Universidad Nacional de Educación a Distancia, UNED

Máster Universitario en I.A. Avanzada: fundamentos, métodos y aplicaciones

TRABAJO DE FIN DE MÁSTER

Preservación de la privacidad en procesos de minería de datos basados en grafos

---

Autor: Jordi Casas

Directores: Dr. Vicenç Torra y Dr. Jordi Herrera

Supervisor: Dr. José Luis Aznarte

Barcelona, 27 de junio de 2011

Resumen

Los grafos son un formato de representación complejo y flexible, que permite representar de una forma natural una gran diversidad de realidades. Algunos ejemplos de estos datos son: redes sociales, redes de comunicaciones, estructuras biológicas, etc.

En paralelo a la explotación de este tipo de datos, aparecen los problemas de seguridad asociados a su difusión. Cuando se difunde un grafo se están difundiendo datos de los individuos que aparecen en él, y algunos de ellos pueden ser datos sensibles o privados. Es necesario detectar y proteger las identidades de los individuos que aparecen en los grafos antes de proceder a su difusión.

En este trabajo se realiza una breve revisión del estado del arte en métodos de anonimización de grafos. Para poder ver la problemática en toda su dimensión, también se revisan conceptos relacionados como las medidas de calidad o los métodos de re-identificación y conocimiento del adversario. También se realiza una breve revisión sobre algunos métodos de minería de datos aplicada a grafos (*graph mining*).

A continuación se escogen dos métodos de anonimización y se analiza su comportamiento ante distintos conjuntos de datos reales. Se evalúa el grado de perturbación introducido a partir de las propiedades estructurales y el grado de afectación que pueda tener en el resultado de los procesos de *graph mining* aplicados sobre los datos. Por otro lado, también se evalúa el nivel de seguridad de los datos anonimizados.

A partir de las deficiencias observadas en los dos métodos de anonimización, se implementa un método basado en anteriores estudios de Liu y Terzi. El nuevo método es analizado con los mismos conjuntos de datos y demuestra superar algunas de las deficiencias detectadas en los métodos anteriores.

Palabras clave: privacidad, anonimización, grafos, minería de datos, *graph mining*.

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Introducción

1.1. Descripción general del problema

En la actualidad, los procesos de minería de datos requieren grandes cantidades de datos, que en muchas ocasiones contienen información personal y privada de usuarios o personas. Aunque se realicen procesos básicos de anonimización sobre los datos, es decir, eliminación de los nombres u otros identificadores clave, existen multitud de técnicas de re-identificación que permiten volver a identificar a un usuario dentro de este conjunto de datos. En la Figura 1.1 se presenta un mapa donde es posible contextualizar los procesos de anonimización y re-identificación dentro de un proceso de minería de datos.

Para solucionar este problema se han desarrollado métodos que realizan operaciones de introducción de ruido en los datos originales (después del proceso de anonimización) con el fin de dificultar los procesos posteriores de re-identificación [14, 34]. Sin embargo, estos procesos de ofuscación o confusión de los datos presentan un problema importante: pueden producir pérdida de información en los datos. Evidentemente, es interesante mantener los datos con un conjunto de propiedades similares, para que el proceso de minería de datos no sea dependiente del método

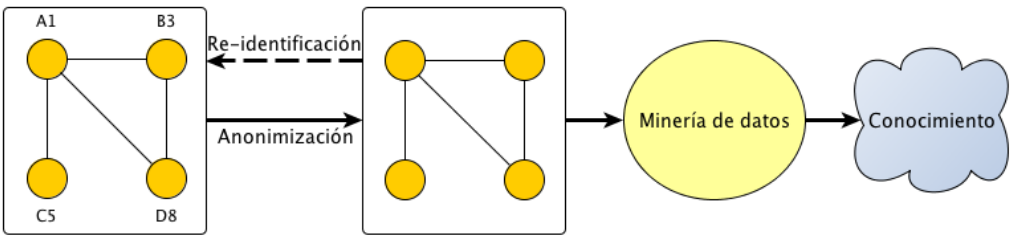

Figura 1.1: Contextualización de los procesos de anonimización y re-identificación dentro de un proceso de minería de datos.

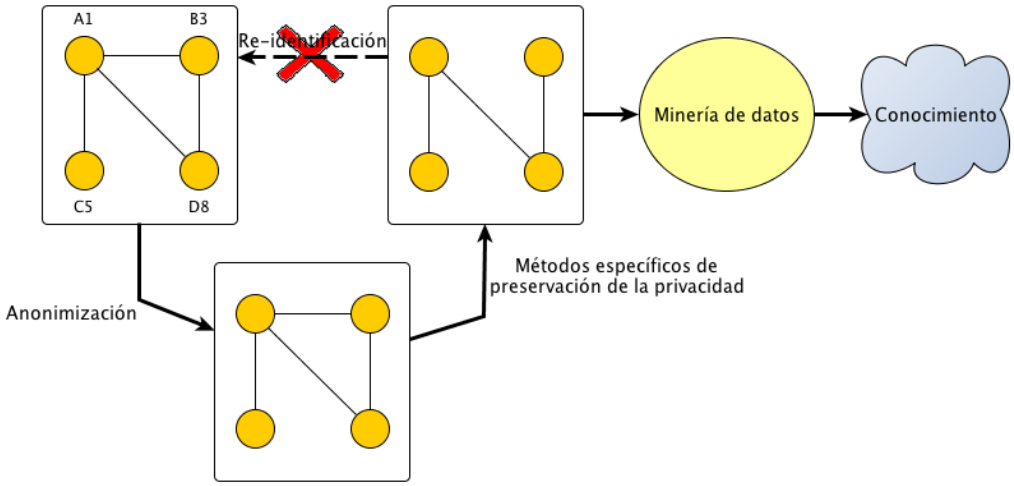

Figura 1.2: Contextualización del proceso de preservación de la privacidad dentro de un proceso de minería de datos.

de ofuscación o confusión aplicado y no se vea demasiado afectado por él. En este sentido, se debe encontrar un equilibrio entre un método que reduzca al máximo la perdida de información y que, por otro lado, dificulte al máximo los procesos de re-identificación que se puedan producir posteriormente sobre los datos. La Figura 1.2 presenta un mapa de contextualización que incluye los métodos específicos de preservación de la privacidad.

En este trabajo se pretende realizar un estudio de la problemática presentada, pero tratando con datos semi-estructurados en formato de grafo como extensión de los resultados sobre datos relacionales. Los grafos permiten una representación más rica de la información, incluyendo datos sobre la relación entre las distintas entidades o nodos. Mediante grafos se pueden representar, por ejemplo: redes sociales, organigramas empresariales, redes de computadoras o de comunicación, estructuras biológicas, etc. Esta riqueza en la representación de información los convierte en una herramienta muy potente e interesante para los métodos de minería de datos y extracción de conocimiento.

Aunque se han realizado importantes avances en preservación de la privacidad en publicación de datos, tales como el modelo k-anonymity [39] o el modelo l-diversity [28], la mayoría de estos modelos no se pueden aplicar directamente sobre datos semi-estructurados. Aunque los conceptos pueden ser aplicados a la problemática con grafos, se deben desarrollar nuevos algoritmos y métodos que permitan trabajar con este formato de datos. La anonimización en datos semi-estructurados presenta mayores retos que en los datos relacionales. En [53] se presentan tres motivos:

1. En primer lugar, es más complejo determinar el conocimiento que pueda poseer un adversario sobre los datos de una red o grafo que sobre unos datos en formato relacional.

Generalmente, los casi-identificadores (*quasi-identifiers*) sirven para relacionar datos de distintas tablas y poder identificar individuos a partir de este conjunto de atributos. Sin embargo, en el caso de los grafos se pueden usar multitud de elementos para re-identificar de forma positiva a un individuo dentro del grafo. Por ejemplo: etiquetas de los nodos o aristas, estructura de los vecinos de un determinado nodo, subgrafos inducidos, etc. Y las posibles combinaciones de estas y otras características.

2. En segundo lugar, la complejidad para evaluar la pérdida de información que se produce en el proceso de anonimización en grafos es mucho mayor a la misma complejidad en el caso de datos relacionales.

3. Y en tercer lugar, la complejidad para el desarrollo de métodos de anonimización sobre grafos es mayor que para el caso de datos relacionales. En el caso de los datos relacionales, se presupone una independencia entre registros que posibilita, generalmente, aplicar estrategias de división que permiten reducir la complejidad temporal de un algoritmo. Con datos semi-estructurados es más complejo poder aplicar estrategias de división, ya que la división en varios conjuntos altera la relación existente entre los nodos.

1.2. Motivación

En la actualidad la representación de datos en formato semi-estructurado, o formato de grafo, está experimentando un importante auge en todos los niveles. Este formato de representación permite representar estructuras y realidades más complejas que los tradicionales datos relacionales, en formato de tuplas. En un formato semi-estructurado cada entidad puede presentar, al igual que los datos relacionales, una serie de atributos en formato numérico, nominal o categórico. Pero además, el formato semi-estructurado permite representar de una forma más rica las relaciones que puedan existir entre las distintas entidades que forman en conjunto de datos. En este sentido, las relaciones pueden ser dirigidas o simétricas, etiquetadas o no-etiquetadas, etc. Toda esta riqueza añadida permite representar de forma más natural estructuras complejas, como por ejemplo, redes de ordenadores o comunicaciones, estructuras o redes sociales, organigramas empresariales, etc.

Este crecimiento ha favorecido el intercambio y la publicación de datos en formato semi-estructurado y al mismo tiempo ha planteado la problemática sobre la preservación de la privacidad en la publicación de datos en este formato. Los modelos utilizados hasta ahora con los datos relaciones no son aplicables, de forma directa, a los nuevos datos. Por lo tanto, aparecen nuevos desafíos que permitan publicar de forma segura datos semi-estructurados para su análisis o estudio.

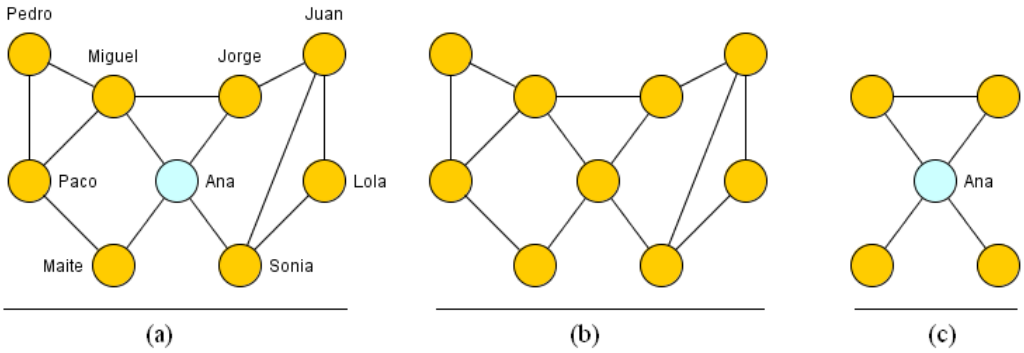

Figura 1.3: (a) Ejemplo de grafo de una red social. (b) Grafo de la misma red anonimizada. (c) Grafo de vecindad a 1 de Ana.

Un claro ejemplo de esta situación lo presentan las redes sociales. En la actualidad las redes sociales se han convertido en un fenómeno muy popular, que hace que millones de usuarios de todo el mundo estén presentes en una o varias redes sociales. Las hay de índole muy distinta: redes profesionales con la finalidad de compartir experiencias laborales y posibles contactos, redes de amistades y de ocio, redes para compartir fotografías u otros recursos entre usuarios, etc. Independientemente de su temática u objetivo, las redes sociales presentan una gran cantidad de información muy interesante para estudios en distintos ámbitos (psicología, ciencias sociales, etc). En este sentido, la explotación de estos datos es de gran interés para científicos y empresas de todo el mundo. La problemática nace con la necesidad de preservar la privacidad de los individuos que aparecen en estas redes sociales. Es decir, se deben poder estudiar las relaciones entre los individuos, la topología que forman los grupos, etc. Pero al mismo tiempo, se debe poder garantizar que la privacidad de los individuos no se verá comprometida.

Inicialmente, y por analogía con la problemática con datos relacionales, se aplica un sistema de anonimización simple que consiste en eliminar cualquier información que permita re-identificar de forma única a un usuario dentro de los datos explotados. Este proceso incluye cualquier número de identificación personal (DNI, pasaporte, etc), dirección, número de teléfono, email, etc.

En la Figura 1.3a se puede ver un ejemplo reducido de una red social. Cada uno de los nodos representa a un individuo y cada una de las aristas representa la relación de amistad entre dos individuos. Si se pretende publicar los datos de esta red para ser estudiados, se debe preservar la privacidad de los individuos que aparecen. En la Figura 1.3b se puede ver la red después de un proceso de anonimización simple, en donde la topología de la red se ha mantenido inalterable, pero se han eliminado los identificadores de los nodos.

Un atacante que posea cierta información sobre los vecinos de un nodo, puede comprometer la privacidad de este método. Por ejemplo, si el atacante sabe que Ana tiene un total de 4

amigos y que dos de ellos, además, son amigos entre si, puede construir el grafo de vecindad a 1 (*1-neighborhood*) de Ana, representado en la Figura 1.3c. A partir de este grafo se puede identificar de forma única a Ana dentro del grafo anonimizado, y por lo tanto, comprometer la privacidad de este usuario.

Aunque la preservación de la privacidad es un tema sobre el que se ha estudiado mucho y se han realizado importantes contribuciones como el modelo k-anonimity [39] y l-diversity [28], la mayoría de estudios realizados y modelos desarrollados sólo son aplicables a los modelos de datos relacionales.

1.3. Objetivos

El principal objetivo de este trabajo es evaluar distintos métodos de anonimización de grafos, prestando especial atención a como afectan a posteriores procesos de minería de datos. Es decir, se pretende evaluar el grado de degeneración introducido en los datos y como se comportan los procesos de minería de datos ante ellos. Se establecen los siguientes objetivos específicos:

- Analizar los distintos tipos de métodos de anonimización de grafos existentes.
- Analizar las medidas utilizadas para evaluar la pérdida de información asociada a este proceso.
- Analizar las medidas utilizadas para medir el riesgo de re-identificación de un grafo.
- Diseñar un conjunto de experimentos que permitan evaluar la pérdida de información producida por los métodos de anonimización en los procesos de *graph mining* .
- Evaluar la pérdida de información y el riesgo de re-identificación asociados a cada uno de los métodos de anonimización implementados.

1.4. Estructura de la memoria

Este trabajo esta estructurado en base a dos bloques. El primer bloque se centra en establecer un marco teórico, describiendo los conceptos básicos de grafos y el estado actual de la problemática de la anonimización y la preservación de la privacidad. El segundo bloque establece un marco experimental, evaluando los resultados de aplicar distintos métodos de anonimización sobre un conjuntos de datos reales.

El trabajo está organizado como sigue: en el capítulo 2 se introducen los conceptos básicos sobre grafos y se realiza un análisis sobre el estado actual de la preservación de la privacidad

en procesos de *graph mining*. Los principales puntos incluidos en el análisis son las medidas de calidad, los métodos de anonimización, el riesgo de re-identificación y los procesos de *graph mining*.

A continuación, en el capítulo 3 se evalúan distintos métodos de anonimización basados en la modificación aleatoria de aristas.

En el capítulo 4 se evalúa un método de anonimización basado en la modificación de aristas para preservar el modelo de la k-anonimidad. Este algoritmo se evalúa utilizando las mismas métricas y conjuntos de datos que en el caso anterior, con el fin de poder comprar los resultados obtenidos.

Para finalizar, en el capítulo 5 se detallan las conclusiones derivadas de este trabajo.

Capítulo 2

Estado del arte

Las técnicas de anonimización y preservación de la privacidad en procesos de minería de datos forman parte de un escenario mayor y más complejo. Este escenario se debe considerar en su globalidad para poder evaluar las repercusiones que puedan producir los procesos de anonimización o preservación de la privacidad.

El proceso parte de un conjunto de datos en formato de grafo. En este conjunto inicial, los datos permiten asociar de forma directa cada uno de los nodos del grafo con el usuario o entidad que representan. Por ejemplo, en el caso de un grafo que represente una red social, cada uno de los nodos contendrá información identificativa como el nombre, DNI, email, login u otra información clave para identificar de forma única al usuario dentro de la red social. Evidentemente, no es posible publicar este grafo para su estudio posterior, ya que se estaría comprometiendo de forma muy grave la privacidad de todos los usuarios representados en él.

Para poder publicar estos datos y poder beneficiarse de la información o conocimientos extraídos, será necesario aplicar un proceso de anonimización sobre estos datos. El objetivo de este proceso es permitir la publicación de los datos sin comprometer la privacidad de los usuarios. Es decir, no permitir que se pueda identificar de forma única a un usuario con el nodo del grafo que lo representa, ya que esto comprometería la privacidad del mismo.

En este punto aparece el rol de atacante, quien busca poder comprometer la anonimidad del sistema, consiguiendo asignar una identidad individual a los nodos del grafo anonimizado. Es decir, volviendo a obtener (de forma parcial o total) el grafo original. Este proceso se llama re-identificación, ya que consiste en re-identificar individuos o entidades que han sido anonimizadas en el paso anterior.

Por otro lado, los datos anonimizados serán utilizados para procesos posteriores de minería de datos basados en grafos (a partir de ahora, *graph mining*). Estos procesos deben permitir extraer información y conocimiento de los datos.

Dadas estas consideraciones se pueden definir las dos clases de medidas que permitirán

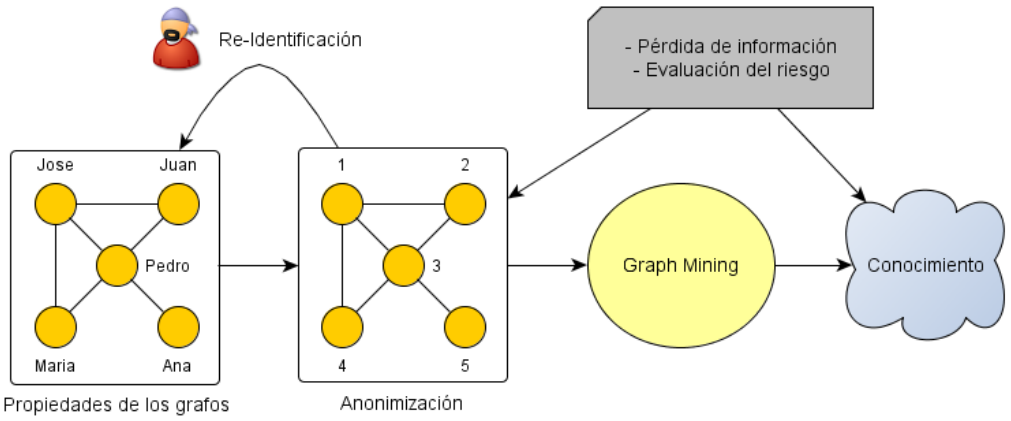

Figura 2.1: Escenario de un proceso de anonimización y los procesos relacionados.

evaluar la calidad de un proceso de anonimización. Estas son:

- Pérdida de información: El objetivo principal de todo este proceso es poder procesar los datos, mediante alguna técnica de *graph mining*, para obtener información o conocimiento a partir de los datos. Dado que se producen modificaciones en el paso de anonimización, se debe considerar que éstas pueden afectar a la calidad de los datos, y por lo tanto, al resultado de los procesos de *graph mining* aplicados. En este sentido, se debe establecer alguna medida para evaluar la pérdida de información sufrida por los datos en el proceso de anonimización. Posiblemente, esta medida dependerá del proceso o procesos que se apliquen posteriormente, ya que según el proceso de *graph mining* que se desee aplicar se deberían de considerar distintas propiedades de los grafos.

- Evaluación del riesgo: Se debe establecer alguna medida para poder evaluar y cuantificar el riesgo de una posible re-identificación. En este sentido, se pueden considerar múltiples escenarios, según distintos contextos en base al conocimiento del atacante.

Todos los factores comentados, inducen a considerar un escenario como el de la Figura 2.1. A partir de este mapa contextual del problema a abordar, se pueden definir los siguientes puntos clave que se deben considerar:

1. Propiedades de los grafos: Es necesario conocer las propiedades elementales de los grafos, para poder evaluar la afectación de los cambios producidos por el proceso de anonimización y los posibles procesos de re-identificación.

2. Las medidas de calidad del proceso de anonimización o preservación de la privacidad. Como se ha comentado, se consideran dos:

   a) Pérdida de información.

b) Riesgo de re-identificación.

3. Métodos de anonimización o preservación de la privacidad.

4. Re-identificación y posibles ataques. Se deben considerar distintos posibles escenarios y adversarios con distintos conocimientos. Con todos estos parámetros, se pueden modelar múltiples contextos y tipologías de ataques.

5. Técnicas y objetivos del *graph mining*: Como se ha comentado, la evaluación de la pérdida de información puede depender del tipo de proceso de *graph mining* aplicado. Por lo tanto, se debe de considerar qué propiedades del grafo se deben mantener inalterables (o minimizar su grado de alteración) para que el proceso de *graph mining* se vea afectado (o su afectación sea mínima).

En las siguientes secciones se desarrollaran cada uno de estos puntos claves.

2.1. Introducción y propiedades de los grafos

Un grafo es una pareja de conjuntos $G(V, E)$, donde $V = \{v_1, v_2..., v_n\}$ es el conjunto de *nodos* o *vértices* y $E = \{e_1, e_2..., e_m\}$ es un conjunto de pares de la forma $e_i = (u, v)$ tal que $u, v \in V$ y $u \neq v$, de manera que cada par une a dos de los nodos.

Si los pares unen dos nodos de forma bidireccional o simétrica se llaman *aristas* y el grafo, *simétrico*. Es decir, la arista (u, v) permite unir al nodo u con el nodo v y también al nodo v con el nodo u. Por el contrario, si los pares están orientados se llaman *arcos* y el grafo, *dirigido*. En este caso el arco (u, v) permite unir al nodo u con el nodo v, pero no en el sentido contrario.

Se llama *orden* de G a su número de nodos, $|V|$, que por convenio es referenciado por la letra n. Asimismo, el número de aristas o arcos, $|E|$, es referenciado por la letra m y se le llama *tamaño* del grafo.

En un grafo dirigido G se define a los *sucesores* de un nodo v_i, $\Gamma(v_i)$, como el conjunto de nodos a los cuales se puede llegar usando un arco desde v_i. Se define el *grado exterior* de un nodo como el número de sucesores. De forma similar, se puede definir a los *antecesores* de un nodo v_i, $\Gamma^{-1}(v_i)$, como el conjunto de nodos desde los cuales es posible llegar a v_i usando un arco. Se define el *grado interior* de un nodo como el número de antecesores. En el caso de que G sea un grafo simétrico, no tiene sentido diferenciar entre los nodos antecesores y sucesores, y se habla de nodos *adyacentes* o vecinos, $\Gamma(v_i)$, que se define como el conjunto de nodos unidos a v_i a través de una arista. En este caso se define el *grado* de un nodo como el número de nodos adyacentes.

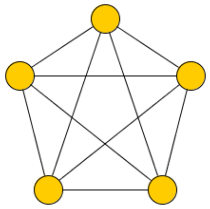

Figura 2.2: K_5, grafo completo de 5 nodos.

Un *lazo* o bucle es una arista (o arco) que relaciona al mismo nodo; es decir, una arista donde el nodo inicial y final coinciden.

Un *camino* es una secuencia ordenada de aristas (o arcos) donde el nodo final de una arista es el nodo inicial de la siguiente. La *distancia* entre dos nodos, $d(v_\mathrm{i}, v_\mathrm{j})$, se define como el número de aristas del camino más corto entre los nodos v_i y v_j. Formalmente se define la distancia entre dos nodos u y v como $d_G(u,v) = min\{\ell(C)|C\ es\ un\ camino\ u-v\}$. Un camino es simple si no repite aristas (o arcos), y elemental en caso de no repetir nodos. A un camino cerrado, es decir, que empieza y termina en el mismo nodo, se le llama *circuito*.

Se define el *diámetro* de un grafo G como la mayor de las distancias mínimas entre cualquiera de los nodos del grafo. Formalmente, $D(G) = max\{d_G(u,v)|u,v \in V\}$.

Tipos básicos de grafos

Existen multitud de tipos de grafos distintos, pero en esta sección sólo se destacaran algunos por su especial relevancia en los temas tratados de este trabajo.

En la sección anterior ya se han definido dos tipos básicos muy importantes de grafos: los *grafos simétricos* y los *grafos dirigidos*.

Se llama *grafo subyacente* al grafo que se obtiene eliminando la orientación de los arcos de un grafo dirigido.

Un grafo $G_\mathrm{p}(V_\mathrm{p}, E_\mathrm{p})$ es un *grafo parcial* o *subgrafo* de $G(V,E) \iff E_\mathrm{p} \subseteq E\ y\ V_\mathrm{p} \subseteq V$.

Se llama *multigrafo* a un grafo que contiene aristas (o arcos) repetidos. Si además se permite que un nodo esté relacionado consigo mismo mediante *bucles* o *lazos*, entonces se le llama *pseudografo*.

Un grafo G cuyas aristas tienen asociados valores reales, llamados *pesos*, se denomina *grafo ponderado*.

Un *grafo completo* es un grafo con el mayor número posible de aristas, sin lazos y sin repeticiones. El grafo simétrico completo de t nodos se denota como K_t. La Figura 2.2 muestra el grafo K_5. Se puede ver que dado $K_t = (V,E)$, entonces $|V| = t$ y $|E| = \binom{t}{2}$.

Un grafo $\overline{G}(V,\overline{E})$ es el *grafo complementario* de $G(V,E)$ si está formado por todas las aristas que le faltan a $G(V,E)$ para poder ser un grafo completo.

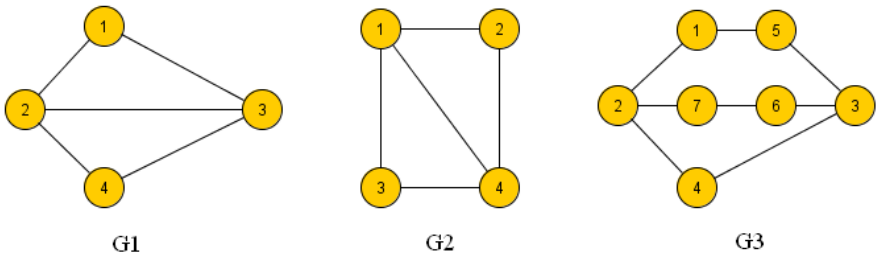

Figura 2.3: Ejemplo de grafos isomorfos (G_1 y G_2) y homomorfos (G_1 y G_3)

Un *grafo conexo* es un grafo simétrico donde para cada par de nodos existe un camino que los une. A cada subgrafo maximal conexo en un grafo simétrico G se le llama *componente* de G. Se puede expresar a un grafo G como la unión de los distintos componentes que lo forman, es decir, $G = G_1 \cup \ldots \cup G_k$ donde cada $G_i | i \in (1 \ldots k)$ es una componente de G.

Un grafo $G = (V, E)$ se llama *bipartito* si existe una partición del conjunto de nodos $V = V_1 \cup V_2$ con $V_1 \cap V_2 = \emptyset$ de tal forma que las aristas conectan los nodos de V_1 con los nodos de V_2. Es decir, $\{u, v\} \in E$ implica que $u \in V_1$ y $v \in V_2$ o viceversa. Un grafo bipartito es completo si todas las aristas posibles conectan nodos de V_1 con nodos de V_2. En tal caso, siendo $|V_1| = n$ y $|V_2| = m$, el grafo bipartito completo se denota como $K_{n,m}$. Generalizando se obtiene el concepto de los grafos k-partitos. En este caso se tiene una partición $(V_1, \ldots, V_k)$ del conjunto de nodos, de tal forma que las aristas conectan nodos que pertenecen a distintas particiones.

Isomorfismo

Dos grafos, G_1 y G_2, son *isomorfos* si existe una correspondencia biunívoca entre los nodos de ambos grafos que preserva las adyacencias entre todos los nodos. Dos grafos, G_1 y G_2, son *homomorfos* si G_2 se puede obtener a partir de G_1 mediante un proceso repetitivo de sustitución de una arista a_j en G_1 por un camino donde los únicos nodos comunes con G_1 son los nodos terminales de a_j, los cuales son los nodos terminales del camino. En la Figura 2.3 los grafos G_1 y G_2 son isomorfos, mientras que G_1 y G_3 son homomorfos.

Formas de representación de los grafos

Una forma muy habitual de representar un grafo es mediante la matriz de adyacencia. La *matriz de adyacencia* es una matriz cuadrada que se utiliza para representar relaciones binarias. Dado un grafo G su matriz de adyacencia $A = (a_\mathrm{ij})$ es cuadrada, de orden n y viene dada por:

$$a_\mathrm{ij} = \begin{cases} 1 & si\, \exists (v_i, v_j) \\ 0 & caso\, contrario \end{cases}$$

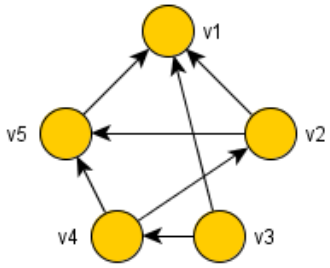

Figura 2.4: Ejemplo de grafo G para el cálculo de la matriz de adyacencia.

La matriz de adyacencia del grafo G, representado en la Figura 2.4, es:

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

Se observa que si $a_{\mathrm{ij}}^{(2)}$ es un elemento de A^2, entonces este representa el número de caminos de dos aristas (o arcos) que hay entre v_{i} y v_{j}. Esto es así dado que:

$$a_{ij}^{(2)} = \sum_{k=1}^{n} a_{ik} a_{kj}$$

Donde el producto vale 1 sólo en el caso de que ambos términos valgan 1. Por analogía se puede calcular $A^3, A^4 ...$ y en general A^k, siguiendo la regla $A^k = A^{k-1}A$, siendo $A^1 = A$. En general, A^k evalúa el número de caminos de k aristas que contiene un grafo.

Otra forma de representación matricial de un grafo G sin lazos es la matriz de incidencia. La *matriz de incidencia* $B = (b_{ij})$ de un grafo $G(V, E)$ contiene n filas y m columnas, y viene dada por:

$$b_{ij} = \begin{cases} 1 & si\ v_i\ es\ el\ vértice\ inicial\ de\ a_j \\ -1 & si\ v_i\ es\ el\ vértice\ final\ de\ a_j \\ 0 & si\ a_j\ no\ afecta\ v_i \end{cases}$$

Si el grafo es simétrico, es habitual cambiar todos los -1 por 1.

La matriz de incidencia del grafo G, representado en la Figura 2.4, es:

$$B(G) = \begin{pmatrix} -1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 \end{pmatrix}$$

La matriz de adyacencia presenta la ventaja de ser cuadrada y binaria, lo cual permite realizar operaciones de una forma sencilla y rápida. En cambio, la matriz de incidencia no presenta esta ventaja, por lo cual no suele ser tan utilizada en las operaciones sobre grafos.

Métodos para recorrer grafos

Para examinar la estructura de un grafo, es común tener que realizar un recorrido por todo el grafo. Las dos estrategias más comunes para tal fin son:

- BFS (*Breath First Search*) o búsqueda en anchura prioritaria: Este método prioriza la búsqueda en paralelo de todas las alternativas posibles desde el nodo actual.
- DFS (*Depth First Search*) o búsqueda en profundidad prioritaria: Este método prioriza la apertura de una única vía de exploración a partir del nodo actual.

Algunas propiedades relevantes en grafos

En este texto se entiende por *grafo simple* aquel grafo que no contiene atributos en los nodos ni etiquetas o pesos en las aristas o arcos. Es decir, un grafo simétrico o dirigido, sin atributos y sin etiquetas o pesos. Por el contrario, se entiende por grafo complejo (o *rich graph*) aquel grafo que contiene atributos en los nodos y etiquetas o pesos en las aristas o arcos, permitiendo así, distintas formas de relación entre los nodos del grafo.

A continuación se describen de forma breve algunas propiedades importantes de los grafos, en el dominio espacial, que frecuentemente se usan para describir las propiedades generales de una red o grafo:

- Número de nodos: Algunos métodos pueden modificar el número de nodos del grafo, por ejemplo, aplicando generalizaciones sobre los nodos del grafo. Es una medida básica y muy importante para la topología del grafo.
- Número de aristas: El número total de aristas del grafo es otra medida muy básica e importante, ya que modificando el número de aristas se afecta enormemente la conectividad del grafo.

- Grado máximo: Grado máximo de todos los nodos del grafo.

- Grado mínimo: Grado mínimo de todos los nodos del grafo.

- Grado medio: Se obtiene a partir del número de nodos y del número de aristas. Su fórmula de cálculo es: $\frac{\sum aristas}{\sum nodos}$.

- Distancia media: Se define como la media de las distancias entre cada par de nodos del grafo. Mide el número medio mínimo de aristas que hay entre cualquier par de nodos. Su fórmula de cálculo es: $\bar{D}(G) = \frac{\sum_{u,v} d_G(u,v)}{\binom{n}{2}}$.

- Diámetro: Es una medida directamente relacionada con la distancia media, ya que mide la mayor de las distancias mínimas entre cualquier par de nodos del grafo ($D(G) = max(d_G(u,v))$).

- Secuencia de grados: Es una representación del grado de los nodos, donde la posición i indica el grado del nodo v_i. Por ejemplo, la secuencia de grados [1, 2, 2, 1] define un grafo con 4 nodos, donde los nodos $v0$ y $v3$ tienen grado 1 y los nodos $v1$ y $v2$ tienen grado 2.

- Histograma de grados: Es otra representación del grado de los nodos, donde se indica la frecuencia de aparición de cada posible valor del grado entre 0 y el grado máximo de cualquier nodo del grafo. Por ejemplo, el histograma de grados [1, 2, 2, 1] define a un grafo con 6 nodos, de los cuales un nodo tiene grado 0, dos nodos tienen grado 1, dos nodos tienen grado 2 y un nodo tiene grado 3.

- *Betweenness centrality*: Este parámetro mide la frecuencia con la que cada vértice aparece en el conjunto de caminos cortos (*shortest paths*) dentro de un grafo. Por lo tanto, esta medida sirve para indicar la centralidad de un nodo basada en el flujo entre los demás nodos del grafo. Un nodo con un valor alto indica que este nodo forma parte de muchos caminos cortos del grafo, con lo cual será un nodo clave en la estructura del grafo. Su eliminación puede repercutir de forma muy directa en la conectividad del grafo.

