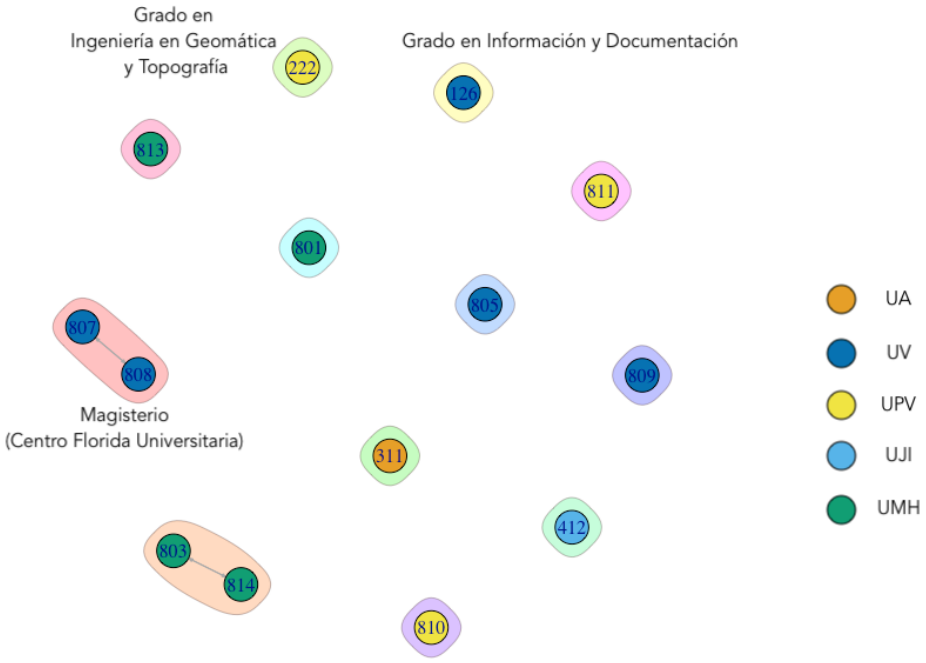


Figura 6.8: Comunidades pequeñas del grafo de acceso a la universidad

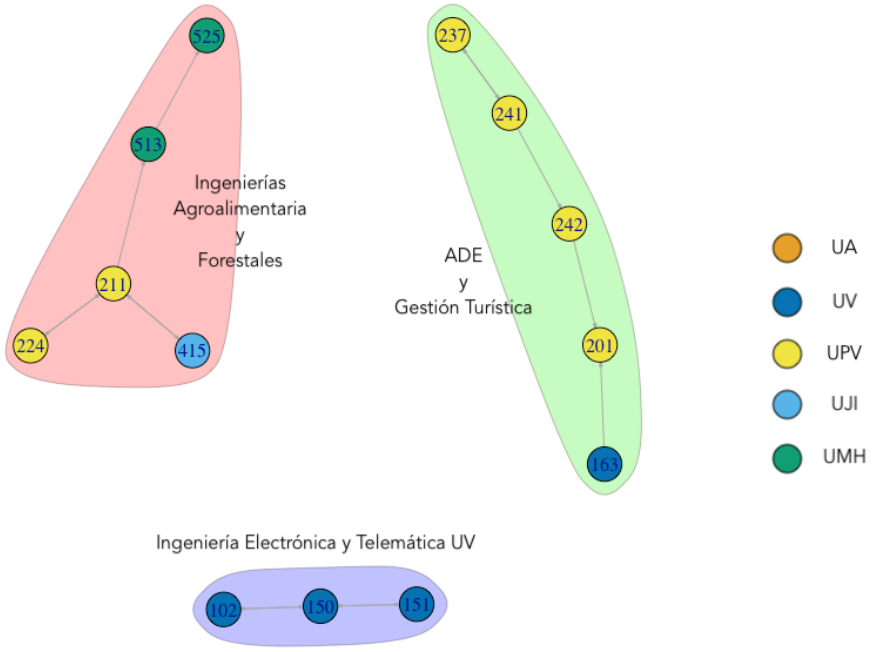


Figura 6.9: Comunidades de tamaño medio del grafo de acceso a la universidad

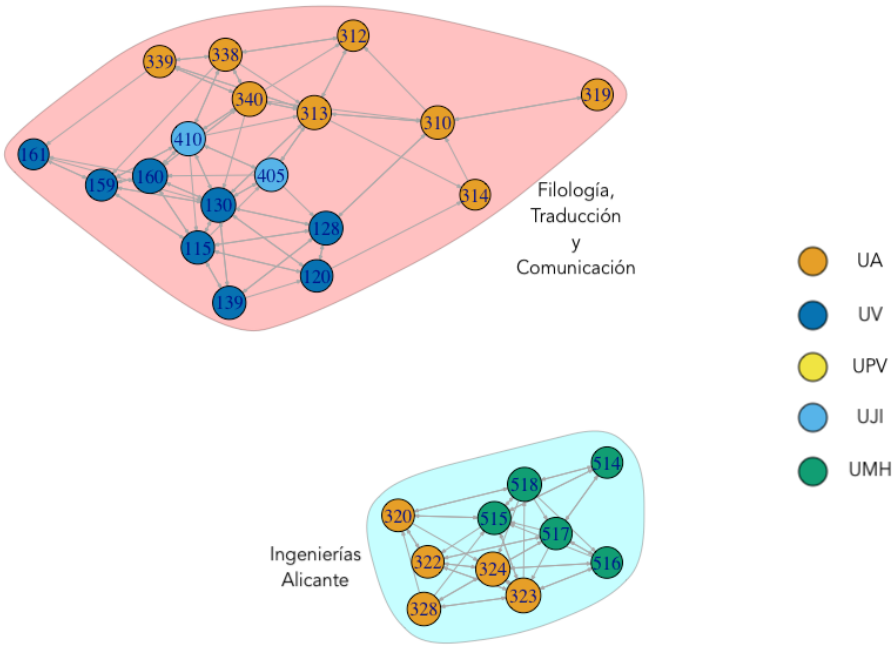


Figura 6.10: Comunidades de Humanidades e Ingeniería del grafo de acceso a la universidad