- *Closeness centrality*: Este parámetro, que se aplica a cada vértice de forma individual, se define como la inversa de la distancia media a todos los vértices accesibles. Se presenta normalizada en el rango (0,1). Un valor elevado indica poca centralidad del vértice.

- *Degree centrality*: Este parámetro considera la centralidad de cada nodo asociada a su grado. Es decir, un mayor grado indica mayor centralidad en el grafo.

Dentro de las investigaciones recientes en grafos o redes, las centradas en redes sociales ocupan una parte muy importante de las mismas. En este sentido, cabe destacar algunas características propias de las redes sociales que no tienen porqué darse en otro tipo de redes. Entre las principales, cabe destacar:

- *Small world*: El fenómeno conocido como *small world* es la hipótesis de que la cadena de conocidos necesarios para conectar a dos personas arbitrarias en cualquier parte del mundo es generalmente corta. El concepto se relaciona con la famosa frase *six degrees of separation* que surgió del experimento [40] que realizó el psicólogo Stanley Milgram y otros en el año 1967. Aplicado a los grafos en general y las redes sociales en particular, implica que la separación entre dos nodos cualquiera del grafo suele ser pequeña.

- *Power law*: La ley de la potencia aplicada a los grafos, y en particular a las redes sociales, implica que la distribución de los nodos en función de su grado sigue la ley de la potencia. Es decir, pocos nodos concentran muchas relaciones, mientras que la mayoría de los nodos tienen pocas relaciones.

2.2. Medidas de calidad del proceso de anonimización

Las medidas utilizadas para evaluar la calidad de un proceso de anonimización o preservación de la privacidad son básicamente dos:

- La pérdida de información sufrida por el grafo en el proceso de anonimización.

- La posibilidad de re-identificación del grafo anonimizado.

Ambas medidas son dependientes del proceso de anonimización aplicado, y además, la pérdida de información puede ser dependiente del posterior proceso de *graph mining* al que sean sometidos los datos.

2.2.1. Pérdida de información

Para evaluar la posible pérdida de información sufrida por un grafo durante un proceso de anonimización se pueden usar medidas generales o medidas específicas que tengan en cuenta la utilidad que se pretende dar al grafo anonimizado.

Las medidas generales se basan en cuantificar, de alguna forma, la distorsión producida en el grafo por el proceso de anonimización. Estas medidas son independientes de la utilidad que se quiera dar a los datos. Generalmente se cuantifican comparando distintas características del grafo original y el grafo anonimizado.

Por el contrario, las medidas específicas consideraran el proceso de *graph mining* asociado, y evalúan la perdida de información ocasionada sobre el proceso de *graph mining*. El objetivo ideal sería poder evaluar la diferencia entre los resultados del grafo original y del grafo anonimizado obtenidos por el proceso de *graph mining*. Sin embargo, en muchos casos esta aproximación no será posible, y se debe de intentar preservar las propiedades del grafo que explota cada una de las distintas técnicas de *graph mining*, teniendo en cuenta, por tanto, el uso que se dará a los datos anonimizados.

Hay et al., en [20], proponen evaluar la pérdida de información con la evaluación de métricas comunes en la teoría de grafos. Los autores consideran que si consiguen mantener estas cinco medidas próximas a los valores originales, la pérdida de información será mínima. Proponen evaluar para cada nodo las siguientes métricas:

- *Closeness centrality*: media del camino más corto del nodo a todos los demás nodos.
- *Betweenness centrality*: proporción de todos los caminos más cortos que pasan a través del nodo.
- *Path length distribution*: se calcula a partir del camino más corto entre cada par de nodos.

Y las dos métricas globales para todo el grafo son:

- Diámetro (*diameter*): se calcula como la distancia máxima de los caminos más cortos entre cada par de nodos.
- *Degree distribution*: distribución de los grados de los nodos.

En [8], Campan y Truta han realizado un trabajo con grafos simples y simétricos, pero con etiquetas en los nodos. En este trabajo se evalúan dos tipos de pérdida de información de forma independiente: por un lado se evalúa la pérdida de información producida por la generalización de las etiquetas de los nodos de la red; y por otra parte se evalúa la pérdida de información producida por las modificaciones estructurales que se hayan producido en la red durante el proceso de anonimización.

Ying et al., en [47], utilizan un conjunto de características de los grafos para evaluar la pérdida de información producida en el proceso de anonimización. Estas son:

- h: la media armónica de la distancia más corta.
- Q: la medida de la modularidad. Indica la bondad de la estructura de la comunidad.
- C: la medida de transitividad.

- SC: la medida de la centralidad. Cuantifica la centralidad de un nodo i basándose en los subgrafos:

$$SC = \frac{1}{n}\sum_{i=1}^{n} SC_i = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=0}^{\infty}\frac{P_i^k}{k!}$$

  Donde P_i^k es el número de caminos que empiezan y finalizan en el nodo i con longitud k.

- λ_1: los valores propios de la matriz de adyacencia.

- μ_2: los segundos valores propios de la matriz de Laplace.

De las seis características presentadas, las cuatro primeras hacen referencia al dominio espacial, mientras que las dos últimas pertenecen al dominio espectral. Un año antes, en 2008, Ying y Wu [48] realizaron un estudio comparativo entre distintos métodos de anonimización basados en la modificación aleatoria de aristas. Para evaluar el grado de afectación producido en el grafo también se basaron en las mismas características del dominio espacial y espectral.

En [55] Zou et al. definen un método sencillo para la evaluación de la pérdida de información en redes simétricas y no-etiquetadas. El método se basa en la diferencia de aristas entre el grafo original y el grafo anonimizado. Cuanto mayor sea esta diferencia, mayor será la pérdida de información sufrida por el grafo. Si G es el grafo original y G^* es el grafo anonimizado, su fórmula de cálculo es:

$$Coste(G, G^*) = (E(G) \cup E(G^*)) - (E(G) \cap E(G^*))$$

Donde $E(G)$ representa el conjunto de aristas de G.

Cai et al. realizan una comparación entre tres algoritmos de *clustering* [7]. Los métodos utilizados para comparar el resultado del proceso de *clustering* entre ellos se pueden utilizar, también, para evaluar la pérdida de información sufrida por el proceso de anonimización. Estas técnicas permiten comparar dos resultados de un proceso de *clustering*. En el texto citado se han utilizado para comparar los resultados entre varios algoritmos y el mismo conjunto de datos. Pero bien pueden usarse sobre los resultados del conjunto original y el conjunto anonimizado sobre el mismo proceso de *clustering*. Los resultados proporcionan una medida de como ha afectado el proceso de anonimización al resultado del *clustering*. Los métodos utilizados en el texto son tres:

- *Precision*: Se define como la fracción de nodos correctamente agrupados sobre el número total de nodos. Es necesario conocer la clase de los nodos, es decir, es necesario un conjunto de datos etiquetados. Su valor oscila en el intervalo [0,1].

- *Normalized Mutual Information* (NMI): Mide la cantidad de información mutua compartida entre dos particiones. Su valor oscila en el intervalo [0,1].

- *Modularity*: Propuesta en 2003, es una métrica importante para evaluar la calidad de un proceso de *clustering*. Mide la proporción de aristas dentro de cada partición sobre esta misma cantidad si las aristas fueran aleatorias.

2.2.2. Riesgo de re-identificación

Evaluar las posibilidades de riesgo asociadas a la re-identificación de los datos es un proceso complejo y con muchas variables. En primer lugar, y como factor más importante, se debe evaluar el conocimiento que pueda poseer el adversario, ya que este conocimiento es básico para poder estimar el tipo de ataque que se puede producir sobre los datos publicados, y por consiguiente, poder evaluar si el proceso de anonimización ofrece una protección suficiente.

En esta linea, Hay et al. proponen, en [20], un modelo para evaluar *a-priori* la posibilidad de re-identificación sobre un grafo anonimizado. En este trabajo los autores presuponen un conocimiento del adversario variable que modelan a través de las *vertex refinement queries*. Según el modelo de *vertex refinement queries*, todos los nodos que forman el conjunto de candidatos para una determinada consulta $\mathcal{H}_i$ comparten el mismo valor de $\mathcal{H}_i$, es decir, son indistinguibles al nivel de $\mathcal{H}_i$. Por lo tanto, la protección de los nodos no será comprometida si el conjunto de candidatos es suficientemente grande. A partir de esta asunción, los autores realizan el cálculo del tamaño de los distintos conjuntos que forman las respuestas a $\mathcal{H}_1$, $\mathcal{H}_2$, $\mathcal{H}_3$ y $\mathcal{H}_4$. Para cualquiera de las consultas, si un grupo contiene sólo un nodo significa que se ha re-identificado este nodo, mientras que en los grupos con menos de 5 nodos se considera que hay un riesgo muy alto de re-identificación. Para que se pueda considerar que la privacidad de un grupo está asegurada ante la consulta $\mathcal{H}_i$, el grupo debe contener más de 20 nodos que respondan positivamente a $\mathcal{H}_i$.

El estudio ha sido realizado sobre cuatro grafos reales:

- Hep-Th: Describe autores y artículos. El subgrafo analizado está compuesto de 2.510 nodos y 4.737 aristas, con un grado medio de 3,77.

- Enron: Describe comunicaciones vía e-mail en la Corporación Enron. El subgrafo analizado está compuesto de 111 nodos y 287 aristas, con un grado medio de 5,15.

- Net-trace: Describe una red informática de una universidad a nivel de comunicaciones exteriores e interiores. El subgrafo analizado está compuesto de 4.213 nodos y 5.507 aristas, con un grado medio de 2,61.

- Net-common: Extraída a partir de Net-trace, describe sólo en tráfico interno de la universidad. El subgrafo analizado está compuesto de 187 nodos y 5.398 aristas, con un grado medio de 57,73.

En la red Enron, en su versión original, se consiguen conjuntos de candidatos en riesgo muy alto (menos de 5 nodos) en cerca del 15 % de los conjuntos con $\mathcal{H}_1$. Con un conocimiento superior, $\mathcal{H}_2$, se consigue re-identificar cerca del 70 % de los nodos, dejando al resto en un riesgo muy alto de re-identificación. La red Net-common obtiene unos resultados aún peores que Enron, mientras que Hep-Th demuestra resistencia a $\mathcal{H}_1$ pero la tasa de re-identificación se dispara con $\mathcal{H}_2$ llegando al 40 %. Net-trace es la que mejor resistencia demuestra, aunque con la consulta $\mathcal{H}_2$ se consiguen cerca de un 10 % de re-identificaciones.

En [47], Ying et al. realizan un estudio comparativo entre dos métodos de anonimización, el primero basado en la modificación de la estructura de la red (*Rand Add/Del*) y el segundo basado en la generalización (*k-degree generalization*). Los autores cuantifican el riesgo de re-identificación de un individuo o nodo (*identity disclosure*) y el riesgo de re-identificación de relaciones sensibles entre nodos (*link disclosure*) de forma independiente. El método utilizado se basa en el cálculo de probabilidades a partir de un conocimiento del adversario basado en el grado de todos los nodos del grafo.

2.3. Métodos de anonimización o preservación de la privacidad

El objetivo de los métodos que se presentan aquí es preservar la privacidad de los usuarios o individuos involucrados en un proceso de publicación de datos. Es decir, se pretende publicar un conjunto de datos, en formato de grafo, para poder extraer cierto conocimiento a partir de estos datos. El conocimiento se extraerá a partir de procesos de minería de datos, que utilizarán los datos disponibles para extraer información y conocimiento sobre el dominio presentado.

Cualquier método de anonimización o preservación de la privacidad persigue un doble objetivo:

- Evitar al máximo los posibles procesos de re-identificación que se puedan producir sobre los datos anonimizados.
- Minimizar la pérdida de información que se pueda producir entre los datos originales y los datos anonimizados.

Se llama anonimización simple o *naive anonimization* al proceso más básico para conseguir tal propósito. Este proceso consiste en eliminar o reemplazar las etiquetas que puedan resultar clave para la identificación de los usuarios o individuos de los nodos del grafo. Es decir, se deben identificar las etiquetas que puedan permitir relacionar a un nodo con el usuario o individuo que representan y eliminar o reemplazar dichas etiquetas. Por ejemplo, en el caso de un grafo

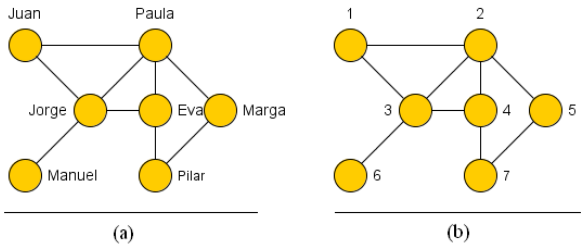

Figura 2.5: (a) Grafo de una red social, G. (b) Resultado de *naive anonymization* del grafo G.

que representa una red social, cada nodo representa a un usuario. En este caso, se deben identificar las etiquetas que puedan relacionar el nodo con el usuario de la red. Algunas de estas etiquetas son, por ejemplo, el nombre, DNI, correo electrónico o teléfono. En un proceso de anonimización simple se eliminarán las etiquetas que puedan comprometer la privacidad del usuario y en su lugar se generará, si es necesario, un identificador único para el nodo. El nodo debe poder ser referenciado de forma única dentro del grafo, entre otros, por el mismo proceso de minería de datos. Una técnica habitual es generar identificadores sintéticos o aleatorios para cada nodo, eliminando cualquier otra información que pueda ser relacionada con el usuario de la red social. En la Figura 2.5 se muestra un grafo G y un posible resultado de un proceso de anonimización simple. Este proceso genera un mapa de anonimización, en el cual se muestra la correspondencia entre cada nombre y la etiqueta sintética que lo representa. Evidentemente, el mapa de correspondencia no se hará público en ningún caso.

Pero como se ha visto anteriormente, un atacante con cierta información externa sobre la topología del grafo puede comprometer la privacidad de los usuarios, consiguiendo re-identificar de forma satisfactoria a algunos usuarios con los nodos que los representan en el grafo anonimizado.

La anonimización o preservación de la privacidad es consecuencia directa de la información contenida en el grafo y del conocimiento que pueda poseer el adversario. En ambos conceptos resulta muy importante el tipo de grafo que se quiere anonimizar y los datos que contiene. En este sentido, siguiendo la propuesta de [3], se distingue entre los grafos simples y los grafos complejos.

Zhou y Pei en [53] afirman que para definir el problema de la preservación de privacidad en la publicación de datos de redes sociales es necesario analizar los siguientes puntos:

1. Identificar la información privada que se debe preservar.

2. Modelar el conocimiento del adversario, es decir, el conocimiento que se presupone tendrá el posible adversario.

3. Definir el uso que se pretende dar a los datos que se harán públicos. Esto es, conocer la

tipología del proceso de *graph mining* que se aplicará para determinar sobre qué características se debe minimizar la pérdida de información.

Básicamente se consideran tres grandes bloques de propiedades que interesa preservar en los procesos de anonimización, para que la afectación en los procesos posteriores de minería de datos sea mínima:

- Propiedades topológicas del grafo (*graph topological properties*): En este grupo se pueden encontrar las propiedades relacionadas con la estructura o topologia de una red. Se pueden considerar dentro de este grupo cualquier medida basada en el grado de los nodos, las relaciones de vecindad, densidad de aristas, etc.

- Propiedades espectrales del grafo (*graph spectral properties*): Considera las propiedades relacionadas con el espectro de un grafo. Generalmente estas medidas se basan en el conjunto de valores propios (*eigenvalues*) obtenidos de la matriz adyacencia y otras matrices derivadas de ésta.

- Consultas agregadas (*aggregate network queries*): Las consultas agregadas responden a una pregunta de forma unificada sobre una parte o la totalidad del grafo. Por ejemplo, la distancia media entre dos tipos de nodos de una red.

Este estudio se centra en los grafos simples, ya que son un tipo de grafo muy utilizado y permiten representar una gran variedad de redes. A casi cualquier red se le puede quitar la información no deseada y se puede mostrar como un grafo simple. Se entiende por grafo simple aquel que no contiene atributos en los nodos ni etiquetas en las aristas o arcos. En este caso el conocimiento del adversario se basa, principalmente, en la estructura del grafo (grados de los nodos, subestructuras, etc). El tratamiento en grafos complejos se deja como un trabajo futuro.

Por ejemplo, en [27], Liu y Terzi presuponen un conocimiento del atacante basado en el grado de los nodos para comprometer la privacidad de los individuos. En [20, 53, 19, 50, 5, 55] los autores se basan en la información estructural, facilitada por las relaciones de vecindad entre nodos y las subestructuras dentro de la red, para atacar la privacidad de algunos individuos de la red.

Se consideran tres categorias que agrupan los métodos de anonimización en grafos simples:

- Modificación para la preservación del modelo k-anonimidad.

- Modificación aleatoria de aristas.

- Generalización basada en agrupaciones de nodos.

2.3.1. Modificación para la preservación del modelo k-anonimidad

El objetivo en estos modelos es modificar el grafo para que no sea posible identificar a un nodo o grupo de nodos de forma única dentro del grafo. Los nodos o grupos de nodos similares modifican su estructura para ser iguales o equivalentes y dificultar de esta forma el proceso de re-identificación a los posibles adversarios. Un adversario intenta localizar un nodo dentro del grafo anonimizado partiendo de su conocimiento sobre la topología de la red, y en concreto, del nodo objetivo y sus alrededores. El éxito del ataque depende del conocimiento del atacante y de la topología de la red.

En 2002, Sweeney presenta en [39] el concepto de k-anonimidad (*k-anonymity*). Inicialmente se diseña y presenta para datos relacionales, pero posteriormente ha sido adaptado para trabajar con datos semi-estructurados.

Formalmente, se define el modelo k-anonimidad como: Si $RT(A_1, \ldots, A_n)$ es una tabla y QI_{RT} es un conjunto de casi-identificadores (*quasi-identifiers*) asociado a RT, se dice que RT safisface el modelo k-anonimidad si y sólo si cada una de las secuencias de valores en $RT\,[QI_{RT}]$ aparecen como mínimo k veces en $RT\,[QI_{RT}]$.

De esta forma el modelo indica que un atacante no puede diferenciar k registros entre si, aunque consiga encontrar un grupo de casi-identificadores. Por lo tanto, no podrá re-identificar a un individuo con una probabilidad superior a $\frac{1}{k}$.

Aplicando este modelo a los grafos, se pueden utilizar distintos conceptos como casi-identificador. Una primera opción adoptada en múltiples trabajos es utilizar el concepto de grado de los nodos como casi-identificador. De esta forma, se supone que el atacante intentará identificar nodos en el grafo original que tengan un grado único en todo el grafo, es decir, $grado(v_i) \neq grado(v_j) \forall j \neq i$. A partir de este conjunto de nodos con grado único, el atacante deberá buscar nodos en el grafo anonimizado con el mismo grado que los nodos identificados sobre el grafo original. Si la correspondencia es única entre ambos, el atacante habrá conseguido identificar positivamente un conjunto de nodos. Por el contrario, si no existe ningún nodo en el grafo anonimizado con grado único, el atacante no podrá re-identificar los nodos. En estos casos se dice que el grafo es k-anónimo en el grado, siendo k el cardinal del menor de los conjuntos de nodos del mismo grado. Por ejemplo, un grafo es 5-anónimo en el grado si la cardinalidad del menor de los conjuntos de nodos del mismo grado es 5. Es decir, si se agrupan los nodos según su grado no puede haber ningún conjunto con cuatro o menos nodos, y debe haber uno o más conjuntos con cinco nodos.

Estos modelos se caracterizan por modificar la estructura del grafo a partir de la inserción y eliminación de aristas. El objetivo que persiguen con estas modificaciones es que cualquier nodo sea igual a otros $k-1$ nodos en términos de patrones estructurales.

En [19] Hay et al. proponen un modelo generalizado del concepto de k-anonimidad, llamado

k-anonimidad de candidatos (*k-candidate anonymity*). En este modelo se define a un nodo x como k-anónimo de candidatos con respecto a una pregunta Q si existen, al menos, $k-1$ otros nodos en el grafo que responden positivamente a la misma pregunta Q. Formalmente, $\mid cand_Q(x) \mid \geq k$ donde $cand_Q(x) = \{y \in V \mid Q(y) = Q(x)\}$. Se dice que un grafo satisface la restricción k-anonimidad de candidatos con respecto a Q si todos sus nodos son k-anónimos de candidatos respecto a Q.

Este concepto permite ampliar el modelo de k-anonimidad según el conocimiento que se presuponga del adversario. Por ejemplo, en [27] Liu y Terzi asumen un conocimiento del adversario basado en el grado de los nodos objetivos. En [53] Zhou y Pei consideran el subgrafo formado por los vecinos a distancia 1 (*1-neigborhood*) de los nodos objetivos. Y en [55] Zhou et al. consideran toda la información estructural posible alrededor del nodo objetivo y proponen un nuevo modelo llamado k-automorfismo (*k-automorphism*) para garantizar la privacidad ante ataques con este tipo de información.

k-anonimidad basada en el grado

En [27], Liu y Terzi investigan como modificar la estructura de un grafo G añadiendo y eliminando aristas, para que el nuevo grafo $\widetilde{G}$ cumpla las restricciones de k-anonimidad en el grado (*k-degree anonymity*). Esta restricción implica que para cada nodo deberán existir, al menos, otros $k-1$ nodos con el mismo grado. Este modelo preserva la privacidad ante ataques de re-identificación basados en el conocimiento del grado de los nodos objetivo. En general, cuanto mayor sea el valor de k, mayor será el grado anonimización de $\widetilde{G}$ y mayor también la pérdida de información. En el caso extremo de $k =\mid G \mid$ todos los nodos en $\widetilde{G}$ tendrán el mismo grado; la posibilidad de re-identificación será prácticamente nula, pero la pérdida de información en $\widetilde{G}$ será muy grande, dejando el grafo prácticamente sin utilidad.

Formalmente, el grafo anonimizado $\widetilde{G}(\widetilde{V}, \widetilde{E})$ a partir las inserciones y eliminaciones de aristas en el grafo original $G(V, E)$ deberá cumplir las siguientes restricciones:

1. $\widetilde{G}$ debe ser k-anónimo en el grado
2. $V = \widetilde{V}$
3. $\widetilde{E} \cap E = E$

El método desarrollado consiste en construir una secuencia de grados k-anónima a partir de la secuencia de grados del grafo original. Una vez obtenida esta secuencia de grados, se deberá implementar minimizando el número de modificaciones (añadir aristas) necesarias para obtener un grafo que cumpla con la secuencia especificada. Por lo tanto, el grafo generado

se obtiene añadiendo aristas al grafo original. En algunas ocasiones puede no ser posible generar el grafo deseado a partir de adiciones de aristas, y es necesario poder eliminar algunas (las mínimas) aristas del grafo original. Este método se conoce como la versión relajada del algoritmo.

Formalmente, estas son las dos fases principales del algoritmo:

1. A partir de la secuencia de grados de los nodos de $G(V, E)$, d, se construye una nueva secuencia $\widetilde{d}$ que sea k-anónima en el grado y se minimiza la distancia $\| \widetilde{d} - d \|$.

2. Se construye un nuevo grafo $\widetilde{G}(\widetilde{V}, \widetilde{E})$ en el cual la secuencia de grados sea igual a $\widetilde{d}$, $\widetilde{V} = V$ y $\widetilde{E} \cap E = E$ (o $\widetilde{E} \cap E \approx E$ en la versión relajada).

En el texto indicado se puede hallar una descripción completa de los diferentes algoritmos desarrollados y de sus bases teóricas, así como un completo test realizado con grafos sintéticos y reales (*prefuse*, *enron*, *powergrid* y un grafo de co-autores). Las evaluaciones empíricas muestran buenos resultados en base a las características topológicas de los grafos anonimizados.

k-anonimidad basada en el subgrafo de vecindad

Zhou y Pei presuponen, en [53], que el atacante posee conocimientos sobre los nodos adyacentes a los nodos objetivo, es decir, que conoce el grafo de vecindad a 1 (*1-neighborhood*) de los nodos objetivos. Proponen un método basado en estrategias *greedy* que permite generalizar las etiquetas de los nodos e insertar aristas para conseguir que que los subgrafos de vecindad a 1 sean iguales en grupos de k miembros.

La definición formal de *k-neighborhood anonymity* es la siguiente: Un nodo u es *k-neighborhood anonymous* si existen, al menos, $k-1$ otros nodos $v_1, \ldots, v_{k-1} \in V$ tales que el subgrafo inmediato (vecindad a 1) de cada nodo $v_1, \ldots, v_{k-1}$ es isomorfo al subgrafo inmediato del nodo u. Un grafo es *k-neighborhood anonymous* si todos sus nodos satisfacen la condición de *k-neighborhood anonymous*.

Este mismo concepto se puede extender ampliando el concepto de vecindad a valores mayores de 1 ($d > 1$).

En [53] los autores se centran en la preservación de la privacidad de los individuos de una red, presuponiendo un conocimiento del adversario basado en la vecindad a 1 de los nodos objetivo. Es decir, se presupone un conocimiento de la estructura o topología local (a distancia $d = 1$) de los nodos objetivo. El objetivo de la red anonimizada será responder a consultas sobre valores agregados (*aggregate network queries).*

Los autores abordan el problema considerando una red social como un grafo simple con los nodos etiquetados, $G = (V, E, L, \mathcal{L})$, donde V es el conjunto de nodos, $E \subseteq V \times V$ es el conjunto

de aristas, L es el conjunto de etiquetas y $\mathcal{L}$ es la función de etiquetado $\mathcal{L} : V \rightarrow L$ que asigna a cada nodo una etiqueta. Se define una jerarquía en el conjunto de etiquetas, L, de tal forma que la relación entre dos etiquetas puede ser: $\ell_1 \prec \ell_2$ si ℓ_1 es más general que ℓ_2, $\ell_1 \succ \ell_2$ si ℓ_1 es más específico que ℓ_2, $\ell_1 = \ell_2$ si ambas etiquetas tienen el mismo orden o $\ell_1 \preceq \ell_2$ y $\ell_1 \succeq \ell_2$ si el orden es parcial. La raíz de la jerarquía es ocupada por la meta-etiqueta $*$, que representa la etiqueta más general posible.

El objetivo es construir un nuevo grafo, G', tal que:

1. G' es k-anónimo.
2. Todos los nodos de G han sido anonimizados a un nodo en G' y no existe ningún nodo falso en G'.
3. Todas las aristas de G han sido conservadas en G'.
4. G' puede ser usado para responder consultas sobre valores agregados.

El algoritmo desarrollado por Zhou y Pei para construir G' consiste en dos fases principales:

1. En la primera fase se extraen los vecinos de todos los nodos de la red. Para facilitar la comparación entre las estructuras adyacentes a cada nodo se propone la técnica *Neighborhood Component Coding Technique*, que permite codificar subgrafos en un formato conciso que facilita su comparación.
2. En la segunda fase se organizan los nodos en grupos y se anonimizan los nodos del mismo grupo.

Estas dos fases se detallan a continuación: En la primera fase se deberá extraer los vecinos a distancia 1 de todos los nodos de G y codificarlos de forma que su comparación pueda ser factible. Para implementar esta operación se utilizará la codificación DFS (*depth-first search tree*) mínima descrita en [45]. Esta notación permite determinar si dos grafos son isomorfos realizando la comparación mediante los dos códigos DFS. Es decir, dos grafos G y G' son isomorfos si, y sólo si, $DFS(G) = DFS(G')$. Mediante esta técnica se pueden agrupar los nodos según conceptos de vecindad. Sobre los grupos generados se aplicará dos tipos de operaciones para anonimizar los nodos: (1) generalización de las etiquetas de los nodos y (2) añadir aristas en caso de ser necesario. Hay que destacar que ambas operaciones pueden producir pérdida de información.

Para el análisis de los resultados se utiliza el modelo R-MAT para generación de datos sintéticos [9] con las dos principales propiedades de los datos de redes sociales: (1) distribución de los nodos según la ley de la potencia (*power law distribution*) y (2) el fenómeno *small-world* (también llamado *six degrees of separation*).

k-automorfismo

En [55] Zou et al. van un paso más allá, y consideran que un atacante puede conocer cualquier subgrafo que contenga a un nodo objetivo α. Si tal grafo puede ser identificado de forma única, se compromete la privacidad del usuario α. El objetivo de los autores es construir un grafo $\widetilde{G}$ tal que para cualquier subgrafo $X \subset G$, $\widetilde{G}$ contiene, al menos, $k-1$ subgrafos isomorfos a X.

El concepto de automorfismo aplicado a un grafo $G(V, E)$ se define como una función automórfica f sobre el conjunto de nodos V, en donde cada arista $e = (u, v)$ de G, $f(e) = (f(u), f(v))$ es también una arista en G.

Formalmente, una red o grafo G se considera *k-automorphism* si:

- Existen $k-1$ funciones automórficas $F_a(a = 1, \ldots, k-1)$ en G.
- Y para cada nodo v en G, $F_{a_1}(v) \neq F_{a_2}(v)(1 \leq a_1 \neq a_2 \leq k-1)$.

A partir de la definición se extrae que para todo nodo v de una red *k-automorphic* no es posible distinguir al nodo v de los demás $k-1$ nodos simétricos basándose en información estructural. Por lo tanto, un adversario no podrá re-identificar al nodo v con una probabilidad mayor de $\frac{1}{k}$.

La idea que subyace en el algoritmo propuesto por Zou et al. es la siguiente: Si se asume que existen $k-1$ funciones automórficas $F_a(a = 1, \ldots, k-1)$ en la red anonimizada G^*, y que para cada nodo v, $F_{a_1}(v) \neq F_{a_2}(v)(a_1 \neq a_2)$, entonces, para cada nodo v en G^* habrá siempre $k-1$ otros nodos simétricos. Es decir, no habrá diferencias estructurales entre v y los demás $k-1$ nodos simétricos usando información estructural. Por lo tanto, un atacante no podrá re-identificar a un individuo con una probabilidad mayor a $\frac{1}{k}$. La clave del problema es determinar las funciones automórficas F_a. En el artículo se presentan tres métodos para desarrollar dichas funciones: *graph partitioning*, *block alignment* y *edge copy*. A partir de estos métodos se desarrolla el algoritmo *K-Match* (KM) que permite generar un grafo G^* en base al concepto *k-automorphism* a partir de un grafo original G.

Los autores realizan distintas pruebas con datos reales (*Prefuse* y grafo de co-autores) y sintéticos (*Pajek*, *Erdos Renyi Model* y *Scale-free Model*). Los datos sintéticos se han generado con un valor por defecto de 1000 nodos, mientras que en los datos reales se trabaja con 129 nodos y 161 aristas en el caso de *Prefuse* y 7955 nodos y 10055 aristas en el grafo de co-autores.

2.3.2. Modificación aleatoria de aristas

Estos modelos se caracterizan por modificar el grafo original a partir de una secuencia aleatoria de inserciones y eliminaciones de aristas. La técnica de añadir ruido aleatorio a los datos originales se ha usado ampliamente en métodos de preservación de la privacidad con datos

relacionales. Para trabajar con datos semi-estructurados en forma de grafo se ha imitado este modelo, adaptándolo a los requerimientos de los grafos. Existen dos estrategias básicas:

- Añadir/eliminar aristas de forma aleatoria (*Random Add/Del*): Se añaden k aristas falsas de forma aleatoria y luego se eliminan k aristas originales del grafo. El número total de aristas se preserva en el grafo anonimizado.
- Intercambio aleatorio de aristas (*Random Switch*): Se intercambian las aristas entre dos pares de nodos, de forma iterativa (k veces). Esta estrategia mantiene el grado de todos los nodos. Por ejemplo, las aristas (t, w) y (u, v) se intercambian por (t, v) y (u, w).

En estos modelos se persigue un doble objetivo: (1) preservar la identidad de los nodos y (2) preservar la anonimidad de las relaciones entre los nodos. En un grafo que cumpla con el modelo k-anonimidad en el grado, se puede producir una rotura en la seguridad que revele información acerca de las relaciones entre distintos nodos. Por ejemplo, se supone que un adversario pretende conocer si existe una relación entre dos nodos v_1 y v_2. El modelo de k-anonimidad en el grado no permitirá identificar de forma única cada uno de los nodos. En su lugar sólo se podrán identificar dos conjuntos de nodos con el mismo grado que v_1 y v_2. Es decir, se obtienen dos conjuntos V_{G1} donde $v_i \in V_{G1} \Leftrightarrow \ grado(v_i) = grado(v_1)$ y V_{G2} donde $v_i \in V_{G2} \Leftrightarrow \ grado(v_i) = grado(v_2)$. En el caso de que existan relaciones (aristas) entre todos los nodos de V_{G1} y todos los nodos de V_{G2}, se puede inferir con total seguridad que existe una relación entre v_1 y v_2, aún sin poder determinar individualmente a los dos nodos. Por lo tanto, en una situación como esta, se podría comprometer la privacidad de las relaciones (*link disclosure*) aún cumpliendo con el modelo de la k-anonimidad.

En [20] Hay et al. proponen un método para la anonimización de grafos no etiquetados. El método, llamado *Random Perturbation*, consiste en dos fases: (1) en primer lugar se eliminan m aristas escogidas de forma aleatoria y uniforme del grafo original, y (2) a continuación se añaden m aristas escogidas también de forma aleatoria entre todas las aristas no existentes en el grafo original. El conjunto de nodos no sufre modificación alguna en el proceso de anonimización. Cabe destacar que si $m = |E|$, entonces el grafo generado será simplemente un nuevo grafo aleatorio con los mismos nodos que el grafo original, con lo cual la pérdida de información habrá sido total.

Para evaluar la pérdida de información inducida por el modelo *Random Perturbation* los autores han considerado cinco métricas. Consideran que si consiguen mantener estas cinco métricas próximas a los valores originales, la pérdida de información será mínima, o expresado de otra forma, la utilidad de los datos se habrá mantenido en un valor adecuado. Las métricas propuestas son: (1) *closeness centrality*: evalúa el camino más corto entre un nodo y todos los demás; (2) *betweenness centrality*: mide la proporción de caminos más cortos que pasan

a través de un nodo determinado. Y para cada par de nodos: (3) *path lenght distribution*: calculado a partir del camino más corto entre cada par de nodos. Finalmente, para todo el grafo se calcularán las métricas: (4) *degree distribution*: distribución de grados de los nodos; y (5) *diameter*: la longitud máxima del camino más corto entre cualquier par de nodos del grafo.

Según los experimentos realizados con el conjunto de datos *Enron*, los valores de m cercanos al 5 % del número total de aristas del grafo original, $|E|$, consiguen niveles aceptables de privacidad (evaluados a partir del valor de *k-anonymity*) y de utilidad en el grafo anonimizado.

Ying et al., en [47], realizan una evaluación sobre cómo las estrategias de modificación basadas en añadir/eliminar nodos de un grafo (*Rand Add/Del* y *k-degree generalization*) preservan la identidad sobre los nodos (*identity disclosure*) y relaciones sensibles (*link disclosure*) entre los nodos de un grafo. En este trabajo se presupone un conocimiento del adversario basado en el grado de los nodos. Las simulaciones realizadas con ambos métodos: (1) *Rand Add/Del* y (2) *k-degree generalization* demuestran que el segundo método preserva mejor las características estructurales de la red.

En [52] se examina la probabilidad de la existencia de una relación entre dos individuos en base a la estimación de la densidad de aristas entre las dos clases. A continuación los autores proponen un algoritmo basado en la modificación de aristas con el objetivo de reducir las probabilidades de rotura en la privacidad de las relaciones entre nodos.

En [48], Ying y Wu estudian los efectos de las estrategias de modificación aleatoria de aristas sobre la privacidad de las relaciones de los grafos. En el texto se incluyen estrategias basadas en añadir/eliminar aristas e intercambio aleatorio de aristas entre nodos. En [50] los mismos autores modelan un ataque que explota la relación entre la probabilidad de la existencia de una arista y una medida de similitud entre pares de nodos del grafo anonimizado.

Se pueden modelar estos métodos como una operación entre matrices: $\widetilde{A} = A + E$, donde $\widetilde{A}$ y A representan las matrices de adyacencia del grafo anonimizado y original, respectivamente, y E representa la matriz de perturbación, en donde cada posición representa:

$$e_{ij} = \begin{cases} 1 & se\ crea\ arista\ (i,j) \\ -1 & se\ elimina\ arista\ (i,j) \\ 0 & otro\ caso \end{cases}$$

En datos relacionales, se conocen métodos que permiten reconstruir la matriz original (A) a partir de un conocimiento *a priori* sobre el tipo de perturbación que se ha aplicado a la red. Aunque se está investigando sobre la posibilidad de aplicar estos métodos en modelos de datos de red, la complejidad para definir la distribución de los datos de la red y la aleatoriedad de los nodos aplicados al proceso, complica de forma significativa estos procesos.

En [44] se realiza un estudio sobre los métodos de aproximación de rango bajo para reconstruir las características estructurales de un grafo anonimizado mediante técnicas de añadir/eliminar aristas aleatoriamente. También se han realizado estudios para aplicar métodos basados en PCA (Análisis de Componentes Principales, *Principal Component Analysis*) y otras características espectrales para conseguir el mismo propósito. Son los llamados *métodos de reconstrucción*, y su objetivo es reconstruir el grafo original partiendo del grafo anonimizado. Hasta la fecha, los resultados obtenidos no son tan buenos como los obtenidos trabajando con datos relacionales.

Los autores de [48] presentan una estrategia de aleatorización que permite preservar las propiedades espectrales del grafo. La matriz del grafo mantiene importantes relaciones con estas propiedades, tales como: diámetro, *clusters* o grupos, caminos entre nodos, centralidad, etc. En el texto, los autores intentan preservar la utilidad de los datos a partir de preservar el mayor de los valores propios (*eigenvalues*) de la matriz de adyacencia y el segundo menor de los valores propios de la matriz de Laplace. Los autores proponen dos algoritmos (*Spctr Add/Del* y *Spctr Switch*) que mantienen las características espectrales próximas a sus valores originales. Estos y otros métodos se incluyen en el grupo de los métodos llamados de *aleatorización preservando el espectro* del grafo. En cualquier operación que modifique la topología de una red, se debe considerar la repercusión que dichas modificaciones puedan causar en la estructura global de la red. Para preservar la utilidad de la red anonimizada se debe minimizar el grado de alteración de las características básicas de la red. Pero, tal y como describen los autores, algunas características importantes se pueden ver demasiado afectadas por el uso de operadores basados en modificación aleatoria de aristas.

Otras tendencias tienen en consideración la secuencia de grados de los nodos o algunas características topológicas del grafo (como la transitividad o la media de la distancia entre dos nodos) como elementos importantes que se deben mantener inalterables en la generación del grafo anonimizado. En [18, 49] se describen métodos diseñados para preservar algunas propiedades implícitas en las redes sociales reales, además de la secuencia de grados. Se llama $\mathcal{G}_{d,S}$ al espacio de grafos que: (1) satisfacen la secuencia de grados d y (2) preservan un conjunto de propiedades S dentro de un margen acotado. Por lo tanto, este espacio contiene todos los grafos que satisfacen ambas propiedades. A partir del grafo original, las permutaciones de las cadenas de Markov permiten explorar este espacio $\mathcal{G}_{d,S}$. En [49] los autores proponen un algoritmo que permite generar cualquiera de los grafos contenidos en $\mathcal{G}_{d,S}$ con igual probabilidad. En [18] se propone un algoritmo que genera un grafo con altas probabilidades de que sea un grafo con propiedades muy cercanas al grafo original.

2.3.3. Generalización basada en agrupaciones de nodos

Los dos modelos vistos anteriormente, k-anonimidad y aleatorización de aristas, se basan en modificar la estructura del grafo mediante la adición y/o eliminación de aristas. En cambio, los modelos basados en generalización se basan en agrupar nodos y aristas, formando subgrafos que se anonimizan como super-nodos o super-aristas. De esta forma, los detalles de los individuos quedan ocultos en el grafo resultante. El grafo resultante deberá ser útil para estudios de macro-propiedades del grafo original.

En [19] Hay et al. describen una técnica de anonimización que protege de los ataques de re-identificación a través de la generalización del grafo original. El proceso de generalización se basa en agrupar los nodos en particiones, para luego publicar las particiones indicando el número de nodos de cada partición y la densidad de aristas dentro de cada partición y entre las distintas particiones. Este esquema permite trabajar posteriormente con la información agregada de la estructura de la red, ocultando los detalles de los individuos a posibles atacantes.

En el extremo, se podría llegar a tener un único super-nodo con todos los nodos del grafo original en su interior. En este caso se consigue una anonimización total, pero también una utilidad nula. Al otro extremo, se puede crear un super-nodo para cada nodo del grafo original, de forma que el grafo anonimizado corresponde topologicamente al grafo original después de un simple proceso de anonimización simple (o *naive anonymization*). Para controlar este aspecto, el método propone el uso de un parámetro para controlar el tamaño de las particiones. De esta forma se establece un umbral mínimo de nodos que contendrá cada super-nodo, estableciendo un modelo similar al k-anonimidad en donde cada super-nodo representa, por lo menos, k nodos. Para cada partición o super-nodo sólo se hará público la densidad de aristas de los nodos interiores.

Para mantener la utilidad, las particiones deben ajustarse a la red original en la mayor medida posible, cumpliendo las condiciones impuestas por la anonimidad. Este modelo se basa en el cálculo de probabilidades para determinar la mejor puntuación entre el conjunto de posibles generalizaciones. Dicha probabilidad se define a partir del tamaño de los conjuntos posibles que implica la partición. Para cada generalización $\mathcal{G}$, el número de aristas en el super-nodo X se denota como $c(X, X)$, el número de aristas entre los super-nodos X e Y se denota como $c(X, Y)$ y el conjunto de todas las especializaciones posibles consistentes con $\mathcal{G}$ se denotan como $\mathcal{W}(\mathcal{G})$, que se calcula como:

$$|\mathcal{W}(\mathcal{G})| = \prod_{X \in V} \binom{\frac{1}{2}|X|(|X|-1)}{c(X,X)} \prod_{X,Y \in V} \binom{|X||Y|}{c(X,Y)}$$

Entonces, la probabilidad para un grafo $g \in \mathcal{W}(\mathcal{G})$ es de $\frac{1}{|\mathcal{W}(\mathcal{G})|}$. La partición de nodos se elige de forma que el grafo generalizado satisface las condiciones de privacidad y maximiza la

utilidad ($\frac{1}{|\mathcal{W}(\mathcal{G})|}$).

El algoritmo busca una aproximación a la partición óptima. El proceso se inicia con una única partición que contiene todos los nodos, y utiliza operaciones de división y combinación de particiones o movimiento de nodos entre particiones hasta hallar un máximo local que cumpla los requerimientos mencionados anteriormente.

En los experimentos realizados, la evaluación de los grafos anonimizados sobre cuestiones estructurales ha producido resultados diversos: las redes sociales y de comunicaciones han demostrado mayor resistencia a los ataques que los grafos generados de forma aleatoria.

2.4. Re-identificación y conocimiento del adversario

Para preservar la privacidad de los grafos es necesario conocer como son atacados y qué conocimiento debe poseer un adversario para poder realizar con éxito estos ataques. En esta sección se presentan algunos métodos relevantes de re-identificación y se discute el conocimiento de puede y debe poseer un adversario para poder ejecutar estos ataques.

Backstrom et al. presentan en [5] una primera clasificación para los ataques basados en redes sociales, aunque se pueden extender la mayoría de las consideraciones a los grafos en general:

- Ataques activos: El atacante tiene acceso a la red antes del proceso de anonimización de la red. Mediante este acceso, podrá crear nuevos nodos y aristas. El objetivo es intentar colocar nodos estratégicos que formen un patrón conocido antes del proceso de anonimización. Este patrón servirá para identificar de forma efectiva los nodos creados junto con otros nodos que tengan relaciones establecidas con los nodos creados. Este caso, por ejemplo, se puede dar en la mayoría de las redes sociales: generalmente será posible crear un número arbitrario de perfiles de usuario y poder generar una topología de red conocida *a priori*.

- Ataques pasivos: El atacante no puede crear nodos o aristas. En este caso sólo puede identificar algunas subestructuras o topologías e intentar identificar el subgrafo dentro del grafo de la red anonimizado, comprometiendo la privacidad de los individuos afectados o sus vecinos más próximos.

La eficacia de un ataque activo se basa en la singularidad del subgrafo H creado por el atacante antes del proceso de anonimización. Este subgrafo H debe satisfacer tres propiedades para que el ataque pueda tener éxito: (1) Sólo debe haber una aparición del subgrafo H dentro del grafo original G. (2) H debe poder ser identificable de una forma eficiente. Y (3), el subgrafo H no debe tener automorfismos no triviales.

A partir de un grafo $G(V, E)$ con n nodos ($|V| = n$), se ha demostrado de forma teórica que cualquier subgrafo generado de forma aleatoria con $O(\sqrt{\log n})$ nodos puede comprometer (con una probabilidad alta) la privacidad de un grupo arbitrario de nodos. Por lo tanto, si H es un subgrafo de orden $O(\sqrt{\log n})$ dentro de un grafo G de orden n, es altamente probable que H sea único dentro de G y sea posible re-identificar de forma positiva los nodos asociados a H.

Por otro lado, el ataque pasivo se basa en la observación que muchos individuos (nodos) de una red social pertenecen a un único subgrafo. Esta consideración básica para los ataques pasivos se apoya en algunas propiedades observadas en distintas redes sociales reales, pero que puede no ser extrapolable a otros tipos de grafos.

En el mismo texto, Backstrom et al. presentan dos ataques activos contra una red no-dirigida y sin etiquetas. El primero de estos ataques, llamado *walk-based attack*, demuestra que con $k = \Theta(\log n)$ nuevos nodos generados en el grafo original, se genera un subgrafo aleatorio H que será único con una probabilidad muy elevada y que será factible recuperar H dentro del grafo anonimizado. Los autores demuestran que en la práctica se pueden usar valores aún menores para k, y realizan experimentos satisfactorios con valores de $k = 7$ en redes de más de 4 millones de nodos. En el segundo de los ataques, llamado *cut-based attack*, se presenta un modelo muy similar, pero en el que el sub-grafo H se une al grafo original con unas pocas aristas. Esto permite que H sea identificable de una forma más sencilla dentro del grafo original, y permite reducir el valor de k. Los experimentos demuestran que estos ataques permiten la re-identificación de algunos nodos de la red (del orden de k^2 para el caso de *walk-based attack* y del orden de k para el caso de *cut-based attack*) con una alta probabilidad.

En [27] los autores presentan una clasificación para ataques de re-identificación basados en el objetivo del ataque:

1. Divulgación de la identidad (*identity disclosure*): Se consigue comprometer, de forma única, la identidad del individuo asociado a un nodo o nodo de la red. La mayoría de los ataques son de este tipo.

2. Divulgación de las relaciones (*link disclosure*): Se consiguen comprometer las relaciones entre dos o más individuos.

3. Divulgación del contenido (*content disclosure*): Se consigue romper la privacidad de la información asociada a cada nodo del grafo. Por ejemplo, obtener el contenido de un mensaje enviado entre dos usuarios de un grafo que muestre las comunicaciones entre distintos usuarios de una empresa o entidad, constituye una ataque de divulgación del contenido.

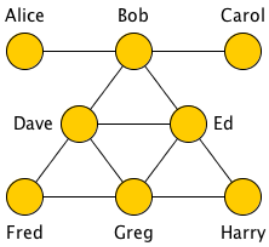

Figura 2.6: Ejemplo de grafo para la demostración de *vertex refinement queries*.

Node ID	$\mathcal{H}_0$	$\mathcal{H}_1$	$\mathcal{H}_2$
Alice	ϵ	1	{4}
Bob	ϵ	4	{1, 1, 4, 4}
Carol	ϵ	1	{4}
Dave	ϵ	4	{2, 4, 4, 4}
Ed	ϵ	4	{2, 4, 4, 4}
Fred	ϵ	2	{4, 4}
Greg	ϵ	4	{2, 2, 4, 4}
Harry	ϵ	2	{4, 4}

Cuadro 2.1: *Vertex refinement queries* aplicados al grafo de la Figura 2.6.

Hay et al., en [20], formalizan dos clases de consultas (*knowledge queries*) que pueden ser utilizadas por un adversario con cierto conocimiento externo. Una consulta Q sobre un grafo G se define como cualquier información que un atacante puede utilizar para extraer información de G. El resultado de una consulta Q es un conjunto de nodos $V' \subseteq V$, en donde cada $v_i \in V'$ es una coincidencia positiva (*match vertex*). Las consultas formalizadas son:

- *Vertex refinement queries*
- *Subgraph knowledge queries*

Las *vertex refinement queries* se basan en un modelo iterativo desarrollado originalmente para resolver de forma eficiente problemas de isomorfismo entre grafos, y que permiten obtener información acerca de la estructura de una red entorno a un nodo determinado. La primera de la consultas, $\mathcal{H}_0$, proporciona la etiqueta del nodo. A partir de esta consulta inicial, las demás consultas proporcionan información más descriptiva de forma iterativa. La segunda consulta, $\mathcal{H}_1(x)$, indica el grado del nodo x. La tercera, $\mathcal{H}_2(x)$, responde con una lista de los grados de los nodos adyacentes a x. De forma general, la consulta $\mathcal{H}_i(x)$ responde con el conjunto de valores que produce la evaluación de $\mathcal{H}_{i-1}$ en todos los nodos adyacentes a x.

La tabla 2.1 muestra el cálculo de $\mathcal{H}_0$, $\mathcal{H}_1$ y $\mathcal{H}_2$ para todos los nodos de la Figura 2.6. Como se puede ver en el ejemplo, la consulta $\mathcal{H}_0$ sobre cualquier nodo responde con la etiqueta

ϵ, indicando que el grafo no contiene etiquetas. La consulta $\mathcal{H}_1$ indica el número de nodos adyacentes (es decir, el grado) y la consulta $\mathcal{H}_2$ responde con un conjunto indicando el grado de los nodos adyacentes.

A partir de este concepto se define la *equivalencia relativa*: dos nodos x, x' en un grafo son equivalentes en relación a $\mathcal{H}_i$, escrito como $x \equiv_{\mathcal{H}_i} x'$ si y sólo si $\mathcal{H}_i(x) = \mathcal{H}_i(x')$.

En el ejemplo anterior se puede ver que Bob, Dave, Ed y Greg son equivalentes en relación a $\mathcal{H}_1$, es decir, son equivalentes en la relación $\equiv_{\mathcal{H}_1}$, mientras que sólo se cumple $Dave \equiv_{\mathcal{H}_2} Ed$.

Entonces, para un adversario con determinado conocimiento que se pueda modelar como una consulta $\mathcal{H}_i$, todos los nodos equivalentes con respecto a $\mathcal{H}_i$ son indistinguibles. Formalmente: Sean $x, x' \in V$ y $x \equiv_{\mathcal{H}_i} x'$ entonces $cand_{\mathcal{H}_i}(x) = cand_{\mathcal{H}_i}(x')$.

Pero el gran inconveniente de este modelo es que no considera las informaciones parciales que pueda poseer un adversario. Es decir, no se puede modelar el conocimiento de un adversario que conozca parte de los nodos adyacentes de un determinado nodo, pero no en su totalidad. Para superar esta limitación se han desarrollado las llamadas *subgraph knowledge queries*.

En [19] Hay et al. realizan un análisis similar, incluyendo tres tipos de conocimiento que puede usar un adversario para atacar un grafo anonimizado de forma simple, es decir, con el método de *naive anonymization*. En el texto se modela el conocimiento del adversario como el acceso a una fuente que proporciona respuestas a preguntas de conocimiento restringido (Q, *restricted knowledge query*) sobre un nodo concreto del grafo original. Se modelan tres tipos de preguntas como base del conocimiento del adversario:

- *Vertex refinement queries*: Las respuestas a estas preguntas permiten describir estructuras locales del grafo alrededor de un nodo de forma iterativa, permitiendo cierto grado de refinamiento.

- *Subgraph queries*: Mediante estas preguntas se puede confirmar la existencia de un subgrafo entorno a un nodo objetivo. Explorando las características de vecindad de un nodo objetivo, el adversario puede obtener información parcial sobre la topología de red alrededor del nodo objetivo.

- *Hub fingerprint queries*: Se llama *hub* a un nodo o nodo de una red con un grado y centralidad muy elevados. El *fingerprint*, $\mathcal{F}_i(x)$, para un nodo objetivo x es una descripción de las conexiones entre el nodo x y un conjunto determinado de *hubs* de la red.

Narayanan y Shmatikov presentan otros dos tipos de conocimiento del adversario en [32]: información auxiliar agregada y información auxiliar individual. En ambos casos se emplea un grafo auxiliar, llamado $G_{aux}(V_{aux}, E_{aux})$. En el primer caso se obtiene información más generalista sobre estructuras o patrones, junto con sus probabilidades de aparición. En el segundo caso,

en cambio, se obtiene información mucho más concreta y específica sobre unos pocos nodos, que serán llamados semillas o *seeds*. En un primer paso, estas semillas serán re-identificadas en el grafo anonimizado, y partir de este punto se inicia una fase de propagación iterativa, en la cual se intenta re-identificar los nodos adyacentes a los nodos re-identificados. En el texto se presenta un algoritmo de de-anonimización basado en estos conceptos, que consigue re-idenitificar de forma satisfactoria una gran cantidad de nodos del grafo anonimizado con la técnica de *naive anonymization*.

En [53] Zhou y Pei realizan un experimento (comentado más ampliamente en 2.3.1) sobre la posibilidad real de efectuar un ataque a una red basándose en el conocimiento sobre la vecindad a 1. Los autores utilizan una red de co-autores con datos reales extraídos de arXiv[1]. Esta red contiene 57.448 nodos (autores) y 120.640 aristas (relaciones de colaboración entre autores), con un grado medio de los nodos próximo a 4. En el experimento los autores anonimizan la red de dos formas distintas y luego evalúan la posibilidad de re-identificación de los nodos suponiendo un conocimiento del adversario basado en la vecindad a distancia 1. En la primera de las formas se aplica un proceso de anonimización simple (*naive anonymization*), es decir, se eliminan las etiquetas de los nodos. En este caso, un 1.3 % de los nodos del grafo incumplen el modelo *k-anonymity* con un valor de k igual a 5. El valor aumenta hasta el 12 % si se aumenta el valor de k hasta 20. En la segunda forma se aplica una generalización sobre las etiquetas de los nodos, de modo que se sustituye el nombre de los autores por el nombre de la institución a la cual pertenecen. En este caso las probabilidades de re-identificación aumentan considerablemente; casi un 13 % de los nodos de la red incumplen el modelo de *k-anonymity* con un valor de k igual a 5. Si se quiere un valor de k igual a 20, el porcentaje de nodos que infringen la regla sube hasta más del 23 %.

2.5. *Graph mining*

Se entiende por *graph mining* o minería de grafos el conjunto de técnicas de minería de datos aplicadas a los grafos. Es un caso especial de la *structured data mining*, que engloba las técnicas de minería de datos aplicadas a cualquier conjunto de datos semi-estructurados.

Dentro de las técnicas de *graph mining* se incluyen tres grandes bloques [3]:

- Búsqueda de patrones frecuentes (*frequent pattern mining*)
- Agrupamiento (*clustering*)
- Clasificación (*classification*)

[1] arXiv.org e-Print archive: http://arxiv.org/

Estas disciplinas presentan un desafío importante en cuanto a las representaciones intermedias y finales, que deben permitir una correcta interpretación de los resultados. Además, presentan un grado elevado de complejidad computacional y temporal.

2.5.1. Búsqueda de patrones frecuentes

La búsqueda de patrones frecuentes (*frequent graph mining*) consiste en identificar subgrafos que están presentes en una colección de grafos o en un grafo mayor con una frecuencia determinada por el usuario mediante un umbral. Las aplicaciones de la búsqueda de patrones frecuentes son múltiples: clasificar componentes químicos, estudiar propiedades biológicas, indexación de características, etc.

En [3, 43, 12] se pueden encontrar exhaustivos trabajos que repasan el estado del arte de la búsqueda de patrones frecuentes en grafos.

2.5.1.1. Búsqueda en un conjunto de grafos

A partir de un conjunto de grafos etiquetados $D = \{G_1, G_2, \ldots, G_n\}$ y un subgrafo g, se determina el conjunto de soporte de g en D como $D_g = \{G_i | g \subseteq G_i, G_i \in D\}$. El soporte de g es $support(g) = \frac{|D_g|}{|D|}$. Un subgrafo frecuente en un determinado conjunto es aquel subgrafo con un soporte igual o superior a un mínimo establecido para el conjunto dado [3].

Entre los primeros estudios sobre la búsqueda de patrones frecuentes [43] en grafos se puede destacar los trabajos realizados por Cook y Holder en 1994 que les permitió desarrollar el algoritmo *SUBDUE*. En el mismo año, Yoshida y Motoda desarrollaron el algoritmo *GBI*. Ambos métodos se basan en aproximaciones *greedy* para superar la complejidad del espacio de soluciones, produciendo soluciones incompletas. Posteriormente, en 1998, Deshape y Toivonen proponen el algoritmo *WARMR* basado en ILP (*inductive logic programming*) que les permite realizar una búsqueda completa en el espacio de soluciones. Dos años después, Nijssen y Kok desarrollan el algoritmo *FARMAR*, que presenta mejoras significativas en el rendimiento.

Durante los primeros años de la década del 2000 se desarrollan un grupo de algoritmos para búsqueda de patrones frecuentes basados en el algoritmo *Apriori*, utilizado para búsqueda de reglas de asociación en un conjunto de datos relacionales. En estos algoritmos la búsqueda se inicia con subgrafos pequeños, que van aumentando su tamaño de forma iterativa. Algunos de los principales algoritmos desarrollados en este grupo son: *AGM algorithm* [23], *FSG algorithm* [25] y *edge-disjoint path-disjoint algorithm* [42].

Otra importante tendencia los constituyen un grupo de algoritmos conocidos como *pattern-growth approach*. En este grupo la técnica consiste en extender un subgrafo frecuente a partir de añadir una arista en las distintas posiciones posibles. Algunos de los principales algoritmos

incluidos en este grupo son: *gSpan* [45], *MoFa* [6], *FFSM* [21], *SPIN* [22], *Gaston* [33],

Los algoritmos de búsqueda de patrones frecuentes pueden incluir una gran cantidad de ítems en los resultados, ya que si un subgrafo es frecuente todos sus subgrafos también lo son. Por lo tanto, un subgrafo frecuente de n aristas tiene un potencial de 2^n posibles subgrafos frecuentes. Esta problemática se puede resolver con los algoritmos que generan conjuntos de subgrafos cerrados (*closed subgraph*), es decir, que en el conjunto de subgrafos no se incluyen dos subgrafos g y g' tales que $g \subset g'$ y $support(g) = support(g')$. Un ejemplo de este modelo es el algoritmo *CloseGraph* [46].

2.5.1.2. Búsqueda en un único grafo

Los métodos presentados hasta ahora permiten la búsqueda de subgrafos frecuentes dentro de una conjunto de múltiples grafos. Otra posibilidad es la búsqueda de patrones frecuentes dentro un único grafo. El concepto es similar, pero se añade la problemática de que dos o más subgrafos se puedan solapar entre sí, generando inconsistencias en la definición de la métrica de soporte. Se puede consultar un análisis en profundidad de la problemática y una posible solución en el texto en que Kuramochi y Karypis [26] presentan dos algoritmos: *HSIGRAM* y *VSIGRAM*.

2.5.2. Agrupamiento o *clustering*

El agrupamiento o *clustering* en grafos contiene dos contextos diferenciados:

1. Agrupamiento de nodos: En el primer contexto se trata de determinar agrupaciones de nodos dentro de un único grafo (*node clustering*). Dentro de este contexto se pueden incluir problemas de partición de grafos (*graph partitioning*) o el problema del corte mínimo (*minimum cut problem*).

2. Agrupamiento de grafos: En el segundo contexto se tiene un conjunto de grafos y se quiere agruparlos en base a un concepto de distancia o similitud (*graph clustering*).

2.5.2.1. Agrupamiento de nodos

Las tareas de *clustering* o agrupamiento consisten en agrupar los datos según una medida de similitud. En el caso del agrupamiento de nodos (*node clustering*) en grafos se trata de agrupar los nodos de un grafo en varios conjuntos según un criterio o medida de similitud específica.

Inicialmente se puede considerar el caso en que se desea agrupar los nodos de un grafo en dos conjuntos disjuntos. Este caso es una especialización del caso para k conjuntos, que se verá a continuación. El problema se define a partir de un grafo con pesos en las aristas

o arcos, y sobre el que se quiere crear dos conjuntos disjuntos de nodos, con la condición de minimizar el peso de las aristas o arcos que unen ambos grupos. Para aplicar el problema a los grafos simples, sin pesos en las aristas, se presupone un valor constante para el peso de todas las aristas. El grafo puede ser simétrico o dirigido. Formalmente, se define un grafo $G(V, E)$ sonde V es el conjunto de nodos y E es el conjunto de aristas o arcos. Cada arista (i, j) tiene asociado un peso, definido por u_{ij}. El objetivo es partir el conjunto de nodos, V, en dos conjuntos: V y $V - S$, de tal forma que la suma de los pesos del conjunto de aristas o arcos que unen un nodo de V y otro de $V - S$, denotado como $C(S, V - S)$, sea mínimo. Es decir, el objetivo es minimizar $\sum_{(i,j)\in C(S,V-S)} u_{ij}$. El problema es análogo a otros problemas clásicos de la teoría de grafos, como por ejemplo, el problema del corte mínimo (*minimum cut problem*) o el problema del flujo máximo (*maximum flow problem*), que han sido ya ampliamente estudiados. En [4] Ahuja et al. presentan diferentes métodos que permiten resolver este problema de forma eficiente con una complejidad temporal polinomial. Existen otras aproximaciones que mejoran el rendimiento temporal a partir de contraer nodos para mejorar el rendimiento del algoritmo, como por ejemplo el algoritmo descrito en [10].

Cuando se quiere partir el grafo original en más de dos conjuntos, $k > 2$, el problema se complica y se convierte en *NP-hard* (o NP-complejo). En [24] se describe una de las técnicas clásicas para partición de grafos, el algoritmo de Kerninghan-Lin, que se basa en estrategias de búsqueda locales (concretamente *hill-climbing*) para determinar las particiones óptimas del grafo. El algoritmo funciona de la siguiente forma:

1. Inicialmente se divide el grafo de forma aleatoria en k conjuntos.
2. Se intercambian un par de nodos entre dos particiones distintas, y se evalúa el resultado del cambio.
3. Si el resultado es positivo, el cambio se realiza; en caso contrario se retrocede.
4. El proceso 2-3 se repite hasta alcanzar la convergencia a una solución óptima.

Cabe destacar que el algoritmo no garantiza que se alcance el óptimo global, y puede terminar dando como solución un óptimo local.

Una de las claves del algoritmo es la selección de los nodos a intercambiar (paso 2). Existen múltiples variaciones en este punto que conducen a comportamientos distintos del algoritmo. En [15] se puede encontrar un estudio sobre los distintos métodos.

Rattigan et al., en [37], proponen un nuevo algoritmo que usa características de los algoritmos *K-means* y *K-medoids*, muy utilizados en la minería de datos con datos relacionales. Al igual que éstos, el algoritmo se inicia tomando k nodos como semillas (*seeds*) del grafo original.

Los principales cambios se encuentran en la forma de calcular la distancia entre nodos y las semillas posteriores. En el caso de grafos etiquetados se puede emplear el peso de las relaciones para el cálculo de las distancias entre nodos, mientras que en los grafos no etiquetados se puede emplear el número de saltos u otras técnicas.

El algoritmo de Girvan-Newman [17] permite una división en *clusters* sin necesidad de utilizar semillas u otros conceptos aleatorios para la inicialización del algoritmo. En este algoritmo se explota el concepto de *edge betweenness centrality*: identifica aristas centrales que forman puentes entre distintos componentes conexos del grafo y los elimina, de manera que se van formando grupos conexos aislados que formaran los distintos conjuntos o *clusters*. Las condiciones de parada del algoritmo pueden ser variadas, dando lugar a distintas configuraciones de conjuntos. La principal ventaja de este algoritmo es que no depende de unos parámetros aleatorios de inicialización.

De forma análoga a los métodos utilizados en minería de datos relacionales, en *graph mining* se han desarrollado métodos que utilizan las propiedades espectrales de los grafos. A partir de la matriz de adyacencia, la matriz de Laplace y otras representaciones de los grafos se analizan los componentes espectrales del grafo que permiten generar las divisiones para el *clustering*. En [11] se pueden encontrar detalles de diferentes métodos que se basan en el espectro de un grafo para realizar el *clustering*.

Las técnicas vistas se basan en crear las particiones a partir de minimizar la densidad de aristas entre grupos o particiones. De forma complementaria, existen otro conjunto de técnicas basadas en maximizar la densidad de aristas dentro de una partición. Estas técnicas son llamadas *quasi-cliques*. Un *clique* se define como un grafo o subgrafo completo. Un *quasi-clique* es una versión relajada en la que se impone un grado mínimo para los nodos que forman parte del *quasi-clique*. Formalmente se llama $\gamma - quasiclique$ a un grafo G de k nodos ($k \geq 1$) si el grado de cada nodo es $\gamma \cdot k$ o superior. El valor de γ está comprendido en el intervalo $(0, 1]$. En [1] Abello et al. presentan el algoritmo GRASP, un algoritmo de búsqueda adaptativo basado en estrategias *greedy* para encontrar un *quasi-clique* dentro de un grafo.

En [16] se estudia el problema de determinar subgrafos de alta densidad en grafos masivos. En el texto los autores presentan un nuevo modelo que permite abordar el problema en grafos de gran envergadura. El algoritmo es aplicado con éxito sobre un grafo que representa las conexiones entres computadores en la World Wide Web. El grafo presenta cerca de 50 millones de nodos y 11 billones de aristas. En [3] se puede encontrar un exhaustivo estudio de algoritmos para esta problemática.

En el año 2000 se presenta MCL (*Markov Cluster Algorithm*) [41], un algoritmo no supervisado para *clustering* basado en la simulación del flujo. Cuatro años después, Pons y Latapy presentan el algoritmo *WalkTrap* [36], desarrollado para la detección de comunidades en grafos

grandes. El algoritmo se basa en el concepto de *random walk* y tiene una complejidad espacial y temporal de orden cuadrática. Más tarde, en el año 2009, Macropol et al. presentan el algoritmo RRW (*Repeated Random Walks*) [29], basado en *random walks* y diseñado para detectar agrupaciones en proteínas y otras estructuras biológicas. En el texto, los autores realizan pruebas empíricas y comparaciones con el algoritmo MCL. Posteriormente, Macropol y Singh [30] desarrollan el agoritmo TopGC (*Top Graph Clusters*) ideado para escalar de forma eficiente con grafos grandes, directos y con pesos en las aristas. El algoritmo no busca todas las particiones del grafo, si no un subconjunto con las principales particiones. Esto le permite ejecutarse con una complejidad temporal lineal.

En [7] los autores realizan una evaluación del algoritmo RRW (*Repeated Random Walks*) en la tarea de detección de comunidades. El algoritmo es comparado con otros dos algoritmos: MCL (*Markov Cluster Algorithm*) y WalkTrap. Para la evaluación se utilizan dos redes sociales ampliamente conocidas: *Zachary's karate club network* y *American college footbal network*. Ambas redes son simétricas y no etiquetadas. Su tamaño es de 78 y 115 nodos, respectivamente. Las simulaciones muestran una precisión más elevada del algoritmo RRW frente a MCL y WalkTrap, aunque también requiere una más tiempo de cálculo.

2.5.2.2. Agrupamiento de grafos

En el agrupamiento de grafos (o *graph clustering*) se considera a los grafos como objetos con ciertas propiedades (estructurales, espectrales, etc) y el problema consiste en dividir un conjunto de grafos en grupos con propiedades similares.

Una de las principales utilidades de este tipo de *clustering* es la agrupación de documentos XML. En este trabajo no se entrará en detalle sobre este tipo de operaciones de *clustering*. Para más información se remite al lector a [13, 2].

2.5.3. Clasificación

Los problemas de clasificación en grafos se pueden dividir en dos categorías diferenciadas:

1. Clasificación de nodos: En esta categoría se pretende etiquetar los nodos de un único grafo a partir de unos datos de entrenamiento.

2. Clasificación de grafos: En esta categoría se tiene un conjunto de grafos y se pretende etiquetar cada uno de ellos como si se tratara de un objeto.

2.5.3.1. Clasificación de nodos

El objetivo de los métodos presentados en esta sección es etiquetar los nodos de un grafo a partir de un subconjunto de los nodos que ya están etiquetados. Es decir, generalmente el problema se presenta como un tipo de aprendizaje supervisado, en el que se utilizarán un conjunto de ejemplos para entrenar el algoritmo.

Tradicionalmente se han empleado métodos basados en *support vector machines* (SVM) y *diffusion kernels* [3]. Estos métodos se basan en una medida de proximidad entre nodos para definir la etiqueta de un nuevo nodo. En grafos no muy grandes obtienen buenos resultados, pero en el caso de grandes grafos su complejidad temporal ($O(n^3)$) y de memoria ($O(n^2)$) los hacen impracticables.