des contienen grados de este último área). La comunidad compuesta por los grados de ingeniería se puede ver en la figura 6.11. Esta comunidad está compuesta por 32 grados, todos de la rama de ingeniería, impartidos en distintas universidades de la Comunidad Valenciana. Una segunda poda de aristas mediante convolución y la detección de comunidades en el grafo resultante devuelve las subcomunidades dentro de esta comunidad de grados en ingeniería, que aportan información añadida a la comunidad ya obtenida inicialmente. Estas subcomunidades se pueden ver en la figura 6.12. Se puede observar como subdivide las ingenierías dentro de sus propias ramas. Los principales subgrupos dentro de esta comunidad son los conformados por los grados en la rama de Ingeniería Industrial y Aeronáutica, y los grados en Arquitectura, Ingeniería Civil y Diseño Industrial. Todos los grados en este subgrupo son impartidos en la UPV, excepto el grado 416 que corresponde al grado en Ingeniería en Diseño Industrial y Desarrollo de Productos impartido en la UJI. Otro de estos subgrupos lo conforman los estudios en Tecnologías de la Información y la Comunicación, formados por los grados en Ingeniería Informática que se imparten en los campus de Vera y Alcoy de la UPV (grados 226 y 225, respectivamente) y por el grado Ingeniería de Tecnologías y Servicios de Telecomunicación y el doble grado con ADE. Del resto de grupos es importante reseñar como se dividen por localización física: en general, los grados impartidos en Valencia, Castellón, Alcoy o Alicante están en grupos distintos. Esto puede indicar que algunos solicitantes priorizan estudiar en una localidad concreta antes que el grado. Esta tendencia es justo al contrario de lo que ocurría en la comunidad de Ingeniería Agroalimentaria y Forestal de la figura 6.9, o con las ingenierías de Alicante de la figura 6.10.

Para la comunidad de los grados de Ciencias de la Naturaleza y de la Salud se procede de la misma manera que con la comunidad anterior: se podan las aristas mediante convolución y se detectan las subcomunidades. El resultado de aplicar este procedimiento se puede ver en la figura 6.13. En este caso, al contrario de lo que ocurría en la comunidad de ingenierías, se puede ver mucha más heterogeneidad de universidades en las distintas comunidades. Un ejemplo claro se puede ver en el caso de Medicina: en la misma comunidad están los grados que se imparten en Valencia, Alicante y Castellón (además