Recientemente han aparecido otros métodos basados en la propagación de etiquetas (*label propagation methods*) que utilizan modelos de cálculo más simples basados en la matriz de adyacencia que les permite obtener mejor rendimiento [54, 31].

2.5.3.2. Clasificación de grafos

Los métodos presentados tienen como objetivo clasificar un conjunto de grafos a partir de un segundo conjunto de entrenamiento que incluye grafos etiquetados. Este es un modelo de clasificación supervisado, aunque existen algoritmos para modelos de clasificación no-supervisados [3].

En [38] Saigo et al. realizan un estudio de distintos métodos supervisados de clasificación de grafos, principalmente: *graph kernels* y *graph boosting*. En el texto se puede hallar una descripción detallada de los fundamentos teóricos de ambos métodos.

Capítulo 3

Evaluación de anonimización basada en aleatoriedad

Como se ha visto en el capítulo anterior, existe una gran diversidad en la tipología de los grafos. Y para cada tipología concreta existen múltiples algoritmos o métodos para anonimizar los datos. Ante tal diversidad, se deberá escoger una tipología concreta de grafos sobre los que realizar el análisis. Para este estudio se ha escogido una tipología de grafos con las siguientes características:

- Nodos sin atributos.
- Aristas no dirigidas (bidireccionales).
- Aristas sin etiquetas o pesos.
- No se permiten lazos (aristas con origen y destino en el mismo nodo).
- No se permiten multi-aristas (más de una arista entre dos nodos).

Esta elección permite poder trabajar con una gran cantidad de datos reales disponibles en la red, ya que a cualquier grafo se le pueden eliminar los datos no deseados y se puede reducir a un grafo de estas características. Esto permite que estos algoritmos sean muy generalistas y se puedan aplicar en multitud de situaciones, siendo, por lo tanto, de amplio uso. Como contrapartida, cabe destacar que este tipo de grafos sólo proporciona información estrictamente topológica. Esto implica: (1) Los métodos, excepto los métodos de generalización, se basan en la alteración de las aristas del grafo. (2) La re-identificación es más compleja que en otros tipos de grafos, ya que el atacante sólo puede disponer de información topológica. Y (3) la perturbación producida por el proceso de anonimización puede reducir la utilidad de los datos

de forma drástica, ya que los procesos de *graph mining* posteriores sólo se pueden basar en la información topológica.

Para esta evaluación se han seleccionado tres conjuntos de datos reales. Estos son:

- *Zachary's Karate Club*: Es un pequeño grafo de 34 nodos y 78 aristas que representa las relaciones entre los miembros de un club de karate.
- *American College Football*: Es un grafo mediano de 115 nodos y 613 aristas. Muestra información de los equipos universitarios de la división IA que jugaron en la temporada regular de 2010.
- *Jazz musicians*: Es un grafo mediano que contiene 198 nodos, que representan músicos de jazz, y 2.742 aristas que representan las relaciones entre ellos.

3.1. Métodos de anonimización seleccionados

Para esta evaluación se analizan algoritmos de anonimización pertenecientes al grupo de anonimización basada en la modificación aleatoria de aristas. Se implementan dos algoritmos:

1. *Random Perturbation*: Se añaden y eliminan el mismo número de aristas del grafo original, de forma que se mantiene el número total de aristas en el grafo. El Algoritmo 1 muestra el pesudocódigo del algoritmo.
2. *Random Switch*: Se intercambian aristas de forma aleatoria. Se escogen dos aristas en el grafo original $e_a(v_{a1}, v_{a2})$ y $e_b(v_{b1}, v_{b2})$, se eliminan y se generan dos nuevas aristas de la forma: $e_c(v_{a1}, v_{b1})$ y $e_d(v_{a2}, v_{b2})$. De este modo, se mantiene el número total de aristas y el grado de cada nodo. El Algoritmo 2 muestra el pesudocódigo del algoritmo.

Para poder comparar de forma correcta ambos algoritmos, se ha introducido una pequeña modificación en el algoritmo *Random Perturbation*: el algoritmo *Random Switch* realiza la modificación de dos aristas en cada paso, ya que una modificación implica eliminar dos aristas existentes y crear dos nuevas aristas a partir de los cuatro nodos implicados. Por lo tanto, el algoritmo *Random Perturbation* eliminará/añadirá las aristas de dos en dos. De este modo, se equipara el número de modificaciones de aristas para ambos algoritmos.

Los experimentos realizados en este trabajo muestran porcentajes de anonimización en un rango variable. Aunque el punto de inicio de este rango siempre es el porcentaje de anonimización del 0 %, que representa el valor original de los datos (sin anonimización alguna), y finaliza en un porcentaje máximo entre el 20 % y el 50 %, dependiendo de cada experimento. Esta variabilidad permite escoger el rango más interesante en cada caso concreto, dependiendo

Algorithm 1 Pseudocódigo del algoritmo *Random Perturbation*

Entrada: El grafo original G y el porcentaje de anonimización p que se desea aplicar.
Salida: El grafo G anonimizado.

```
num = round(G.num_edges() * P)
edges_to_delete = {}
i = 0
while i < num do
  edges_to_delete.append(G.random_edge())
  i = i + 1
end while
i = 0
while i < num do
  G.add_edge(G.random_vertex(), G.random_vertex())
end while
for e in edges_to_delete do
  G.delete_edge(e)
end for
return G
```

Algorithm 2 Pseudocódigo del algoritmo *Random Switch*

Entrada: El grafo original G y el porcentaje de anonimización p que se desea aplicar.
Salida: El grafo G anonimizado.

```
num = round(G.num_edges() * p)
i = 0
while i < num do
  e1 = G.random_edge()
  e2 = G.random_edge()
  new_e1 = (e1.origen, e2.origen)
  new_e2 = (e1.destino, e2.destino)
  if !G.exist(new_e1) and !G.exist(new_e2) then
    G.add_edge(new_e1)
    G.add_edge(new_e2)
    G.delete_edge(e1)
    G.delete_edge(e2)
    i = i + 1
  end if
end while
return G
```

de los datos utilizados. En cada experimento se detalla el rango empleado. El número de aristas a modificar en cada caso se calcula como $round(m \times p_a)$, donde m representa el número de aristas y p_a es el porcentaje de anonimización aplicado. La fórmula de cálculo es la misma para ambos métodos.

Como se verá, no se muestra información del tiempo de ejecución de los algoritmos, ya que esta medida no se considera relevante para la evaluación que se realiza en este trabajo.

3.2. Medidas de calidad

Para evaluar la bondad de los métodos de anonimización se utilizan las dos medidas de calidad ya mencionadas previamente:

- Medidas para evaluar la pérdida de información: Cuantifican de forma objetiva distintas características del grafo con el objetivo de evaluar el grado de deterioro sufrido durante el proceso de anonimización.
- Medidas para evaluar la posibilidad de re-identificación: Se basan en un supuesto conocimiento del adversario para cuantificar el riesgo de re-identificación asociado a un grafo.

3.2.1. Pérdida de información

Las medidas utilizadas en este trabajo para evaluar la pérdida de información se pueden dividir en las que agrupan las propiedades estructurales, y aquellas que hacen referencia a los resultados del proceso de *graph mining*.

3.2.1.1. Propiedades estructurales

En esta evaluación se consideran seis medidas para cuantificar la pérdida de información estructural sufrida por el grafo durante el proceso de anonimización. Estos son:

- Distancia media: Mide el número medio mínimo de aristas que hay entre cualquier par de nodos.
- Diámetro: Está relacionada con la distancia media. Mide la mayor de las distancias mínimas entre cualquier par de nodos.
- Histograma de grados: Es una representación del grado de los nodos, donde se indica la frecuencia de aparición de cada posible valor del grado entre 0 y el grado máximo de cualquier nodo del grafo.

- *Betweenness centrality*: Este parámetro mide la frecuencia con la que cada vértice aparece en el conjunto de caminos cortos (*shortest paths*) dentro de un grafo.

- *Closeness centrality*: Este parámetro se define como la inversa de la distancia media a todos los vértices accesibles.

- *Degree centrality*: Este parámetro considera la centralidad de cada nodo asociada a su grado.

En la sección 2.1 se puede encontrar una descripción más detallada de estas medidas.

No se han considerado como parámetros a evaluar el número de nodos, el número de aristas y el grado medio, dado que en los métodos utilizados estos valores permanecen constantes.

3.2.1.2. Resultados del proceso de *graph mining*

Los métodos de *graph mining* empleados en este estudio se incluyen dentro de las técnicas de *clustering* (agrupamiento) de nodos. Es decir, su función es agrupar los nodos del grafo en distintos conjuntos. Por lo tanto, el resultado de aplicar a un grafo uno de estos métodos es la obtención de múltiples conjuntos de nodos, donde cada uno de ellos representa una clase (*cluster*) de nodos.

Para evaluar como afecta la perturbación introducida por el proceso de anonimización, se compara el resultado de aplicar el método de *clustering* sobre los datos originales con el resultado obtenido utilizando los datos anonimizados. En el caso ideal, los resultados deberían de ser los mismos. Es decir, la misma cantidad de conjuntos con los mismos elementos en cada conjunto. En este caso se podría decir que la anonimización de los datos no ha afectado al proceso de *clustering*. Cuando los conjuntos no coincidan, se deberá poder calcular el error cometido. Para tal fin se emplean los coeficientes de similitud entre grupos.

Los coeficientes de similitud permiten cuantificar de forma objetiva el grado de similitud entre dos o más conjuntos. Existen multitud de coeficientes para este cálculo. En este trabajo se empleará el coeficiente de similitud de Jaccard, más conocido como el índice de Jaccard.

El índice de Jaccard se define como el tamaño de la intersección entre los conjuntos dividido por el tamaño de la unión de los conjuntos. Cuando sólo se tienen dos conjuntos se define como sigue:

$$J(A, B) = \frac{\mid A \cap B \mid}{\mid A \cup B \mid}$$

Pero cuando se tienen dos conjuntos de conjuntos, se debe de buscar la correspondencia que maximiza la intersección entre los conjuntos de conjuntos. En el caso que $A = \{A_0, A_1, ..., A_n\}$

y $B = \{B_0, B_1, ..., B_m\}$, se puede formalizar como:

$$J(A, B) = \frac{\sum_{(i,j)} |A_i \cap B_j|}{|A \cup B|}$$

donde (i, j) son pares de índices que maximizan el número de elementos de las intersecciones, $|A_i \cap B_j|$, y donde cada conjunto $A_k \in A$ y $B_l \in B$ aparecerá, como máximo, una sola vez en el sumatorio.

Por lo tanto, el índice de Jaccard es un valor en el rango [0, 1], que toma el valor 0 cuando no existe ninguna coincidencia entre los conjuntos y el valor 1 cuando la coincidencia entre los conjuntos es total.

3.2.2. Riesgo de Re-identificación

Como se vio en el capítulo anterior, existen varios métodos para evaluar la posibilidad de re-identificación en un grafo anonimizado. Para poder hacer suposiciones en este sentido es clave presuponer el conocimiento de que dispondrá el atacante. Para este estudio se presupone un conocimiento del adversario basado en el número de nodos adyacentes a un nodo concreto, es decir, basado en el conocimiento del grado de uno o más nodos del grafo original.

Cálculo del valor de k-anonimidad

En base a esta definición del conocimiento del adversario, se empleará el modelo de k-anonimidad basada en el grado para evaluar la probabilidad de re-identificación asociada a un grafo. El cálculo del valor de k-anonimidad basada en el grado (en adelante, simplemente k-anonimidad) se realiza a partir del histograma de grados. A continuación se presenta un ejemplo para ilustrar este cálculo.

La Figura 3.1 muestra un grafo $G(V, E)$ con 6 nodos y 7 aristas. La secuencia de grados asociada a este grafo es $Dseq_G = [2, 3, 3, 2, 2, 2]$, e indica que el nodo 0 tiene grado 2, el nodo 1 tienen grado 3, etc. El histograma de grados asociado al grafo es $Hist_G = [0, 0, 4, 2]$, e indica que hay:

- 0 nodos con grado 0.
- 0 nodos con grado 1.
- 4 nodos con grado 2.
- 2 nodos con grado 3.

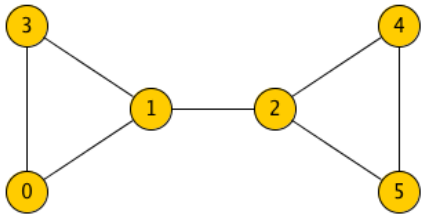

Figura 3.1: Grafo $G(V, E)$ para ejemplificar el cálculo del valor de k-anonimidad.

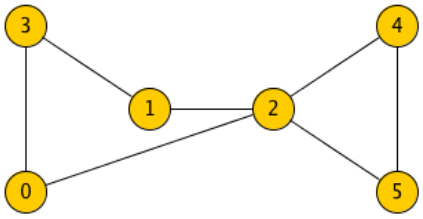

Figura 3.2: Grafo $G'(V, E')$ para ejemplificar el cálculo del valor de k-anonimidad.

A partir del histograma de grados, el cálculo del valor de k-anonimidad es directo y se define como el valor mínimo distinto de cero. Es decir, $k = min(n_i | n_i \in Histograma(G), n_i \neq 0)$. Para este ejemplo se establece un valor de k-anonimidad igual a 2. Este valor indica que existe uno o más conjuntos con sólo dos nodos, y que por lo tanto la probabilidad de re-identificación dentro de estos conjuntos es de $\frac{1}{2}$ ($\frac{1}{k}$ en general).

Cálculo del valor de k-anonimidad en grafos anonimizados

Para el cálculo del grado de k-anonimidad en un grafo anonimizado mediante técnicas de modificación de aristas se debe tener en cuenta un aspecto importante: los datos que servirán de conocimiento a un adversario para realizar un ataque son obtenidos a partir del grafo original. Las implicaciones de esta asunción se discuten a continuación.

Durante el proceso de anonimización mediante técnicas de modificación de aristas no se modifica el número de nodos del grafo, pero sí que se modifica el grado de algunos o todos los nodos del grafo. Este proceso, lógicamente, afecta a la secuencia de grados y al histograma de grados del grafo, que como se ha visto anteriormente, son la base para el cálculo del valor de k-anonimidad.

Las implicaciones que un proceso de anonimización tiene sobre el cálculo del valor de k-anonimidad se basan en el tipo de operaciones que realiza. Estas son: (1) eliminar aristas existentes y (2) añadir nuevas aristas. Ambas operaciones se pueden ver como modificaciones en el histograma de grados, de tal forma que una operación simple de eliminar una arista y crear otra nueva se puede ver como un cambio de grado en la secuencia e histograma de grados. Por ejemplo, continuando con el ejemplo de la Figura 3.1, si se elimina la arista $e_1 = (0, 1)$ y se crea una nueva arista $e'_1 = (0, 2)$ se obtiene el grafo $G'(V, E')$ representado en la Figura 3.2.

La secuencia de grados del grafo modificado es $Dseq_{G'} = [2, 2, 4, 2, 2, 2]$. Y el histograma de grados es $Hist_{G'} = [0, 0, 5, 0, 1]$. La perturbación generada por el algoritmo de anonimización puede ser modelada como una modificación en el histograma de grados.

La Figura 3.3 muestra una representación de los histogramas de grados de $G(V, E)$ y

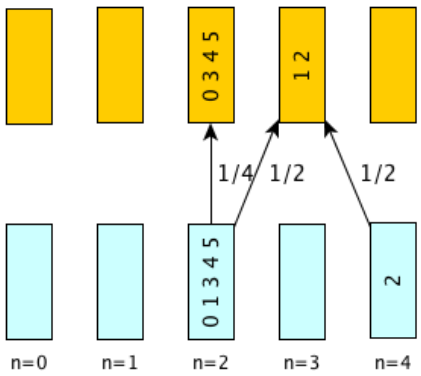

Figura 3.3: Relación entre el histograma de grados de $G(V, E)$ y $G'(V, E')$ (caso 1).

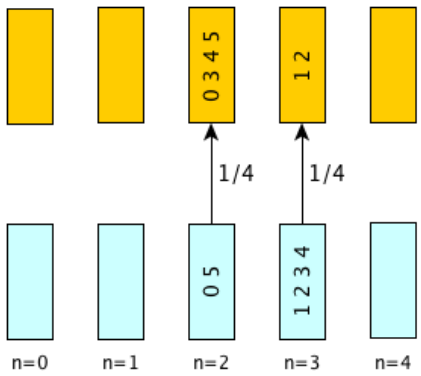

Figura 3.4: Relación entre el histograma de grados de $G(V, E)$ y $G''(V, E'')$ (caso 2).

$G'(V, E')$. La parte superior (color naranja) muestra la disposición de los nodos de G (Figura 3.1) dentro del histograma de grados. Se puede ver que en las dos primeras posiciones (grados 0 y 1) no hay ningún nodo, mientras que existen 4 nodos con grado 2 y 2 nodos con grado 3. La parte inferior (color azul) muestra los datos de G' (Figura 3.2). Si se aplica el cálculo del modelo de k-anonimidad de una forma directa, el valor de k-anonimidad para G' será $k = 1$. Analizando con mayor profundidad se puede ver que:

- El nodo nº 2 es el único nodo con un grado igual a 4 en G', por lo tanto, parece que su re-identificación es directa. Pero si un atacante pretende re-identificar este nodo deberá buscar su correspondencia en el grafo original, G, y en este grafo no encontrará ningún nodo con grado igual a 4. El atacante puede, entonces, presuponer que se ha aplicado algún proceso de anonimización sobre el grafo. En un caso muy simple, como este ejemplo, podría llegar a deducir por la disposición de los nodos, que se le ha añadido una arista y que en el grafo original este nodo tenía un grado igual a 3. En este caso, que desde luego no es trivial, deberá de re-identificar el nodo nº 2 dentro del grupo de nodos con grado igual a 3 en el grafo original, teniendo una probabilidad de re-identificación igual a $\frac{1}{2}$.

- El grupo de nodos de G' con grado igual a 2 (nodos nº 0, 1, 3, 4 y 5) continúan teniendo una probabilidad de re-identificación de $\frac{1}{4}$, excepto el nodo nº 1, que no puede ser re-identificado si el atacante no obtiene información de las modificaciones aplicadas durante el proceso de anonimización. Pero, aún teniendo esta información, la probabilidad de re-identificación es de $\frac{1}{2}$.

El caso complementario se presenta en la Figura 3.4. En este caso se crea una nueva arista $e_8 = (3, 4)$ en el grafo de la Figura 3.1 y se obtiene el grafo $G''(V, E'')$. Relacionando los datos con el grafo original se puede ver que:

- La re-identificación de los nodos de G'' con grado igual a 3 (nodos nº 1, 2, 3 y 4) se puede calcular como la suma de las probabilidades de escoger un nodo y re-identificarlo. Formalmente: $\sum_i (prob(v_i) \times prob_{rei}(v_i))$, que en este caso se convierte en: $\sum 2 \times (\frac{1}{4} \times 0) + 2 \times (\frac{1}{4} \times \frac{1}{2}) = \frac{1}{4}$. Es decir, si el atacante escoge el nodo nº 3 o 4 (caso que puede ocurrir con una probabilidad de $\frac{1}{4} \times 2$) no será capaz de re-identificarlo, ya que no existe en el conjunto original de nodos con grado igual a 3. Mientras que si escoge los nodos nº 1 o 2 (caso que puede ocurrir con una probabilidad de $\frac{1}{4} \times 2$), continua teniendo una probabilidad de re-identificación de $\frac{1}{2}$ en cada caso. Por lo tanto, en este caso, se ha aumentado el valor inicial de k-anonimidad, al insertarse cierta incertidumbre con el proceso de perturbación asociado al proceso de anonimización.

- El grupo de nodos de G'' con grado igual a 2 (nodos nº 0 y 5) mantienen la misma probabilidad de re-identificación, ya que sin el conocimiento de las alteraciones producidas en el grafo durante el proceso de anonimización no es posible reducir la probabilidad de re-identificación por debajo de $\frac{1}{4}$.

La idea que subyace en los casos vistos se puede aplicar a cualquier modificación que se produzca en el histograma de grados. Por lo tanto, se puede concluir que los métodos de anonimización de grafos simples basados en la modificación de aristas producen grafos con igual o mayor grado de k-anonimidad que el grafo original.

El cálculo del valor de la k-anonimidad se puede formular como:

$$k = max(min(Dseq_G), min(Dseq_{G'}))$$

Para poder calcular el valor de k-anonimidad del grafo anonimizado es necesario disponer del grafo original. En caso contrario, sólo se podrá inferir que el valor de k-anonimidad será igual o superior al valor de k-anonimidad del grafo anonimizado.

3.3. Métodos de *graph mining*

Para la evaluación de los resultados del proceso de *graph mining* se han empleado dos algoritmos de *clustering* para grafos:

- MCL Algorithm (Markov Cluster Algorithm)

- RRW (Repeated Random Walk)

Ambos son algoritmos no supervisados de *clustering* para grafos, pero presentan algunas diferencias importantes que hacen interesante el estudio de los resultados aplicados a ambos algoritmos.

El algoritmo MCL (*Markov Cluster Algorithm*) fue desarrollado por Stijn van Dongen [41] en el año 2000, y se basa en la simulación de flujo en los grafos. Este principio considera que se pueden encontrar más aristas entre los nodos de un mismo *cluster* (partición) que entre nodos de distintos *clusters*. Tomando este principio, el algoritmo considera que partiendo de un nodo y moviéndose aleatoriamente hacia uno de los nodos vecinos, existe una probabilidad más elevada de moverse dentro del mismo *cluster* que entre dos *clusters* distintos. Por lo tanto, moviéndose aleatoriamente por el grafo se puede descubrir como se mueve el flujo de forma natural en el grafo, identificando de esta forma los distintos *clusters* que hay. El movimiento aleatorio a través de distintos nodos del grafo se formaliza a través del concepto de *random walk*, que se calcula utilizando las cadenas de Markov aplicadas a la matriz de probabilidades de transición entre los nodos del grafo. Una característica importante de este algoritmo es que clasifica todos los nodos del grafo en alguno de los *clusters* y no permite solapamiento. Es decir, un nodo pertenece a un, y sólo a un, *cluster*. No se dan casos de nodos no asociados a ningún *cluster*. El autor proporciona una implementación en lenguaje C de este algoritmo. [1]

En este trabajo el algoritmo MCL se utiliza con el valor por defecto en todos los parámetros, exceptuando en el parámetro *inflation*, que permite controlar la granularidad de los conjuntos de resultados. Es decir, este parámetro ayudar a controlar si se desea un resultado con: (1) pocos conjuntos de muchos nodos, o (2) muchos conjuntos de pocos nodos. Su valor se ajusta en función de los datos de cada experimento.

El algoritmo RRW (*Repeated Random Walk*) fue desarrollado por Macropol, Can y Singh [29] en el año 2009 dentro del ámbito de la biología para descubrir módulos funcionales dentro de proteínas de gran tamaño. El algoritmo, como su nombre indica, se basa en el concepto de *random walk* aplicado de forma iterativa. Una diferencia importante con el algoritmo MCL es que RRW no clasifica todos los nodos del grafo; sólo algunos conjuntos de nodos son etiquetados como *clusters*, mientras que los demás simplemente no pertenecen a ningún *cluster*. En este caso el algoritmo permite que haya solapamiento entre los distintos conjuntos de nodos, es decir, nodos que pueden pertenecer a más de un conjunto a la vez. Este solapamiento esta controlado mediante un parámetro del algoritmo. Para el estudio que aquí se realiza, este parámetro se configura para no permitir solapamiento entre conjuntos, con el fin de simplificar el cálculo de similitud entre conjuntos, que representa el objetivo final de este estudio. Los autores proporcionan una implementación en lenguaje Java de este algoritmo. [2]

En este trabajo el algoritmo RRW se utiliza con el valor por defecto en todos los parámetros, excepto:

[1] Más información en: http://micans.org/mcl/
[2] Más información en: http://cs.ucsb.edu/~kpm/RRW/

- *overlap*: Este parámetro controla el grado de intersección entre los distintos *clusters*. En todos los experimentos se establece un valor igual a 0 para este parámetro, para no permitir que haya intersección entre *clusters*.
- *min*: Controla el tamaño mínimo de los *clusters* de resultados. Su valor se ajusta en función de los datos de cada experimento.
- *max*: Controla el tamaño máximo de los *clusters* de resultados. Su valor se ajusta en función de los datos de cada experimento.

3.4. Conjunto *Zachary's Karate Club*

Zachary's Karate Club [51] es un grafo ampliamente utilizado en la literatura. El grafo muestra las relaciones entre los 34 miembros de un club de karate en una universidad de los Estados Unidos en el año 1970. La Figura 3.5 muestra una posible disposición de la red.

Sus características básicas son:

- 34 nodos y 78 aristas.
- No dirigido y sin etiquetas en las aristas.
- El grado medio es de 4,588.
- La distancia media es de 2,408.
- El diámetro es de 5.
- El histograma de grados es: [0, 1, 11, 6, 6, 3, 2, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1].

El rango de anonimización utilizado para este experimento está comprendido entre los valores 0 % y 20 %. Para cada valor del porcentaje de anonimización se muestran los resultados promedios de 100 ejecuciones independientes.

A continuación se realiza la evaluación del grafo original y las anonimizaciones obtenidas mediante los algoritmos *Random Perturbation* y *Random Switch*.

3.4.1. Propiedades estructurales

Tal y como se ha comentado, los métodos que se evalúan en este capítulo se basan en la modificación aleatoria de aristas, es decir, eliminar y añadir aristas, manteniendo siempre un número constante de aristas (78 en este caso). Por lo tanto, el número de nodos no se modifica

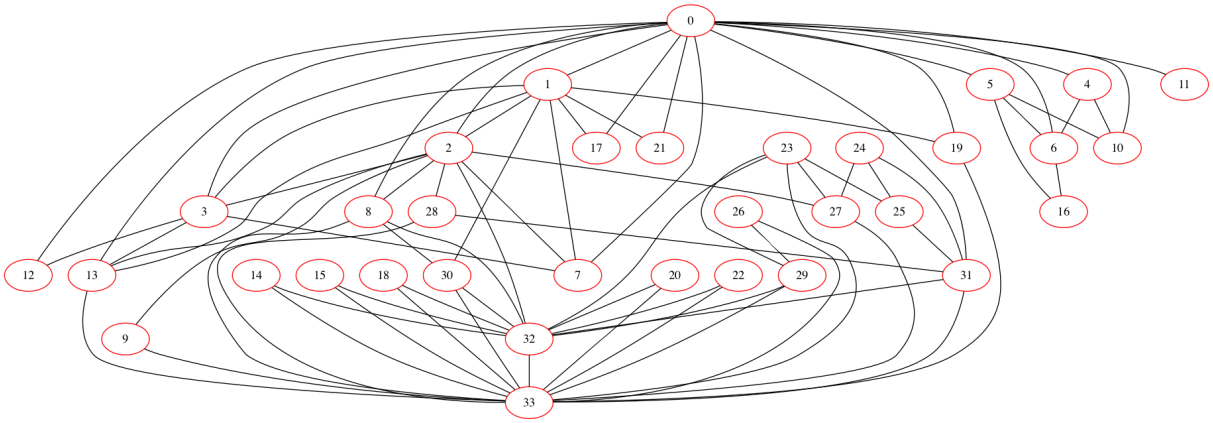

Figura 3.5: Zachary's Karate Club Network.

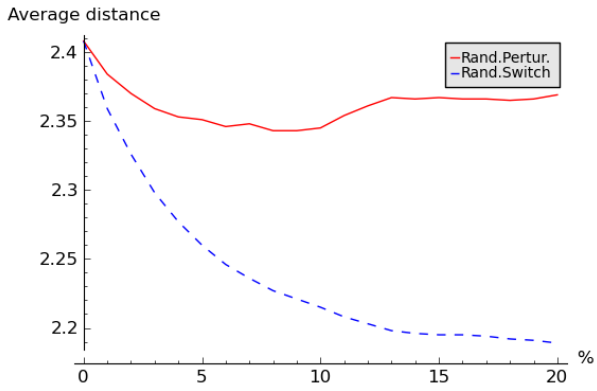

Figura 3.6: Distancia media.

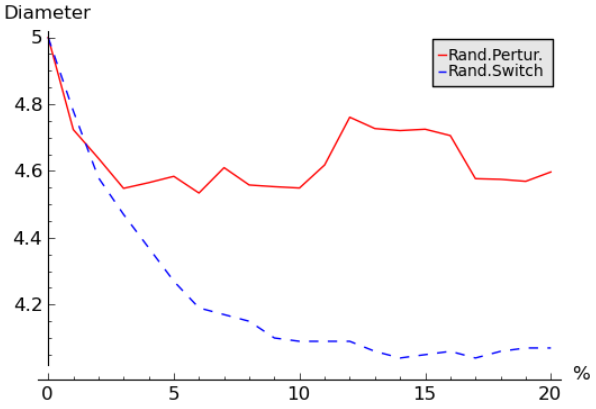

Figura 3.7: Diámetro.

(manteniéndose constante a 34 nodos), así como tampoco se producen modificaciones en el grado medio del grafo.

En lo referente a la **distancia media**, se puede observar en la Figura 3.6 que los grafos anonimizados con el método *Random Switch* padecen un sensible descenso de este valor conforme aumenta el porcentaje de anonimización, llegando a tasas de un 6 % cuando el porcentaje de anonimización supera el 15 %. Las causas que pueden motivar este comportamiento pueden ser: (1) Un aumento en el número de aristas del grafo. No es el caso, dado que el número de aristas no sufre variación alguna. (2) Un aumento en la cohesión entre los nodos del grafo. Si se aumenta la centralidad de los nodos principales del grafo, se reduce la distancia media entre los nodos.

Por otro lado, dado que el **diámetro** es una medida directamente relacionada con la distancia media, en la Figura 3.7 se puede ver una evolución similar a la evolución de la distancia media en ambos métodos.

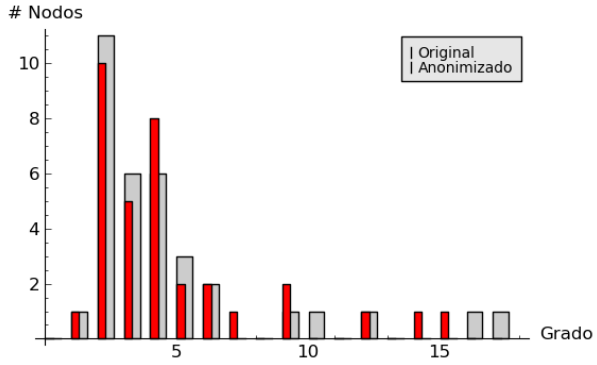

Figura 3.8: Histograma de grados del grafo original y del grafo anonimizado con *Random Perturbation* del 3 %.

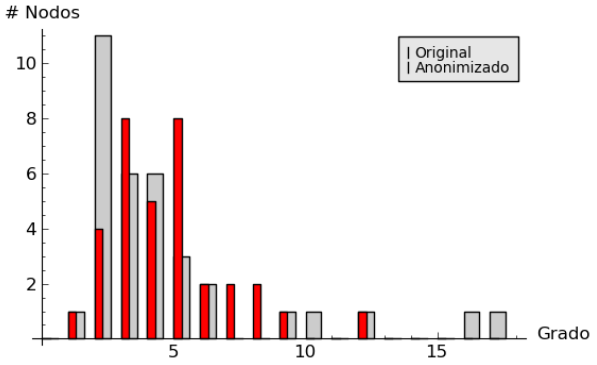

Figura 3.9: Histograma de grados del grafo original y del grafo anonimizado con *Random Perturbation* del 20 %.

La Figura 3.8 muestra en color gris el histograma de grados del grafo original. Se puede ver claramente como el grafo muestra la característica de la ley de la potencia (*power-law*) en el histograma de grados. Las Figuras 3.8 y 3.9 muestran como se deteriora el histograma de grados en *Random Perturbation* (en color gris el histograma del grafo original) cuando se pasa de un porcentaje de anonimización del 3 % al 20 %. Este modelo no preserva el grado de los nodos, ya que las aristas son eliminadas e insertadas de forma aleatoria. Por el contrario, el modelo *Random Switch* intercambia aristas entre nodos, de forma que mantiene el grado de todos los nodos. Por lo tanto, el histograma no sufre variación alguna en este proceso de anonimización.

Las Figuras 3.10 y 3.11 muestran la evolución de la medida de ***betweenness centrality***. Las figuras muestran la variación de estos valores en un 3 % y 20 % de anonimización. En la Figura 3.11 se puede ver como los grafos anonimizados mediante *Random Perturbation* tienden a suavizar este parámetro, mientras que las anonimizaciones mediante *Random Swicth* consiguen mantener mejor la forma que caracteriza el grafo original. Un nodo con un valor alto indica que este nodo forma parte de muchos caminos cortos del grafo, con lo cual será un nodo clave en la estructura del grafo. Por lo tanto, la pérdida de centralidad en los nodos con valores mayores es significativa cuando el porcentaje de anonimización aumenta, indicando que se están modificando caminos importantes para la conectividad del grafo.

Las Figuras 3.12 y 3.13 muestran la evolución de la media de **closeness centrality**. Se puede apreciar un aumento considerable de las perturbaciones en este parámetro a medida que aumenta el porcentaje de anonimización. Aunque esta medida se ve perturbada de forma similar por ambos procesos de anonimización.

Las Figuras 3.14 y 3.15 muestran la última evaluación referente a la centralidad, llamada **degree centrality**. En ellas se puede ver como el método *Random Perturbation* provoca

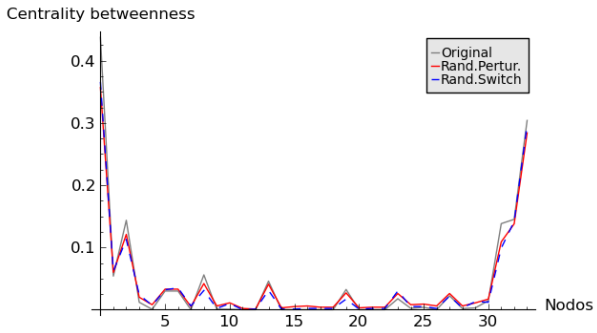

Figura 3.10: *Betweenness centrality* del grafo anonimizado del 3 %.