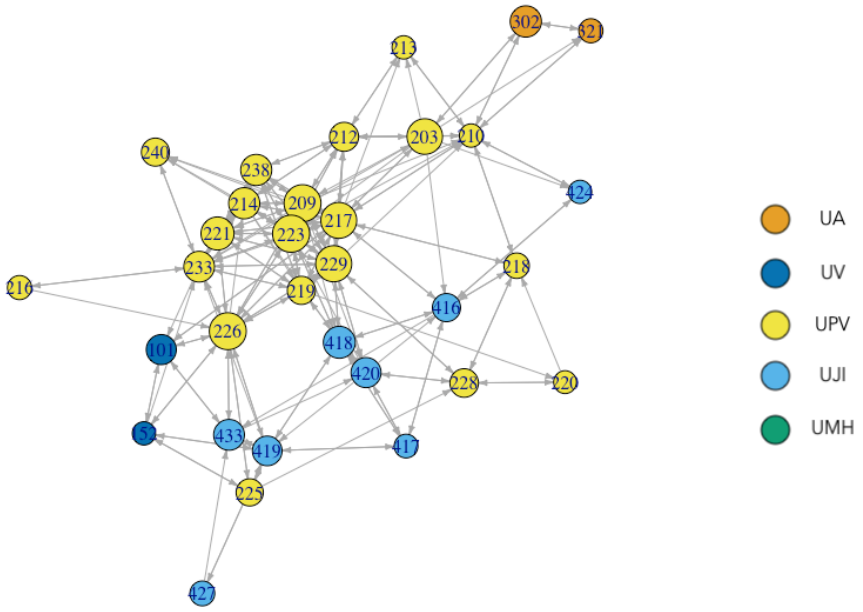


Figura 6.11: Comunidad de Ingeniería del grafo de acceso a la universidad

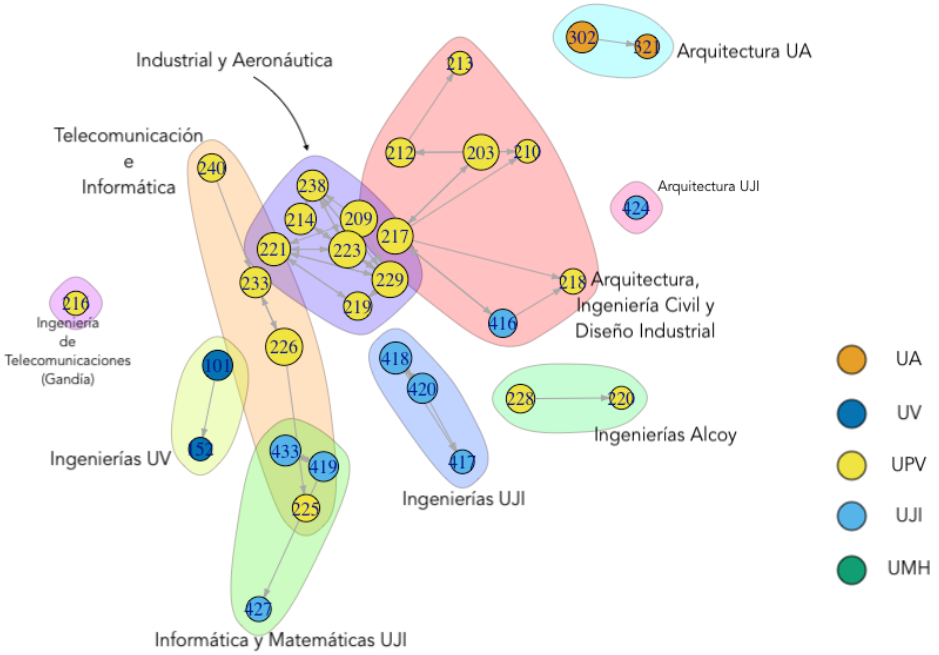


Figura 6.12: Subcomunidades dentro de la comunidad de ingenierías

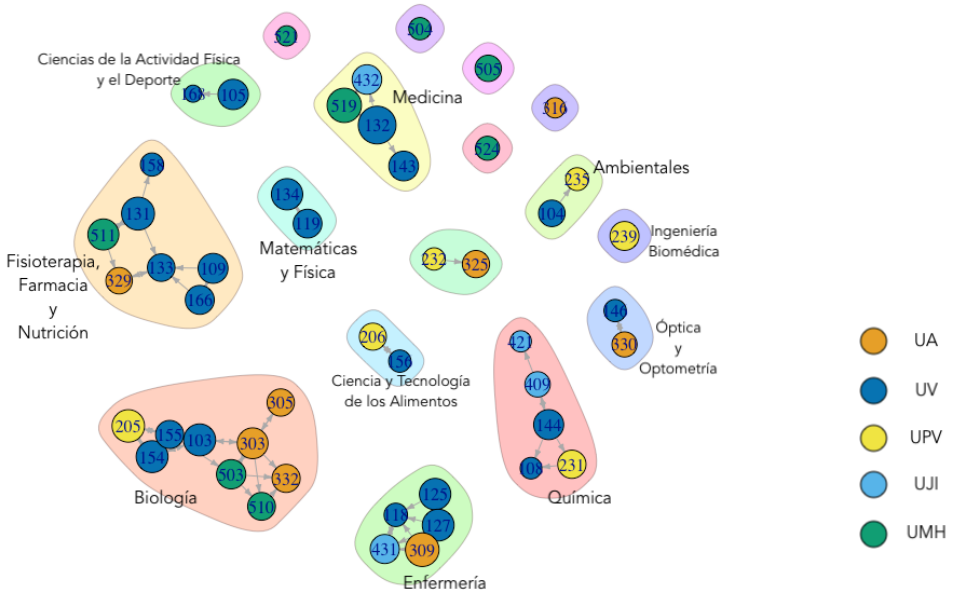


Figura 6.13: Subcomunidades dentro de la comunidad de ciencias

de Odontología, que parece ser un grado alternativo). Esto puede indicar que las carreras científicas son más vocacionales que las de ingeniería, y que a los interesados en estudiar un grado específico les es indiferente la localidad en la que estudien. Esta subdivisión permite distinguir claramente los distintos grados en sus respectivas ramas. También vemos algunas comunidades formadas por un solo grado, que corresponden a grados muy poco solicitados. Todos estos grados aislados se imparten en la provincia de Alicante.

Por último tenemos las dos comunidades de grados de Ciencias Sociales. Una de estas comunidades está formada por los grados impartidos en Valencia y Castellón, mientras que la otra está compuesta por los grados impartidos en Alicante. Esto es una muestra más de que el tráfico entre Valencia y Castellón es mucho mayor que entre Alicante y cualquiera de las otras dos provincias. La comunidad de Ciencias Sociales de Valencia y Castellón se puede ver en la figura 6.14, mientras que la de Alicante se puede ver en la figura 6.15. En ambas comunidades se puede ver una subdivisión por áreas similar, lo mismo que ocurre con las comunidades anteriormente descritas.

No es el objetivo de este trabajo obtener conclusiones de los resultados obtenidos, pero es interesante recalcar algunos aspectos. Se pueden observar dos tipos de comportamiento entre los solicitantes: o bien se prioriza la afinidad académica, o bien la geográfica. Los solicitantes de grados como Medicina o Ingenierías priorizan estudiar un grado en la misma rama aún teniendo que cambiar de localidad, mientras que los solicitantes de grados de, por ejemplo, Ciencias Sociales prefieren cambiar de grado o rama para evitar cambiar de localidad.