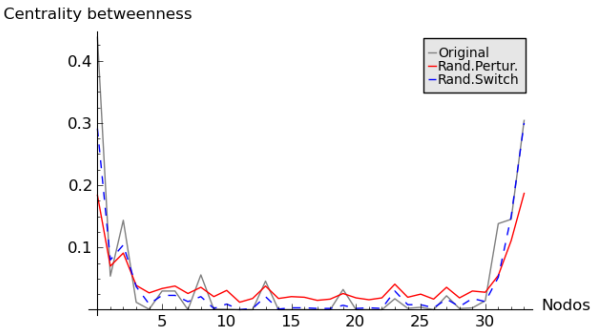

Figura 3.11: *Betweenness centrality* del grafo anonimizado del 20 %.

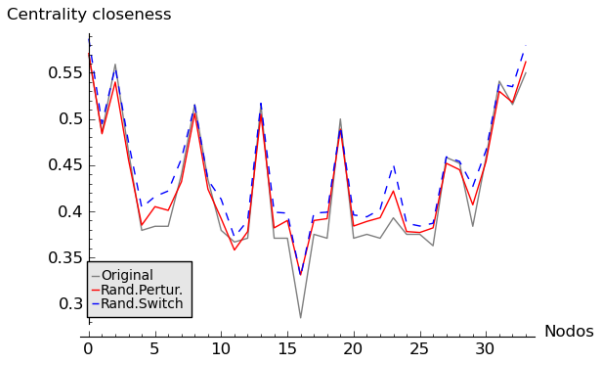

Figura 3.12: *Closeness centrality* del grafo anonimizado del 3 %.

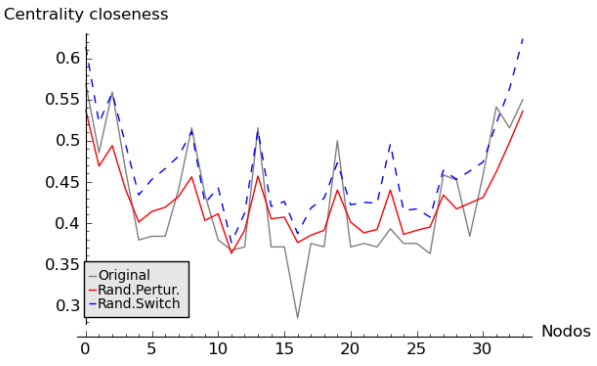

Figura 3.13: *Closeness centrality* del grafo anonimizado del 20 %.

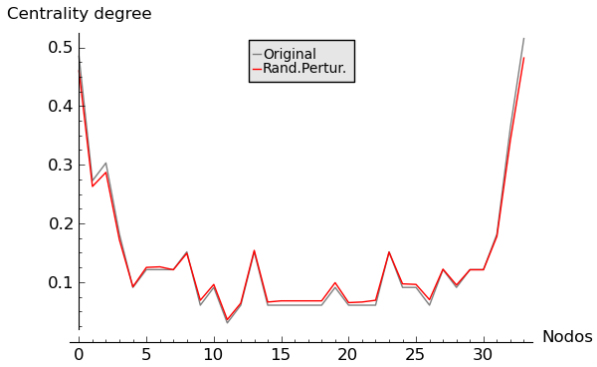

Figura 3.14: *Degree centrality* del grafo anonimizado del 3 %.

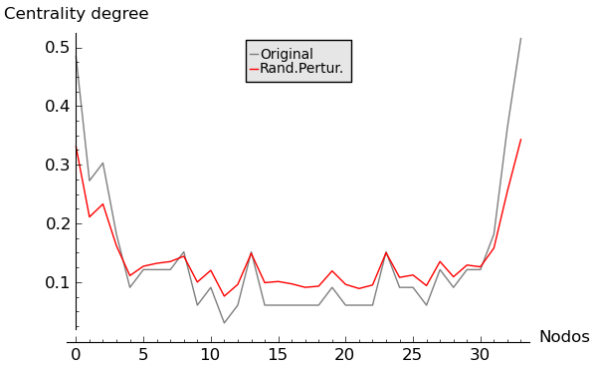

Figura 3.15: *Degree centrality* del grafo anonimizado del 20 %.

una perturbación en todos los nodos del grafo, disminuyendo los nodos de mayor grado y aumentando los nodos de menor grado. Se puede ver la evolución del efecto comparando ambas imágenes. Como ya se ha mencionado, el método *Random Switch* no tiene ningún efecto sobre este parámetro, ya que no modifica el grado de los nodos.

3.4.2. Resultados del proceso de *graph mining*

A continuación se discuten los resultados obtenidos tras el proceso de *graph mining*. En el proceso de *graph mining* se aplican dos algoritmos de *clustering* (agrupamiento) distintos, con el objetivo de poder comparar los resultados obtenidos en dos algoritmos distintos. Estos son: (1) algoritmo MCL (*Markov Cluster Algorithm*) y (2) RRW (*Repeated Random Walk*).

Ambos algoritmos realizan *clustering* de nodos. Por lo tanto, el resultado en todos los casos serán conjuntos de nodos. El algoritmo MCL clasificará todos los nodos (34 en este caso) en dos o más conjuntos. El algoritmo RRW clasificará sólo algunos nodos en uno o más conjuntos.

Como ya se ha comentado, se ha implementado el índice de Jaccard como medida para evaluar la similitud entre los resultados obtenidos. Este índice se presenta en el rango $[0, 1]$, donde un valor igual a 0 indica que no hay ninguna coincidencia entre los dos conjuntos y un valor igual a 1 indica que la coincidencia de ítems es total.

Las Figuras 3.16 y 3.17 muestran los índices de Jaccard para los algoritmos MCL y RRW, respectivamente. En el eje horizontal (eje de las x) se muestra el porcentaje de anonimización de los datos, que varía entre 0 % (los datos originales, sin alteración alguna) y 20 % (el porcentaje máximo de anonimización aplicado en este estudio). En el eje vertical (eje de las y) se muestra el valor del índice de Jaccard. Cada figura muestra en color rojo los valores para el conjunto de datos anonimizado con el método *Random Perturbation* y en color azul los valores para el conjunto de datos anonimizado con el método *Random Switch*. En un caso ideal, las gráficas

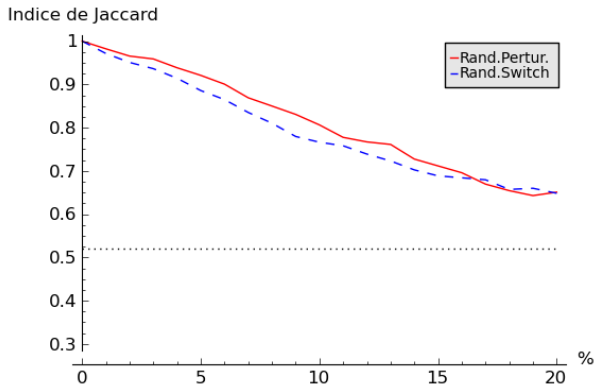

Figura 3.16: Índice de Jaccard en MCL.

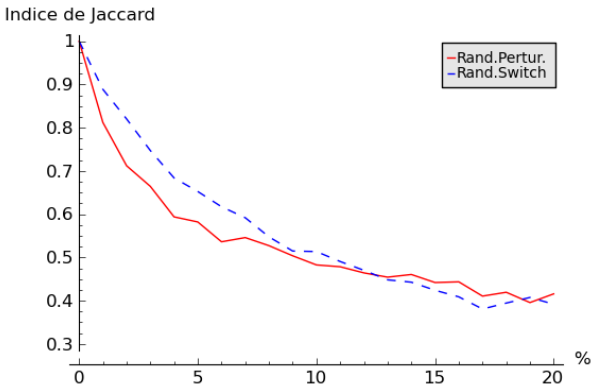

Figura 3.17: Índice de Jaccard en RRW.

deberían mostrar dos líneas horizontales con un valor constante igual a 1. Esto indicaría que el grado de anonimización no altera el resultado del proceso de *graph mining*. Lógicamente, este caso no es posible, ya que ambos métodos alteran los datos originales, degradando el resultado de cualquier proceso de *graph mining*.

El algoritmo MCL, ejecutado sobre los datos originales, forma dos grupos distintos de 18 y 16 nodos respectivamente (todos los nodos son clasificados en uno u otro grupo). La Figura 3.16 muestra como el índice de Jaccard desciende de forma progresiva y casi lineal con el aumento del porcentaje de anonimización. Ambos métodos de anonimización presentan resultados similares, aunque el método *Random Perturbation* mantiene valores algo superiores hasta el 17 % de anonimización. A partir del 25-30 % de anonimización, los valores se estabilizan entorno a un valor de 0,5. Los datos conseguidos mediante el método *Random Switch* consiguen valores superiores en estos rangos, quedando estabilizados en valores próximos a 0,55. En cambio los datos obtenidos mediante el método *Random Perturbation* se estabilizan en valores sensiblemente inferiores, próximos a 0,45.

Es relevante notar que si se realiza una clasificación en donde todos los nodos son colocados con el grupo mayoritario, se obtiene un índice de Jaccard de 0,52 (linea discontinua de la figura 3.16). Por lo tanto, valores cercanos a este límite son totalmente inaceptables y corresponden a degradaciones muy elevadas de los datos. Un grafo generado de forma aleatoria con el mismo número de nodos podría fácilmente obtener estos valores en el índice de Jaccard, lo que indica que la utilidad de los datos anonimizados será prácticamente nula.

El algoritmo RRW, ejecutado sobre los datos originales, forma dos grupos de 5 nodos cada uno. Es decir, sólo un total de 10 nodos (29 %) son clasificados en una de las dos particiones. Los demás nodos no son clasificados. La Figura 3.17 muestra como el índice de Jaccard desciende con mayor intensidad en los primeros ciclos del proceso de anonimización. Esto indica que el algoritmo RRW es menos resistente al ruido introducido por los métodos de anonimización. La

causa principal es que el conjunto de resultados (que es sobre los cuales se realiza la evaluación) es menor que en el caso del algoritmo anterior.

Al contrario que en el caso anterior, con el algoritmo RRW los resultados obtenidos con el algoritmo *Random Switch* son sensiblemente mejores en la primera parte del proceso de anonimización, hasta alcanzar valores valores próximos al 12 % de anonimización. A partir del 25-30 % de anonimización, los valores se estabilizan entorno a un valor de 0,35 en el índice de Jaccard. Los datos conseguidos mediante el método *Random Switch* consiguen valores algo superiores en estos rangos, aunque la variación en los resultados es notable en ambos casos.

Para todas las pruebas realizadas con este conjunto de datos, se ha establecido el parámetro *inflation*, que controla la granularidad del algoritmo MCL, a un valor igual a 1,8. Los parámetros que controlan el número mínimo y máximo de nodos en cada *cluster*, para el algoritmo RRW, se han establecido a unos valores de 5 y 20, respectivamente.

3.4.3. Riesgo de re-identificación

La secuencia del histograma de grados del grafo original es: [0, 1, 11, 6, 6, 3, 2, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1]. La Figura 3.18 muestra una gráfica generada a partir de estos datos.

Como se puede ver, existe un sólo nodo con grado 1, 9, 10, 12, 16 y 17. Por lo tanto, el valor de k-anonimidad basada en el grado se establece a k=1 para este grafo. Es decir, existe uno o más (en este caso 6) nodos que se pueden identificar de forma única a partir de información sobre su grado.

Como ya se ha comentado en este informe, el método *Random Switch* no modifica el grado de los nodos, por lo tanto, la secuencia y el histograma de grados es el mismo que en el grafo original. Y en consecuencia también se establece un valor de k=1 para la k-anonimidad en todos los grafos generados a partir de este método.

El método *Random Perturbation*, en cambio, si que modifica el grado de los nodos y produce variaciones en la secuencia de grados que pueden afectar al histograma de grados.

Se han generado 2.000 grafos anonimizados con este método (100 ejecuciones independientes con anonimizaciones entre el 1 % y el 20 %), de los cuales han conseguido un valor de k superior a 1 sólo:

- k=2: 11 grafos (0,55 %)
- k=3: 1 grafo (0,05 %)

La Figura 3.19 muestra la distribución de los casos en los que aumenta el valor de k-anonimidad según el porcentaje de anonimización que se aplica. Se puede apreciar que durante los 6 primeros pasos de anonimización (en el intervalo 1 %-6 %) no se ha producido ningún caso.

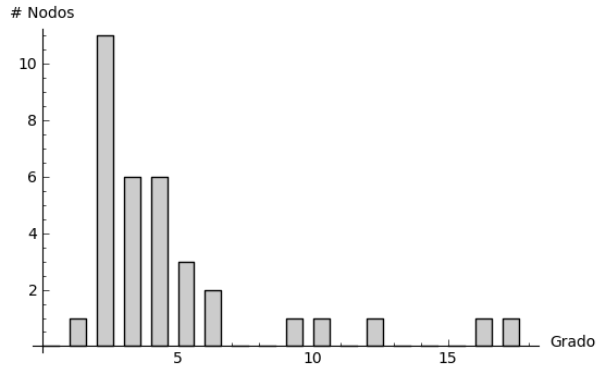

Figura 3.18: Histograma de grados del grafo original.

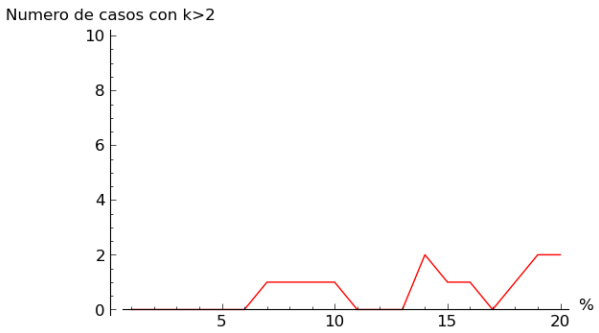

Figura 3.19: Número de casos en los que se consigue un aumento del valor de k-anonimidad superior a 1.

Como se ha comentado anteriormente, el grafo original presenta 6 nodos con grado único. Serán necesarias múltiples modificaciones de aristas para conseguir que todos estos nodos dejen de tener grado único, y que en consecuencia el grado de k-anonimidad del grafo pueda aumentar. Por lo tanto, no es posible que en los primeros pasos del proceso de anonimización se puedan conseguir grafos con un valor de k-anonimidad superior a 1.

Aún así, el número de grafos que consiguen aumentar su valor de k-anonimidad es muy pequeño, llegando a un máximo del 2 % de los grafos anonimizados en tres ocasiones (concretamente en los puntos de 14 %, 19 % y 20 % de anonimización).

Por ejemplo, uno de los grafos que han conseguido un valor de k-anonimidad igual a 2 presenta la siguiente secuencia e histograma de grados:

Secuencia: [14, 14, 9, 9, 6, 6, 5, 5, 5, 5, 5, 5, 5, 5, 4, 4, 4, 4, 4, 4, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 2, 1, 1, 1].

Histograma: [0, 3, 2, 9, 6, 8, 2, 0, 0, 2, 0, 0, 0, 0, 2].

En el histograma se puede apreciar que los grupos con un menor número de nodos son los correspondientes a los grados 2, 6, 9 y 14. Todos ellos contienen un número mínimo de 2 nodos, por lo tanto, el valor de k-anonimidad se establece a 2 y la probabilidad de re-identificación a $\frac{1}{2}$.

3.4.4. Conclusiones

En las evaluaciones realizadas, el método *Random Switch* presenta una menor distorsión de las propiedades estructurales del grafo. El motivo principal, ya comentado, es que el método en cuestión no modifica el grado de los nodos. De esta forma, las medidas basadas en el grado (gradio medio, histograma de grados y *degree centrality*) no sufren modificación alguna duran-

te el proceso de anonimización. En las demás, presenta mejores resultados en la medida de *betweenness centrality*, mientras que provoca una perturbación superior en la distancia media, el diámetro y el *closeness centrality*. A priori puede parecer que esta sensible mejoría en las propiedades estructurales se debe trasladar a los resultados obtenidos por el proceso de *graph mining*. Pero como se ha visto, no es así.

Los resultados observados en el algoritmo MCL muestran datos muy similares entre los grafos anonimizados mediante *Random Switch* y *Random Perturbation*. En este caso, el descenso de la calidad de los datos según el porcentaje de anonimización es casi lineal. Esto permite poder fijar un valor mínimo para el coeficiente de Jaccard y a partir de este valor poder definir el valor máximo de anonimización que se puede aplicar a los datos. De esta forma, se pueden obtener datos con el valor máximo de anonimización y que maximicen la utilidad de los datos.

Los datos observados en el algoritmo RRW se comportan de forma similar, aunque se puede destacar: (1) los resultados del método *Random Switch* son sensiblemente mejores en el rango entre el 1 % y el 10 % de anonimización y (2) el descenso no es lineal respecto al porcentaje de anonimización aplicado; en los primeros ciclos se produce una pérdida de utilidad mayor.

Para poder preservar la utilidad de los datos dentro de un rango aceptable, el grado de anonimización que se puede aplicar debe ser inferior al 10 %. De esta forma se consigue un índice de Jaccard con valores que no desciendan de 0,8 en el algoritmo MCL y 0,5 en el algoritmo RRW. Sin embargo, con estos valores de anonimización es difícil poder obtener grafos con valores de k-anonimidad superiores a 1. En este caso, el proceso de anonimización puede quedar en entredicho, ya que la re-identificación de algunos nodos puede ser perfectamente factible.

3.5. Conjunto *American College Football*

American College Football [17] es un grafo que contiene los equipos universitarios de la división IA que jugaron durante la temporada regular de 2010. No se incluye una figura con una posible disposición de la red, dado que la visualización, en tan reducido espacio, es imposible con la cantidad de nodos y aristas que presenta la red.

Sus características básicas son:

- 115 nodos y 613 aristas.
- No dirigido y sin etiquetas en las aristas.
- El grado medio es de 10,661.
- La distancia media es de 2,508.
- El diámetro es de 4.

- El histograma de grados es: [0, 0, 0, 0, 0, 0, 0, 1, 3, 5, 28, 66, 12].

El rango de anonimización utilizado para este experimento está comprendido entre los valores 0 % y 50 %. Para cada valor del porcentaje de anonimización se muestran los resultados promedios de 50 ejecuciones independientes.

A continuación se realiza la evaluación del grafo original y las anonimizaciones obtenidas mediante los algoritmos *Random Perturbation* y *Random Switch*.

3.5.1. Propiedades estructurales

Tal y como se ha comentado, los métodos que se evalúan en este capítulo se basan en la modificación aleatoria de aristas, es decir, eliminar y añadir aristas, manteniendo siempre un número constante de aristas (613 en este caso). Por lo tanto, el número de nodos no se modifica (manteniéndose constante a 115 nodos), así como tampoco se producen modificaciones en el grado medio del grafo.

En la Figura 3.20 se puede ver como evoluciona el valor de la **distancia media** en función del porcentaje de anonimización aplicado. Ambos métodos producen unos resultados prácticamente iguales. El valor de la distancia media experimenta un descenso exponencial, especialmente en el primer 20 % de anonimización. El valor tiende a estabilizarse a partir del 30-40 % de anonimización, después de un descenso de cerca del 10 % de su valor original. Este parámetro indica que ha aumentado la cohesión entre los nodos del grafos, produciendo una reducción de la distancia entre los nodos del grafo.

Aunque el **diámetro** es una medida relacionada con la distancia media, la Figura 3.21 no muestra, en este caso, unos datos similares. Mientras que los datos anonimizados mediante *Random Perturbation* mantienen, prácticamente, el mismo diámetro durante todo el rango de anonimización, los datos anonimizados mediante *Random Switch* muestran un descenso importante a partir de un porcentaje de anonimización del 20 %. Cuando el porcentaje de anonimización se acerca al 50 % la pérdida en el diámetro es del 15 %.

La Figura 3.22 muestra en color gris el histograma de grados del grafo original. Es importante notar que muestra la característica de la ley de la potencia (*power law*) de forma invertida. Es decir, la mayor concentración de nodos se produce en la parte alta del histograma de grados. Esto indica que existen muchos nodos en el grafo con un grado elevado. Las Figuras 3.22 y 3.23 muestran el deterioro sufrido por el **histograma de grados** (en color gris se muestra el histograma del grafo original) en *Random Perturbation* cuando se aplica una anonimización del 3 % y del 10 %, respectivamente. La Figura 3.22 muestra una importante variación en el grado de los nodos, especialmente en los nodos con grado 11: en el grafo original son 66 y con sólo un 3 % de anonimización pasan a ser sólo 38. Es decir, casi un 50 % de los nodos del grupo

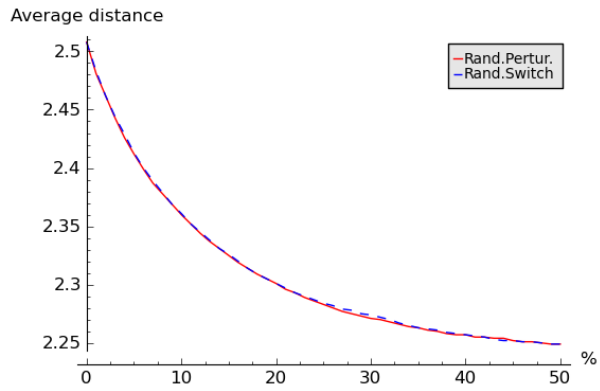

Figura 3.20: Distancia media.

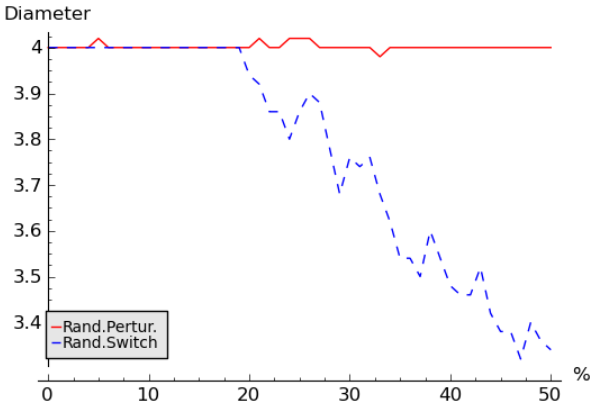

Figura 3.21: Diámetro.

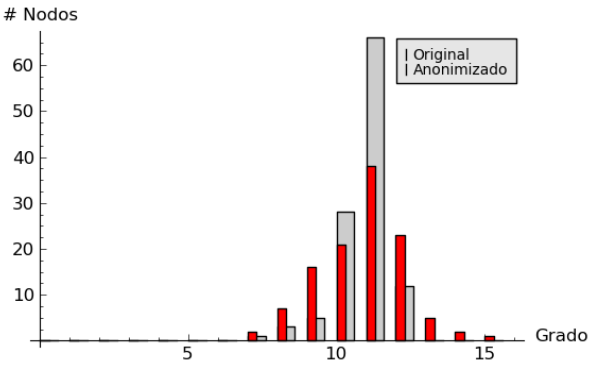

Figura 3.22: Histograma de grados del grafo original y del grafo anonimizado con *Random Perturbation* del 3 %.

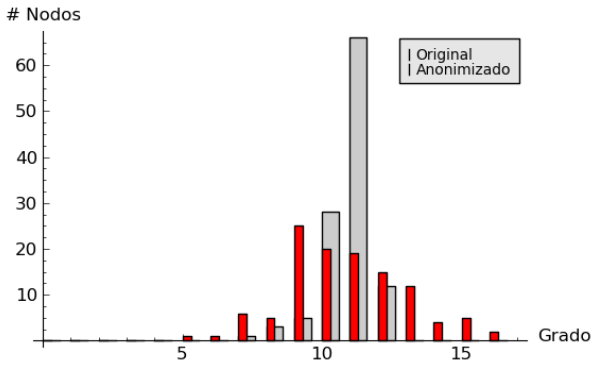

Figura 3.23: Histograma de grados del grafo original y del grafo anonimizado con *Random Perturbation* del 10 %.

mayoritario cambian su grado. De forma global, se observa como el histograma adopta una forma similar a una gaussiana, en lugar de la forma característica de la ley de la potencia. En la Figura 3.23 se acentúa aún más la distorsión sobre el histograma original. El grado mayoritario en el grafo anonimizado pasa el ser el grado 9.

Las Figuras 3.24 y 3.25 muestran la evolución de la medida de **betweenness centrality**. Las figuras muestran la variación de estos valores en un 3 % y 10 % de anonimización. En la primera figura se puede ver que ambos métodos mantienen de una forma aceptable el valor del grafo original, aunque cabe destacar que el grafo original presenta una variabilidad muy notable en esta medida, provocando oscilaciones continuas en todo su rango de nodos. Un nodo con un valor alto indica que el nodo forma parte de muchos caminos cortos del grafo, y al revés para nodos con un valor bajo. En la figura se puede ver como los nodos con un valor alto pierden parte de su valor y los nodos con un valor más bajo aumentan ligeramente su valor. Es

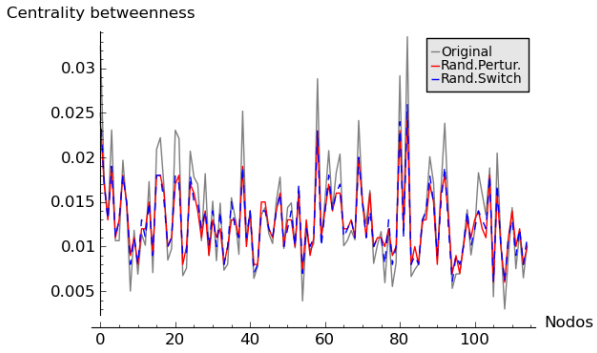

Figura 3.24: *Betweenness centrality* del grafo anonimizado del 3 %.

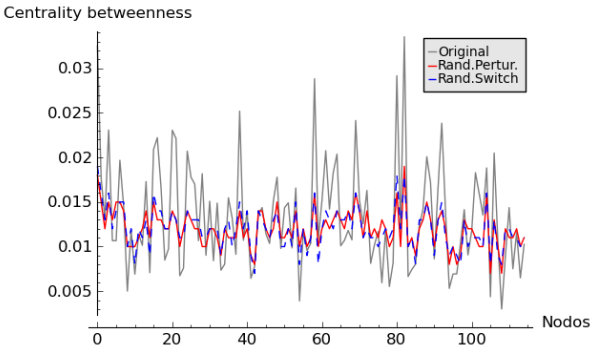

Figura 3.25: *Betweenness centrality* del grafo anonimizado del 10 %.

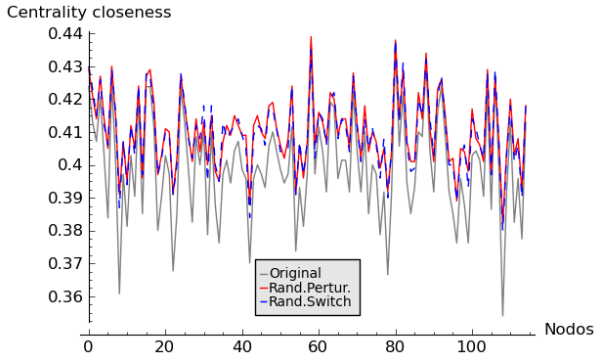

Figura 3.26: *Closeness centrality* del grafo anonimizado del 3 %.

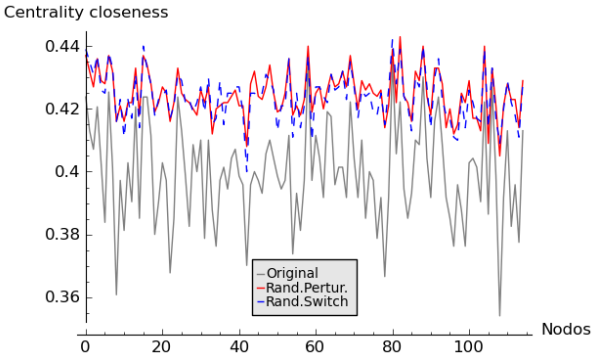

Figura 3.27: *Closeness centrality* del grafo anonimizado del 10 %.

decir, se tiende a estabilizar el valor de *betweenness centrality*, y por lo tanto, a equiparar la centralidad de los nodos del grafo. Esta tendencia se confirma cuando aumenta el porcentaje de anonimización, tal y como se puede observar en la segunda figura. El grado de distorsión es muy similar en ambos métodos de anonimización.

Las Figuras 3.26 y 3.27 muestran la evolución de la medida de **closeness centrality**. El comportamiento de esta medida es muy similar al comportamiento observado de *betweenness centrality*. Al igual que en el caso anterior, a partir del 10 % de anonimización esta medida se muestra muy deteriorada, mostrando un equilibrio de la centralidad entre los nodos del grafo.

Las Figuras 3.28 y 3.29 muestran la última evaluación referente a la centralidad, llamada **degree centrality**. El algoritmo *Random Switch* no muestra variación alguna, ya que no modifica el grado de los nodos. En cambio, el algoritmo *Random Perturbation* introduce cierta perturbación en el grado de los nodos, que aumenta conforme aumenta el porcentaje de anonimización.

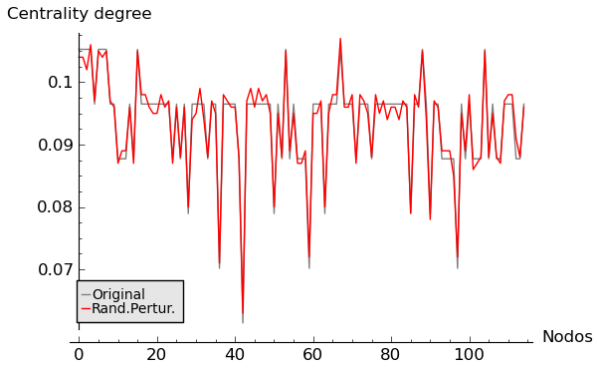

Figura 3.28: *Degree centrality* del grafo anonimizado del 3 %.

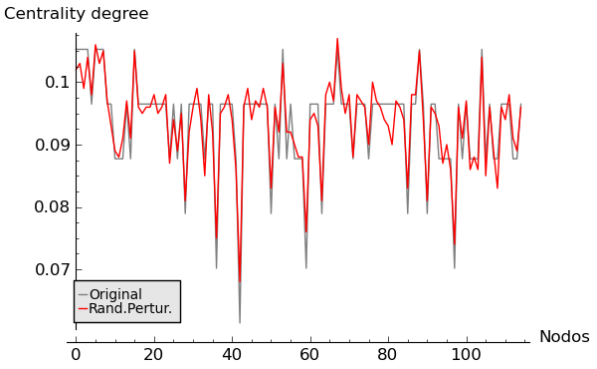

Figura 3.29: *Degree centrality* del grafo anonimizado del 10 %.

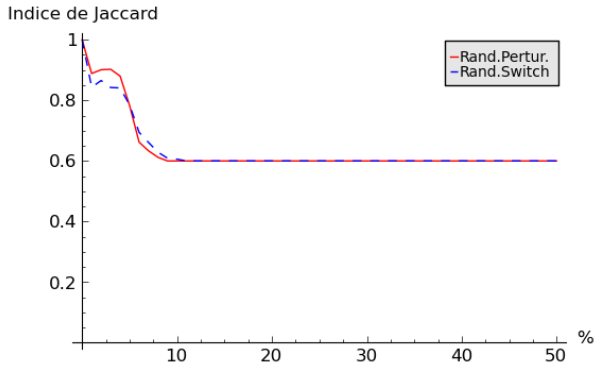

Figura 3.30: Índice de Jaccard en MCL con el parámetro de inflación I=1,4.

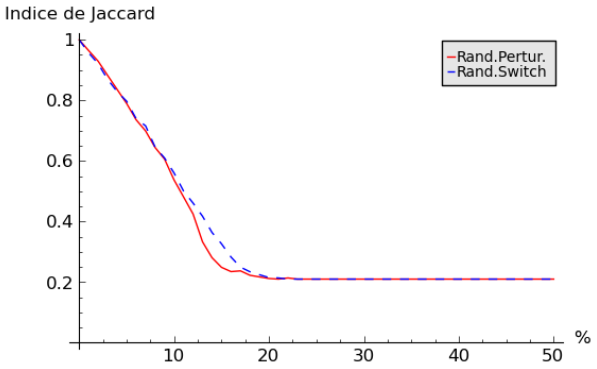

Figura 3.31: Índice de Jaccard en MCL con el parámetro de inflación I=1,5.

3.5.2. Resultados del proceso de *graph mining*

Los resultados de los índices de Jaccard para los algoritmos MCL y RRW se pueden ver en las Figuras 3.30, 3.31 y 3.32. Igual que en el caso anterior, el eje de las x muestra el porcentaje de anonimización de los datos, que varía entre 0 % (los datos originales, sin alteración alguna) y 50 % (el porcentaje máximo de anonimización aplicado en este estudio) y el eje de las y muestra el valor del índice de Jaccard. Cada figura muestra en color rojo los valores para el conjunto de datos anonimizado con el método *Random Perturbation* y en color azul los valores para el conjunto de datos anonimizado con el método *Random Switch*. Como ya se ha comentado, una situación óptima mostraría dos líneas horizontales con un valor constante igual a 1, indicando que el grado de anonimización no altera el resultado del proceso de *graph mining*. Esta situación no es posible, ya que ambos métodos alteran los datos originales y provocan una degradación en el resultado de posteriores procesos de *graph mining*.

El algoritmo MCL, ejecutado sobre los datos originales, forma dos conjuntos de 69 y 46 nodos (todos los nodos son clasificados en uno u otro grupo). La Figura 3.30 muestra cuatro etapas claramente diferenciadas: en los primeros ciclos de anonimización (hasta el 2 %) se experimenta un brusco descenso en el valor del índice de Jaccard. A continuación y hasta el 4 % de anonimización, el valor del índice se estabiliza en valores próximos a 0,9 para el caso del algoritmo *Random Perturbation* y 0,85 para el caso del algoritmo *Random Switch*. A partir del 4 % y hasta el 10 % de anonimización se vuelve a producir un descenso importante en el valor del índice. Y finalmente, se estabiliza a 0,6 a partir del 10 % de anonimización. Analizando los datos se puede ver que a partir de un 9 o 10 % de anonimización el algoritmo MCL clasifica todos los nodos en el mismo conjunto. Es decir, los resultados del *clustering* se agrupan en un único conjunto con 115 nodos, obteniendo una puntuación de 0,6 en todos los casos a partir de una anonimización del 10 %.

Se puede concluir que una anonimización del 9 o 10 % sobre los datos originales, en cualquiera de los dos algoritmos de anonimización, provoca que los datos no tengan utilidad para un proceso posterior de *clustering* empleando el algoritmo MCL. Es decir, las propiedades estructurales que utiliza MCL para calcular los *clusters* quedan totalmente distorsionadas con este grado de anonimización. En este experimento el punto óptimo se obtiene con una anonimización del 4 %, que obtiene un índice de Jaccard de entre 0,85 y 0,9.