Se ha podido demostrar que el método de detección de comunidades presentado en este proyecto funciona correctamente en casos en los que los algoritmos clásicos no tienen buen rendimiento. En este caso, el algoritmo consigue eliminar una parte importante de aristas de ruido, preservando la estructura comunitaria, lo que facilita el trabajo de detección de comunidades a algoritmos como Girvan-Newman o *Walktrap*. Los resultados obtenidos son satisfactorios, y ofrecen información de interés sobre el flujo de solicitantes entre los distintos grados ofertados en las universidades de la Comunidad Valenciana. Como se ha comentado anteriormente, se han empleado datos de 2014, que es el año más recientes dentro de los que se dispone de datos. Sería de gran interés aplicar estos mé-

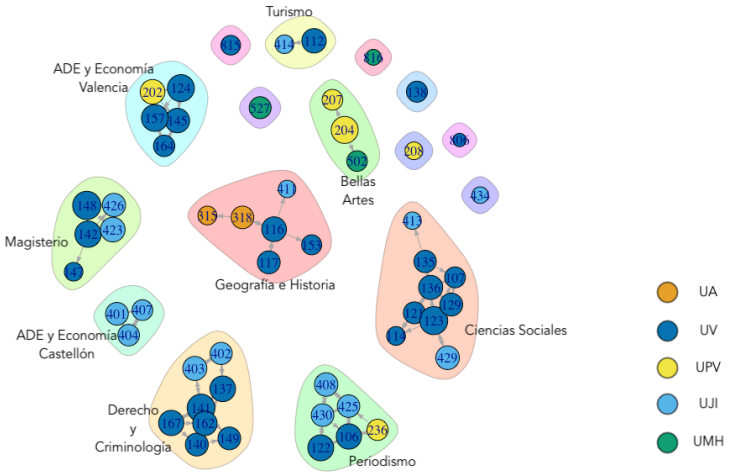


Figura 6.14: Subcomunidades dentro de la comunidad de ciencias sociales (Valencia y Castellón)

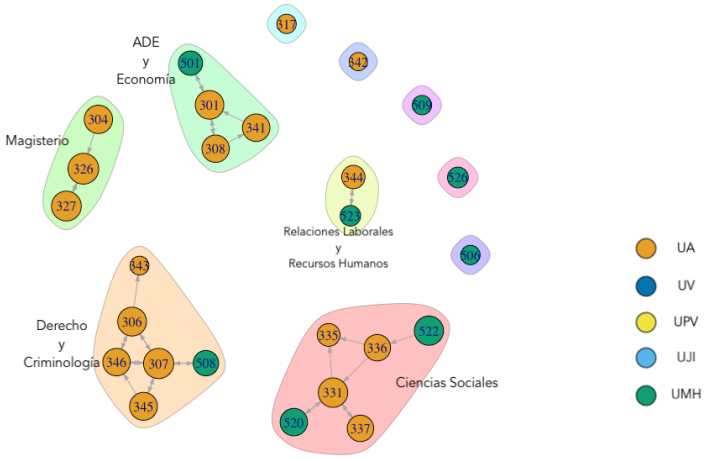


Figura 6.15: Subcomunidades dentro de la comunidad de ciencias sociales (Alicante)

todos en datos actuales, especialmente debido a la introducción de nuevos grados y al cambio de paradigma que se ha originado en los últimos años.

CAPÍTULO 7
Conclusiones

En este proyecto se han presentado los conceptos básicos y la importancia hoy en día de la detección de comunidades en grafos, además de los algoritmos más comúnmente empleados en la actualidad. También se han descrito algunos algoritmos surgidos con el propósito de mejorar los resultados obtenidos hasta el momento en detección de comunidades en grafos dirigidos débilmente conectados. Además, se ha propuesto un nuevo algoritmo de detección de comunidades que emplea técnicas usadas en tratamiento de imágenes para la detección de bordes.

Los métodos definidos se han aplicado sobre dos casos reales: el tráfico de vuelos comerciales entre aeropuertos de Estados Unidos, y el flujo de solicitantes entre grados en el sistema de acceso a la universidad pública en España. Se ha podido comprobar que el nuevo método propuesto en este proyecto consigue mejorar los resultados obtenidos con los algoritmos clásicos de detección de comunidades. Por tanto, según los resultados obtenidos, el uso del producto de convolución para la detección de aristas de ruido parece apropiado. No obstante, aún queda mucho margen de mejora en este ámbito.

A partir de este proyecto pueden surgir muchas líneas de trabajo futuro. Una posibilidad sería usar distintos tipos de filtros de convolución que puedan mejorar los resultados obtenidos hasta el momento. También se podría analizar distintos modos de aplicar al convolución sobre el grafo.

Bibliografía

[1] L. Euler, "Solutio problematis ad geometriam situs pertinentis," *Commentarii academiae scientiarum Petropolitanae*, págs. 128-140, 1741.

[2] P. Erdos, A. Rényi y col., "On the evolution of random graphs," *Publ. Math. Inst. Hung. Acad. Sci*, vol. 5, n.º 1, págs. 17-60, 1960.

[3] M. Girvan y M. E. Newman, "Community structure in social and biological networks," *Proceedings of the national academy of sciences*, vol. 99, n.º 12, págs. 7821-7826, 2002.

[4] E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai y A.-L. Barabási, "Hierarchical organization of modularity in metabolic networks," *Science*, vol. 297, n.º 5586, págs. 1551-1555, 2002.

[5] S. Maslov y K. Sneppen, "Specificity and stability in topology of protein networks," *Science*, vol. 296, n.º 5569, págs. 910-913, 2002.

[6] J. J. Bechtel, W. A. Kelley, T. A. Coons, M. G. Klein, D. D. Slagel y T. L. Petty, "Lung cancer detection in patients with airflow obstruction identified in a primary care outpatient practice," *Chest*, vol. 127, n.º 4, págs. 1140-1145, 2005.

[7] X. Chu, J. Guan, Z. Zhang y S. Zhou, "Epidemic spreading in weighted scale-free networks with community structure," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2009, n.º 07, P07043, 2009.

[8] H. Zhao y Z. Gao, "Modular effects on epidemic dynamics in small-world networks," *EPL (Europhysics Letters)*, vol. 79, n.º 3, pág. 38 002, 2007.

[9] M. John, T. Ikuta y J. Ferbinteanu, "Graph analysis of structural brain networks in Alzheimer's disease: beyond small world properties," *Brain Structure and Function*, vol. 222, n.º 2, págs. 923-942, 2017.

[10] A. Raj, A. Kuceyeski y M. Weiner, "A network diffusion model of disease progression in dementia," *Neuron*, vol. 73, n.º 6, págs. 1204-1215, 2012.

[11] T. Xu, K. R. Cullen, B. Mueller y col., "Network analysis of functional brain connectivity in borderline personality disorder using resting-state fMRI," *NeuroImage: Clinical*, vol. 11, págs. 302-315, 2016.

[12] S. Fortunato, "Community detection in graphs," *Physics reports*, vol. 486, n.º 3-5, págs. 75-174, 2010.

[13] S. Wasserman, K. Faust y col., "Social network analysis: Methods and applications," 1994.

[14] R. D. Luce y A. D. Perry, "A method of matrix analysis of group structure," *Psychometrika*, vol. 14, n.º 2, págs. 95-116, 1949.

[15] R. D. Luce, "Connectivity and generalized cliques in sociometric group structure," *Psychometrika*, vol. 15, n.º 2, págs. 169-190, 1950.

[16] F. Luccio y M. Sami, "On the decomposition of networks in minimally interconnected subnetworks," *IEEE Transactions on Circuit Theory*, vol. 16, n.º 2, págs. 184-188, 1969.

[17] M. E. Newman y M. Girvan, "Finding and evaluating community structure in networks," *Physical review E*, vol. 69, n.º 2, 2004.

[18] A. Lancichinetti y S. Fortunato, "Community detection algorithms: a comparative analysis," *Physical review E*, vol. 80, n.º 5, 2009.

[19] Z. Yang, R. Algesheimer y C. J. Tessone, "A comparative analysis of community detection algorithms on artificial networks," *Scientific reports*, vol. 6, n.º 1, págs. 1-18, 2016.

[20] J. M. Montañana, A. Hervás y P. P. Soriano Jiménez, "A Proposal for a Benchmark Generator of Weakly Connected Directed Graphs," *Open Journal of Modelling and Simulation*, vol. 8, n.º 1, págs. 18-34, 2020.

[21] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, págs. 35-41, 1977.

[22] J. R. Tyler, D. M. Wilkinson y B. A. Huberman, "Email as spectroscopy: Automated discovery of community structure within organizations," en *Communities and technologies*, Springer, 2003, págs. 81-96.

[23] D. M. Wilkinson y B. A. Huberman, "A method for finding communities of related genes," *proceedings of the national Academy of sciences*, vol. 101, n.º suppl 1, págs. 5241-5248, 2004.

[24] M. E. Newman, "Fast algorithm for detecting community structure in networks," *Physical review E*, vol. 69, n.º 6, pág. 066 133, 2004.

[25] A. Clauset, M. E. Newman y C. Moore, "Finding community structure in very large networks," *Physical review E*, vol. 70, n.º 6, pág. 066 111, 2004.

[26] P. Pons y M. Latapy, "Computing communities in large networks using random walks," en *J. Graph Algorithms Appl*, Citeseer, 2006.

[27] J. H. Ward Jr, "Hierarchical grouping to optimize an objective function," *Journal of the American statistical association*, vol. 58, n.º 301, págs. 236-244, 1963.

[28] S. Zhang, R.-S. Wang y X.-S. Zhang, "Identification of overlapping community structure in complex networks using fuzzy c-means clustering," *Physica A: Statistical Mechanics and its Applications*, vol. 374, n.º 1, págs. 483-490, 2007.

[29] M. Gong, J. Liu, L. Ma, Q. Cai y L. Jiao, "Novel heuristic density-based method for community detection in networks," *Physica A: Statistical Mechanics and its Applications*, vol. 403, págs. 71-84, 2014.

[30] A. Rodriguez y A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, n.º 6191, págs. 1492-1496, 2014.

[31] M. Ester, H.-P. Kriegel, J. Sander, X. Xu y col., "A density-based algorithm for discovering clusters in large spatial databases with noise.," en *kdd*, vol. 96, 1996, págs. 226-231.

[32] L. Ertöz, M. Steinbach y V. Kumar, "Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data," en *Proceedings of the 2003 SIAM international conference on data mining*, SIAM, 2003, págs. 47-58.

[33] T. You, H.-M. Cheng, Y.-Z. Ning, B.-C. Shia y Z.-Y. Zhang, "Community detection in complex networks using density-based clustering algorithm and manifold learning," *Physica A: Statistical Mechanics and its Applications*, vol. 464, págs. 221-230, 2016.

[34] X. Xu, N. Yuruk, Z. Feng y T. A. Schweiger, "Scan: a structural clustering algorithm for networks," en *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2007, págs. 824-833.

[35] J. Montañana, A. Hervás y P. Soriano, "Communities' detection in weakly connected directed graphs," *Modelling for Engineering & Human Behaviour*, págs. 72-77, 2020.