En vista de los resultados obtenidos en este experimento, se ha realizado otro experimento similar modificando el parámetro del algoritmo MCL (*inflation*) que ayuda a controlar la granularidad de los conjuntos de resultados. Para este segundo experimento se modifica el valor de este parámetro a 1,5 (en el anterior experimento tenía un valor de 1,4) con el objetivo de ver si el comportamiento será similar en caso de haber más de dos conjuntos en el resultado del *clustering*. La Figura 3.31 muestra los resultados de este segundo experimento. La clasificación de los datos originales forma un total de 8 *clusters* con un repartimiento bastante equitativo de nodos: el *cluster* mayor tiene 24 nodos asociados y el menor 9. La figura muestra un descenso importante y constante en el valor del índice de Jaccard hasta un porcentaje de anonimización próximo al 16 %. A partir de este valor se produce, al igual que en el caso anterior, todos los nodos son asociados a un único *cluster*, que produce siempre el mismo valor en el índice. Por lo tanto, modificando la granularidad de los *clusters* de resultados no se consigue evitar este efecto indeseado, y que indica, sin duda, que la utilidad de los datos ha quedado totalmente corrompida.

El algoritmo RRW, ejecutado sobre los datos originales, forma 8 *clusters* con el siguiente número de nodos en cada uno: 11, 13, 13, 11, 12, 11 ,14 y 11. Es decir, un total de 96 nodos (85 %) son clasificados en una de las 8 particiones. Los demás nodos no son clasificados. La Figura 3.32 muestra un descenso en todos los ciclos de anonimización, obteniendo valores de

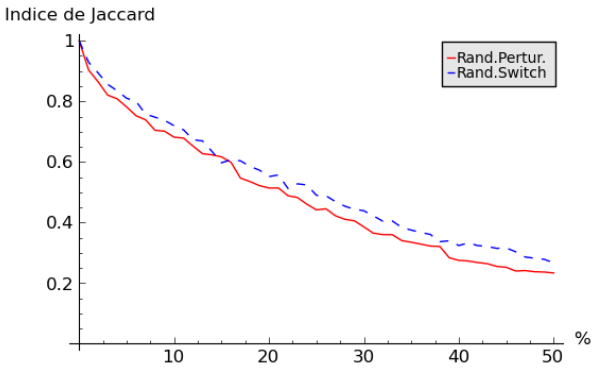

Figura 3.32: Índice de Jaccard en RRW.

0,8 con un 5 % de anonimización y de 0,7 cerca del 10 % de anonimización. Los resultados obtenidos con el algoritmo *Random Switch* son sensiblemente mejores durante todo el rango del proceso de anonimización. A diferencia del caso anterior, el algoritmo RRW demuestra más resistencia que el algoritmo MCL a las perturbaciones introducidas por los procesos de anonimización.

Los parámetros que controlan el número mínimo y máximo de nodos en cada *cluster*, para el algoritmo RRW, se han establecido a unos valores de 10 y 50, respectivamente.

3.5.3. Riesgo de re-identificación

La secuencia del histograma de grados del grafo original es: [0, 0, 0, 0, 0, 0, 0, 1, 3, 5, 28, 66, 12]. La Figura 3.33 muestra una gráfica generada a partir de estos datos. Como se puede ver, existe un sólo nodo con grado 7. Por lo tanto, el valor de k-anonimidad basada en el grado se establece a k=1 para este grafo. Es decir, existe un nodo que se pueden re-identificar de forma única a partir de información sobre su grado.

Como ya se ha comentado en este trabajo, el método *Random Switch* no modifica el grado de los nodos, por lo tanto, la secuencia y el histograma de grados es el mismo que en el grafo original. Y en consecuencia también se establece un valor de k=1 para la k-anonimidad en todos los grafos generados a partir de este método.

El método *Random Perturbation*, en cambio, si que modifica el grado de los nodos y produce variaciones en la secuencia de grados que pueden afectar al histograma de grados. Se han generado 2.500 grafos anonimizados con este método (50 ejecuciones independientes con anonimizaciones entre el 1 % y el 50 %), de los cuales han conseguido un valor de k superior a 1:

- k=2: 206 grafos (8,24 %)

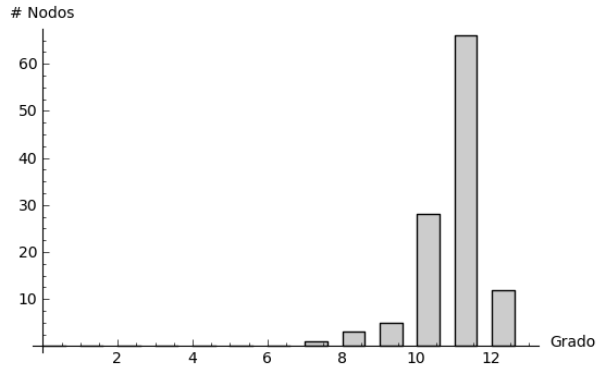

Figura 3.33: Histograma de grados del grafo original.

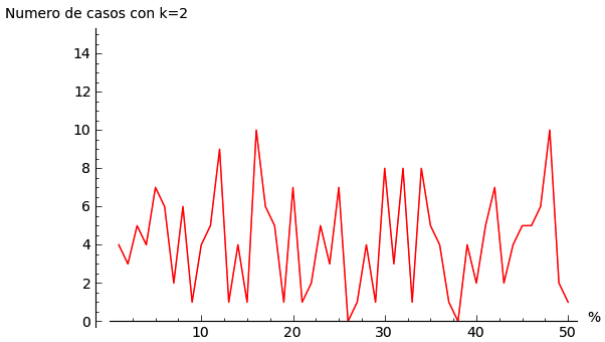

Figura 3.34: Número de casos en los que se consigue un valor de k-anonimidad igual a 2.

- k=3: 30 grafos (1,20 %)
- k=4: 6 grafos (0,24 %)
- k=5: 5 grafos (0,20 %)

La Figura 3.34 muestra la distribución de los casos en los que se consigue un valor de k-anonimidad igual a 2, según el porcentaje de anonimización que se aplica. Como se puede observar, esta medida presenta una gran oscilación. No se obtiene un número mayor de grafos con un valor de k-anonimidad igual a 2 si se aplica un porcentaje de anonimización mayor. Aplicando un porcentaje de anonimización de sólo un 1 % ya se obtienen 4 grafos con un valor de k-anonimidad igual a 2. Estos grafos son muy interesantes, ya que preservan gran parte de la utilidad de los datos y consiguen aumentar el valor de la k-anonimidad.

En este grafo es sencillo que cualquier método basado en la modificación aleatoria de aristas consiga aumentar el valor de k-anonimidad con un porcentaje de anonimización bajo. Es así porqué en los datos originales existen pocos grupos con muchos nodos cada uno, exceptuando dos grupos que contienen menos de 5 nodos.

3.5.4. Conclusiones

Las evaluaciones realizadas sobre este grafo, considerado de tamaño medio en este trabajo, muestran una gran perturbación en las características estructurales de los grafos anonimizados. El método *Random Perturbation* muestra una degradación muy importante del histograma de grados a partir de un porcentaje de anonimización muy bajo. Aunque el método *Random Switch* no modifica el histograma de grados, ambos métodos muestran una perturbación importante en las medidas de centralidad de los nodos.

Esto se traduce en los resultados del proceso de *graph mining*, especialmente con el algoritmo MCL. Según estas evaluaciones se produce un deterioro muy significativo en los datos anonimizados, ya desde los primeros ciclos de anonimización. Según los parámetros aplicados al algoritmo, se obtienen datos sin prácticamente utilidad a partir de porcentajes de anonimización superiores al 8 %.

En este caso el algoritmo RRW muestra mayor robustez a ambos procesos de anonimización, aunque cuando el porcentaje de anonimización supera el 10 % se obtienen coeficientes de Jaccard inferiores a 0,75. En este caso el método *Random Switch* presenta unos resultados ligeramente superiores durante prácticamente todo el rango de anonimización.

Dadas las características de este grafo, es sencillo obtener un valor de k-anonimidad igual a 2 aplicando un porcentaje de anonimización bajo (del orden del 2 %). Esto permite obtener grafos con una probabilidad de re-identificación inferior y con un grado perturbación aceptable. Si el nivel de seguridad deseado es bajo, los grafos generados mediante estas técnicas pueden ser válidos. En caso de desear un nivel de seguridad superior, y por lo tanto un valor de k-anonimidad superior, es necesario un porcentaje de anonimización mayor, lo cual provoca una reducción importante de la utilidad de los datos.

3.6. Conjunto *Jazz musicians*

Jazz musicians [35] es un grafo que contiene 198 nodos, que representan músicos de jazz, y 2.742 aristas que representan las relaciones entre ellos. No se incluye una figura con una posible disposición de la red, ya que la visualización en tan reducido espacio es imposible con la cantidad de nodos y aristas que presenta la red.

Sus características básicas son:

- 198 nodos y 2.742 aristas.
- No dirigido y sin etiquetas en las aristas.
- El grado medio es de 27,697.
- La distancia media es de 2,235.
- El diámetro es de 6.
- El histograma de grados es: [0, 5, 3, 3, 3, 4, 5, 3, 3, 3, 2, 1, 3, 6, 4, 1, 4, 5, 4, 7, 8, 2, 2, 9, 5, 5, 3, 5, 3, 5, 2, 9, 3, 4, 3, 1, 2, 3, 1, 6, 5, 4, 3, 3, 1, 3, 5, 0, 2, 2, 0, 1, 2, 2, 1, 2, 2, 1, 0, 2, 2, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1].

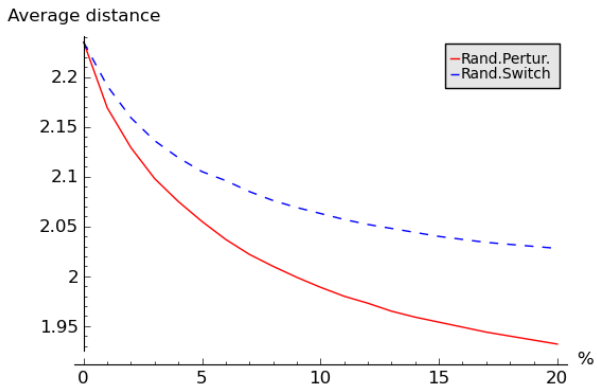

Figura 3.35: Distancia media.

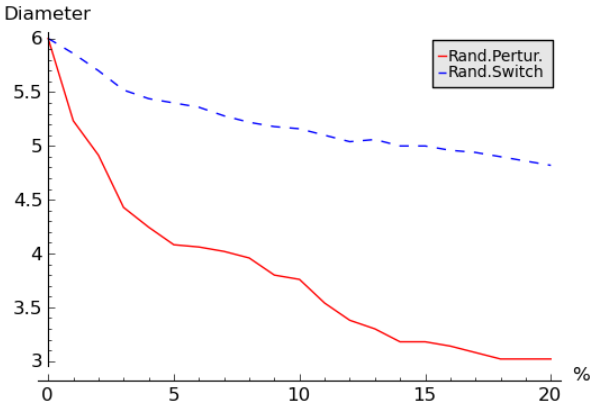

Figura 3.36: Diámetro.

El rango de anonimización utilizado para este experimento está comprendido entre los valores 0 % y 20 %. Para cada valor del porcentaje de anonimización se muestran los resultados promedios de 50 ejecuciones independientes.

A continuación se realiza la evaluación del grafo original y las anonimizaciones obtenidas mediante los algoritmos *Random Perturbation* y *Random Switch*.

3.6.1. Propiedades estructurales

Tal y como se ha comentado, los métodos que se evalúan en este capítulo no modifican el número de nodos ni el número de aristas, que se mantienen constantes a 198 y 2.742, respectivamente. En consecuencia, el grado medio tampoco sufre variación alguna en estos métodos de anonimización, manteniéndose constante a 27,697 durante todo el proceso.

En la Figura 3.35 se puede ver como evoluciona el valor de la **distancia media** en función del porcentaje de anonimización aplicado. Ambos métodos producen un cierto descenso en esta medida, aunque el método *Random Switch* produce un menor descenso, manteniéndose en unos valores más próximos al valor del grafo original.

Tal y como se ha comentado, el **diámetro** es una medida relacionada con la distancia media. En la Figura 3.36 se puede ver como la evolución del diámetro es similar a la evolución de la distancia media para ambos métodos. Aunque en este caso, la diferencia entre ambos métodos es mayor. El método *Random Perturbation* produce un descenso importante en el diámetro. Cuando el porcentaje de anonimización llega al 20 % el diámetro se reduce a la mitad del diámetro del grafo original.

Los resultados de la distancia media y el diámetro indican que al aumentar el porcentaje de anonimización, aumenta la cohesión entre los nodos del grafo, que se refleja en una reducción de la distancia media y del diámetro.

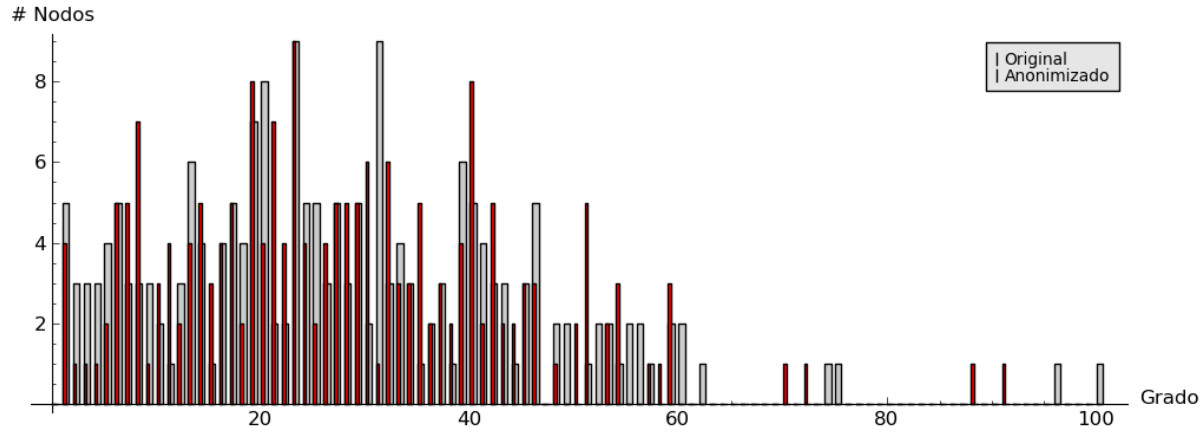

Figura 3.37: Histograma de grados del grafo original y del grafo anonimizado con *Random Perturbation* al 3 %.

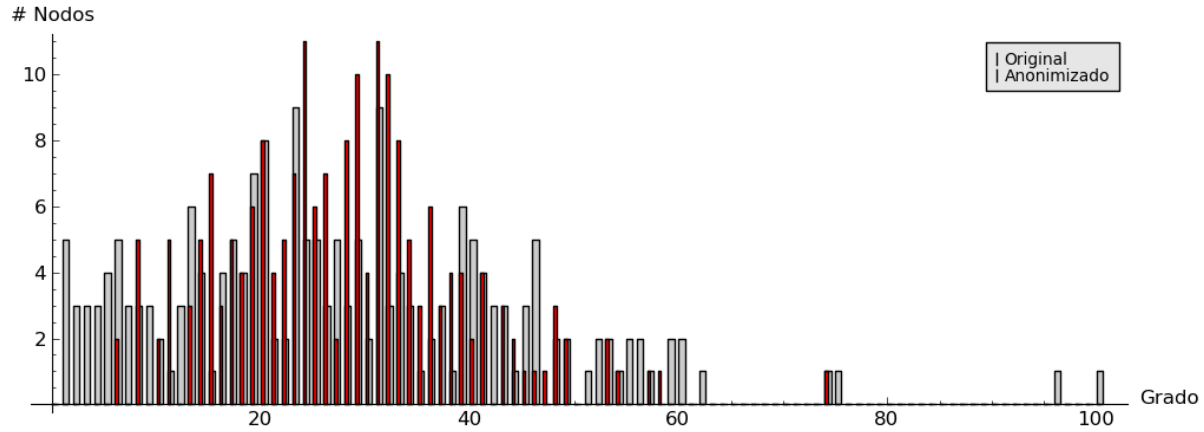

Figura 3.38: Histograma de grados del grafo original y del grafo anonimizado con *Random Perturbation* al 20 %.

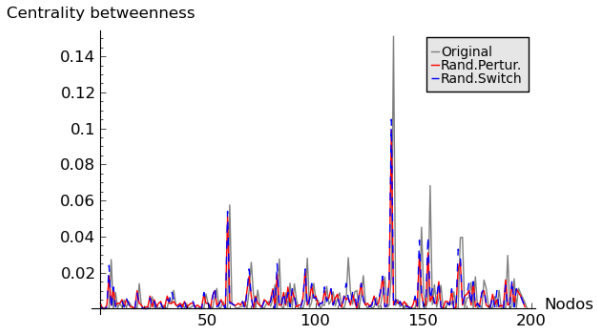

Figura 3.39: *Betweenness centrality* del grafo anonimizado del 3 %.

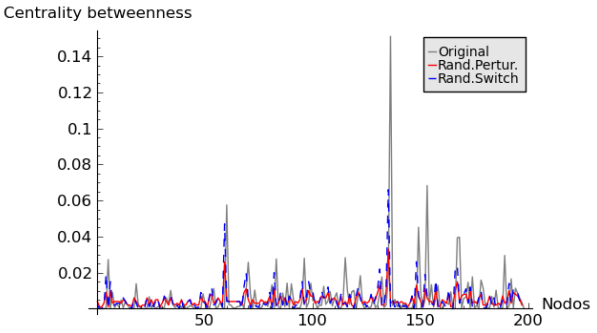

Figura 3.40: *Betweenness centrality* del grafo anonimizado del 20 %.

La Figura 3.37 muestra en color gris el **histograma de grados** del grafo original. Se puede percibir una cierta aproximación a la ley de la potencia (*power law*), pero no de una forma muy clara. Según el grado, la concentración más elevada de nodos en el grafo original se produce en valores en el rango entre 20 y 40. Las Figuras 3.37 y 3.38 muestran el deterioro sufrido por el histograma en *Random Perturbation* cuando se aplica una anonimización del 3 % y del 20 %, respectivamente. En la Figura 3.38 se puede ver como ha aumentado la perturbación cuando el porcentaje de anonimización se sitúa en el 20 %, aumentando de forma considerable los nodos con un grado en el rango 20-40 y disminuyendo los nodos con grado inferior a 10 y superior a 50.

Las Figuras 3.39 y 3.40 muestran la evolución de la medida de **betweenness centrality** en un porcentaje de anonimización del 3 % y del 20 %. En la primera figura se puede ver que ambos métodos mantienen de forma correcta los valores del grafo original. Sólo los nodos con un valor muy superior a la media se ven perturbados de una forma considerable durante el proceso de anonimización. Esto indica que algunos nodos que forman parte de muchos caminos cortos en el grafo original, pierden centralidad durante el proceso de anonimización y dejan de estar presentes en algunos caminos cortos de los grafos anonimizados. Aún así, el grado de perturbación introducido es aceptable.

La medida de **closeness centrality** se puede ver en las Figuras 3.41 y 3.42. La Figura 3.41 muestra unos resultados correctos y similares para ambos métodos con un porcentaje del 3 % de anonimización. En la Figura 3.42 se puede ver como el aumento al 20 % del porcentaje de anonimización ha producido mayor perturbación en ambos métodos, especialmente para el método *Random Perturbation*.

Las Figuras 3.43 y 3.44 muestran la medida de **degree centrality** en un porcentaje de anonimización del 3 % y del 20 %, respectivamente. En ambas figuras sólo se muestran los resultados para el método *Random Perturbation*, ya que el método *Random Switch* no modifica

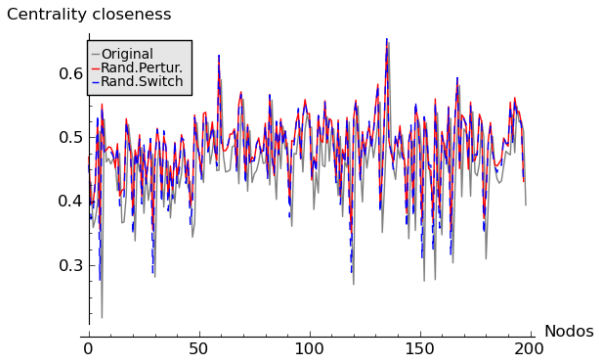

Figura 3.41: *Closeness centrality* del grafo anonimizado del 3 %.

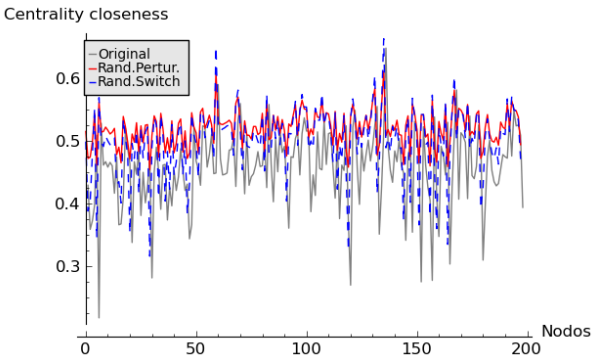

Figura 3.42: *Closeness centrality* del grafo anonimizado del 20 %.

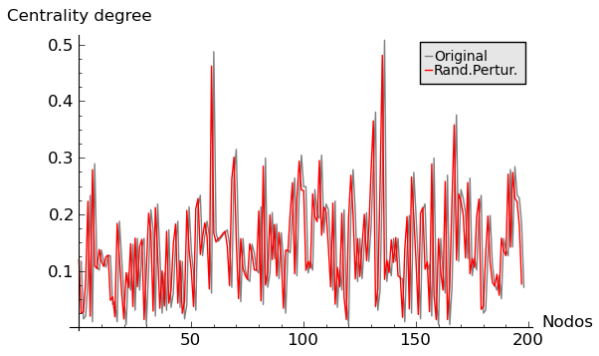

Figura 3.43: *Degree centrality* del grafo anonimizado del 3 %.

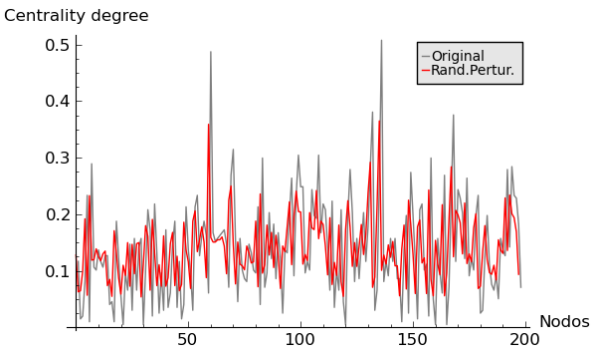

Figura 3.44: *Degree centrality* del grafo anonimizado del 20 %.

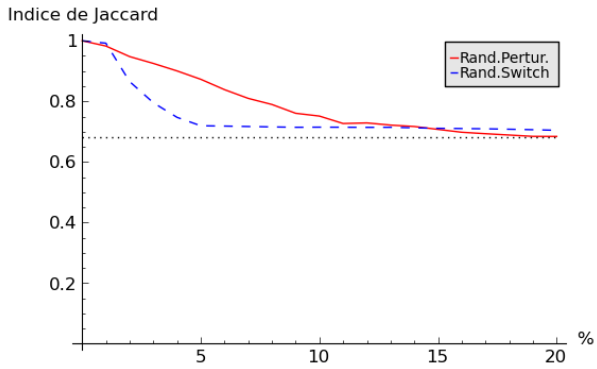

Figura 3.45: Índice de Jaccard en MCL.

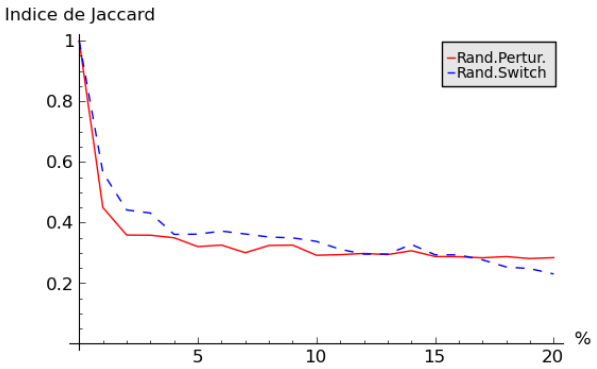

Figura 3.46: Índice de Jaccard en RRW.

el grado de los nodos, y por lo tanto, no muestra variación alguna en esta medida. Como se puede ver en ambas figuras, el grado de perturbación es aceptable, incluso cuando el porcentaje de anonimización llega al 20 %.

3.6.2. Resultados del proceso de *graph mining*

Las Figuras 3.45 y 3.46 muestran los índices de Jaccard para los algoritmos MCL y RRW. En el eje horizontal se muestra el porcentaje de anonimización de los datos, que varía entre 0 % (los datos originales, sin alteración alguna) y 20 % (el porcentaje máximo de anonimización aplicado en este estudio). En el eje vertical se muestra el valor del índice de Jaccard. Cada figura muestra en una linea continua de color rojo los valores para el conjunto de datos anonimizados con el método *Random Perturbation* y en una linea discontinua de color azul los valores para el conjunto de datos anonimizados con el método *Random Switch*.

La Figura 3.45 muestra los resultados del índice de Jaccard para el algoritmo MCL. Este algoritmo, ejecutado sobre los datos originales, forma un total de cinco *clusters* con 134, 56, 4, 2 y 2 nodos. Esta distribución irregular se produce independientemente de los parámetros aplicados al algoritmo MCL, y produce que una clasificación en un único conjunto con todos los nodos obtenga un índice de Jaccard de 0,67. Este valor se muestra en la figura como una linea punteada de color gris. Lógicamente, valores próximos a este valor no son admisibles e indican que el grado de distorsión introducido en los datos es demasiado elevado. Es decir, que los datos resultantes no son útiles para procesos de *clustering* con el algoritmo MCL. La figura muestra como el método *Random Switch* obtiene valores próximos a 0,67 a partir del 5 % de anonimización, mientras que el método *Random Perturbation* obtiene valores superiores hasta el 10 % de anonimización. A partir de este punto, ambos métodos han introducido demasiado distorsión y los datos han quedado inutilizados para esta tarea de *clustering*.

El algoritmo RRW, ejecutado sobre los datos originales, forma cinco *clusters*, cada uno de los cuales contiene entre 11 y 18 nodos. En total, se clasifican 60 nodos (30,3 %). Los demás nodos no son clasificados. La Figura 3.46 muestra un descenso muy importante en los primeros ciclos de anonimización, obteniendo valores de 0,4 en el índice de Jaccard con tan sólo un 1 % o 2 % de anonimización. En los siguientes ciclos de anonimización, el valor del índice de Jaccard desciende ligeramente hasta alcanzar valores de 0,3 cuando se aplica una anonimización del 20 %.

Ambos resultados indican que esta red presenta poca resistencia a la perturbación introducida por los métodos de anonimización basados en la modificación aleatoria de aristas.

Para todas las pruebas realizadas con este conjunto de datos, se ha establecido el parámetro *inflation*, que controla la granularidad del algoritmo MCL, a un valor igual a 1,8. Los parámetros que controlan el número mínimo y máximo de nodos en cada *cluster*, para el algoritmo RRW, se han establecido a unos valores de 10 y 100, respectivamente.

3.6.3. Riesgo de re-identificación

La secuencia del histograma de grados del grafo original es: [0, 5, 3, 3, 3, 4, 5, 3, 3, 3, 2, 1, 3, 6, 4, 1, 4, 5, 4, 7, 8, 2, 2, 9, 5, 5, 3, 5, 3, 5, 2, 9, 3, 4, 3, 1, 2, 3, 1, 6, 5, 4, 3, 3, 1, 3, 5, 0, 2, 2, 0, 1, 2, 2, 1, 2, 2, 1, 0, 2, 2, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1].

El valor de k-anonimidad basada en el grado se establece a k=1 para este grafo.

Tal y como se ha comentado, el método *Random Switch* no modifica el grado de los nodos, por lo tanto, la secuencia y el histograma de grados es el mismo que en el grafo original. Y en consecuencia también se establece un valor de k=1 para la k-anonimidad en todos los grafos generados a partir de este método.

El método *Random Perturbation* si que modifica el grado de los nodos, pero en ninguno de los 1.000 grafos anonimizados con este método (50 ejecuciones independientes con anonimizaciones entre el 1 % y el 20 %) se ha conseguido un valor de k superior a 1. El histograma de grados (ver Figura 3.47) muestra la existencia de cuatro nodos con grado único y valores 75, 76, 97 y 101. Para que el grafo pueda tener un valor de k-anonimidad superior a 1, todos estos nodos deben modificar su grado de tal forma que deje de ser único. Estos nodos presentan un valor de grado muy superior a los demás nodos del grafo (*outliers*), lo cual dificulta que las modificaciones aleatorias de aristas puedan producir que su grado deje de ser único. Existen otros nodos con grado único en el grafo, pero al no presentar valores de grado tan superiores a los demás es más probable que dejen de ser únicos mediante modificaciones aleatorias de aristas.

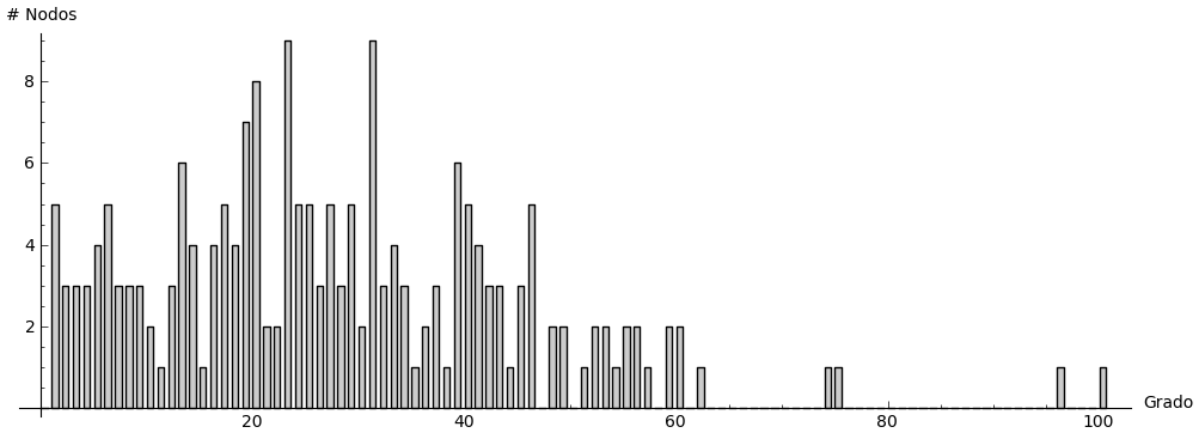

Figura 3.47: Histograma de grados del grafo original.

3.6.4. Conclusiones

Las evaluaciones realizadas sobre las propiedades estructurales de este grafo muestran como se produce una perturbación aceptable en la mayoría de las medidas analizadas. Ambos métodos consiguen resultados aceptables, especialmente el método *Random Switch*, que produce resultados sensiblemente mejores en la distancia media, diámetro y *closeness centrality*.

Aunque las medidas sobre las propiedades estructurales se han mantenido de una forma aceptable en todo el rango de anonimización, los resultados del proceso de *graph mining* no han sido satisfactorios. Con el algoritmo MCL se obtienen valores para el índice de Jaccard que demuestran la no utilidad de los datos para a partir de un 5 % (*Random Switch*) y de un 10 % (*Random Perturbation*) de anonimización. Es decir, aplicando un porcentaje de anonimización bajo, los grafos generados no son útiles para procesos de *clustering* posteriores. Con el algoritmo RRW los resultados son similares: con un porcentaje de anonimización de sólo el 2 % el índice de Jaccard baja hasta un valor de 0,4. Ambos resultados no son aceptables e indican que los datos quedan demasiado afectados por el proceso de anonimización, reduciendo de forma drástica su utilidad posterior para procesos de minería de datos.

En referencia al riesgo de re-identificación asociado a los grafos, es importante indicar que no se ha producido ninguna mejora en el valor de la k-anonimidad en ninguno de los grafos generados por los métodos de anonimización.

3.7. Conclusiones

Las evaluaciones de los grafos analizados indican, de forma general, que el método *Random Switch* mantiene los valores de las propiedades estructurales ligeramente más próximos a sus valores originales que el método *Random Perturbation*. Como se ha comentado, el método

Random Switch no modifica el grado de los nodos, lo cual le permite obtener valores iguales a los originales en las medidas basadas en el grado de los nodos.

Aún así, los resultados del proceso de *graph mining* han sido muy similares para ambos métodos. Es difícil poder identificar a un método como mejor en los procesos de *clustering* vistos. Es importante notar que la ligera superioridad demostrada por el método *Random Switch* en referencia a las propiedades estructurales no se ha trasladado en la evaluación de los procesos de *clustering*.

Las evaluaciones anteriores indican que el método de anonimización de grafos *Random Perturbation* permite obtener, en algunos casos, grafos con un valor de k-anonimidad ligeramente superior al valor del grafo original. La condición necesaria es poder generar grafos de forma iterativa hasta conseguir el grafo deseado. Cabe destacar que no es posible conseguir un grafo con el valor de k-anonimidad deseado, y sólo es posible conseguir un valor ligeramente superior. Los grafos anonimizados con el modelo *Random Switch* no experimentan ninguna modificación en su histograma de nodos, lo cual se traduce en que no se modifica su valor de k-anonimidad. Aunque es cierto que la perturbación añadida a los datos puede dificultar en cierta medida la re-identificación de los nodos, no se puede asegurar que no sea posible la re-identificación en un grafo anonimizado mediante esta técnica.

A partir de estas reflexiones generales, se puede afirmar que ambos métodos introducen un grado similar de perturbación durante el proceso de anonimización. También se puede afirmar que este grado es, en general, bastante elevado. Como punto a favor, el método *Random Perturbation* consigue algunos grafos con un nivel de k-anonimidad ligeramente superior al grafo original.

Capítulo 4

Evaluación de anonimización basada en k-anonimidad

En el capítulo anterior se ha analizado el comportamiento de los métodos *Random Perturbation* y *Random Switch*, que constituyen dos métodos básicos en la anonimización de grafos. Ambos pertenecen al grupo de métodos basados en la modificación aleatoria de aristas. Para analizar y evaluar los métodos se han empleado tres conjuntos de datos reales de distinto tamaño y con características significativamente distintas.