[36] A. Hervás, P. Soriano-Jiménez, A. Jiménez, J. Peinado, R. Capilla y J. Montanana, "A case study of directional communities in a directed graph: the accesing procedure to the Spanish public university system," *Modeling Human Behavior*, pág. 9,

[37] D. Marr y E. Hildreth, "Theory of edge detection," *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 207, n.º 1167, págs. 187-217, 1980.

[38] H. Munoz, E. Vicente, I. Gonzalez, A. Mateos y A. Jimenez-Martin, "ConvGraph: Community Detection of Homogeneous Relationships in Weighted Graphs," *Mathematics*, vol. 9, n.º 4, pág. 367, 2021.

[39] A. Hervás, J. Montañana, S. Morillas y P. Soriano, "A procedure for detection of border communities using convolution techniques," *Modelling for Engineering & Human Behaviour*, págs. 267-271, 2021.

[40] R. Guimera, S. Mossa, A. Turtschi y L. N. Amaral, "The worldwide air transportation network: Anomalous centrality, community structure, and cities' global roles," *Proceedings of the National Academy of Sciences*, vol. 102, n.º 22, págs. 7794-7799, 2005.

APÉNDICE A
Lista de grados

- 101: Grado en Ingeniería Multimedia, UV
- 102: Grado en Ingeniería Telemática, UV
- 103: Grado en Biología, UV
- 104: Grado en Ciencias Ambientales, UV
- 105: Grado en Ciencias de la Actividad Física y del Deporte (U. Valencia), UV
- 106: Grado en Comunicación Audiovisual, UV
- 107: Grado en Sociología, UV
- 108: Grado en Ingeniería Química, UV
- 109: Grado en Farmacia, UV
- 112: Grado en Turismo, UV
- 114: Grado en Logopedia, UV
- 115: Grado en Lenguas Modernas y sus Literaturas, UV
- 116: Grado en Historia, UV
- 117: Grado en Historia del Arte, UV
- 118: Grado en Enfermería, UV
- 119: Grado en Matemáticas, UV
- 120: Grado en Filología Catalana, UV
- 121: Grado en Pedagogía, UV
- 122: Grado en Periodismo, UV
- 123: Grado en Psicología, UV
- 124: Grado en Negocios Internacionales (International Business), UV
- 125: Grado en Enfermería, UV
- 126: Grado en Información y Documentación, UV
- 127: Grado en Enfermería, UV

- 128: Grado en Estudios Hispánicos: Lengua Española y sus Literaturas, UV
- 129: Grado en Educación Social, UV
- 130: Grado en Estudios Ingleses, UV
- 131: Grado en Fisioterapia, UV
- 132: Grado en Medicina, UV
- 133: Grado en Nutrición Humana y Dietética, UV
- 134: Grado en Física, UV
- 135: Grado en Relaciones Laborales y Recursos Humanos, UV
- 136: Grado en Trabajo Social, UV
- 137: Grado en Criminología (Presencial), UV
- 138: Grado en Filosofía, UV
- 139: Grado en Filología Clásica, UV
- 140: Grado en Derecho + Ciencias Politicas y de la Administración Pública, UV
- 141: Grado en Derecho, UV
- 142: Grado en Maestro en Educación Infantil, UV
- 143: Grado en Odontología, UV
- 144: Grado en Química, UV
- 145: Grado en Economía, UV
- 146: Grado en Óptica y Optometría, UV
- 147: Grado en Maestro en Educación Infantil, UV
- 148: Grado en Maestro en Educación Primaria, UV
- 149: Grado en Ciencias Políticas y de la Administración Pública, UV
- 150: Grado en Ingeniería Electrónica de Telecomunicación, UV
- 151: Grado en Ingeniería Electrónica Industrial, UV
- 152: Grado en Ingeniería Informática, UV
- 153: Grado en Geografía y Medio Ambiente, UV
- 154: Grado en Bioquímica y Ciencias Biomédicas, UV
- 155: Grado en Biotecnología, UV
- 156: Grado en Ciencia y Tecnología de los Alimentos (U. Valencia), UV
- 157: Grado en Administración y Dirección de Empresas, UV
- 158: Grado en Podología, UV
- 159: Grado en Traducción y Mediación Interlingüística (Francés), UV

- 160: Grado en Traducción y Mediación Interlingüística (Inglés), UV
- 161: Grado en Traducción y Mediación Interliguïstica (Alemán), UV
- 162: Grado en Derecho + Criminología, UV
- 163: Grado en Administración y Dirección de Empresas, UV
- 164: Grado en Finanzas y Contabilidad, UV
- 166: Grado en Farmacia /Nutrición Humana y Dietética, UV
- 167: Grado en Administración y Dirección de Empresas+Derecho, UV
- 168: Grado en Ciencias de la Actividad Física y el Deporte (Sección Onteniente), UV
- 201: Grado en Administración y Dirección de Empresas, UPV
- 202: Grado en Administración y Dirección de Empresas, UPV
- 203: Grado en Fundamentos de la Arquitectura, UPV
- 204: Grado en Bellas Artes, UPV
- 205: Grado en Biotecnología, UPV
- 206: Grado en Ciencia y Tecnología de los Alimentos (U. Politécnica de Valencia), UPV
- 207: Grado en Conservación y Restauración de Bienes Culturales, UPV
- 208: Grado en Gestión y Administración Pública, UPV
- 209: Grado en Ingeniería Aeroespacial, UPV
- 210: Grado en Arquitectura Técnica, UPV
- 211: Grado en Ingeniería Agroalimentaria y del Medio Rural, UPV
- 212: Grado en Ingeniería Civil, UPV
- 213: Grado en Ingeniería de Obras Públicas, UPV
- 214: Grado en Ingeniería de Organización Industrial, UPV
- 216: Grado en Ingeniería de Sistemas de Telecomunicación, UPV
- 217: Grado en Ingeniería en Diseño Industrial y Desarrollo de Productos, UPV
- 218: Grado en Ingeniería en Diseño Industrial y Desarrollo de Productos, UPV
- 219: Grado en Ingeniería Eléctrica, UPV
- 220: Grado en Ingeniería Eléctrica, UPV
- 221: Grado en Ingeniería Electrónica Industrial y Automática, UPV
- 222: Grado en Ingeniería en Geomática y Topografía, UPV
- 223: Grado en Ingeniería en Tecnologías Industriales, UPV
- 224: Grado en Ingenieria Forestal y del Medio Natural, UPV

- 225: Grado en Ingeniería Informática, UPV
- 226: Grado en Ingeniería Informática, UPV
- 228: Grado en Ingeniería Mecanica, UPV
- 229: Grado en Ingeniería Mecanica, UPV
- 231: Grado en Ingeniería Química, UPV
- 232: Grado en Ingeniería Química, UPV
- 233: Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación, UPV
- 235: Grado en Ciencias Ambientales, UPV
- 236: Grado en Comunicación Audiovisual, UPV
- 237: Grado en Gestión Turística, UPV
- 238: Grado en Ingeniería de la Energía, UPV
- 239: Grado en Ingeniería Biomédica, UPV
- 240: Grado en ADE + Ingeniería de Tecnologías y Servicios de Telecomunicación, UPV
- 241: Grado en ADE + Gestión Turística (E.P.S. GANDIA), UPV
- 242: Grado en ADE + Gestión Turística (E.P.S. ALCOY), UPV
- 301: Grado en Administración y Dirección de Empresas, UA
- 302: Grado en Fundamentos de la Arquitectura, UA
- 303: Grado en Biología, UA
- 304: Grado en Ciencias de la Actividad Física y del Deporte (U. Alicante), UA
- 305: Grado en Ciencias del Mar, UA
- 306: Grado en Criminología (Presencial), UA
- 307: Grado en Derecho, UA
- 308: Grado en Economía, UA
- 309: Grado en Enfermería, UA
- 310: Grado en Español: Lengua y Literaturas, UA
- 311: Grado en Estudios Árabes e Islámicos, UA
- 312: Grado en Estudios Franceses, UA
- 313: Grado en Estudios Ingleses, UA
- 314: Grado en Filología Catalana, UA
- 315: Grado en Geografía y Ordenación del Territorio, UA
- 316: Grado en Geología, UA

- 317: Grado en Gestión y Administración Pública, UA
- 318: Grado en Historia, UA
- 319: Grado en Humanidades, UA
- 320: Grado en Ingeniería Civil, UA
- 321: Grado en Arquitectura Técnica, UA
- 322: Grado en Ingeniería en Sonido e Imagen en Telecomunicación, UA
- 323: Grado en Ingeniería Informática, UA
- 324: Grado en Ingeniería Multimedia, UA
- 325: Grado en Ingeniería Química, UA
- 326: Grado en Maestro en Educación Primaria, UA
- 327: Grado en Maestro en Educación Infantil, UA
- 328: Grado en Matemáticas, UA
- 329: Grado en Nutrición Humana y Dietética, UA
- 330: Grado en Óptica y Optometría, UA
- 331: Grado en Publicidad y Relaciones Públicas, UA
- 332: Grado en Química, UA
- 335: Grado en Sociología, UA
- 336: Grado en Trabajo Social, UA
- 337: Grado en Turismo, UA
- 338: Grado en Traducción e Interpretación (Francés), UA
- 339: Grado en Traducción e Interpretación (Alemán), UA
- 340: Grado en Traducción e Interpretación (Inglés), UA
- 341: Estudios Simultáneos de Grado en Turismo y Grado en ADE, UA
- 342: Grado en Relaciones Laborales y Recursos Humanos, UA
- 343: Grado en Criminología (On-line), UA
- 344: Grado en Relaciones Laborales y Recursos Humanos, UA
- 345: Estudios Simultáneos de Grado en Derecho y Grado en ADE, UA
- 346: Estudios Simultáneos de Grado en Derecho y Grado en Criminología, UA
- 401: Grado en Administración de Empresas, UJI
- 402: Grado en Criminologia y Seguridad, UJI
- 403: Grado en Derecho, UJI
- 404: Grado en Economía, UJI

- 405: Grado en Estudios Ingleses, UJI
- 407: Grado en Finanzas y Contabilidad, UJI
- 408: Grado en Publicidad y Relaciones Públicas, UJI
- 409: Grado en Química, UJI
- 410: Grado en Traducción e Interpretación, UJI
- 411: Grado en Historia y Patrimonio, UJI
- 412: Grado en Humanidades: Estudios Interculturales, UJI
- 413: Grado en Relaciones Laborales y Recursos Humanos, UJI
- 414: Grado en Turismo, UJI
- 415: Grado en Ingeniería Agroalimentaria y del Medio Rural, UJI
- 416: Grado en Ingeniería en Diseño Industrial y Desarrollo de Productos, UJI
- 417: Grado en Ingeniería Eléctrica, UJI
- 418: Grado en Ingeniería en Tecnologías Industriales, UJI
- 419: Grado en Ingeniería Informática, UJI
- 420: Grado en Ingeniería Mecanica, UJI
- 421: Grado en Ingeniería Química, UJI
- 423: Grado en Maestro en Educación Primaria, UJI
- 424: Grado en Arquitectura Técnica, UJI
- 425: Grado en Comunicación Audiovisual, UJI
- 426: Grado en Maestro en Educación Infantil, UJI
- 427: Grado en Matematica Computacional, UJI
- 429: Grado en Psicología, UJI
- 430: Grado en Periodismo, UJI
- 431: Grado en Enfermería, UJI
- 432: Grado en Medicina, UJI
- 433: Grado en Diseño y Desarrollo de Videojuegos, UJI
- 434: Grado en Gestión y Administración Pública, UJI
- 501: Grado en Administración y Dirección de Empresas, UMH
- 502: Grado en Bellas Artes, UMH
- 503: Grado en Biotecnología, UMH
- 504: Grado en Ciencias Ambientales, UMH
- 505: Grado en Ciencias de la Actividad Física y del Deporte (Elche), UMH