Los resultados demuestran que ambos métodos introducen una cantidad de ruido importante, que puede llegar a perturbar de forma significativa los datos anonimizados. Esta problemática es inherente al tipo de grafos con los que se ha trabajo: grafos simples sin pesos en las aristas ni atributos en los nodos. En estos grafos, la propia estructura es quien contiene toda la información de la red. Por lo tanto, la perturbación producida por el proceso de anonimización afecta directamente a los datos utilizados por los procesos de *graph mining* posteriores.

Los resultados de los métodos de anonimización han sido más o menos buenos, dependiendo de las características concretas de cada grafo y del porcentaje de anonimización aplicado. Pero ha habido un factor constante en todos los experimentos: las probabilidades de re-identificación, asociadas al valor de k-anonimidad basada en el grado, no se pueden escoger a priori. Es decir, en dos de los tres experimentos se han producido algunos grafos con un valor de k-anonimidad superior al valor del grafo original, pero quizás no sea un valor suficiente o simplemente se desea poder fijar un valor de k-anonimidad concreto en función de los datos y de la utilidad posterior que se les quiera dar.

Es importante que se pueda controlar el grado de seguridad en los datos anonimizados. En este sentido, los métodos de anonimización basados en la modificación de aristas para preservar el modelo de k-anonimidad pueden ofrecer una alternativa mejor. En estos métodos se obtiene un grafo que cumple con un valor de k-anonimidad fijado a priori.

En este capítulo se implementa un método de anonimización basado en la modificación de aristas para preservar el modelo de k-anonimidad. El algoritmo implementado es una variación del algoritmo *Relaxed Graph Construction* (RGC), en el que se ha introducido el concepto de algoritmos genéticos para resolver una de las tareas, que los autores originales resuelven mediante técnicas de programación lineal.

4.1. Algoritmo *Relaxed Graph Construction*

El algoritmo que se implementa es una variación del algoritmo *Relaxed Graph Construction* (RGC) desarrollado por Liu y Terzi en [27] y pertenece al grupo de métodos de anonimización basados en la modificación de aristas para preservar el modelo de k-anonimidad. El objetivo de estos algoritmos es modificar las aristas de un grafo para conseguir un valor de k-anonimidad específico. Este algoritmo presenta dos fases:

1. A partir de la secuencia de grados (d) del grafo original ($G(V, E)$), aplicar las modificaciones necesarias para que la secuencia sea k-anónima ($\widehat{d}$).

2. Generar un grafo, $\widehat{G} = (V, \widehat{E})$, a partir de la secuencia de grados obtenida en el paso anterior ($\widehat{d}$), y reconstruir el grafo mediante intercambios de aristas válidos, de tal forma que se minimicen las aristas que no existían en el grafo original ($E \cap \widehat{E} \approx E$). Es decir, transformar el grafo obtenido a partir de la secuencia del paso anterior para que sea tan similar al grafo original como sea posible.

4.1.1. Obtención de la secuencia de grados k-anónima

La primera parte consiste en obtener, a partir de la secuencia de grados original, una secuencia k-anónima para un valor específico de k. Además, se debe realizar el número mínimo de cambios posibles en la secuencia original, ya que de esta forma se minimizan las modificaciones de aristas en el grafo. Liu y Terzi resuelven esta parte mediante técnicas de programación lineal. La propuesta de mejora que se propone en este trabajo es un nuevo algoritmo basado en algoritmos genéticos que permite obtener una secuencia de grados k-anónima.

El problema de obtener una secuencia de grados k-anónima tiene ciertas particularidades que se deben considerar:

- Número de elementos de la secuencia de grados: El número de elementos de la secuencia determina el número de nodos que formaran un grafo obtenido a partir de esta secuencia. El grafo generado debe tener el mismo número de nodos que el que grafo original, por lo tanto, este valor no se puede alterar.

- Valores de la secuencia de grados: Cada elemento de la secuencia de grados puede tener un valor entero en el rango $[0, n-1]$, donde n indica el número de nodos del grafo.

- Preservar el número de aristas del grafo: El número de aristas del grafo es la mitad del sumatorio de la secuencia, ya que cada arista se contabiliza dos veces en la secuencia de grados (por ejemplo, la arista (i, j) se contabiliza una vez para el nodo v_i y otra para el nodo v_j). Para preservar el número de aristas, el sumatorio de la secuencia obtenida debe ser igual al sumatorio de la secuencia original.

- Minimizar la distancia a la secuencia original: Los cambios realizados en la secuencia de grados se transforman en modificaciones de aristas en el grafo. Es necesario realizar el mínimo número de modificaciones en las aristas del grafo, para minimizar el ruido introducido en los datos. Por lo tanto, es necesario realizar el número mínimo de modificaciones en la secuencia de grados, o lo que es lo mismo, que la distancia entre ambas secuencias sea mínima.

El nuevo algoritmo recibe como parámetros la secuencia de grados del grafo original y el valor deseado de k-anonimidad. Como cualquier algoritmo genético, se compone de tres fases principales: En la primera fase, la población se inicializa a partir de la secuencia de grados del grafo original. La segunda fase del algoritmo constituye la fase principal, que se ejecuta repetidamente y donde los candidatos son modificados, evaluados y se selecciona los que pasan a la siguiente generación. En la tercera fase se selecciona el mejor candidato, que será identificado y devuelto como solución.

La generación de los nuevos candidatos, que permite a la población evolucionar, se basa únicamente en la mutación. La recombinación de parejas de padres, el otro método usado para la generación de nuevos candidatos en algoritmos genéticos, incumpliría de forma sistemática la regla que preserva el número de aristas del grafo, y por lo tanto, generaría candidatos no válidos. Una operación de mutación básica, dadas las particularidades de este problema, es la siguiente: sumar uno a un elemento de la secuencia de grados y restar uno a otro elemento. Esta operación representa la modificación de uno de los nodos de una arista. Por ejemplo, si a la arista $e_1 = (v_0, v_1)$ se le modifica un nodo, se puede obtener $e'_1 = (v_0, v_2)$. Este cambio se representa en la secuencia de grados como restar uno al nodo v_1 y sumar uno al nodo v_2.

Cuando la generación de candidatos ha terminado, se evalúa la bondad de los candidatos obtenidos. La función de *fitness* que realiza este cálculo se basa en tres parámetros:

1. El valor de k-anonimidad de la secuencia. Se debe conseguir un valor igual o superior al valor deseado.

2. La distancia entre la secuencia de grados y la secuencia de grados original. El objetivo es minimizar este valor.

3. En el caso de obtener un valor de k-anonimidad inferior al valor deseado, se considera el número de grupos de nodos que, agrupados según su grado, presentan una cardinalidad inferior al valor de k deseado. Cuando el valor de k-anonimidad sea el deseado, este parámetro vale 0.

Para la selección de los supervivientes, es decir, los individuos que pasan a la siguiente generación, se utiliza el modelo de estado progresivo (*steady-state model*). Según este modelo, se seleccionan los peores individuos de la población actual y se reemplazan por los mejores individuos del conjunto de candidatos. Esta valoración se realiza basándose en la puntuación obtenida en la función de *fitness*.

El Algoritmo 3 muestra el pesudocódigo del algoritmo asociado a esta primera parte.

Algorithm 3 Pseudocódigo del algoritmo para obtener una secuencia k-anónima

Entrada: La secuencia de grados original, d, y el valor de k-anonimidad deseado, k.
Salida: La secuencia de grados, d', que cumple la k-anonimidad.

```
INICIALIZAR poblacion ⇐ d
while iter < iter_max or iter_sin_mejora < iter_sin_mejora_max do
  MUTAR poblacion
  EVALUAR nuevos candidatos
  poblacion ⇐ SELECCIONAR individuos
end while
d' ⇐ SELECCIONAR mejor individuo
return d'
```

4.1.2. Reconstrucción del grafo

En la segunda fase, se debe crear un grafo a partir de la secuencia k-anónima generada en la fase anterior. Luego se aplica un proceso de intercambio válido de aristas para conseguir que el nuevo grafo sea lo más parecido posible al grafo original. Liu y Terzi definen este intercambio válido de aristas como una operación entre cuatro nodos i, j, k y l de $G_i(V, E_i)$ tales que (i, k) y $(j, l) \in E_i$ y (i, j) y $(k, l) \notin E_i$ o (i, l) y $(j, k) \notin E_i$. Este proceso de intercambio es iterativo. Entonces, una operación de intercambio válida de aristas que producirá $G_{i+1}(V, E_{i+1})$ a partir de $G_i(V, E_i)$ es:

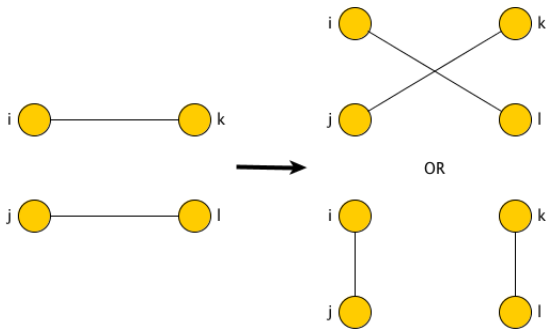

Figura 4.1: Representación del intercambio válido entre aristas.

$$E_{i+1} \leftarrow E_i \setminus \{(i,k),(j,l)\} \cup \{(i,j),(k,l)\}$$

$$o$$

$$E_{i+1} \leftarrow E_i \setminus \{(i,k),(j,l)\} \cup \{(i,l),(j,k)\}$$

Donde el objetivo es conseguir que el conjunto de aristas del nuevo grafo sea tan similar como sea posible al conjunto de aristas del grafo original ($E \cap \widehat{E} \approx E$). Es importante notar que el conjunto de nodos es el mismo, ya que la modificación de la secuencia de grados no produce ninguna modificación en el número de nodos del grafo. La Figura 4.1 ejemplifica de forma gráfica esta transformación.

El Algoritmo 4 muestra el pesudocódigo del algoritmo asociado a esta segunda parte.

Algorithm 4 Pseudocódigo del algoritmo para construir el grafo a partir de la secuencia k-anónima

Entrada: El grafo obtenido de secuencia k-anónima $\widehat{G}_0(V,\widehat{E})$ y el grafo original $G(V,E)$.
Salida: El grafo $\widehat{G}(V,\widehat{E})$ donde se ha modificado $\widehat{E}$ para cumplir $\widehat{E} \cap E \approx E$.

```
Ĝ(V, Ê) ⇐ Ĝ_0(V, Ê)
(c, (e_1, e_2, e'_1, e'_2)) = Find_Max_Swap(Ĝ, G)
while c > 0 do
    Ê = Ê \ {e_1, e_2} ∪ {e'_1, e'_2}
    (c, (e_1, e_2, e'_1, e'_2)) = Find_Max_Swap(Ĝ, G)
end while
return Ĝ
```

4.2. Medidas de calidad y métodos de *graph mining*

Las medidas de calidad utilizadas en el análisis del algoritmo RGC son las mismas que se han utilizado en el capítulo anterior, y que están descritas en la sección 3.2. Al igual que en el capítulo anterior, no se han considerado como parámetros a evaluar el número de nodos, el número de aristas y el grado medio, dado que en el método RGC estos valores permanecen constantes.

De forma adicional, en este estudio se considera como medida el número de aristas que se han modificado durante el proceso de anonimización de cada grafo. En los métodos descritos en el capítulo anterior esta información no es relevante, ya que directamente se obtiene a partir del porcentaje de anonimización y del número de aristas del grafo original. En el método RGC, y en general en cualquier método de modificación de aristas para la preservación de la k-anonimidad, el número de aristas modificadas en cada proceso de anonimización es un dato importante que puede servir como medida para cuantificar el grado de perturbación que se ha introducido en el grafo anonimizado.

Los algoritmos de *graph mining* aplicados son los mismos que en el capítulo anterior. Para cada conjunto de datos se aplican los mismos valores en los parámetros de configuración que en el capítulo anterior.

4.3. Conjunto *Zachary's Karate Club*

Las principales características de este grafo son:

- 34 nodos y 78 aristas.
- No dirigido y sin etiquetas en las aristas.
- El grado medio es de 4,588.
- La distancia media es de 2,408.
- El diámetro es de 5.
- El histograma de grados es: [0, 1, 11, 6, 6, 3, 2, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1].
- Valor de k-anonimidad igual a 1.

Valor de k	2	3	4	5
Núm. aristas	29	40	44	45
% aristas	37,2 %	51,3 %	56,4 %	57,7 %

Cuadro 4.1: Número de aristas modificadas según el valor de k-anonimidad.

	Original	$k = 2$	$k = 3$	$k = 4$	$k = 5$
Distancia media	2,408	2,422	2,522	2,643	2,62
Diámetro	5	6	5	6	6

Cuadro 4.2: Distancia media y diámetro según el valor de k-anonimidad.

4.3.1. Propiedades estructurales

Para esta evaluación se han generado grafos con un valor de k-anonimidad igual a 2, 3, 4 y 5. Es importante notar que muy pocos (cerca de 0,55 %) en la anonimización mediante *Random Perturbation* llegaron a obtener un valor de k-anonimidad igual a 2, y sólo un grafo (que representa el 0,05 %) obtuvo un grado de k-anonimidad igual a 3.

La Tabla 4.1 muestra el número de aristas que se han modificado en cada paso para conseguir los distintos valores de k-anonimidad. Para conseguir un valor de $k = 2$, el algoritmo RGC ha modificado 29 aristas del grafo original. Como se indica en la tabla, representa un 37,2 % del número total de aristas del grafo. Para poder comparar con los resultados vistos en el capítulo anterior, los algoritmos *Random Perturbation* y *Random Switch* modifican 29 aristas en el grafo cuando aplican un porcentaje de anonimización aproximado del 17 %.

La Tabla 4.2 muestra la evolución de la **distancia media** y el **diámetro**. Como se puede ver, la distancia media presenta un aumento progresivo en función del valor de k-anonimidad. En el grafo original se obtiene un valor de 2,408 para esta medida, que aumenta hasta alcanzar un valor máximo de 2,643 para el grafo con $k = 4$. El diámetro presenta unos resultados similares, aumentando ligeramente con el valor de k-anonimidad. Ambas medidas indican un descenso en la cohesión de los nodos de los grafos anonimizados.

La Figura 4.2 muestra el **histograma de grados** del grafo anonimizado con valor $k = 2$. En ella se puede apreciar como se han modificado los grados de los nodos que incumplían la k-anonimidad, y como el algoritmo modifica la secuencia de grados de tal forma que se respeta al máximo su forma original. Se puede comparar con la Figura 3.9, que corresponde a una perturbación *Random Perturbation* del 20 %, lo que implica que se han modificado unas 34 aristas, es decir, 5 más que en este caso. La Figura 4.3 muestra un deterioro mucho más elevado cuando se modifica el grafo para conseguir un valor de k-anonimidad igual a 5.

Las Figuras 4.4 y 4.5 muestran la evolución de la medida ***betweenness centrality***. Ambas medidas presentan una distorsión superior a las mostradas por los algoritmos *Random*

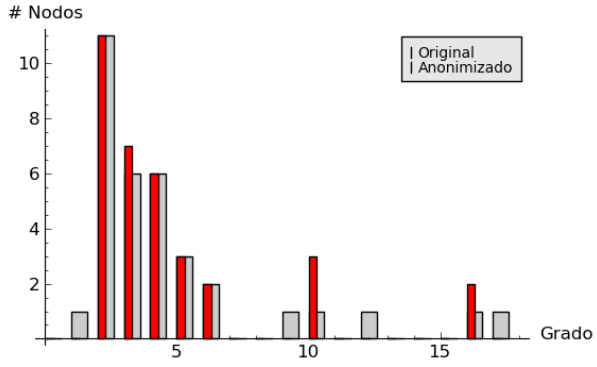

Figura 4.2: Histograma de grados para el grafo con valor $k = 2$.

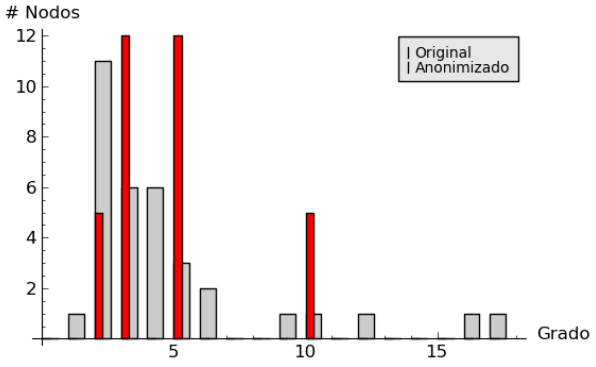

Figura 4.3: Histograma de grados para el grafo con valor $k = 5$.

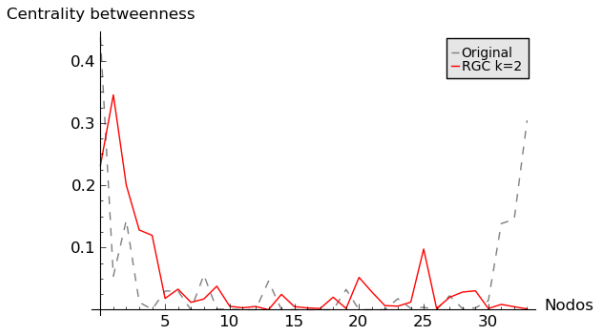

Figura 4.4: *Betweenness centrality* para el grafo con valor $k = 2$.

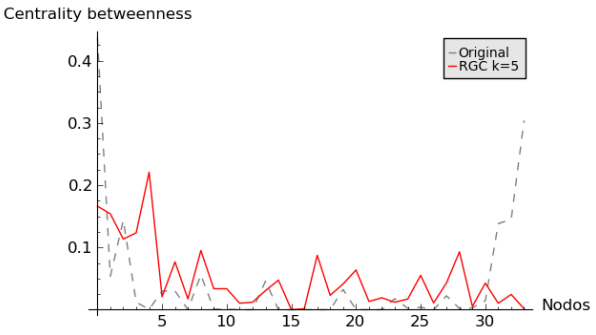

Figura 4.5: *Betweenness centrality* para el grafo con valor $k = 5$.

Perturbation y *Random Switch*.

Las Figuras 4.6 y 4.7 muestran la evolución de la medida ***closeness centrality***. Al igual que en el caso anterior, ambas figuras presentan una distorsión muy superior a las presentadas por los algoritmos de modificación aleatoria de aristas.

Las Figuras 4.8 y 4.9 muestran la evolución de la medida ***degree centrality***. Al igual que en las dos medidas anteriores, el grado de perturbación de estas medidas es alto, especialmente en los nodos 30-34 del grafo.

4.3.2. Resultados del proceso de *graph mining*

A continuación se discuten los resultados obtenidos tras el proceso de *clustering*.

Las Figuras 4.10 y 4.11 muestran los índices de Jaccard para los algoritmos MCL y RRW, respectivamente. En el eje horizontal (eje de la x) se muestra el valor de k-anonimidad conseguido en los datos, que varía entre $k = 1$ (los datos originales, sin alteración alguna) y $k = 5$

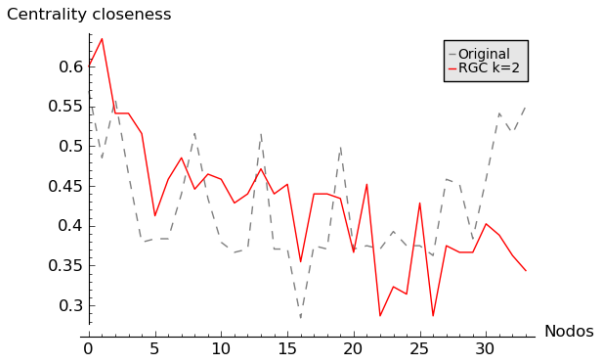

Figura 4.6: *Closeness centrality* para el grafo con valor $k = 2$.

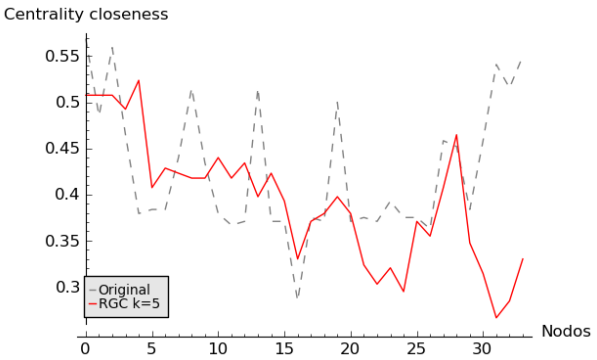

Figura 4.7: *Closeness centrality* para el grafo con valor $k = 5$.

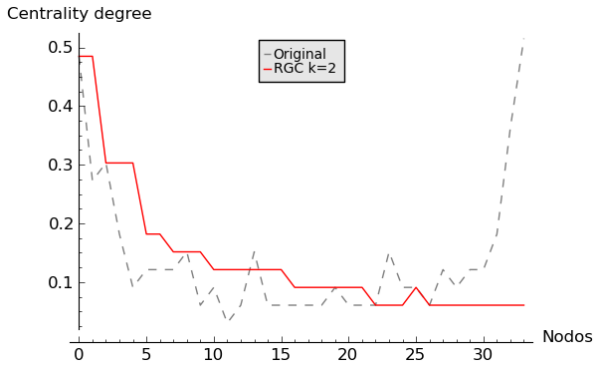

Figura 4.8: *Degree centrality* para el grafo con valor $k = 2$.

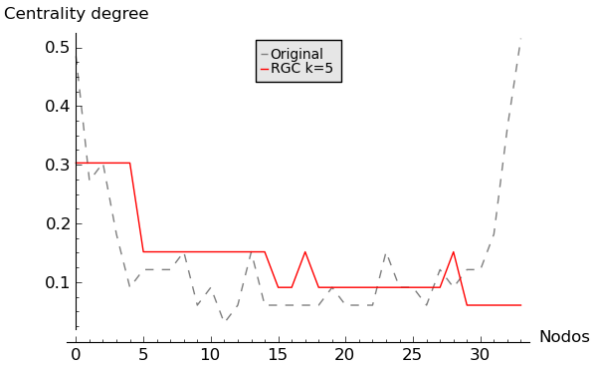

Figura 4.9: *Degree centrality* para el grafo con valor $k = 5$.

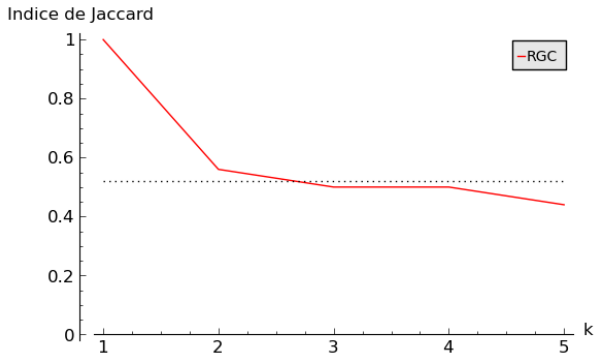

Figura 4.10: Índice de Jaccard en MCL.

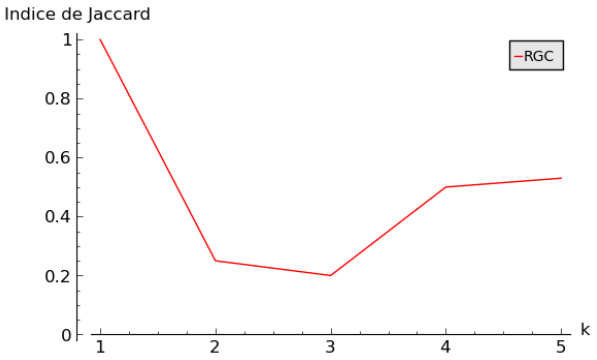

Figura 4.11: Índice de Jaccard en RRW.

(el valor máximo de anonimización aplicado en este estudio). En el eje vertical (eje de la y) de muestra el valor del índice de Jaccard.

La Figura 4.10 muestra como el resultado obtenido para el caso de $k = 2$ es sensiblemente peor al obtenido con un porcentaje de anonimización similar con los datos *Random Perturbation* o *Random Switch*. Concretamente, del orden del 10 % peor. Las anonimizaciones con valores de k superiores a 2 se encuentran en valores próximos al 0.5, que como se vio en el apartado anterior, corresponden a valores similares a los obtenidos con grafos aleatorios.

Al igual que la evaluación anterior, la evaluación utilizando el algoritmo RRW es peor, aunque en este caso se agudiza, obteniendo valores muy bajos para las anonimizaciones con valores de $k = 2$ y $k = 3$.

4.3.3. Riesgo de re-identificación

El cálculo del riesgo de re-identificación, basado en el valor de la k-anonimidad en el grado, es directo, ya que los grafos generados mediante el algoritmo RGC tienen un valor de k-anonimidad igual a 2, 3, 4 y 5.

Es importante notar que el aumento en la protección de los datos generados con el algoritmo RGC es muy superior a la protección que han demostrado los algoritmos anonimizados con los dos métodos anteriores. *Random Perturbation* sólo ha conseguido generar un conjunto muy pequeño de grafos con un valor de k-anonimidad igual a 2, mientras que *Random Switch* no genera variación en la secuencia de grados, y por lo tanto, no ha generado ningún grafo con un valor de k-anonimidad mayor que 1.

4.3.4. Conclusiones

En esta sección se han realizado anonimizaciones mediante el método RGC para obtener grafos con valores de k-anonimidad superiores al valor del grafo original. Se han generado grafos con valores de k-anonimidad iguales a 2, 3, 4 y 5. Para poder obtener los grafos anonimizados ha sido necesario modificar una gran cantidad de aristas del grafo, perturbando en gran medida las características estructurales del grafo original.

Las medidas relacionadas con la centralidad han mostrado una gran perturbación en cualquiera de los grafos generados. Los resultados obtenidos en el proceso de *clustering* no son los deseados. El grafo anonimizado con un valor de $k = 2$ sólo obtiene un valor de 0,56 en el índice de Jaccard. Una clasificación simple con un único *cluster* que contiene todos los nodos obtiene un valor de 0,52 en el índice de Jaccard. Por lo tanto, el valor obtenido para el grafo con un valor de $k = 2$ no se puede considerar satisfactorio.

Si se comparan los resultados obtenidos por el método RGC con los resultados obtenidos

Valor de k	4	5	10
Núm. aristas	30	35	27
% aristas	4,89 %	5,70 %	4,40 %

Cuadro 4.3: Número de aristas modificadas según el valor de k-anonimidad.

por *Random Perturbation* y *Random Switch* se puede ver que el descenso en el índice de Jaccard se sitúa aproximadamente en el 10 %. A cambio, el método RGC obtiene un grafo con un valor de k-anonimidad superior, que proporciona un nivel de seguridad mayor.

4.4. Conjunto *American College Football*

Las principales características de este grafo son:

- 115 nodos y 613 aristas.
- No dirigido y sin etiquetas en las aristas.
- El grado medio es de 10,661.
- La distancia media es de 2,508.
- El diámetro es de 4.
- El histograma de grados es: [0, 0, 0, 0, 0, 0, 0, 1, 3, 5, 28, 66, 12].
- Valor de k-anonimidad igual a 1.

4.4.1. Propiedades estructurales

Para esta evaluación se han generado grafos con un valor de k-anonimidad igual a 4, 5 y 10. Es importante notar que muy pocos grafos (cerca de 0,44 %) en la anonimización mediante *Random Perturbation* llegaron a obtener un valor de k-anonimidad igual a 4 o 5, y en ningún caso se obtuvo un grafo con un valor de k-anonimidad igual a 10.

La Tabla 4.3 muestra el número de aristas que se han modificado en cada paso para conseguir los distintos valores de k-anonimidad. Para conseguir un valor de $k = 4$, el algoritmo RGC ha modificado 30 aristas del grafo original. Como se indica en la tabla, representa un 4,89 % del número total de aristas del grafo. De forma similar, se consigue una valor de $k = 5$ con la modificación del 5,70 %. Para conseguir un valor de $k = 10$ sólo es necesario modificar un total de 27 aristas del grafo original, que representa sólo un 4,40 % del número total de aristas.

	Original	$k = 4$	$k = 5$	$k = 10$
Distancia media	2,508	2,460	2,450	2,465
Diámetro	4	4	4	4

Cuadro 4.4: Distancia media y diámetro según el valor de k-anonimidad.

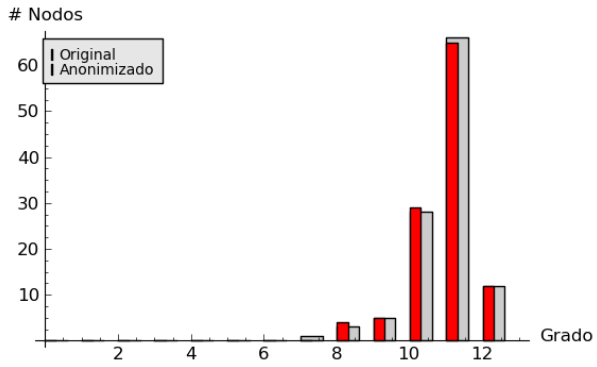

Figura 4.12: Histograma de grados para el grafo con valor $k = 4$.

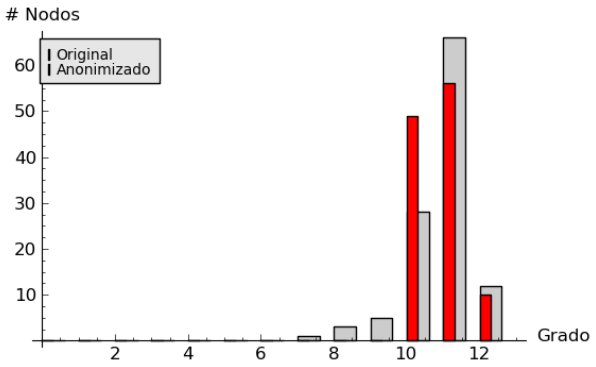

Figura 4.13: Histograma de grados para el grafo con valor $k = 10$.

La Tabla 4.4 muestra la evolución de la **distancia media** y el **diámetro**. Ambas medidas presentan un grado de distorsión muy bajo. La distancia media sufre una ligera disminución, aunque se mantiene en valores muy próximos al valor original, mientras que el diámetro se mantiene constante.

La Figura 4.12 muestra el **histograma de grados** del grafo anonimizado con valor $k = 4$. En ella se puede apreciar como se ha realizado un pequeño número de cambios en el histograma de grados que permite cumplir con un valor de k-anonimidad igual a 4. La Figura 4.13 muestra el histograma de grados del grafo anonimizado con valor $k = 10$. El histograma permite ver, de una forma gráfica, qué cambios se han aplicado para permitir este valor de k.

Las Figuras 4.14 y 4.15 muestran la evolución de la medida ***betweenness centrality***. Ambas figuras presentan una distorsión inferior a las mostradas por los algoritmos *Random Perturbation* y *Random Switch*.

Las Figuras 4.16 y 4.17 muestran la evolución de la medida ***closeness centrality***. Al igual que en el caso anterior, ambas figuras presentan una distorsión aceptable e inferior a las presentadas por los algoritmos de modificación aleatoria de aristas.

Las Figuras 4.18 y 4.19 muestran la evolución de la medida ***degree centrality***. A diferencia de las dos medidas precedentes, en este caso se percibe una distorsión más notable, debido a que en estos grafos se han modificado los nodos que presentaban un grado menor, y que presentan valores extremos (*outliers*) en las gráficas.

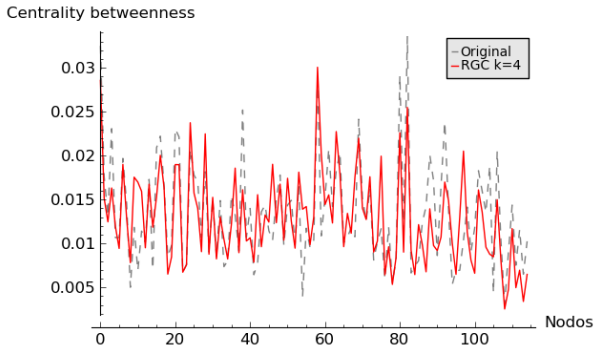

Figura 4.14: *Betweenness centrality* para el grafo con valor $k = 4$.

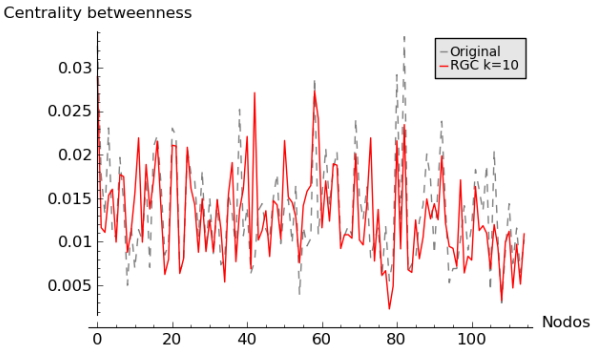

Figura 4.15: *Betweenness centrality* para el grafo con valor $k = 10$.

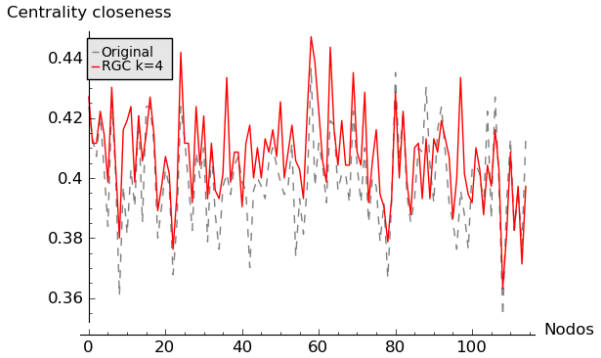

Figura 4.16: *Closeness centrality* para el grafo con valor $k = 4$.

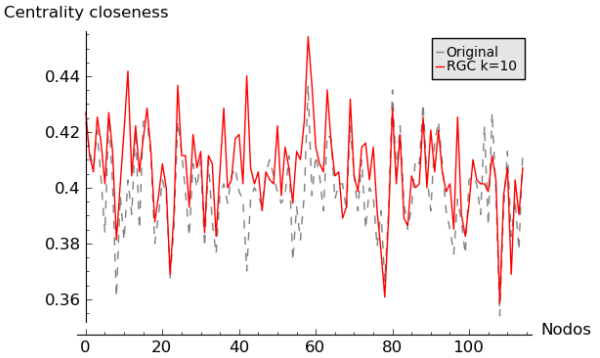

Figura 4.17: *Closeness centrality* para el grafo con valor $k = 10$.

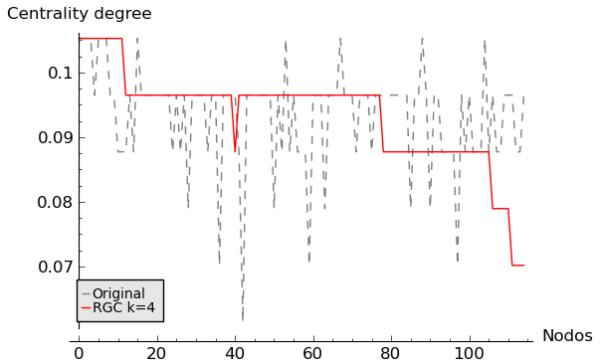

Figura 4.18: *Degree centrality* para el grafo con valor $k = 4$.

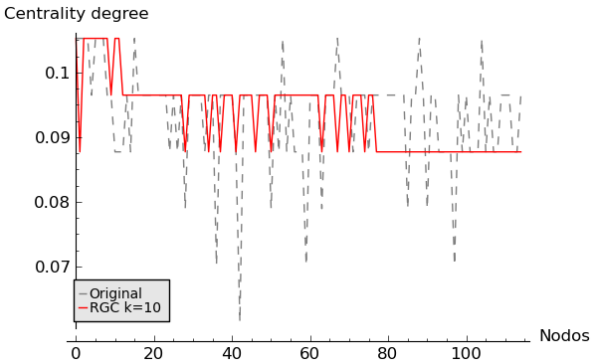

Figura 4.19: *Degree centrality* para el grafo con valor $k = 10$.

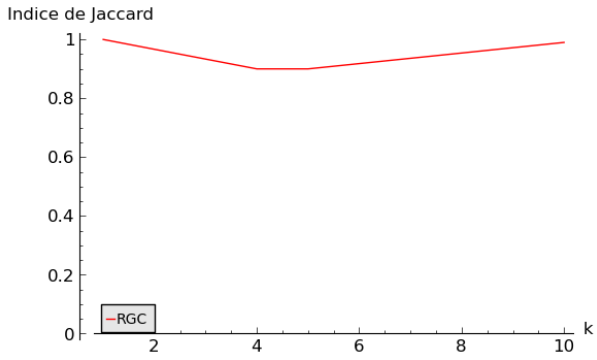

Figura 4.20: Índice de Jaccard en MCL.

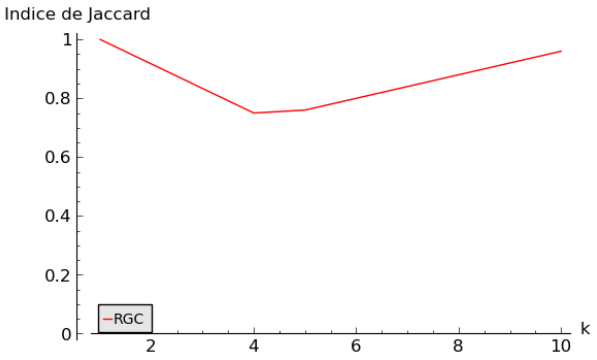

Figura 4.21: Índice de Jaccard en RRW.

4.4.2. Resultados del proceso de *graph mining*

A continuación se discuten los resultados obtenidos tras el proceso de *clustering*.

Las Figuras 4.20 y 4.21 muestran los índices de Jaccard para los algoritmos MCL y RRW, respectivamente. En el eje horizontal se muestra el valor de k-anonimidad conseguido en los datos, que varía entre $k = 1$ (los datos originales, sin alteración alguna) y $k = 10$ (el porcentaje máximo de anonimización aplicado en este estudio). En el eje vertical de muestra el valor del índice de Jaccard.

Ambas figuras muestran unos resultados excelentes, especialmente en el grafo con un valor de $k = 10$. En los grafos anonimizados con un valor de k igual a 4 y 5 se consiguen muy buenos resultados. El índice de Jaccard se sitúa alrededor de 0,9 para el algoritmo de *clustering* MCL y de 0,75 para RRW. Pero el mejor resultado se consigue con el grafo anonimizado con un valor de k igual a 10. Para este grafo se consiguen valores del índice de Jacard de 0,99 para el algoritmo MCL y 0,96 para el algoritmo RRW.

4.4.3. Riesgo de re-identificación

El cálculo del riesgo de re-identificación, basado en el valor de la k-anonimidad en el grado, es directo, ya que los grafos generados mediante el algoritmo RGC tienen un valor de k-anonimidad igual a 4, 5 y 10.

Es importante notar que el aumento en la protección de los datos generados con el algoritmo RGC es muy superior a la protección que han demostrado los algoritmos basados en la modificación aleatoria de aristas. En ningún caso el proceso de modificación aleatorio ha construido un grafo con valor de k-anonimidad cercano a 10.

4.4.4. Conclusiones

En esta sección se han realizado anonimizaciones mediante el método RGC para obtener grafos con valores de k-anonimidad igual a 4, 5 y 10.

El método RGC presenta unos resultados excelentes en todas las pruebas. La clave para este éxito es el reducido número de modificaciones que deben realizarse para obtener el grafo con el valor de k-anonimidad deseada. Esto facilita que las propiedades estructurales se mantengan de una forma más que correcta, produciendo unos excelentes resultados en el proceso de *graph mining*.

Merece especial atención el grafo con un valor de k-anonimidad igual a 10. En este caso se obtiene un grafo con un nivel de seguridad muy elevado y se preserva de una forma excelente la utilidad de los datos en los dos procesos de minería de datos aplicados.

4.5. Conjunto *Jazz musicians*

Las principales características de este grafo son:

- 198 nodos y 2.742 aristas.
- No dirigido y sin etiquetas en las aristas.
- El grado medio es de 27,697.
- La distancia media es de 2,235.
- El diámetro es de 6.
- El histograma de grados es: [0, 5, 3, 3, 3, 4, 5, 3, 3, 3, 2, 1, 3, 6, 4, 1, 4, 5, 4, 7, 8, 2, 2, 9, 5, 5, 3, 5, 3, 5, 2, 9, 3, 4, 3, 1, 2, 3, 1, 6, 5, 4, 3, 3, 1, 3, 5, 0, 2, 2, 0, 1, 2, 2, 1, 2, 2, 1, 0, 2, 2, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1].
- Valor de k-anonimidad igual a 1.

4.5.1. Propiedades estructurales

Para esta evaluación se ha generado un único grafo con un valor de k-anonimidad igual a 2. En la anonimización mediante *Random Perturbation* no se ha obtenido ningún grafo con un valor de k-anonimidad superior a 1, por lo tanto, este grafo presenta una mejora importante en términos de seguridad frente a los grafos anonimizados mediante las técnicas de modificación aleatoria de aristas.

Valor de k	2
Núm. aristas	1.286
% aristas	46,90 %

Cuadro 4.5: Número de aristas modificadas según el valor de k-anonimidad.

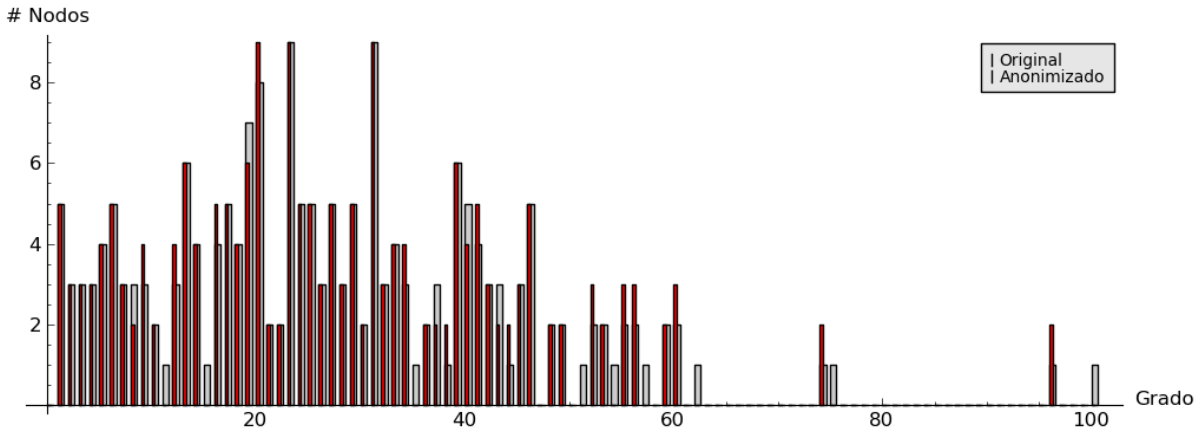

Figura 4.22: Histograma de grados para el grafo con valor $k = 2$.

La Tabla 4.5 muestra el número de aristas que se han modificado para conseguir un grafo con un valor de k-anonimidad igual a 2. Como se puede ver, el número de aristas que se han modificado es muy grande, del orden del 47 % del número total de aristas. El histograma de grados muestra que existe gran cantidad de nodos con grado único (valor igual a 1 en el histograma de grados). Para aumentar el valor de k-anonimidad del grafo es necesario modificar las aristas de estos nodos, de forma que su grado deje de ser único dentro del grafo.

La **distancia media** y el **diámetro** no se pueden calcular en el grafo anonimizado, ya que resulta inconexo.

La Figura 4.22 muestra el **histograma de grados** del grafo anonimizado. Como se puede ver, las modificaciones aplicadas minimizan los cambios a realizar para conseguir el valor de k-anonimidad deseado, pero aún así, son necesarios gran cantidad de cambios para conseguir un valor de k-anonimidad igual a 2 en este grafo.

La Figura 4.23 muestra el valor de ***betweenness centrality***. La figura muestra un grado de perturbación muy elevado, que ha deteriorado de forma muy importante los valores originales para esta medida de centralidad.

La Figura 4.24 muestra el valor de ***closeness centrality***. Al igual que en el caso anterior, la figura muestra un grado de perturbación muy elevado.

La Figura 4.25 muestra el valor de la última medida de centralidad incluida en este trabajo, la ***degree centrality***. A igual que en las dos medidas precedentes, en este caso también se muestra un grado de perturbación muy elevado. Los valores de esta medida en el grafo

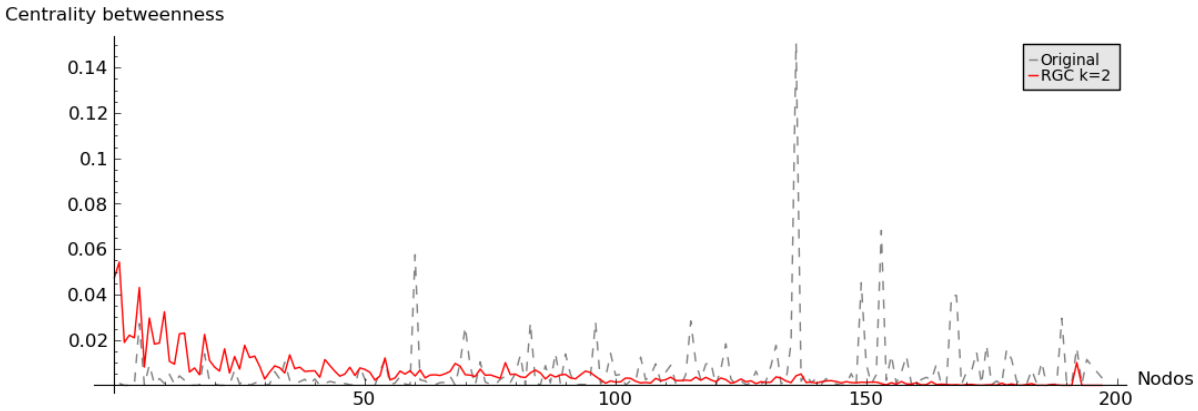

Figura 4.23: *Betweenness centrality* para el grafo con valor $k = 2$.

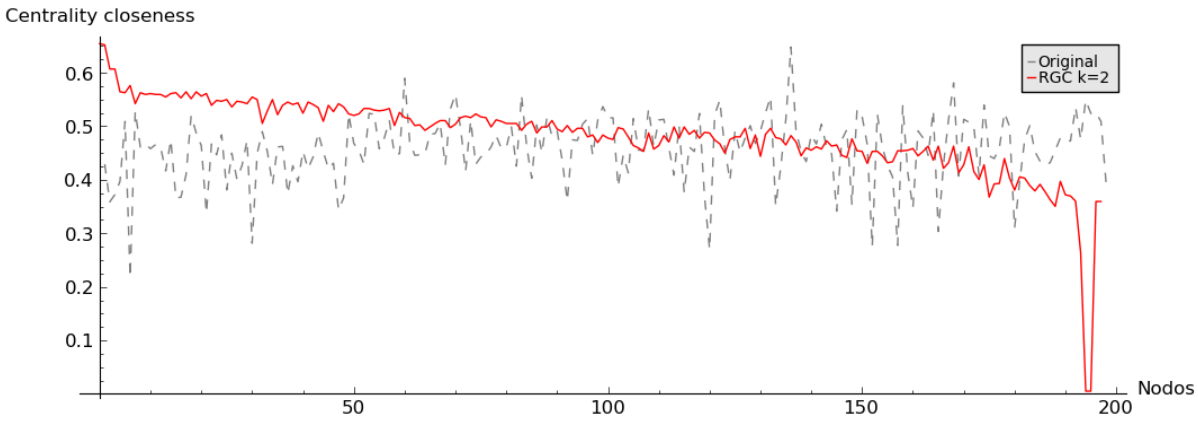

Figura 4.24: *Closeness centrality* para el grafo con valor $k = 2$.

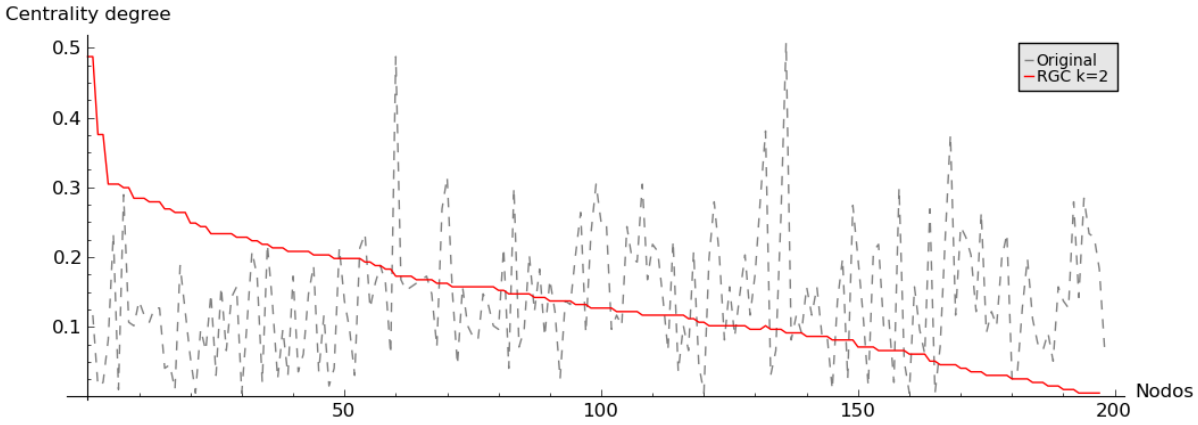

Figura 4.25: *Degree centrality* para el grafo con valor $k = 2$.

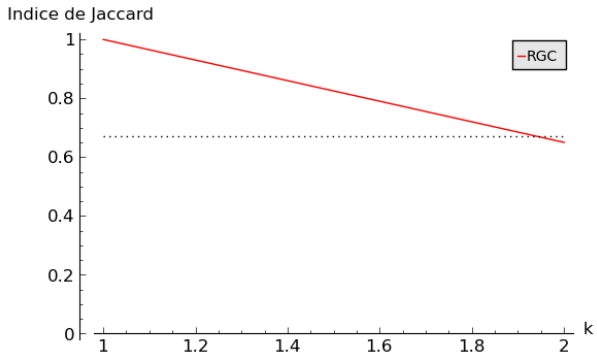

Figura 4.26: Índice de Jaccard en MCL.

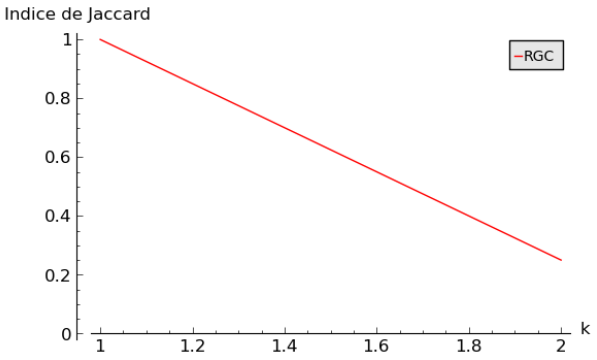

Figura 4.27: Índice de Jaccard en RRW.

anonimizado son totalmente distintos a los valores presentados por el grafo original.

4.5.2. Resultados del proceso de *graph mining*

A continuación se discuten los resultados obtenidos tras el proceso de *clustering*.

Las Figuras 4.26 y 4.27 muestran los índices de Jaccard para los algoritmos MCL y RRW, respectivamente. En el eje horizontal se muestra el valor de k-anonimidad de los datos originales ($k = 1$) y de los datos anonimizados ($k = 2$). El eje vertical muestra el valor del índice de Jaccard.

La Figura 4.26 muestra un índice de Jaccard de 0,65 para el del grafo anonimizado. Tal y como se ha comentado en la sección 3.6.2, dada la distribución de los *clusters* de resultados, es posible obtener un índice de Jaccard de 0,67 (linea punteada de color gris) situando todos los nodos en un mismo *cluster*. Por lo tanto, el resultado obtenido por el grafo anonimizado con $k = 2$ no consigue un resultado satisfactorio.

La Figura 4.27 muestra el índice de Jaccard para el algortimo RRW. Al igual que en el caso anterior, el resultado para el grafo anonimizado no es satisfactorio, obteniendo un valor de 0,25.

4.5.3. Riesgo de re-identificación

El cálculo del riesgo de re-identificación, basado en el valor de la k-anonimidad en el grado, es directo, ya que el grafo generado mediante el algoritmo RGC tiene un valor de k-anonimidad igual a 2.

Es importante notar que el aumento en la protección de los datos generados con el algoritmo RGC es superior a la protección que han demostrado los algoritmos basados en la modificación aleatoria de aristas. En ningún caso el proceso de modificación aleatorio ha construido un grafo con valor de k-anonimidad superior a 1.

4.5.4. Conclusiones

En esta sección se han realizado anonimizaciones mediante el método RGC para obtener un grafo con un valor de k-anonimidad igual a 2.

Las medidas estructurales, y en especial las medidas relacionadas con la centralidad, han mostrado un grado de perturbación muy elevado en el grafo anonimizado. Los resultados obtenidos en el proceso de *clustering* no son satisfactorios. El grafo anonimizado obtiene un valor de 0,65 en el índice de Jaccard para el algoritmo de *clustering* MCL, mientras que una clasificación simple con un único *cluster* que contiene todos los nodos obtiene un valor de 0,67.

Los resultados obtenidos por el método RGC son similares a los obtenidos por *Random Perturbation* y *Random Switch* con un porcentaje de anonimización del 10 %. Aunque es importante notar que el método RGC obtiene un grafo con un valor de k-anonimidad superior, que proporciona un nivel de seguridad mayor.

4.6. Conclusiones

En este capítulo se ha detallado una propuesta de mejora sobre el algoritmo *Relaxed Graph Construction* que presentan Liu y Terzi en [27]. Esta propuesta se basa en utilizar algoritmos genéticos para obtener una secuencia de grados k-anónima, que será utilizada para la construcción del grafo anonimizado. Este algoritmo pertenece al grupo de algoritmos basados en modificar las aristas de un grafo para preservar el modelo de k-anonimidad. Su funcionamiento es ligeramente distinto a los algoritmos presentados en el capitulo anterior, y su objetivo es conseguir un grafo con un valor de k-anonimidad específico.

En este capítulo se ha evaluado el nuevo algoritmo RGC utilizando los mismos conjuntos de datos que en el capítulo anterior.

En dos de los tres conjuntos de datos evaluados se han conseguido resultados similares en distorsión de las propiedades estructurales y resultados del proceso de *graph mining*. Pero aún así, el algoritmo RGC siempre ha conseguido grafos con un valor de k-anonimidad superior a los valores conseguidos por los métodos evaluados en el capítulo anterior.

En el conjunto de datos *American College Football*, los resultados obtenidos por el método RGC han sido muy superiores a los resultados obtenidos por los métodos *Random Perturbation* y *Random Switch*, evaluados en el capítulo anterior.

Capítulo 5

Conclusiones y trabajo futuro

En este trabajo se han evaluado algunos métodos de anonimización o preservación de la privacidad en entornos de minería de datos basados en grafos (*graph mining*). Para su evaluación se han considerado tres tipos de medidas distintas: en primer lugar se han considerado medidas referentes a las propiedades estructurales de los grafos, como por ejemplo, histograma de grados, medidas basadas en la centralidad de los nodos, etc. En segundo lugar se han considerado medidas relacionadas con los resultados de procesos de *clustering*, con el fin de poder evaluar el impacto real que producen los métodos de anonimización sobre los procesos posteriores de *graph mining*. Y en tercer lugar, se han considerado medidas para evaluar el riesgo de re-identificación asociado al grafo anonimizado.

En el capítulo 3 de este trabajo se han evaluado dos métodos de anonimización de grafos basados en la modificación aleatoria de aristas. Se ha visto que ambos métodos introducen una cantidad considerable de ruido en los datos, provocando una distorsión notable en sus propiedades estructurales. Los resultados asociados a los procesos de *graph mining* no han sido, en general, satisfactorios. Los índices utilizados para evaluar la bondad de estos resultados han demostrado poca resistencia de los datos al ruido introducido por estos métodos de anonimización. Además, el nivel de protección asociado al riesgo de re-identificación no presenta mejoras notables; sólo en algunos casos se consigue una cierta mejora, pero que puede ser insuficiente para determinados usos de los datos.

En el capítulo 4 de este trabajo se ha implementado una propuesta de mejora en el algoritmo *Relaxed Graph Construction* que presentan Liu y Terzi en [27]. Esta propuesta se basa en utilizar algoritmos genéticos para obtener una secuencia de grados k-anónima, que será utilizada para la construcción del grafo anonimizado. Este algoritmo pertenece al grupo de algoritmos basados en modificar las aristas para preservar el modelo de k-anonimidad. El algoritmo RGC obtiene resultados similares, en cuanto a pérdida de información, en dos de los tres conjuntos de datos analizados. En el otro conjunto de datos analizado obtiene unos resultados excelentes. Pero en

todos los casos evaluados, el algoritmo RGC obtiene grafos con un nivel de seguridad mayor o mucho mayor que el grafo original.

En base a los resultados obtenidos, se puede decir que los métodos analizados de anonimización basados en la modificación aleatoria de aristas no consiguen aumentar de forma significativa el nivel de seguridad de los datos anonimizados, mientras que introducen una cantidad importante de ruido en los datos. Alternativamente, el método analizado de modificación de aristas para preservar el modelo de k-anonimidad, el algoritmo RGC, consigue un aumento significativo del nivel de seguridad de los datos, aunque el nivel de ruido introducido es también demasiado elevado en algunos casos.

Independientemente del método analizado, en este trabajo se ha podido observar que no existe una correlación clara entre los resultados de analizar las propiedades estructurales y los resultados reales de los procesos de *graph mining* aplicados sobre los datos anonimizados.

El trabajo deja una multitud de caminos abiertos para seguir con esta investigación. El primer lugar, se podrían implementar y evaluar otros métodos de anonimización basados en la modificación de aristas para preservar el modelo de k-anonimidad. También sería muy interesante implementar y evaluar algunos métodos basados en la generalización, aunque implicaría un cambio circunstancial en las medidas de evaluación y re-identificación.

En segundo lugar, resultaría muy interesante abrir el estudio a otros tipos de grafos. El uso de grafos con pesos en las aristas o atributos en los nodos abre nuevos retos a la anonimización. En este caso se deberían implementar y evaluar otros tipos de métodos de anonimización, así como replantear las medidas de evaluación y re-identificación.

En tercer lugar, se podría ampliar este trabajo explorando otras características que permitieran evaluar el grado de perturbación asociado a un proceso de anonimización. Una posibilidad muy interesante seria considerar las propiedades espectrales de los grafos. Los valores y vectores propios de las matrices de adyacencia o de Laplace pueden ayudar a evaluar como afecta a un grafo la perturbación introducida por los procesos de anonimización.

Bibliografía

[1] James Abello, Mauricio G. C. Resende, and Sandra Sudarsky. Massive quasi-clique detection. In *Proceedings of the 5th Latin American Symposium on Theoretical Informatics*, LATIN '02, pages 598–612, London, UK, UK, 2002. Springer-Verlag.

[2] Charu C. Aggarwal, Na Ta, Jianyong Wang, Jianhua Feng, and Mohammed Zaki. Xproj: a framework for projected structural clustering of xml documents. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '07, pages 46–55, New York, NY, USA, 2007. ACM.

[3] Charu C. Aggarwal and Haixun Wang. *Managing and Mining Graph Data*. Springer Publishing Company, Incorporated, 1st edition, 2010.

[4] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, united states ed edition, February 1993.

[5] Lars Backstrom, Cynthia Dwork, and Jon Kleinberg. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 181–190, New York, NY, USA, 2007. ACM.

[6] Christian Borgelt and Michael R. Berthold. Mining molecular fragments: Finding relevant substructures of molecules. In *Proceedings of the 2002 IEEE International Conference on Data Mining*, ICDM '02, pages 51–, Washington, DC, USA, 2002. IEEE Computer Society.

[7] Bing-Jing Cai, Hai-Ying Wang, Hui-Ru Zheng, and Hui Wang. Evaluation repeated random walks in community detection of social networks. In *Machine Learning and Cybernetics (ICMLC), 2010 International Conference on*, volume 4, pages 1849 –1854, 2010.

[8] Alina Campan and Traian Marius Truta. A clustering approach for data and structural anonymity in social networks. In *In Privacy, Security, and Trust in KDD Workshop (PinKDD)*, 2008.

[9] Deepayan Chakrabarti, Yiping Zhan, and Christos Faloutsos. R-mat: A recursive model for graph mining. In *In SDM*, 2004.

[10] Joseph Cheriyan, Torben Hagerup, and Kurt Mehlhorn. An o(n3)-time maximum-flow-algorithm: Can a maximum flow be computed in o(nm) time? *SIAM Journal on Computing*, 25(6):1144–1170, 1996.

[11] Fan R. K. Chung. *Spectral Graph Theory*. AMS, 1994.

[12] Diane J. Cook and Lawrence B. Holder. *Mining Graph Data*. John Wiley & Sons, 2006.

[13] T. Dalamagas, T. Cheng, K. Winkel, and T. K. Sellis. Clustering XML Documents Using Structural Summaries. In *EDBT Workshops*, pages 547–556, 2004.

[14] Josep Domingo-Ferrer and Vicenç Torra. Ordinal, continuous and heterogeneous k-anonymity through microaggregation. *Data Min. Knowl. Discov.*, 11:195–212, September 2005.

[15] Per O. Fjällström. Algorithms for graph partitioning: A Survey. In *Linkoping Electronic Atricles in Computer and Information Science, 3.*, 1998.

[16] David Gibson, Ravi Kumar, and Andrew Tomkins. Discovering large dense subgraphs in massive graphs. In *Proceedings of the 31st international conference on Very large data bases*, VLDB '05, pages 721–732. VLDB Endowment, 2005.

[17] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America*, 99(12):7821–7826, June 2002.

[18] Sami Hanhijärvi, Gemma Garriga, and Kai Puolamäki. Randomization techniques for graphs. *Proc of the 9th SIAM Conference on Data Mining*, pages 780–791, 2009.

[19] Michael Hay, Gerome Miklau, David Jensen, Don Towsley, and Philipp Weis. Resisting structural re-identification in anonymized social networks. *Proc. VLDB Endow.*, 1:102–114, August 2008.

[20] Michael Hay, Gerome Miklau, David Jensen, Philipp Weis, and Siddharth Srivastava. Anonymizing social networks. Technical report, SCIENCE, 2007.

[21] Jun Huan, Wei Wang, and Jan Prins. Efficient Mining of Frequent Subgraphs in the Presence of Isomorphism. *Data Mining, IEEE International Conference on*, 0:549+, 2003.

[22] Jun Huan, Wei Wang, Jan Prins, and Jiong Yang. Spin: mining maximal frequent subgraphs from graph databases. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, pages 581–586, New York, NY, USA, 2004. ACM.

[23] Akihiro Inokuchi, Takashi Washio, and Hiroshi Motoda. An apriori-based algorithm for mining frequent substructures from graph data. In *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, PKDD '00, pages 13–23, London, UK, 2000. Springer-Verlag.

[24] B. W. Kernighan and S. Lin. An Efficient Heuristic Procedure for Partitioning Graphs. *The Bell system technical journal*, 49(1):291–307, 1970.

[25] Michihiro Kuramochi and George Karypis. Frequent subgraph discovery. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, ICDM '01, pages 313–320, Washington, DC, USA, 2001. IEEE Computer Society.

[26] Michihiro Kuramochi and George Karypis. Finding frequent patterns in a large sparse graph*. *Data Min. Knowl. Discov.*, 11:243–271, November 2005.

[27] Kun Liu and Evimaria Terzi. Towards identity anonymization on graphs. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, SIGMOD '08, pages 93–106, New York, NY, USA, 2008. ACM.

[28] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. L-diversity: Privacy beyond k-anonymity. *ACM Trans. Knowl. Discov. Data*, 1, March 2007.

[29] Kathy Macropol, Tolga Can, and Ambuj Singh. RRW: repeated random walks on genome-scale protein networks for local cluster discovery. *BMC Bioinformatics*, 10(1):283+, September 2009.

[30] Kathy Macropol and Ambuj Singh. Scalable discovery of best clusters on large graphs. *Proc. VLDB Endow.*, 3:693–702, September 2010.

[31] Sara Mostafavi, Debajyoti Ray, David W. Farley, Chris Grouios, and Quaid Morris. GeneMANIA: a real-time multiple association network integration algorithm for predicting gene function. *Genome biology*, 9 Suppl 1(Suppl 1):S4+, 2008.

[32] Arvind Narayanan and Vitaly Shmatikov. De-anonymizing social networks. In *Proceedings of the 2009 30th IEEE Symposium on Security and Privacy*, pages 173–187, Washington, DC, USA, 2009. IEEE Computer Society.

[33] Siegfried Nijssen and Joost N. Kok. A quickstart in frequent structure mining can make a difference. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, pages 647–652, New York, NY, USA, 2004. ACM.

[34] Jordi Nin, Javier Herranz, and Vicenç Torra. On the disclosure risk of multivariate microaggregation. *Data Knowl. Eng.*, 67:399–412, December 2008.

[35] P.Gleiser and L. Danon. Adv. complex syst.6, 565, 2003.

[36] Pascal Pons and Matthieu Latapy. Computing communities in large networks using random walks. *J. of Graph Alg. and App. bf*, 10:284–293, 2004.

[37] Matthew J. Rattigan, Marc Maier, and David Jensen. Graph clustering with network structure indices. In *Proceedings of the 24th international conference on Machine learning*, ICML '07, pages 783–790, New York, NY, USA, 2007. ACM.

[38] Hiroto Saigo, Sebastian Nowozin, Tadashi Kadowaki, Taku Kudo, and Koji Tsuda. gboost: a mathematical programming approach to graph classification and regression. *Mach. Learn.*, 75:69–89, April 2009.

[39] Latanya Sweeney. k-anonymity: a model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10:557–570, October 2002.

[40] Jeffrey Travers, Stanley Milgram, Jeffrey Travers, and Stanley Milgram. An experimental study of the small world problem. *Sociometry*, 32:425–443, 1969.

[41] Stijn van Dongen. *Graph Clustering by Flow Simulation*. PhD thesis, University of Utrecht, 2000.

[42] N. Vanetik, E. Gudes, and S. E. Shimony. Computing frequent graph patterns from semistructured data. In *Proceedings of the 2002 IEEE International Conference on Data Mining*, ICDM '02, pages 458–, Washington, DC, USA, 2002. IEEE Computer Society.

[43] Takashi Washio and Hiroshi Motoda. State of the art of graph-based data mining. *SIGKDD Explor. Newsl.*, 5:59–68, July 2003.

[44] Leting Wu, Xiaowei Ying, and Xintao Wu. Reconstruction from randomized graph via low rank approximation. In *Proceedings of the SIAM International Conference on Data Mining*, SDM 2010, pages 60–71, Columbus, Ohio, USA, 2010.

[45] Xifeng Yan and Jiawei Han. gspan: Graph-based substructure pattern mining. In *Proceedings of the 2002 IEEE International Conference on Data Mining*, ICDM '02, pages 721–, Washington, DC, USA, 2002. IEEE Computer Society.

[46] Xifeng Yan and Jiawei Han. Closegraph: mining closed frequent graph patterns. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '03, pages 286–295, New York, NY, USA, 2003. ACM.

[47] Xiaowei Ying, Kai Pan, Xintao Wu, and Ling Guo. Comparisons of randomization and k-degree anonymization schemes for privacy preserving social network publishing. In *Proceedings of the 3rd Workshop on Social Network Mining and Analysis*, SNA-KDD '09, pages 10:1–10:10, New York, NY, USA, 2009. ACM.

[48] Xiaowei Ying and Xintao Wu. Randomizing Social Networks: a Spectrum Preserving Approach. In *SDM*, pages 739–750. SIAM, 2008.

[49] Xiaowei Ying and Xintao Wu. Graph generation with prescribed feature constraints. In *Proceedings of the SIAM International Conference on Data Mining*, SDM 2009, pages 966–977, Sparks, Nevada, USA, 2009.

[50] Xiaowei Ying and Xintao Wu. On link privacy in randomizing social networks. In *Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, PAKDD '09, pages 28–39, Berlin, Heidelberg, 2009. Springer-Verlag.

[51] W.W. Zachary. Information-flow model for conflict and fission in small-groups. *Journal Of Anthropological Research*, 33:452—473, 1977.

[52] Lijie Zhang and Weining Zhang. Edge anonymity in social network graphs. In *Proceedings of the 2009 International Conference on Computational Science and Engineering - Volume 04*, pages 1–8, Washington, DC, USA, 2009. IEEE Computer Society.

[53] Bin Zhou and Jian Pei. Preserving privacy in social networks against neighborhood attacks. In *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering*, pages 506–515, Washington, DC, USA, 2008. IEEE Computer Society.

[54] Dengyong Zhou, Olivier Bousquet, Thomas N. Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*, volume 16, pages 321–328, 2004.

[55] Lei Zou, Lei Chen, and M. Tamer Ozsu. k-automorphism: a general framework for privacy preserving network publication. *Proc. VLDB Endow.*, 2:946–957, August 2009.