- 506: Grado en Ciencias Politicas y Gestión Pública, UMH
- 508: Grado en Derecho, UMH
- 509: Grado en Estadística Empresarial, UMH
- 510: Grado en Farmacia, UMH
- 511: Grado en Fisioterapia, UMH
- 513: Grado en Ingeniería Agroalimentaria y Agroambiental, UMH
- 514: Grado en Ingeniería Eléctrica, UMH
- 515: Grado en Ingenieria Electrónica y Automática Industrial, UMH
- 516: Grado en Ingeniería Informática en Tecnologías de la Información, UMH
- 517: Grado en Ingeniería de Tecnologías de Telecomunicación, UMH
- 518: Grado en Ingeniería Mecanica, UMH
- 519: Grado en Medicina, UMH
- 520: Grado en Periodismo, UMH
- 521: Grado en Podología, UMH
- 522: Grado en Psicología, UMH
- 523: Grado en Relaciones Laborales y Recursos Humanos, UMH
- 524: Grado en Terapia Ocupacional, UMH
- 525: Grado en Ciencia y Tecnología de los Alimentos (Orihuela), UMH
- 526: Grado en Administración y Dirección de Empresas, UMH
- 527: Grado en Comunicación Audiovisual, UMH
- 801: Grado en Administración y Dirección de Empresas, UMH
- 803: Grado en Comunicación y Relaciones Públicas, UMH
- 805: Grado en Administración y Dirección de Empresas, UV
- 806: Grado en Turismo, UV
- 807: Grado en Maestro en Educación Primaria, UV
- 808: Grado en Maestro en Educación Infantil, UV
- 809: Grado en Finanzas y Contabilidad, UV
- 810: Grado en Ingeniería Electrónica Industrial y Automática, UPV
- 811: Grado en Ingeniería Mecanica, UPV
- 813: Grado en Organización de Eventos, UMH
- 814: Grado en Gestión Comercial y Marketing, UMH
- 815: Grado en Administración y Dirección de Empresas, UV
- 816: Grado en Negocios Internacionales (International Business), UMH

APÉNDICE B
Objetivos de Desarrollo Sostenible

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1. **Fin de la pobreza.**				X
ODS 2. **Hambre cero.**				X
ODS 3. **Salud y bienestar.**		X		
ODS 4. **Educación de calidad.**		X		
ODS 5. **Igualdad de género.**				X
ODS 6. **Agua limpia y saneamiento.**				X
ODS 7. **Energía asequible y no contaminante.**		X		
ODS 8. **Trabajo decente y crecimiento económico.**				X
ODS 9. **Industria, innovación e infraestructuras.**		X		
ODS 10. **Reducción de las desigualdades.**				X
ODS 11. **Ciudades y comunidades sostenibles.**		X		
ODS 12. **Producción y consumo responsables.**				X
ODS 13. **Acción por el clima.**			X	
ODS 14. **Vida submarina.**				X
ODS 15. **Vida de ecosistemas terrestres.**				X
ODS 16. **Paz, justicia e instituciones sólidas.**				X
ODS 17. **Alianzas para lograr objetivos.**		X		

Este trabajo puede relacionarse en cierta medida con algunos de los Objetivos de Desarrollo Sostenible. Por un lado, este trabajo, debido a sus posibles aplicaciones prácticas, puede vincularse con los ODS Salud y bienestar, y Educación de calidad. Muchas de las herramientas y técnicas desarrolladas en el contexto del análisis de redes, entre las que se encuentra la detección de comunidades, se emplean, por ejemplo, en el estudio de enfermedades y en el análisis de redes epidemiológicas, por lo que avances en métodos de detección de comunidades puede repercutir positivamente en este campo. Estos métodos también se pueden aplicar en Educación: ejemplo de ello es la aplicación de la detección de comunidades en el sistema de acceso a la universidad desarrollado en el capítulo 6. De manera similar se puede relacionar este trabajo con el ODS Alianzas para lograr objetivos, ya que la detección de comunidades tiene muchas aplicaciones en ciencias sociales y en el estudio de relaciones internacionales.

Por otra parte, el uso de los métodos definidos en este trabajo en el análisis del sistema de tráfico aéreo puede relacionarse con varios ODS. La búsqueda de soluciones para una mejora de la eficiencia de este sistema puede relacionarse con los ODS Energía asequible y no contaminante, y Ciudades y comunidades sostenibles. Además, también puede relacionarse en menor medida con el ODS Acción con el clima. El análisis del sistema de tráfico aéreo puede resultar en una reducción de vuelos innecesarios que no solo tiene una repercusión económica, sino que también puede afectar positivamente al clima y reducción de la contaminación.